Information State Based Speech Recognition

GOTHENBURG MONOGRAPHS IN LINGUISTICS 41

Information State Based Speech Recognition

Rebecca Jonson

UNIVERSITY OF GOTHENBURG
GRADUATE SCHOOL OF LANGUAGE TECHNOLOGY

Doctoral dissertation in Linguistics, University of Gothenburg

Printed by Reprocentralen, University of Gothenburg, 2010.
Dissertation edition, May 2010
Published at: http://hdl.handle.net/2077/22169

Distribution:
Insititutionen för filosofi, lingvistik och vetenskapsteori, Göteborgs Universitet
Box 200, 405 30 Göteborg

Abstract

Ph.D dissertation in Linguistics at University of Gothenburg, Sweden, 2010

Title: *Information State Based Speech Recognition*
Author: Rebecca Jonson
Language: English
Department: Department of Philosophy, Linguistics and Theory of Sciences, University of Gothenburg, Box 200, SE-405-30 Göteborg
Series: Gothenburg Monographs in Linguistics 41
Published at: http://hdl.handle.net/2077/22169

One of the pitfalls in spoken dialogue systems is the brittleness of automatic speech recognition (ASR). ASR systems often misrecognize user input and they are unreliable when it comes to judging their own performance. Recognition failures and deficient confidence estimation affect the performance of a dialogue system as a whole and the impression it makes on a user. Humans outperform ASR systems on most tasks related to speech understanding. One of the reasons is that humans make use of much more knowledge. For example humans appear to take a variety of knowledge-based aspects of the current dialogue into account when processing speech. The main purpose of this thesis is to investigate whether speech recognition also can benefit from the use of higher level knowledge sources and dialogue context when used in spoken dialogue systems.

One of the major contributions of this thesis is to provide more insight into what type of knowledge sources in spoken dialogue systems would be potential contributors to the task of ASR and how such knowledge can be represented computationally. In the framework of *information state based* dialogue management we have an important source of semantic and pragmatic knowledge represented in the *information state*. We will investigate if the knowledge in the information state can help to alleviate the search problem and reliability estimation in speech recognition. We call this knowledge and context aware approach to speech recognition *information state based speech recognition*.

The first part of this thesis investigates approaches to obtaining better initial language models more rapidly for spoken dialogue systems and ways of dynamically selecting the most appropriate models based on the dialogue context.

The second part of this thesis concerns the use of the speech recognition output and investigates how additional knowledge sources can enhance a dialogue system's decision-making on how to proceed and make use of speech recognition hypotheses.

The thesis presents several experimental studies addressing the issues described above and proposes an integration of the explored techniques into the GoDiS dialogue system.

Keywords: dialogue systems, speech recognition, language modelling, dialogue move, dialogue context, ASR, higher level knowledge, linguistic knowledge, N-Best re-ranking, confidence scoring, confidence annotation, information state, ISU approach.

Acknowledgements

A long journey is coming to an end and it is time to express my gratitude to everyone travelling with me along the way. This journey would not even have started if it was not for my supervisor Prof. Robin Cooper, who was the first to encourage me to become a researcher, and gave me the opportunity to carry out my PhD studies remotely from Spain. Moreover, without Robin holding the compass and helping me to navigate I would probably be lost on the seven seas. Thanks for your guidance, support, patience and never-ending encouragement. Thanks for reading and re-reading every single paragraph of the manuscript. Also my deep gratitude for contributing to this thesis with your perspicaciousness, brilliant ideas and insightful comments.

In fact, the inspiration for this voyage started long before my PhD studies with my first encounter with speech recognition and dialogue systems. Thanks to Prof. Jim Hieronymous and Prof. Luis Hernández for introducing me to the world of speech recognition and to Staffan Larsson for evoking my interest in dialogue systems and introducing me to the GoDiS "club". The research questions in this thesis started to grow while working at Telefónica R&D (TID). I am therefore sincerely grateful to my dear TID colleagues for all interesting discussions. Gracias!

This thesis has clearly benefitted from all constructive feedback and insightful comments from my second supervisor Rolf Carlson, from Oliver Lemon (Chapter 8), from Steve Young and Karl Weilhammer (Chapter 4 and Chapter 7), from Joakim Nivre (Chapter 4, Chapter 6 and Chapter 8), from Stina Ericsson and from David Hjelm as well as from all valuable comments from anonymous reviewers of my published papers.

For a voyage to be able to reach its end it is also necessary to have the machinery working. A big thank you to Robert Andersson for being a brilliant remote system administrator keeping the boat's engine healthy and up to date. Warm thanks to the Gothenburg Dialogue lab people (Aarne, Ann-Charlotte, Björn, David, Håkan, Jessica, Peter, Staffan, Stina) who have contributed in many different ways and given me a helping hand with GF and GoDiS whenever needed. A special thanks to David Hjelm for your technical help, for inspiring discussions about speech and dialogue and for being a brilliant colleague both at the university and at Artificial Solutions. I also owe a great deal to all students and colleagues who participated in the experiments.

An asset when travelling is all the people you meet. I am so lucky to have had the opportunity, thanks to GSLT[1], to get to know all amazing GSLT PhD students. I am also happy to have met so many wonderful colleagues, both at the Department of Linguistics and at Artificial Solutions. I am sincerely grateful to have had the opportunity to participate in the TALK project[2] and being able to meet so many interesting people sharing the same research interests, as well as receiving funding for my work and for the conference travels to present my work.

This journey would not have been possible without the support of my family and friends. Thanks to my parents for always believing in me and encouraging whatever journey I decide to embark on. Besos for Bella for being an extraordinary sister. Kramar to my brother Joakim and his wonderful kids. Lots of love to my everlasting friends (Annica, Kristin, Magdalena, Mira, Sofia, Therese) for being so close although we are so far away. My deepest thanks to Oscar for being the best possible travelmate and making every new day a thrilling adventure.

Finally, thanks beforehand to everyone that has the intention of reading parts of this thesis.

Enjoy the journey!

[1]The Swedish Graduate School of Language Technology

[2]TALK (Talk and Look, Tools for Ambient Linguistic Knowledge), IST-507802

Contents

List of Figures

List of Tables

List of Abbreviations

ASR	automatic speech recognition, page 1
CER	concept error rate, page 33
CFG	context-free grammar, page 18
DM	dialogue move, page 70
DME	dialogue move engine, page 66
DMER	dialogue move error rate, page 34
DMSeqER	dialogue move sequence error rate, page 34
DMSLM	dialogue move specific SLM, page 110
DomNewsSLM	SLM based on the domain selected part of the GNC corpus, page 96
DSP	digital signal processing, page 11
DTMF	Dual-tone multi-frequency signaling, page 29
FA	false acceptance, page 59
FC	false confirmation, page 77
FR	false rejection, page 59
GF	the Grammatical Framework, page 71
GMM	Gaussian Mixture Model, page 15
GSLC	Gothenburg Spoken Language Corpus, page 93
GSLCSLM	SLM based on the GSLC corpus, page 96
HMM	Hidden Markov Model, page 14
ICMs	interactive communication management moves, page 75
IS	information state, page 65
ISU	information state update, page 65
MBT	memory-based tagger, page 164
MIV	module interface variable, page 67
MixDomNews	Interpolation of DomNewsSLM and MP3GFSLM, page 97
MixGSLC	Interpolation of GSLCSLM and MP3GFSLM, page 96
MixNews	Interpolation of NewsSLM and MP3GFSLM, page 97
MP3GFSLM	SLM based on the GF grammar-generated MP3 corpus, page 96
MP3GFSRG	SRG compiled from the MP3 GF grammar, page 96
NewsSLM	SLM based on the GNC News corpus, page 96
OAA	Open Agent Architecture, page 67
OOG	out-of-grammar, page 19
OOVs	out-of-vocabulary words, page 3

PCFG	probabilistic context-free grammar, page 49
POS	part-of-speech, page 47
PPL	perplexity, page 97
QUD	questions under discussion, page 68
RIV	resource interface variable, page 68
SA	sentence accuracy, page 32
SDS	spoken dialogue system, page 11
SER	sentence error rate, page 32
SLM	statistical language model, page 17
SLU	spoken language understanding, page 23
SNR	speech-to-noise-ratio, page 37
SRG	speech recognition grammar, page 18
TIS	total information state, page 68
Triple SLM	Interpolation of MP3GFSLM, GSLCSLM and DomNewsSLM, page 97
TTS	Text-to-Speech, page 23
WA	word accuracy, page 31
WER	word error rate, page 31
WOz	Wizard of Oz, page 53

Chapter 1

Introduction

Automatic recognition of speech converts human spoken language into written words. It is a 50 year old technique that has now matured to the point where it is being applied commercially. One of the most promising applications of automatic speech recognition (ASR) is in spoken dialogue systems. Spoken dialogue systems are opening up big opportunities not only as 24/7 customer service interfaces but also as they enable new types of services and applications. With spoken dialogue systems users can carry out a spoken dialogue with a machine to among other things retrieve information (e.g. timetable schedule), carry out some task (e.g. book a flight) or request a process (e.g. a bank transaction). It is not unusual today that people's first encounter with ASR is with these types of systems. Moreover, it is not unusual that people's first impression of speech recognition is that it leaves much to be desired. Adding speech to a dialogue system has proved to be much more complicated than was first thought.

Although ASR has made important improvements during the last decades, recognition performance for dialogue systems is still deficient. Recognition failures affect the performance and impression of the dialogue system as a whole. It is not just that speech recognizers are error prone. It is also difficult to get a system to judge how well the speech recognizer is performing. The deficiencies in current speech recognizers means that users are restricted in what they can say and the commercial systems which have been built have often very little flexibility. Despite these restrictions, speech has been widely introduced into commercial systems. However, it has not been possible to take full advantage of the flexibility offered by the use of a natural language interface. Many advanced techniques used in text-based dialogue systems have therefore been abandoned. Unfortunately the brittle performance of ASR not only limits the use of spoken interfaces but also the complexity of tasks that can be performed. Ways of improving ASR performance is therefore not only a (long term) goal for researchers but an immediate requirement from industry. Current speech recognition techniques in use make use of very little knowledge about language and dialogue. This is knowledge that is easily accessible in dialogue systems. The main concern in this thesis is in what ways we can make use of such knowledge to enhance the recognition performance in spoken dialogue systems.

1.1 Why is speech so difficult for dialogue systems?

Why is the addition of speech to dialogue systems so difficult? One of the key concepts is variability: variability of the acoustic environment, variability of the channel capturing the audio, variability of users' voices, variability of the acoustic realization of phones, variability in pronunciation of words, variability of the vocabulary and expressions users make use of and variability in their dialogue behaviour. Such great variability makes it hard to create accurate computational models.

What are the factors in spoken language that make it so difficult for machines to recognize speech? First of all, analysis of continuous speech is *per se* a difficult task as there are no clear boundaries between words (like the spaces we use in text). The ambiguity and non-discreteness of the speech signal is often exemplified with the phrase: "How to wreck a nice beach" which acoustic realization is very similar to the one for the phrase: "How to recognize speech" (Gold and Morgan, 2000; Lieberman *et al.*, 2005). Acoustically similar words (homophones) or phrases are very common in spoken language and the speech signal is therefore highly ambiguous which makes the segmentation into words and phonemes a very difficult task. In addition, the simplified definition we make of phonemes is difficult to apply directly to the speech signal due to phenomena such as co-articulation, reduction and assimilation. Human speech production does not seem to focus on absolute phonetic targets as this would require an energy consuming articulation effort. Focus is rather on maintaining enough phonetic contrast between speech sounds to enable them to be distinguished perceptually or rather enough contrast to enable words to be distinguished conceptually. This constant balancing act by speakers, trying to minimize speech production effort while maintaining intelligibility for the listener, is according to Lindblom (1990) the reason for the large phonetic variation in speech. As the ultimate goal is understanding and there are many cues outside the speech signal to derive the meaning there is not always a need for a clear phonetic realisation. This explains why humans put less articulation effort on the most predictive patterns which therefore also turn out to have the most varied acoustic forms.

Speech is not only a stream of words used to convey meaning but it also conveys much more such as emotions, attitudes and information about the speaker. In addition, the acoustic signal that a speech recognizer receives does not come as plain speech. A speech recognizer has to be able to distinguish speech from other acoustic signals such as noise. If it was the case that the acoustic environment were held constant this could be modelled, but systems are exposed to many different environments and channels which are hard to handle acoustically. Channel bandwidth, noise, room acoustics and telephone and microphone frequency are some of the factors which disturb recognition of the speech signal. The problem becomes even more complex with current requirements of ubiquitous computing where a user is expected to connect to a system from any location via any of a variety of channels. This means even more channel and acoustic environment variability. In this view, the acoustic signal that a speech recognizer has to decipher is packed with information, some of it of importance for the task and some of it only a disturbing factor that distorts the speech signal.

Another aspect is that the variation of human voices is immense and in most spoken dialogue systems we have to try to cope with all kinds of speakers who say the same thing in acoustically quite distinct ways. Some of them will be easier than others to recognize. Ideally, the system should be able to cover all types of users so as not, for example, to discriminate against minorities. Even the same speaker varies acoustically depending on physical and emotional factors such as having a cold or being stressed. Not only do different speakers differ in their pronunciation of words but even the same speaker will pronounce the same words differently in different contexts. Pronunciation in spoken dialogue is highly varied especially when it comes to highly frequent words which are realized acoustically in many ways and often reduced. Speakers also have different dialogue behaviours, such as more or less disfluent speech production. People do not speak as clearly and eloquently as they think they do. They produce filled pauses, repetitions, repairs and truncated words (reductions). They make false starts, mistakes and slips of the tongue and they even change their minds during speech production. How can we cope with all these disfluencies? Some small comfort is gained from the fact that it seems that users modify their speech in dialogues with machines and speak more clearly than they would have done with a human dialogue partner. The occurrence of disfluencies is nevertheless common in human-machine dialogues as the Adapt Corpus shows (Gustafsson *et al.*, 2000) but less frequent than in human-human dialogues as the disfluency study made by Eklund and Shriberg (1998) shows. People also produce a lot of extralinguistic sounds such as inhalation, throat clearing, clicking, lip smacking or coughing. We have pointed out that it is difficult to handle irrelevant non-vocal sounds. It is even more difficult to make a machine distinguish between linguistic and extra-linguistic vocal signals. Speakers are often not aware of all disfluencies and non-words they produce vocally and will therefore be unable to consciously control their production.

An additional problem is the difficulty in modelling vocabulary and idiomatic expressions. People will always come up with words that developers had not thought of or that the system had not been exposed to before, so called out-of-vocabulary words (OOVs). Although there exist techniques which handle unknown words in text-based dialogue systems, for example (Purver, 2002), speech recognizers cannot recognize words that are not in its predefined vocabulary. The lack of good methods to identify OOVs, which instead may lead to the incorrect recognition of an in-vocabulary word also affects the recognition of surrounding words. This means a whole phrase can be misrecognized due to the use of an OOV word.

Speech recognizers are not ready-made systems but need to be adapted and fine-tuned to the task and domain. This means developers need to provide domain vocabularies and a model of the domain language. The high productivity and variability of language makes it difficult for developers to predict user expressions or to collect enough material to capture possible variants in the application domain. It is a chicken and egg problem where the developer needs a working system to collect realistic user expressions but cannot achieve this without any good starting material. Then why not use all the quantity of text available electronically to capture the variation of language? The problem is that text is not speech. Some people claim that speech is ungrammatical and fragmentary. Others would probably

say that it just does not follow the grammatical rules formalized for written language. At any rate, speech is distinct from written text due to its real-time processing with shorter, less complex sentences, disfluencies and it may look fragmentary if you do not take into account all the contextual cues available. In addition, the distribution of words in speech is quite different from their distribution in text (Allwood, 1998). The differences between spoken and written media make most available written corpora inappropriate for speech recognition in spoken dialogue systems. For speech dictation, written corpora play a more important role as we have a particular situation where the objective is to write (through speech). This results in a spoken style closer to writing in its structure.

According to Lindblom (1990) humans vary their speech production from hyper- to hypospeech depending on communicative and situational demands. The less information to be found outside the speech signal the more information is needed in the signal. Hyperspeech has stricter pronunciation patterns, acoustically more distinctive sounds, is clearer with less reductions and thereby also has a slower pace. Reading out loud and speech dictation could be considered close to hyperspeech. Hypospeech on the other hand appears in spontaneous dialogue and has an increased speech rate and increased quantity of reductions, co-articulations and assimilations. Hypospeech is much harder to perceive than hyperspeech without additional knowledge sources as it relies much more on the context for intelligibility. In spoken dialogue systems we are much closer to hypospeech than hyperspeech.

When using speech as an interface in dialogue systems we also have to take into account that dialogue is an interactive and collaborative process with different speakers taking turns to speak and sometimes actually overlapping each other. In text-based dialogue systems turn-taking is built into the system automatically whereas in spoken dialogue systems we need to decide when to consider a user turn as finished and be aware that the user may interrupt the system or start speaking before the system has finished its turn. A recognizer needs to detect the "end points" of the acoustic signal, i.e. when the speaker is considered to have finished her spoken turn. Dialogue system and user then have to agree who should speak when. This is much more difficult than it seems. Commercial systems therefore often choose to limit the turn-taking flexibility and restrict users to talk in explicit turns without the user interrupting or overlapping the system. In these systems, a user turn is often considered as finished whenever a certain period of silence has occurred or the user has passed a fixed time limit. In systems that do accept user interruption, so called *barge-in*, the system will stop speaking and give the floor to the user whenever the user says anything. However, turn-taking is more complex than this as the user could well only have given positive feedback such as a "uhm" without intending to take the turn. Finally, most spoken dialogue systems will be exposed to more or less experienced users who will behave quite differently on all levels and a spoken dialogue system will thereby need to be prepared for both experts and novices.

Humans often manage to compensate for many of these disturbing factors in the speech signal whereas today's speech recognizers cannot. Humans in contrast to ASR easily distinguish speech from noise or extralinguistic sounds, smoothly adapt their listening to any speaker, can recognize and identify unknown words and understand ungrammatical or frag-

mentary utterances with the help of contextual interpretation. Lippmann (1997) shows in a study how humans outperform machines in recognition tasks and how humans do not seem to be affected by noise to any great extent. In spontaneous human-human dialogue spoken turns frequently overlap and different speakers produce speech at the same time without any apparent problems. This does not seem to affect the perception and understanding of the speech and many people are not even aware of these overlaps. A clear example of the excellence of human speech processing is the so called *cocktail party effect*. You are at a bar, the music is loud and people are talking. Somehow you manage to hold a conversation anyway. Although the actual acoustic speech signal that you perceive is heavily distorted and sometimes even fragmentary you are able to catch the speaker's message and follow the conversation. This is only possible as we make use of many other cues apart from the acoustic signal in spoken dialogue, such as gesture, lip reading, facial expressions, dialogue context, situational context, knowledge about language, knowledge about the speaker and general world knowledge. As Lippmann (1997) points out further studies need to be made to clarify how humans compensate for all the disturbance in the speech signal.

Humans are not only good at compensating for disturbing factors in the speech signal but also very good at identifying them, especially in spoken dialogue. Humans do not seem to have any difficulty in knowing when they have recognition problems, why they have a problem and what problem they have. In this way they can easily solve those problems together with their dialogue partner. This is something spoken dialogue systems have a hard time with. However, we have to remember that humans too misrecognize and misunderstand what is said, which indicates that even for humans speech processing is a very complex task.

In summary, there are many things going on in human spoken language processing that have been disregarded in speech recognition for dialogue systems. In this thesis we will explore ways of integrating speech recognition more closely with other parts of a dialogue system. This will hopefully enable systems to imitate some of the human ability of using other knowledge sources in the disambiguation task of the speech signal.

1.2 Research questions

At a workshop in the 80s Frederik Jelinek is reported to have stated that: "Every time we fire a linguist, the performance of our system goes up" (Moore, 2005). What he probably referred to was that when they tried to integrate linguistically based techniques into their statistical recognition framework this normally degraded the speech recognition performance in the form of more word errors. However, this has not prevented researchers, including Jelinek, from looking for new methods to incorporate linguistic knowledge into ASR in the hope of improved speech recognition performance. I will give a survey of some of their attempts in Section 2.4. Most attempts have been focussed on achieving language models that take into account more knowledge about language. Unfortunately, what has been achieved is often only slight improvements if compared to improvements achieved by, for example, better noise robustness techniques. Quite early, Jelinek (1991) demonstrated

the weaknesses of trigrams, i.e. the statistical language models normally used in ASR, and his commitment to the search for alternatives. Brill *et al.* (1998) urged the use of linguistic and world knowledge to improve ASR and showed abilities that human subjects seemed to have used to improve output from a speech recognizer. Moore (1999) presented a survey of grammar approaches for language modelling and the possibility of hybrid approaches where linguistic structure can be modelled statistically. Glass (1999) included finding a way to incorporate linguistic constraints at an early stage in the recognition search process as one of the challenges in spoken dialogue system research. Rosenfeld (2000a) pointed out how statistical language models take little advantage of the nature of language and lack the incorporation of basic linguistic theory. More recently, Shriberg (2005) described fundamental properties of human speech that violate some of the assumptions current speech technology is based on. McTear (2002) included in his list of issues of importance in future spoken dialogue research, the issue of investigating how language understanding and dialogue management can compensate for deficiencies in speech recognition. Although, speech recognition accuracy has gone up considerably during the last decades the performance of speech recognition in spoken dialogue systems is far from optimal. Despite the fact that many people argue for the use of more knowledge, even today speech recognizers are using very little knowledge about language and dialogue.

Does this mean that the hypothesis is wrong, that speech recognition cannot benefit from the use of additional knowledge sources? As Rosenfeld (2000a) points out, it may have been the attempts to encode the knowledge of these methods into the current statistical framework that has failed and not the methods *per se*. This means that it is not necessarily the assumption that is wrong but the difficulty of combining more structured linguistic knowledge into the current probabilistic framework. Although statistical methods with minimal linguistic knowledge is the leading approach in speech recognition it is evident that it is not accurate enough.

I adhere to the view that speech recognition may profit from the use of other knowledge sources and that by no means all attempts in this area have been in vain. In this thesis we will continue this challenging line of investigation where good proofs seem rare but indications are many. One of the contributions of this thesis is to show possible new knowledge sources and how to incorporate them into the speech recognition process of a dialogue system. In spoken dialogue systems we actually have access to much more information than for other applications of speech recognition. Therefore, we should not see ASR as an external, separate process in a dialogue system framework but rather we need a tighter coupling where language understanding modules and dialogue management share knowledge with ASR. ASR should take into account the information available at runtime in a dialogue system to alleviate its search problem. This can be done in two ways: by incorporating this knowledge into the recognizer or by making use of this knowledge on the recognition output. This means we can try to improve the recognition process or improve upon the recognizer's output.

One of the goals of this thesis is to present methods that fulfil the requirement of immediately applicable solutions and that can be applied without interfering with internal speech recognition processes. Recent speech recognition enhancements have been obtained

by for example improving acoustic models, improving robustness to noise, better digital signal processing etc. All these methods involve altering the ASR system which is seldom a possibility for a dialogue system developer using a commercial black-box ASR system. In this thesis I will therefore present methods that can be used by a dialogue system developer to improve the performance of the ASR that she uses without needing to alter the ASR system *per se*. In this way the experiments are an attempt to show how you can make the most of the speech recognizer which is available.

On the one hand, we want to prevent initial errors and improve ASR performance by e.g creating and applying better language models that are more suited to the current context. We want to capture better what users say but also predict in what situation they might say it. To be able to do this we need to make use of knowledge about the dialogue. We also need to get round the chicken and egg problem when no suitable data is available so that we can get a better first version of a dialogue system. These issues will be the focus of the first experiments of this thesis presented in **Part II**.

On the other hand, we want to identify errors better to know what to do with the ASR hypotheses. An error-prone ASR introduces uncertainty that a dialogue system needs to handle. We want to more accurately predict when the ASR will fail and be able to select the most appropriate hypothesis which will enable us to proceed with the dialogue. To be able to do this we need to introduce more knowledge sources than the ones currently used in the selection process. This will be the focus of the second suite of experiments in this thesis presented in **Part III**.

The main purpose of this thesis is to investigate how we can benefit from the use of more knowledge sources in speech recognition for dialogue systems. We will experiment with different sources of knowledge on different levels and investigate how to encode this knowledge. Dialogue system developers in the framework of *information state based* dialogue management have an important source of semantic and pragmatic knowledge in the *information state*. We will attempt to make use of this knowledge from the dialogue system for speech recognition and we will call this more tightly coupled approach *information state based speech recognition*.

1.3 Thesis outline

This thesis is structured into four main parts. **Part I** starts with a brief non-technical introduction to speech recognition and spoken dialogue systems followed by a discussion and survey of earlier approaches to the enhancement of ASR. This first chapter, **Chapter 2**, also presents the evaluation metrics used in the various experiments in the thesis. All approaches presented in this thesis have been considered for the information state based dialogue system GoDiS and evaluated in two different domains: the MP3 player domain and the Calendar domain. A pilot experiment has been carried out for each approach in the MP3 player domain followed by a more extensive and thorough experiment in the Calendar domain. This experimental setting has been chosen in order to avoid domain-dependence. Nevertheless, we should bear in mind that both domains are quite small. The information

state update approach, the GoDiS dialogue system and these two baseline systems are introduced in **Chapter 3**. **Chapter 3** also presents the two machine learning toolkits used.

Part II embarks upon the problem of providing a speech recognizer for a dialogue system with good initial language models to enhance a dialogue system's ASR performance. **Chapter 4** considers the generation of statistical language models from grammars to investigate whether such models give more robust behaviour than speech recognition grammars. This would be a way to alleviate the chicken and egg dilemma in language modelling. **Chapter 5** pursues this idea further by generating statistical language models where certain dialogue moves are boosted that, when used in the appropriate context, may improve ASR performance further. **Chapter 6**, investigates how such *dialogue move specific statistical language models* could be predicted using the information state.

Part III concerns the use of the speech recognition output and investigates how additional knowledge sources can enhance a dialogue system's decision-making of how to proceed and make use of speech recognition hypotheses. **Chapter 7** shows how the bootstrapping approach from **Part II** is applied to semantic decoding in order to be able to use the resulting dialogue move tagger for the subsequent experiment in **Chapter 8**. **Chapter 8** outlines an approach to confidence classification of N-Best hypotheses for later re-ranking that makes use of additional knowledge sources such as dialogue context.

Part IV constitutes the integration of some of these techniques in the GoDiS dialogue system and a consideration of future research. **Chapter 9** describes how the implementation has proceeded and been planned. In **Chapter 10** we discuss possible future directions based on the results of this thesis.

The thesis finally ends with a summary of results and a concluding discussion of the information state based recognition approach.

Part I

Preliminaries

Chapter 2

Background

The first part of this background chapter is principally aimed at readers who are unfamiliar with automatic speech recognition (ASR) and spoken dialogue systems (SDSs). I will give a brief introduction to ASR and SDSs followed by an introduction to some common metrics in ASR that have been used to evaluate the experimental results throughout this thesis. The remainder of the chapter consists of a survey of the problems that arise when applying ASR to SDSs and the attempts in research to overcome the deficiencies of ASR inspired by knowledge about human speech recognition and language.

2.1 A brief introduction to ASR

It is beyond the scope of this thesis to describe in detail how ASR works so I will rather give a brief overview of the main issues for a better understanding of the objectives of this thesis. This non-technical description will be focussed rather on how knowledge about speech and language is used in ASR than on the technical engineering aspects. For a more thorough introduction to ASR see Jurafsky and Martin (2008), Huang *et al.* (2001) or Young (1996).

The typical architecture of an ASR system is illustrated in Figure 2.1. An ASR system captures speech through some access device, for instance a telephone, and then passes the speech to the digital signal processing (DSP) that will output a digital representation of the speech signal. The decoder is the central part that with help of acoustic models (HMMs and GMMs) and language models (SLMs) will try to hypothesize what the speaker might have said based on the acoustic evidence (the digital representation) and output a ranked list of hypotheses in written form. In this example the speaker said "hej" ("hi" in Swedish) and the system recognized this correctly (see rank 1). The following sections will describe this process further.

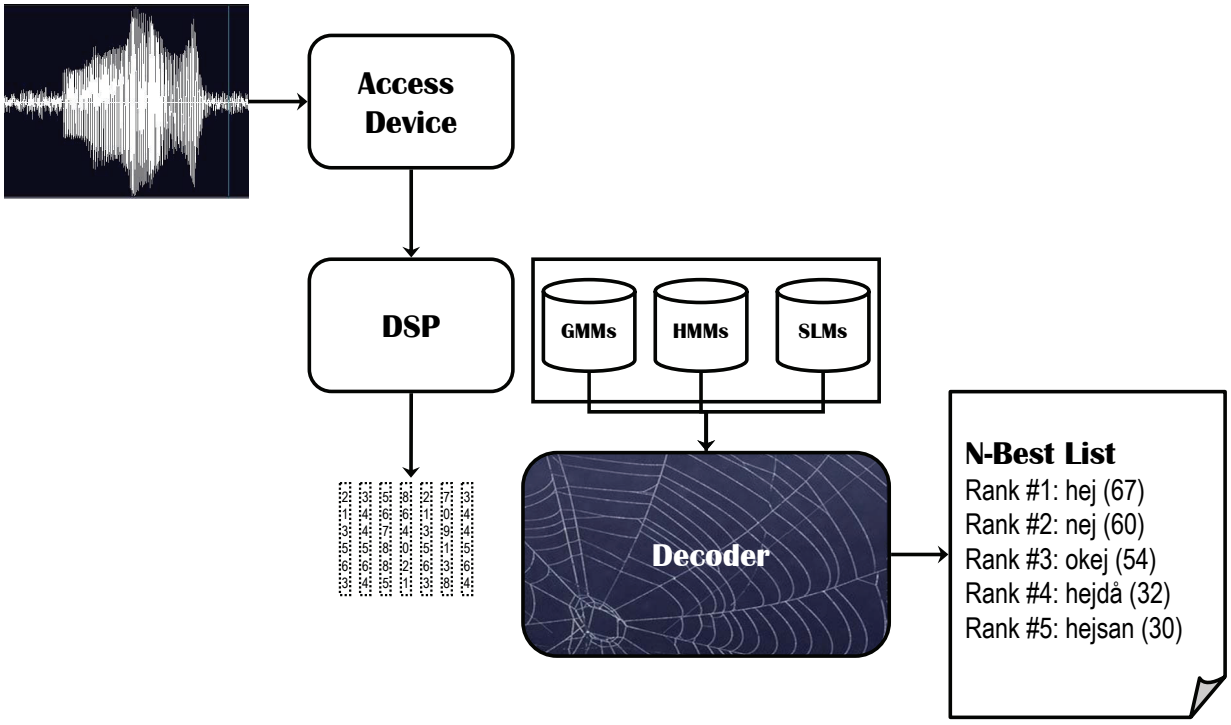

Figure 2.1: A typical ASR system architecture

2.1.1 Digital signal processing

First of all, the analog continuous speech signal needs to be digitized to a discrete representation for use in computers. The Front End in speech recognizers that carries out this task is the *Digital Signal Processing* (DSP) component that converts an acoustic spoken signal into its digital representation.

The analog to digital (A/D) conversion consist of *sampling* and *quantization*. As the analog signal is continuous with infinitely many data points and computers can only keep track of finite sets we need to decide how many points to store or rather with what time interval we need to sample the signal to capture frequencies. In fact, human speech is normally below 10000 Hz which means we do not need to consider frequencies above that. Moreover, as computers can only hold an approximation of numeric values we need to decide with what precision values should be stored. This rounding off of numeric values is the quantization step. In this conversion we will evidently have an information loss. Sampling and quantization need to be optimized to maintain faithfulness to the analog signal while keeping the digital print small enough. The digital representation needs to be kept relatively small partly for computer storage purposes but also for better use in subsequent processes in the recognizer.

There are different techniques for extracting and compressing the information in the digital signal, *Cepstral Analysis* and *Linear Predictive Coding* (LPC) being the most common. The information which has proved to be most valuable is that concerning the frequency components of a wave. Normally, a digital representation of an acoustic signal in ASR consists of a sequence of acoustic feature vectors where a feature vector represents acoustic characteristics detected in a small time span (a frame of 10-20ms). The assumption is that a signal is constant over very short time intervals because the vocal tract shape does not change that rapidly. A continuous speech signal can therefore be segmented into short frames represented by a feature vector. The features represent the spectral information considered important and unique to the acoustic signal fragment.

In spoken language processing the goal is to extract the information in the acoustic signal which is invariant over different utterances of the same linguistic material. For example, loudness is of less importance as I could utter the same word almost whispering or screaming and you would still perceive it as the same word. What we need to identify are those features that are similar when we perceive two apparently different acoustic signals as the same word but differ for acoustic signals perceived as different words. We need to focus on the perceptual information which is important to make phonetic distinctions and try not to be sensitive to acoustic variation that is irrelevant. The acoustic feature vectors used in ASR normally consist of 39 features representing spectral information and energy. For a detailed overview of how feature extraction is carried out and what features are normally extracted and used in ASR see Huang *et al.* (2001).

Speech sounds are constrained by the limitations of the human speech and hearing apparatus. It is important to take such limitations into account in the design of ASR systems. The range of frequencies used in speech is much more limited than the range of frequencies of other sounds in the world. Moreover, human hearing is not equally sensitive

to all frequencies but has a more refined perception of frequencies on some frequency levels commonly used in speech. Most ASR systems use techniques that are inspired by this non-linear human perception of frequency bands. Applying knowledge of human auditory processing has led to considerable improvements in ASR. The application of findings from human hearing is further discussed in Section 2.4.2.

As discussed in the introduction, speech does not come alone but jointly with other surrounding sounds. That means that the first thing a DSP component needs to do is to filter out or to compensate for noise and other non-verbal sounds. An ASR Front End therefore needs to model noise and how it may affect the speech signal. ASR performance is very brittle when it comes to noise and there exist various techniques to make ASR systems more robust to it. Noise robustness will be addressed in Section 2.4.1.1. As mentioned in the introduction users do not always utter verbally meaningful sounds but also produce many extra-linguistic sounds such as inhalations, clicks, laughs etc. These also have to be distinguished from the sounds that carry linguistic information (see Section 2.4.1.2).

Speech carries much more information than just words and the same features seem to carry information about different things at the same time. When someone speaks we not only perceive the words she says but we also get information about the person's gender, age, mood, health and even her attitude towards what is said. The digitization of an analog signal means a loss of information, but not all parts of the signal are relevant for the task of ASR. While speech recognition focuses on extracting the information necessary to recognize words there are other systems that focus on extracting information about speaker identity, speaker age or speaker emotion. On the other hand, there is information that could be useful for dialogue systems which is unfortunately not extracted with current ASR techniques. Such an example is prosody which is an important factor in spoken dialogue (attempts to make use of prosody in ASR will be further discussed in Section 2.4.2.3).

Indeed, in SDSs, the main focus is to recognize the semantic content of the acoustic signal rather than the individual words. However, current systems do not map directly to semantic content but use the intermediate level of transcribed speech. There is no sound to sense process. What humans actually do is still an open question, though it seems clear, given the existence of illiterate speakers, languages without writing systems and the long existence of spoken language without any written language, that they are not necessarily representing what they hear in terms of a written transcription.

2.1.2 Acoustic modelling

To represent how people pronounce sounds, words and transitions between words, we need what is called an *acoustic model*. To capture the huge variation of speech, where not only different speakers say the same things acoustically differently but also the same speaker may vary from one time to the next, knowledge-based techniques have been abandoned in favour of statistical methods. The most common statistical technique used in ASR is that of *Hidden Markov Models (HMMs)*. The acoustic modelling can be seen as the phone recognition stage in ASR. Given the acoustic observation (from the DSP) represented as a

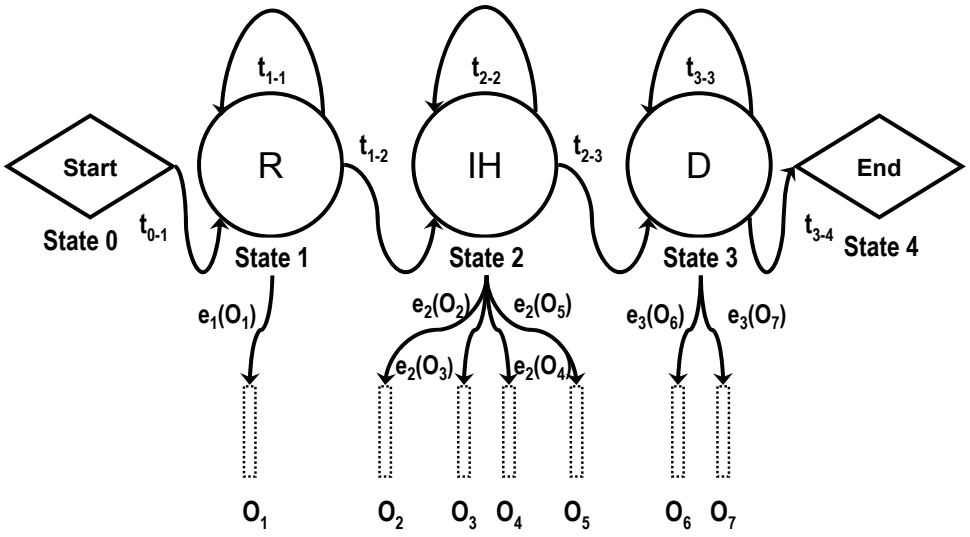

Figure 2.2: An HMM representing the pronunciation of the word "read" with transition probabilities (t_{nn}), emission probabilities ($e_n(O_n)$) and acoustic observations (O_n) in the form of vectors.

sequence of acoustic feature vectors we need to model the likelihood that a word, a phone or a subpart of a phone has given rise to such an acoustic observation. For this we use HMMs to model the probability of words being realized as certain sequences of phones and *Gaussian Mixture Models (GMMs)* to model the probability of phones being represented as certain acoustic features.

2.1.2.1 Hidden Markov Models (HMMs)

An HMM consists of states, transition probabilities between states and state emission probabilities (also called "observation likelihoods"). In ASR a hand-crafted HMM pronunciation dictionary specifies how words are modelled as HMMs where a phone (or sub-phone) is represented as a state and the move from one phone to another is represented as a transition. A word like "read" could for example be represented by an HMM of three states corresponding with the three sounds in the word and transitions between these states as in Figure 2.2. To model the duration of phones, an HMM permits self looping, i.e. a state with a transition to itself (see $t_{2\,2}$ in Figure 2.2). In this way the same HMM would be able to model different durations of a vocal sound, e.g. the variant "reeeead" of the word above. The orthographic word "read" is a homograph as also the past tense of the verb is written the same way but pronounced differently. In ASR these two words would be considered one single word with two pronunciations. This can be modelled by branching the HMM state graph from the "r"-sound with transitions to two different states representing the

two different vocal sounds. Apart from the pronunciation variants we are usually aware of, many more variants normally appear in speech due to co-articulation, assimilation and reduction. Consider the final sound in "read" and how it varies to adapt to surrounding sounds in the following two phrases when uttered rapidly: "to read a book" and "to read the paper".

Phones often last much longer than the time slice an acoustic feature vector represents and a phone would therefore often correspond to several vectors. In addition, the acoustic patterns of a phone may not be held the same during its realization. In acoustic modelling it is therefore common to match HMM states to smaller entities than phones, *sub-phones*, and represent a phone as three states (a *triphone*): start, middle and end of the phone. Such an HMM phone model can be coupled with other HMM phone models and thereby model entire words. In such a framework the word "read" could be represented by an HMM sequence consisting of three phone models with a total of nine states (plus initial and final state for the word).

2.1.2.2 Gaussian mixture models (GMMs)

HMMs also consist of state emissions which model the probability of acoustic observations in states. To model the probability that a certain HMM state (a phone or sub-phone) would give rise to a certain acoustic feature vector ASR systems make use of *Gaussian Models*. Gaussians are normal distributions which can graphically be represented as a bell curve. A Gaussian represents an acoustic feature in the vector with a mean and a variance. The further away from the mean value, the lower the probability for a value to occur. However, in many cases acoustic features are not normally distributed. Therefore what is normally modelled is several Gaussians for each feature. Thereby the name Gaussian Mixture Models (GMMs). It should be noted that some ASR systems use other methods than GMMs such as neural nets to model acoustic features.

An acoustic model based on HMMs and GMMs is trained with real speech to learn how different people say the same things. For this we need a well distributed and large acoustic corpus for training. This audio corpus does not need to be hand-labelled on the phone or the sub-phone level but only needs to be transcribed on the word level. For training we feed our system with audio files and transcriptions of these to train how words are pronounced by different speakers, in different contexts, and in different parts of a sentence. The audio files will be converted with the DSP component to acoustic feature vectors. The word transcriptions will be used to create possible HMM sequences by using the pronunciation lexicon to get the possible phones in the words and thereby pick the corresponding phone HMMs to build up HMM model sequences. The training consists of going through these HMM sequences using the feature vectors as observations and estimate probabilities for each state to have produced an observation. With enough acoustic material the system will learn certain patterns and probabilities of pronunciations. The closer the speech conditions used for collecting the training data are to the speech conditions the ASR system will be exposed to, the better the performance of the acoustic models. Common algorithms used for training acoustic models are *Baum-Welch* and *Forward-Backward* which are for example

described in Huang *et al.* (2001).

In dictation systems, part of the training of the acoustic models is made by the user to adapt the acoustic models to the user's way of pronouncing phones, so called *enrollment*. This improves recognition considerably.

The acoustic models in ASR systems normally include hand-crafted pronunciation dictionaries which model common ways of pronouncing words, e.g. the two standard ways of saying tomato (with [ei] or [a]). Developers using ASR systems often have the possibility to extend these dictionaries with pronunciations for new words and additional pronunciation for known words. For languages with clear pronunciation patterns, such as Spanish, these dictionaries will be rule-based models and the ASR can normally propose pronunciations for new words with high accuracy. For other languages the developer will probably need to specify or modify pronunciations of additional words. Addition of accurate pronunciations can improve recognition accuracy.

2.1.3 Language modelling

The way to put some more language into ASR is by providing ASR with knowledge of words, word order and word occurrences. In language syntactic, semantic and pragmatic constraints make some word sequences more likely than others. These constraints help us minimize the search space of plausible words in a certain context. Based on previously seen words humans seem to put expectations on the next word and can thereby predict that the word following "to be or not to" is more likely to be the word "be" than the word "not".

The predominant approach to modelling language in ASR is to use N-grams which model the probability of words and word sequences. It is quite a crude linguistic model where the probability of a word only depends on the immediately preceding words (normally up to two). N-grams which are often referred to as *statistical language models* or *SLMs* are trained on corpora from which probabilities are estimated by counting word and word sequence occurrences. In a *bigram* model a word is predicted based only on the previous word. The *trigram* model, which is the most common form in ASR, takes as its basis two preceding words to estimate the probability of the next. It is a simplistic and efficient way of modelling the language that the speech recognizer is expected to be exposed to. Commonly used toolkits for statistical language modelling are SRILM (Stolcke, 2002) and the CMU toolkit (Clarkson and Rosenfeld, 1997). These toolkits are fed with corpora and generate SLMs for example in ARPA format. Both toolkits are used in this thesis. Some speech recognizers also supply their own SLM toolkit.

Consider the following minimal corpus example where the symbols $< s >$ and $< /s >$ stand for sentence beginning and sentence ending:

(1) $< s >$ CAN YOU HEAR $< /s >$
$< s >$ CAN YOU SPEAK $< /s >$
$< s >$ CAN YOU SEE $< /s >$
$< s >$ CAN YOU THINK $< /s >$

An SLM trained on this minimal corpus would estimate that the words "can" and "you" are more probable than the words "hear" or "speak" and that "can" is a probable start of a sentence whereas "think" is not. It would also estimate that "you" is more likely to follow the word "can" than the word "can" again. The four phrases in the corpus would be equiprobable. However, due to the minimal size of the corpus this model would also predict that "you can" or "see you" are uncommon word sequences in the language. To be able to get good statistical estimates of words and word sequences vast amounts of domain-related material is needed. The nature of the distribution of words in language is such that we have a large number of rare words and a smaller number of very frequent words. This means that even with an extremely large corpus the estimates of rare words will be poor and a number of words will be missing entirely. To overcome this type of data sparseness statistical language modelling applies *smoothing techniques*. Smoothing techniques attempt to take into account the probability of n-grams seen only once to estimate the probability of unseen n-grams. Smoothing techniques also attempt to move probability mass from more frequent words and word sequences to infrequent words and word sequences to "smooth" the probability estimations. This can be illustrated by the Robin Hood metaphor: to take (probability mass) from the rich (highly frequent words and word sequences) and give to the poor (low frequent words and word sequences) to equalize (smooth) the distribution. For an introduction to smoothing techniques see Jurafsky and Martin (2008). Smoothing techniques will also be further discussed in Section 2.4.4.1.

It is often hard for dialogue system developers to get a sufficient amount of data to achieve good statistical estimates even using smoothing techniques. When statistical techniques are inconvenient to use, because of the lack of appropriate or sufficient training data, developers can opt for a rule-based language modelling technique: speech recognition grammars (SRGs). These context-free grammars (CFGs), are hand-crafted by the developer in an attempt to describe the domain language. The example corpus above could be formalised in a grammar as follows (using the GSL format by Nuance (2006)):

```
.Grammar SENTENCE

 SENTENCE (AUXVERB PRONOUN VERBINF)
 AUXVERB can
 PRONOUN you
 VERBINF [hear see speak think]
```

Capitalised strings stand for grammar rules whereas lower case strings are terminals, i.e. words. Strings inside brackets are disjunctions. Parentheses group strings together, disallowing other strings in between and imposing a fixed order. All words and phrases in this grammar would be equally probable in contrast to the SLM. When using a grammar it is easy to generalize by adding words to a category (rule) and by that cope with things not seen in a corpus. An example for this grammar would be to add the pronoun "we" which would result in a grammar that accepts four additional phrases. We could also easily add new verbs to the category VERBINF such as "understand", "smell" and "feel" or more

auxiliary verbs (AUXVERB) such as “do”, “may” or “will”. In this way we would suddenly capture more phrases. For the SLM approach we would need to collect more phrases until we capture all these expressions. An SRG is therefore a compact and generalized way of describing a language.

When an SRG is applied, the speech recognizer can only recognize the utterances that are explicitly formalised in the grammar. It is often stated that SLMs are more robust than SRGs. What is normally referred to with this robustness is that SLMs can recognize utterances not found in the training corpus whereas grammars can only recognize exactly what has been defined in the grammar. Say that we use a speech recognizer with the minimal SLM and the minimal SRG above. Imagine the speaker saying: “you can hear”. This phrase is not found either in the corpus or defined in the grammar. However, the SLM would actually be able to recognize this phrase because the individual words exist in the corpus (albeit in the wrong order). The language model probability would be low but the acoustic probability could be high which could lead to a correct result. Respecting the grammar, if the user says something outside the scope of the grammar, a so called out-of-grammar (OOG) utterance, the speech recognizer will either misrecognize it as another valid grammar utterance or reject the audio completely. It would therefore not be able to recognize this phrase. This makes the grammar approach much more restricted. Now, imagine the speaker saying: “you cannot hear”. As this phrase has a word unknown (or unseen) to both models neither of them would be able to recognize this phrase but would either reject the audio or recognize a phrase consisting of words found in the vocabulary. Speech recognizers cannot recognize unknown words but are limited to the scope of the vocabulary of the language model. Approaches to overcome this restriction will be discussed in Section 2.4.4.3.

An alternative approach provided by some ASR systems is *keyword spotting* where a recognizer does not try to recognize whole sentences but only keywords (Yu *et al.*, 2006). Consider a speech solution where the system asks the following: *Please, say the name of the city.* It is highly plausible that some users will not keep only to the expected city vocabulary but will say something as: *Uhm, Barcelona please.* The keyword spotting technique uses *filler* (or garbage) models for all possible surrounding sounds and words and tries to identify whether there is some part (or parts) of the speech signal that can be identified as one of the specified keywords. This technique is only applicable when the dialogue system task is simple enough to suffice by recognition of some slots.

In dictation systems the SLMs come with the ASR system and can then be fine-tuned by re-training the models on user documents. For SDSs it is up to the developer to model what words are expected in the domain and in what way they may appear. This can be done either by training SLMs or by writing SRGs. Language modelling is further discussed in Section 2.4.4.

2.1.4 Decoding

Given an acoustic observation, represented as a set of feature vectors (from the DSP), an ASR system needs to find a sequence of words that is likely to have led to such an acoustic

signal representation. This means we have a challenging search problem as we need to search a huge space of potential word sequences that could have resulted in the acoustic observation. To delimit this search an ASR system makes use of three models: a model of probabilities of phones being represented as certain acoustic features (GMM), a model of probabilities of words being realized as certain sequences of phones (or sub-phones) (HMM) and a model of probabilities of words and word sequences (SLM). In ASR the search for a potential word sequence given an acoustic observation is called *decoding*.

A common view of the ASR process is as a noisy channel problem. The *source* sentence, the sentence the speaker had in mind, is considered to have been sent through a noisy channel that has modified and distorted the source sentence. The noisy channel corresponds both to the vocal realization of the sentence, the channel and the DSP part. We need to break the code of this noisy sentence to recover the original message. The way we do this is to generate potential source sentences, word sequence candidates, using the language model and examine what noisy sentences they would have led to if passed through a similar noisy channel. The system will then choose the potential source sentence that seems to generate a noisy sentence close enough to the observed noisy sentence.

As ASR systems normally need real-time response the search or decoding techniques need to be fast and efficient. Common techniques are *Viterbi Search* and *Stack Decoding (A*)*. To make the Viterbi algorithm more efficient by avoiding considering all states and hence constrain the search space a method called *beaming* is used that prunes (cuts off) low probability paths (transitions through states). Therefore the term *Viterbi beam search* is often used. A good description of these algorithms can be found in Huang *et al.* (2001) and Jurafsky and Martin (2008).

To sum up: a decoder will use the digital representation of the audio, i.e. the sequence of acoustic feature vectors, as acoustic evidence and use the acoustic model and the language model to hypothesize the word sequences in the domain language that most probably could have given rise to the acoustic evidence. Current statistical techniques are governed by the amount and adequacy of the training data: the more and more adequate the better. The drawback of this data dependency is that ASR systems will always miss some words and word sequences (scarceness of the language model), some pronunciation patterns (scarceness of the HMMs), some acoustic realizations of phones (scarceness of the GMMs). Even when a user realizes an utterance that consists of words and word sequences in the language model, uses defined pronunciations and produces similar acoustic realizations of the phones the ASR may fail. The system just may not be able to constrain the huge search space sufficiently due to ambiguity on all levels and insufficient information and knowledge sources.

2.1.4.1 N-Best lists and confidence scores

In the end the decoder will output the most probable hypothesis given the acoustic observation. As we are dealing with probability there is nothing that prevents us from outputting not only the one most probable hypothesis but the N most probable hypotheses. Most ASR systems have this capability and can output *N-Best lists*. These are ordered lists of

ASR hypotheses in the following form:

(2) Transcription: när är middagen imorgon *Eng. when is the dinner tomorrow*
Rank 1: nej middag imorgon (conf: 63) *Eng. no dinner tomorrow*
Rank 2: nej middagen imorgon (conf: 60) *Eng. no the dinner tomorrow*
Rank 3: nä middag imorgon (conf: 60) *Eng. nope dinner tomorrow*
Rank 4: nä middagen imorgon (conf: 60) *Eng. nope the dinner tomorrow*
Rank 5: när middag imorgon (conf: 60) *Eng. when dinner tomorrow*
Rank 6: när är middag imorgon (conf: 60) *Eng. when is dinner tomorrow*
Rank 7: när middagen imorgon (conf: 60) *Eng. when the dinner tomorrow*
Rank 8: när är middagen imorgon (conf: 60) *Eng. when is the dinner tomorrow*
Rank 9: nä en middag imorgon (conf: 52) *Eng. nope a dinner tomorrow*
Rank 10: nej middag imorrn (conf: 52) *Eng. no dinner tomorrow*

This real Swedish example (from the AgendaTalk system presented in this thesis) first shows the transcription which corresponds to what the user actually said and then 10 hypotheses that the ASR system proposed ordered by probability. This time, the ASR system did not manage to rank the correct word sequence as the most probable hypothesis. However, it did have it as one of its top 10 possibilities at rank 8. N-Best lists can be used as a basis for more sophisticated techniques using additional knowledge sources to make the final decision of which hypothesis to select (see Section 2.4.5). In this way we add a subsequent step in the recognition without interfering with the internal recognition process.

What the ASR outputs is a hypothesis (or hypotheses) of the most probable word sequence it can find with the evidence and knowledge provided. However, there are many sources for errors due to the variability of speech and because information can be missing due to the scarceness of knowledge sources. This introduces uncertainty in ASR results. Therefore, ASR systems also estimate a *confidence score* (also confidence measure) to measure the reliability of the recognized transcription. Different ASR systems have different estimation techniques but normally estimate a number on a 0-1 scale. They usually take into account both the acoustic resemblance and the language model probability of a word sequence as opposed to competing hypotheses. The idea is that the higher the confidence score the more probable that the ASR system was correct in its choice. In the N-Best list example above we can see confidence scores given on a scale from 0-100 for each of the hypotheses by the Nuance 8.5 recognizer (Nuance, 2006). Confidence scores make it possible for dialogue systems to decide how to proceed with the recognized user input, e.g. reject it if the confidence score is low as it was most likely a misrecognition or accept it if the confidence score is high. Section 2.4.6 will discuss confidence scoring further.

2.1.5 The three fundamental ranges

Speech recognition is governed by three fundamental factors: the vocabulary size, the number of different speakers it accepts and the fluency of the speech. We can see this

as three different ranges: the vocabulary range, the speaker range and the fluency range. These three can not be simultaneously exploited to a maximum and we therefore have to choose which to prioritize. We will always have to set one of the following restrictions: fewer speakers, smaller vocabulary or less spontaneity.

The first speech recognizers appeared more than half a century ago (Gold and Morgan, 2000). In the beginning, ASR meant having a minimal vocabulary (up to 10 words), one single speaker and it was restricted to isolated one-word speech. In this case all these three ranges had been restricted to a minimum to facilitate the task. It should be mentioned that it is not only the size of a vocabulary that matters but also the acoustic confusability of the words. If the words are very much alike, such as the sounds of the following letters in the English alphabet; P, B, D, G, T, the task is more complicated. This may for instance complicate the task of a spoken interface to a cellphone address book if the names of your friends sound similar, e.g. Kim, Jim, Tim. The tasks speech recognizers are confronted with nowadays are expanding all these ranges. What we see today is either dictation systems with big vocabularies (more than 100 000 words), quickly adapted to one single speaker (in around 10 minutes) which requires continuous clear read speech. In this case the restriction lies primarily on the number of speakers where the recognizer has been adapted through training to one single speaker. The fluency range has been pushed forward as the technology has matured and nowadays the user does not need to say the words with pauses in between as in the first commercial systems. The other possibility is what is often used in SDSs: medium-vocabulary systems (hundreds or thousands of words), accepting most speakers (but probably worse for atypical or non-native speakers) and requiring the speaker to use clear continuous speech. In these systems the restriction is the vocabulary size which is much smaller than for dictation systems. The fluency range is also here being pushed forward to accept more fluent speech and not the command-like speech which the first systems required. Still, there is no ASR system that can at the same time handle a large vocabulary and accept any speaker speaking spontaneously without any constraints. Such a system seems far ahead. Actually, all ASR systems have a restricted vocabulary as they cannot recognize unknown words.

To conclude, ASR systems are not ready-made systems that you just plug in. When it comes to dictation, you will need to train it on your voice and perhaps adapt the vocabulary and the language models to your written style. ASR used for dialogue systems needs to be adapted and fine-tuned to the purpose at hand. This normally means providing a good language model, an appropriate vocabulary and deciding on how to manage the uncertainty of ASR hypotheses that the ASR system proposes. This poses a great challenge for dialogue system developers.

2.2 A brief introduction to spoken dialogue systems

A *dialogue system* is a system that can engage in a conversation (a dialogue) in a restricted domain with a user and have not only some sort of understanding of what is being said but also a way to keep track of the conversation. In this way, dialogue systems are distinct from

other natural language interfaces such as Question-Answer (QA) systems, that are normally constrained to responding directly to the previous question, or Chatbots, that only simulate understanding in open domains. Dialogue systems typically follow an architectural system design as in Figure 2.3 with five basic processes or modules and an access interface in a pipeline structure. The basic processes are: input, interpretation, dialogue management, generation and output.

How these modules work and what techniques they apply differ considerably in different frameworks but their basic purposes are the same. The input module detects and processes the user input and sends a textual representation of it to the interpretation module. The interpretation module attempts to extract semantic concepts from the textual representation or parse it into some semantic representation which is then sent to the dialogue manager. The dialogue manager will use the semantic interpretation and knowledge about the current dialogue to decide what action to take. A dialogue manager can also make use of or control external resources and applications such as databases, web services, devices, ontologies or external applications. If the dialogue manager decides to respond to the user in for example spoken language it will send a proposed semantic representation of the next action to the generation module. The generation module will generate an appropriate message from this to send to the output module that will render the message.

A dialogue system can make use of different modalities for input and output such as speech, text, graphics or gestures. Originally, dialogue systems were only text-based. Commercially, it has been *Spoken Dialogue Systems* (SDSs) that have resulted in the widest uptake. SDSs use speech as their input and output modality. In these systems, the access interface to the system can either be over a telephone line or using a headset connected to the audio device on a computer. Speech recognition is used as the input module to recognize the speech input. To render speech output, SDSs make use either of Text-to-Speech (TTS) systems or pre-recorded human speech utterances (so called *voice prompts*). A *multimodal dialogue system* can combine several modalities in parallel or switch between them. Multimodal dialogue systems may therefore need various output and input modules and multiple access interfaces that enables the use of all modalities (for example a touch screen and a headset). In addition, to be able to combine user input from distinct modalities or distribute system output over modalities there is normally a need for additional modules for multimodal fusion and fission (Kruijff-Korbayová *et al.*, 2006). In this thesis the focus is on the spoken modality of dialogue systems although the baseline systems that have been used are actually multimodal (see Section 3.4).

2.2.1 Spoken language understanding

Independent of the input modality, dialogue systems, need to extract the meaning of the user input by interpreting it into some semantic representation. In SDSs this process is normally called *Spoken Language Understanding* (SLU). There are widely differing methods for SLU ranging from detecting keywords to more advanced parsing techniques. There are no standardised semantic representations so the outcome of the techniques applied therefore differ considerably, ranging from attribute-value pairs to logical formula. The differences

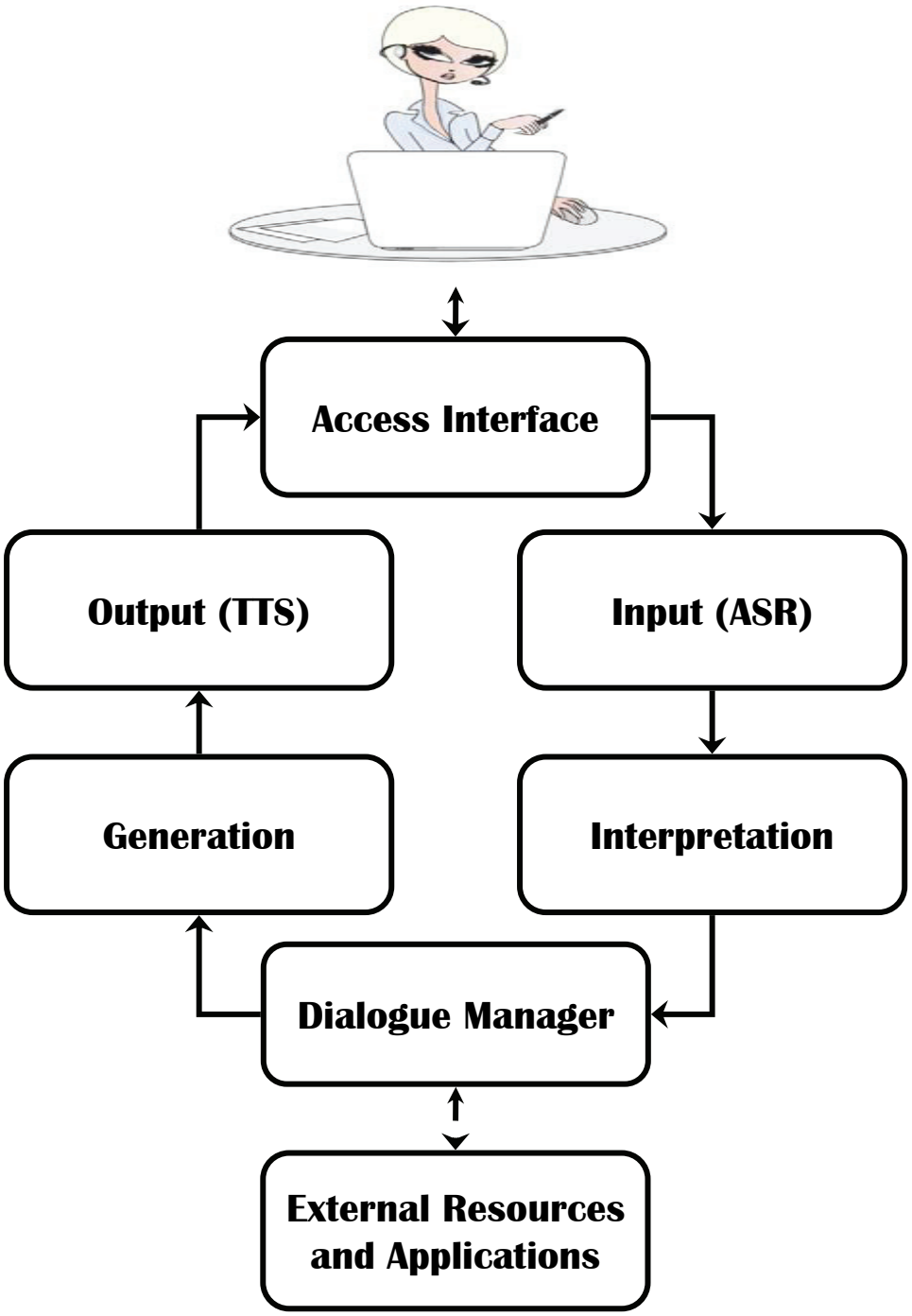

Figure 2.3: A typical dialogue system architecture

in representations also have an impact on the interface between the SLU component and the dialogue manager.

The choice of technique depends on what is needed to be interpreted and what semantic representation is to be achieved. Many SDSs only need to extract the most important semantic concepts from the user input. Consider the user utterance "I want to order broadband for my summer house". In a directed dialogue where the user is asked to choose between two options (e.g. broadband or phone line) it is not necessary to extract a semantic meaning of the utterance but it is sufficient to spot the expected keywords (e.g. broadband). *Keyphrase spotting* techniques interpret utterances based on the existence of some keywords (or keyphrases) in the utterance. In some systems, such as call routing, it is only necessary to classify utterances into tasks or call type to be able to route the call to the right person. We may for example want to distinguish between "orders" and "enquiries". In these systems the interpretation will suffice with the use of some classification technique of utterances into predefined classes. For the previous example such a task classifier would classify the utterance as an order and would not care about the specifiers "broadband" or "summer house".

However, in many dialogue systems it is necessary to identify what subject or task the utterance is referring to (such as "order") as well as extracting information specifying the task (such as "broadband"). A very common approach is to use *frames* with *slot-value pairs*. A semantic frame correspond to a task (or subtask) that represents the issue (subject) of the utterance (e.g. an order). Slot-value pairs are named entities (e.g. product type) to be filled with values (e.g. broadband or phone line). Both data-driven techniques and rule-based approaches have been used to interpret utterances into frames and slots. One rule-based technique is to extend a CFG used for speech recognition with semantics and create a *semantic grammar*. By using a grammar we rely also on syntactic patterns in addition to lexical choices for the semantics. The advantage with semantic CFGs is that no training data is needed and the same grammar can be used both for recognition and interpretation. This saves work and assures that these grammars are in sync. However, just as with SRGs, semantic grammars need to be hand-crafted and require expertise and maintenance. They also suffer from coverage problems. Moreover, the use of semantic grammars with an ordinary CFG parsing does not fit well with the use of SLMs as the interpretation model will be too restricted in comparison with the output from the SLMs.

Existing parsing techniques have primarily been developed for parsing text and to detect and reject ungrammatical input. They are often not suitable for spoken language as they expect features of the input that spoken language does not live up to, such as full sentences and grammatically correct input. The output from a speech recognizer is often fragmentary and ungrammatical due to the nature of spoken language and because information gets lost in the recognition process. Research in SLU has therefore been focussed on more robust parsing techniques which can handle spoken language. Rather than parsing sentences robust semantic parsers try to detect meaningful phrases. A property that robust parsers explore is therefore to relax the constraint of fully parsed sentences and focus on parseable phrase chunks. A common approach is to use semantic grammars together with a robust parsing algorithm such as chart parsing (Ward, 1991; Ward and Issar, 1994). Hybrid

approaches have also been suggested that combine deep parsing with shallow (partial) parsing as a back off when deep parsing fails (Wang *et al.*, 2002b; Noord *et al.*, 1999).

Data-driven approaches to SLU are trained with example sentences with semantic concepts anchored to the words in the sentences. HMMs or FSTs have often been used. They are more robust and work better in conjunction with SLMs than grammar-based approaches. The disadvantage is that they require a large amount of annotated corpus which is seldom available for spoken dialogue domains. To be able to obtain more complex hierarchical semantic structures even more complex annotated data is required. Statistical approaches have therefore had most success in frame or task classification whereas rule-based approaches have been used for slot-value identification or when a more complex semantic representation is needed.

Depending on what type of interpretation is needed simpler methods such as frame and slot-filling may actually be sufficient for many applications. However, if we aim for more advanced human-language interaction we need more sophisticated methods. In more advanced dialogue systems it is necessary not only to capture the task and the information but also the intention of the user. Using ideas going back to speech act theory (e.g. Searle (1969)), utterances in dialogue systems are often classified as *dialogue acts* according to their communicative function. An utterance such as "Hey you!" would for example be classified as a greeting dialogue act. There are several distinct dialogue act taxonomies (Traum, 2000). Dialogue acts are sometimes also called conversation acts, conversational moves or dialogue moves. In this thesis we will use the term *dialogue moves* (see Section 3.3.2). Both rule-based and machine learning techniques have been applied to retrieve the user's intended dialogue move automatically. Different knowledge sources such as lexical, syntactical and prosodic knowledge have been used. Chapter 6 gives an overview of approaches to *dialogue move recognition*. In Chapter 7 a simplistic approach to bootstrap a machine-learnt dialogue move classifier is described.

How to approach spoken language interpretation is determined by the type of application, type of dialogue management technique and the type of representation required. Whatever method is chosen for interpretation we need more research on how interpretation and recognition can be further integrated. Somehow we need to take into account semantics at an earlier stage. Some of the current problems apart from robustness concern portability to new domains or to new semantic frameworks. Finally, as will be discussed in the following section, full semantic interpretation is not possible without more knowledge from the dialogue manager.

2.2.2 Dialogue management

The semantic representation rendered by the interpretation module is used by the dialogue manager to anchor the semantics to the dialogue context. Without dialogue context it is impossible to do anaphora resolution or interpret elliptical utterances. For example when a user says a city name in isolation in a travel domain it is only with the help of the dialogue context that we can assume whether the user is referring to a destination or departure city for a trip. In the same way we need the dialogue context to decide what the user is

rejecting when saying "no" or to what previously mentioned object she is referring when using the word "it".

The main task of a dialogue manager is to decide what action to take next and to keep track of the course of the dialogue. The dialogue manager is the decision-maker that controls the dialogue system's behaviour. To do this a dialogue manager will make use of different knowledge sources such as some representation of the dialogue state and the dialogue context, knowledge of the domain, a task model and a user model. The dialogue manager may also need to call external resources to either get the information needed from a database or webservice or to control an external device or application. A task model consists of the tasks (or plans or frames) that can be performed with the dialogue system in the domain. An example would be the tasks of "booking" and "cancelling" in a travel domain. The tasks are modelled with the pieces of information that are necessary (or useful) to carry out a specific task.

There have been many ways of modelling dialogue and the decision-making of the dialogue manager. The dialogue management techniques used in research have normally been more sophisticated than the ones used commercially. However the trend in commercial systems has been to go towards the more advanced techniques used in research. The most basic dialogue management technique is *finite state based* dialogue modelling. In these systems the dialogue flow is explicitly designed using dialogue states, transitions between these states and conditions to fulfil on making a transition. A task is here modelled explicitly as a set of states and possible transitions and the dialogue context is often limited to being aware of the current dialogue state. Such a methodology has severe limitations if we want to give users the right to initiate dialogue and to allow more flexibility in the order in which transitions to various dialogue states are traversed. To achieve such behaviour, by adding more states and transitions between most states, the graph of states and transitions will become complex and will be hard to get an overview of and maintain.

This simple approach has therefore given way to the *form-based* approach (also frame-based) to dialogue modelling. This approach does not model dialogue states explicitly but models the dialogue context as a data structure to operate on. A dialogue state is here an instance of the dialogue context data structure with values. In the form-based approach we design tasks as forms (or frames) with fields to be filled. These fields can be filled in in any order and the user can even fill in several fields in one dialogue turn. The dialogue manager will ask the user for the missing fields until the form is completed with help of the dialogue context representation. The form-based approach is therefore more adequate to design dialogues with mixed initiative and less constrained order. *VoiceXML* which is a markup language to design spoken dialogue applications uses this form-based approach and has been very successful commercially[1]. VoiceXML systems normally use restrictive SRGs for each dialogue state to simplify the ASR task.

Due to the limited nature of ASR commercial dialogue systems have had to put restrictions on dialogue behaviour and have therefore often made do with the use of the simpler dialogue management strategies such as Finite State and Frame-based as well as simple

[1]A specification of VoiceXML is given at http://www.w3.org/TR/voicexml21/

SLU techniques. These systems therefore often apply a cautious approach to dialogue management with little flexibility. The system normally has the initiative in order to keep control over the dialogue. In this way the user responses are more predictable and more easily modelled in the ASR and SLU modules. In research where the goals are more long term and an optimal end-to-end performance is not of such immediate importance it has been possible to focus on more complex dialogue management strategies. Such systems put less restriction on the dialogue flow to give users more flexibility and allow a mixed initiative dialogue behaviour. However, more freedom for users also leads to a much harder ASR and SLU problem. One thing that distinguishes more sophisticated dialogue management techniques from the finite state or form-based approach is how the dialogue state and dialogue context is modelled and the information about the dialogue is stored. In research systems it is very common not only to model the current dialogue state but also the dialogue history. Many research approaches use a blackboard methodology where all information about the dialogue is stored in a more advanced data structure. Furthermore, in research systems the role of the dialogue manager is normally more prominent with ability to control the other modules.

One research approach to dialogue management is the *information state update* approach (Traum and Larsson, 2003). It goes away from the classical pipeline structure in Figure 2.3 (see page 24) to enable asynchronous dialogue and more complex and flexible processing. In such an architecture the modules can access information in a less restricted manner to achieve a more complex dialogue processing behaviour. In the information state update approach the dialogue context is modelled as a rich formal representation (such as a feature structure or a record) called the *information state*. The information state update approach will be further described in Chapter 3.

2.2.3 Natural language generation

The generation models used in SDSs are often very simple and do not make use of all technologies from the field of natural language generation (NLG). For systems using voice prompts it is for example necessary to know beforehand exactly what the system is going to say at each moment in order to be able to record these utterances in advance. For systems using TTS it is possible to be more flexible and dynamic which can make the system seem less predictable.

The principal task of the generation module is to map a semantic representation generated by the dialogue manager to a textual utterance. The most common way is to use predefined text chunks, so called *templates*. These templates can have slots to fill in with content words or expressions as in the following example:

(3) What time is the *EVENT* on *DATE*?

It is important when it comes to generation not only to model how things will be said but also to design what is going to be said. A SDS needs to present its capabilities correctly. Also the way of posing questions affects the way users will respond to the system. The use of more open questions will lead to a more difficult recognition task. It is therefore

important to model this correctly to avoid recognition traps. For example, the vocabulary used by the system should match the vocabulary expected in the recognition (see also Section 2.4.7).

2.2.4 A brief historical background

The history of SDSs is fairly brief with the first systems not appearing until the end of the 70s. These first SDSs, as for example the voice driven chess system Hearsay (Lesser *et al.*, 1975), were rather spoken language understanding systems than dialogue systems as they did not have any dialogue management. The US government-funded ATIS project (89-95) contributed to a huge leap in the SLU research involving research centres such as CMU, MIT, AT&T and SRI. All participating centres developed SLU systems in the same air travel domain. The main outcome of this project was the development of advanced spoken language parsers such as CMU's Phoenix system (Ward, 1991) or MIT's TINA parser (Seneff, 1992). An additional important result was the amount of spoken language data that was collected and was to be used in future research. Although the emphasis in the project had been on spoken language understanding and not on dialogue many SDSs started to appear in the participating research centres. A precursor was the MINDS system developed at CMU (Young *et al.*, 1989). It was far ahead of its time being the first SDS to exploit higher level knowledge to alleviate the speech recognition process. MIT developed the Voyager system in the same time period (Zue *et al.*, 1991).

The move of focus from SLU to dialogue management came first with large-scale government-funded projects such as the DARPA Communicator project in the US or the SUNDIAL (Peckham, 1991) project in Europe. These projects resulted in common dialogue system architectures and sharing of components which would simplify and reduce the effort of building dialogue systems. In this way many different research prototypes appeared. Most of the systems were related to the travel domain as this had been the focus of the large-scale projects.

This late appearance of more advanced SDSs was partly due to the limits of the ASR and SLU technology. Furthermore, we should not forget that a great deal of the fundamental theories and models of dialogue that underlie the current dialogue management techniques were not formed until the 70s-80s (Sacks *et al.*, 1974; Searle, 1975; Grice, 1975; Perrault *et al.*, 1978; Lewis, 1979; Allwood, 1981; Grosz and Sidner, 1986; Clark and Wilkes-Gibbs, 1986; Clark and Schaefer, 1989). The computational applications of some of these theories thus came surprisingly fast and were applied in systems such as MINDS Young *et al.* (1989), Trains (Allen, 1991), SUNDIAL (Peckham, 1991) and TrindiKit (Larsson and Traum, 2000).

The ASR technology matured sufficiently in the 90s to enable the first SDSs to enter the commercial market. These first systems were primarily a substitute for menu-based DTMF solutions and made used of simple SRGs. Slightly more sophisticated solutions were soon to appear. An early commercialization in Europe was the Philips train timetable system that served customers calling to railway customer services in several European countries (Aust *et al.*, 1995). Since then there has been a commercial explosion of the use of SDSs especially

as part of customer service automatization. Meanwhile there has been intense research on improved dialogue management, better SLU techniques and better language modelling among other things. At the end of the 90s and the beginning of the new millennium several research toolkits for developing dialogue systems and reusable architectures and dialogue managers appeared such as MIT's Galaxy architecture (Seneff *et al.*, 1998), the CMU Communicator system (Rudnicky and Xu, 1999), the CSLU toolkit (McTear, 1999), TrindiKit Toolkit (Larsson and Traum, 2000) and Jaspis (Turunen and Hakulinen, 2003). Several commercial toolkits also started to appear. Although the heaviest influence on the field, at least commercially, was the appearance of VoiceXML which contributed to the widespread development of SDSs. Today, many commercial systems are still built using VoiceXML to some extent. However, many commercial providers are now also offering more advanced toolkits that go beyond the limits of VoiceXML. The exposure to real users and the vast collection of data has given commercial research on spoken dialogue systems an advantage over academic research.

However, academic research prototypes, are finally finding their way out of the labs in order to confront real users and collect valuable data (Raux *et al.*, 2005, 2006). In 2005, the Clarissa system actually reached space when deployed at the International Space Station (Rayner *et al.*, 2005). The encounter with real users in more realistic settings than labs will most certainly inspire further academic research.

There is no doubt that the use of SDSs will soon become an everyday telephone occurrence. However, for this to happen and to be able to provide users with better solutions a great deal of research is needed in order to overcome the current deficiencies. Indeed, many of the current problems arise from the persistent limitations of the ASR systems. For more snapshots from the history of SDSs the reader is referred to Smith and Hipp (1995); Glass (1999); McTear (2004); Jurafsky and Martin (2008); Jokinen (2009).

2.3 Evaluation metrics in ASR

The most widely used evaluation metrics to measure the performance of ASR systems are *perplexity* and *word error rate*. These metrics together with metrics on the semantic level will be introduced in the following sections.

2.3.1 Perplexity

Perplexity is a measure borrowed from information theory based on entropy. For the mathematical foundations behind this measure I point the reader to Manning and Schütze (1999). The speech recognition community normally report perplexity rather than entropy presumably because the figures look nicer and give a better understandability. Perplexity can be seen as a branching factor since at each step a value "k" in perplexity means that the model is guessing between "k" equally likely words. To give an example we consider the following utterance pronounced at the annual Oscar Awards: "And the award goes to...". The branching factor most people in the audience would have after the word "to" would

probably be the number of nominees rather than all the words they know. Although, as there certainly are some favourites to win the award the nominees will not be equiprobable and thereby the perplexity will fall. Consider that some people in the audience would already have been informed of the name in the envelope, i.e. they would be certain about the winner. For those people the perplexity would fall to one as all other nominees would be impossible. However, it could also be the case that the presenter wanted to create even more suspense and continues the utterance with something like: "the eminent actor..." so the real branching factor is probably higher than only considering the number of nominees and their probabilities. Perplexity can in this way be seen as the average surprise factor and is used to measure how well a language model models the expectation of coming words. In studies by Lippmann (1997); Jelinek (1991) it has been shown that humans seem to model this tremendously well. The aim is to attain language models that are good at expecting the next word and only choose from a smaller part of the total vocabulary at each step. A low perplexity is therefore preferable.

We should not forget that perplexity is task dependent, meaning that in some tasks the domain language will inherently have a high perplexity. An example would be the recognition of flight booking reference numbers. In this thesis we will use perplexity to compare models of the same domain language and therefore the differences in perplexity are more important than the actual figures. Unfortunately, perplexity does not take into account acoustic confusability, i.e. how alike words sound (remember the cellphone address book example in Section 2.1.5), which is something ASR must account for. Therefore, low perplexity figures only give an indication that one language model is better than another but we need to look at ASR performance to evaluate the actual benefit of its use.

2.3.2 Word error rate and sentence error rate

In speech recognition it is error rates rather than accuracy rates that are normally presented. On the other hand, in dialogue act (move) recognition (tagging) researchers normally report accuracy rates. I will follow this practice in this thesis.

Word error rate (WER) measures the number of word errors a speech recognizer makes by comparing the speech recognition result with what the user actually said, i.e. the manual transcription of the utterance. WER is calculated by estimating the number of errors in the form of substitutions (SUBS), deletions (DEL) and insertions (INS) of words in comparison to reference words in a transcription of the audio as in Equation 2.1.

$$\boldsymbol{WER} = \frac{\text{SUBS} + \text{DEL} + \text{INS}}{\text{REF WORDS}} \tag{2.1}$$

Most speech recognizers provide the user with evaluation programs that calculate WER using dynamic programming. For the interested reader, I recommend Jurafsky and Martin (2008) for an introduction of the algorithms used. The Word accuracy (WA) can thereafter be obtained following Equation 2.2.

$$\boldsymbol{WA} = \mathbf{100} - \boldsymbol{WER} \tag{2.2}$$

Since WER considers all word errors as equally serious the acoustic or semantic similarity of confused words (substitutions), or the impact of a deletion or insertion are not taken into consideration.

An additional measure is Sentence error rate (SER), which gives a measure of the number of utterances that contain some word error (see Equation 2.3).

$$\boldsymbol{SER} = \frac{\text{Sentences with word errors}}{\text{Total number of sentences}} \tag{2.3}$$

From this we can calculate the Sentence accuracy (SA) using Equation 2.2 but with SER to show how many utterances the ASR recognized completely correct. By taking into consideration N-Best lists it is possible to calculate the lowest possible WER or SER from those lists by selecting the hypotheses in the lists that minimize the error rate. The result is called the *N-Best error rate*. This error rate gives an estimation of the possibility of improvement by considering whole N-Best lists.

What factors may then lead to a high WER or SER? First of all, test conditions affect ASR performance heavily as ASR systems are very sensitive to noise and channel distortions. Also, performance varies dependent on speakers. Some speakers will get much lower WER than others for the same task. Speech recognizers seem to work better for women than for men, better for adults than for children or elderly people and better for native than for non-native speakers (Adda-Decker and Lamel, 2005; Goldwater *et al.*, 2008; Raux *et al.*, 2005). Also, experienced users seem to be able to perform better than naive users (Knight *et al.*, 2001). Another factor is the speaking style, where more spontaneous and less clear speech (hypospeech) is harder to recognize. A study by Goldwater *et al.* (2008) shows that factors that increase the error rate of a single word is its acoustic resemblance to other words (homophony), its prosodic values, its situation in the sentence, its position in relation to disfluencies, its speech rate, its frequency rate and its word class. Open class words without any extreme prosodic values, without homophones, with a normal speech rate, highly frequent and not positioned close to disfluencies were found to be more easily recognized. WER also depends on the complexity of the recognition task, the size of the vocabulary and the quality of the language model. Therefore, it is hard to say what a good WER is. WER figures reported for different SDSs will apparently differ. However, reported WERs for SDSs normally lie between 15-30% (Riccardi *et al.*, 1998; Xu and Rudnicky, 2000b; Hazen *et al.*, 2002; Moore, 2003; Bangalore and Johnston, 2004; Bohus and Rudnicky, 2005b) (sometimes more if non-native speakers are included).

In general, ASR works better the closer the conditions and the speech are to the original training situation. The recognizer will more easily recognize words and sentences that have been seen more often in the training data. For SRGs, the more utterances that follow the grammar the better the result. Furthermore, an ASR system will more easily recognize speakers that speak in a similar manner to the speakers in the training data (style, dialect, voice). In addition, if the recording environment is more alike the recordings of the training data it will work better.

In work in this thesis, users have been both experienced and naive users, adult speakers, native and non-native. Recordings have been conducted outside the laboratory with some

background noise. The speaking style has been rather spontaneous. This has resulted in less controlled data than for laboratory experiments. What we want to investigate is how different methods perform in these conditions. Taking these models into an even more realistic setting may lead to a degradation of the results.

2.3.3 Word error rate vs concept error rate

In the context of dialogue systems we need a way to measure that we do not only lower the rate of incorrectly recognized words but that we also manage to interpret more of the user's message with the help of the higher number of correctly recognized words. There are various ways of measuring ASR errors that lead to incorrect semantic interpretations and different names have been used: Semantic Error Rate, Understanding Error Rate, Slot Error Rate, Concept String Accuracy and Concept Error Rate. Concept Error Rate (CER) was proposed by Boros *et al.* (1996) and measures the number of incorrectly recognized concepts using the same formula as for WER (see Equation 2.1) by replacing words for concepts (semantic units). In this way CER measures, as Boros *et al.* (1996) define it, "the degree of system understanding". However, some researchers instead present figures on the total number of incorrectly interpreted sentences in analogy with SER. This has sometimes been called Semantic Error Rate (Rayner *et al.*, 2006) and in other cases misleadingly Concept Error Rate.

An interesting issue is whether there is a correlation between WER and the semantic metric. If we manage to lower WER can we assume that the error rate of our semantic metric also will fall and that the semantic understanding of our system will thereby improve? So far, this correlation has not been proved which makes reporting a semantic metric important. Researchers have not even agreed on a general metric which makes it hard to compare studies and draw any conclusions. According to Wang *et al.* (2003) WER often correlates badly with semantic metrics while results from other studies seem to indicate a correlation (Chotimongkol and Rudnicky, 2001; Boros *et al.*, 1996; Boye *et al.*, 2006). This depends firstly on the type of semantic metric used, i.e. if we measure correctly interpreted utterances or the degree of correctness. As Rayner *et al.* (2006) point out, it also depends on the type and complexity of the semantic representation, e.g. if determiners or articles are taken into consideration or only concept words. In this thesis we will introduce a new semantic metric bound to our semantic representation to investigate how the WER affects understanding. If we can find a correlation between this metric and WER and/or SER it would mean that our efforts at improving speech recognition would be valuable also for improving speech understanding.

2.3.4 Dialogue move error rate

We will use two different metrics to evaluate system understanding. The first one follows the Semantic Error Rate (SemErr) approach by calculating exact semantic matches. In our framework where utterances are interpreted as dialogue move sequences (see Chapter

3) we will then compare whether two dialogue move sequences are exactly the same or not. We will call this metric *Dialogue Move Sequence Error Rate* (DMSeqER).

However, the Semantic Error Rate (SemErr) measure does not give any indication whether the joint speech recognizer and understanding model is mostly correct or wildly wrong whenever an ASR hypothesis cannot be identically interpreted as the transcription. I have therefore chosen to also follow the original proposal of CER (as in Boros *et al.* (1996)) where the degree of correctly recognized concepts is considered. In our case we measure the degree of correctly recognized dialogue moves, the *Dialogue Move Error Rate* (DMER). Example(4) shows an example of how a user utterance has been partially misrecognized and how only some of these errors have propagated to the dialogue move interpretation.

(4) USR> Yes I want to add a meeting on Thursday

USR DM: answer(yes) + request(add_event) + answer(event(meeting)) + answer(date(Thursday))

ASR HYP> Yeah I want to add meeting on Tuesday

DM HYP: answer(yes) + request(add_event) + answer(event(meeting)) + answer(date(Tuesday))

In this example, we have a WER of 33% with two substitutions and one deletion. However the substitution of the word "Yeah" for "Yes" is not a conceptual error as both can be interpreted as the same dialogue move (answer(yes)). Neither does the deletion of the determiner "a" affect the dialogue move interpretation. However, the misrecognition of the weekday "Thursday" will actually lead to a dialogue move interpretation error. Although the dialogue move is the correct one the value of the date differs. This means that one out of four dialogue moves is wrongly interpreted. The DMER for this example is thereby 25%, whereas the DMSeqER is 100%. In this way we get a figure on the degree of system understanding and can evaluate if the recognizer misrecognizes parts relevant for understanding. In the context of dialogue systems an improvement of ASR performance is only interesting if we can show that the better performance also will propagate to the understanding model, i.e. that the system improves its ability to recognize content words.

2.3.5 Significance testing

As WER is influenced by so many factors and dependent on the test data, the task and the ASR system in use there is no real use of reporting a single WER result. Instead what is normally reported are comparisons of performance on the same task for different techniques or models. Error rates are normally compared against a *baseline* which is the starting point of performance setting the lower level. For some tasks a 100% accuracy is not possible. In these cases it is common to estimate an *Oracle* rate which is the maximally achievable performance using an ideal technique, i.e. an *Oracle method*. An achieved lower WER for one technique over another is normally reported as a relative improvement rather than absolute difference in WER. This means that if we achieve 10% WER with

our new technique whereas the baseline only achieved 20% WER we will report a relative improvement of 50% (a 10% absolute difference).

A better ASR result gives an indication that there has been an improvement although it does not justify the conclusion that it occurred because of the use of a better method as the better result could well have been achieved by mere coincidence. We therefore need to provide figures on the significance of the results to exclude the role of chance. Depending on the size of the evaluation test set smaller or bigger percentage differences are needed for significance. In ASR normally small differences are reported many times for small evaluation test sets which make significance testing important. There are several different significance metrics which put more or less hard constraints on significance. The two most widely used in the speech community are the McNemar test and the χ^2 (Chi-square) test. As stated in Grönqvist (2006) the McNemar test only compares the cases where two systems differ which makes it easier to prove significance on small data sets. However, to use the McNemar test we need paired observations, i.e. we need to compare only the ASR hypotheses where the two systems differ. On the other hand the χ^2 test calculates significance on the whole data set. This makes it harder to prove significance, i.e. prove that one system is better than another. As paired observations from the systems we compare will not always be available I have chosen to use the χ^2 test throughout this thesis to calculate the significance of the improvements. Significance will be reported on the following levels (according to Manning and Schütze (1999)):

Table 2.1: Critical significance levels

Degree of freedom	Significance levels				
1	0.10	0.05	0.01	0.005	0.001
1	2.71	3.84	6.63	7.88	10.83

Normally, a significance on the $p<0.05$ level is necessary for a result to be considered significant. We will henceforth report results where $p>0.05$ as *not significant*.

2.4 Survey of attempts to compensate for ASR deficiencies in spoken dialogue systems

Speech recognition performance has steadily improved over the past twenty years or so. The main reason is presumably more computing power and more computing storage for more training data. Improvements of the ASR technology have also been made with improved feature extraction techniques, new noise robustness techniques, improved acoustic modelling and improved language modelling. Still, as discussed in the introduction, humans outperform ASR on all types of recognition tasks. Most difference in performance is found when it comes to spoken dialogue and noisy environments. On the Switchboard corpus (Godfrey and Holliman, 1997), which is a spoken dialogue corpus, Lippmann (1997) reports a 43% WER for ASR but only 4% WER for human subjects. Today, the best ASR error rate

for the Switchboard corpus is probably a bit lower but still orders of magnitude higher than the reported human error rate above. This comparison is not truly fair as the vocabulary size and the knowledge sources used are not comparable for humans and machines. Humans can make use of higher level knowledge such as meanings of words, situational context or dialogue context. Even when these sources are not involved, however, such as in the recognition of non-sense words and non-sense sentences, human performance is unrivalled (Lippmann, 1997). This indicates that humans use other acoustic cues than the ones that are used in ASR. Humans can use all the information of the acoustic signal while ASR is restricted to the features we extract in the DSP phase. Therefore, there seem to be some important cues missing even in the feature extraction step. The superiority of human performance in comparison to ASR indicates that there is much room for improvement on different levels.

Most of today's ASR systems are statistically built using HMMs and SLMs. This predominant technology has some inherent limitations as it does not imitate human spoken processing too well, and makes use of minimal knowledge of language and speech and almost no knowledge of dialogue (Hermansky, 1998; Moore, 2007; Bilmes, 2004). However, according to a survey carried out by Moore (2005) the speech community does not expect any superior model in the near future but predictions are that HMMs and SLMs will continue to be dominate. Alternative approaches have been proposed but are still more on a theoretical level (Moore, 2007; Bilmes, 2003). In the meantime, researchers endeavour to compensate for the deficiencies in ASR by incorporating knowledge about human spoken processing and language into the current statistical framework. The following sections consist of a survey of such ideas, techniques and approaches that have been used by researchers on different levels in the ASR framework in an attempt to enhance ASR. Although, some of the techniques and problems are equally applicable for any speech recognition task the perspective of the survey is for speaker independent ASR for SDSs. The survey will also focus on the problems arising when using ASR in dialogue systems.

2.4.1 Improving the front end

Dictation systems, in contrast to speech recognizers for dialogue systems, are sold with a microphone, usually include some guidance of how to use the system and also require a training procedure with the user. Guiding users on how to give speech input to a computer can save many misrecognitions. A training procedure to adapt to a speaker's voice leads to important recognition performance improvement. In most speech solutions for customer service systems, where any customer should be able to call, it is hard to include any training procedure and adaptation as many of the callers will not use the system again. Also, such a procedure would require some sort of speaker identification for successive calls to be able to reuse adapted models. In addition, as customers expect an immediate service they are probably reluctant to lose time training the system or listening to instructions on how to best give input to ASR.

Speech recognizers for dialogue systems get their input in various ways, depending on the application. Most commercial systems get their input via a telephone line. In these

systems, the speech recognizer must take into account the distortion of the signal in the telephone line, the limited bandwidth and be prepared to receive signals both from the terrestrial network, the mobile network and more recently also IP telephony. This limits the recognition performance in contrast to dictation systems with a direct input line from a headset.

2.4.1.1 Robustness to noise

When taking SDSs into the wild, from a laboratory setting to real environment conditions, performance is reported to degrade heavily. Raux *et al.* (2005) report a WER increase from 17% (43% for non-native) to 68% for their SDS "Let's Go Public" when comparing its laboratory performance with its real performance. Hazen *et al.* (2002) report a WER of 10% for clean speech (speech without noise, disfluencies, OOVs etc.) in comparison to a 50% WER on the remaining data. One of the reasons why this happens is that ASR is not very robust to noise and channel changes (Lippmann, 1997). Dictation is normally performed on a desktop with a headset in a more or less quiet and stable acoustic environment. SDSs normally get input from more noisy and less stable environments. In the Let's Go Public system incoming calls were made from diverse conditions such as cell phones while on the street, noisy rooms as well as more quiet indoor places (Raux *et al.*, 2005). ASR is very sensitive to adverse environments and performance degrades considerably. Humans on the other hand, easily adapt to unexpected conditions. We identify two kinds of noise: stationary and non-stationary. Stationary noise, such as a computer fan or a car engine is easier to model than non-stationary noise such as a door slam or a mobile ringing. The attempts to make ASR less sensitive to noise, so called *noise robustness techniques*, constitute an intensive research area. Huang *et al.* (2001) gives a good overview of noise robustness techniques in ASR with primary focus on robustness to stationary noise. Noise robustness techniques include everything from coping with noise by developing better microphones, handling echo-cancelling, finding noise resistant signal features in DSP, subtracting noise from the speech signal in the DSP, to adapting acoustic models to different noise conditions. Three principal strategies for a speech recognizer to handle noise can be identified:

1. Ensure that the same acoustic model performs similarly in different conditions by using noise resistant acoustical features

2. Transform the incoming speech by removing noise while maintaining the same acoustic model

3. Transform the acoustic model to the noisy condition

Noise conditions are measured in ASR with a speech-to-noise-ratio (SNR). SNR measures how intelligible speech is expected to be by comparing the speech and the noise level (in dB). The lower the SNR (the less difference between both levels) the higher the expected error rate. Lippmann (1997) reports a study where human subjects and ASR were exposed

to speech with additive automobile noise for different SNRs including speech in almost quiet conditions. It was shown that human performance held steady, independent of the SNR, while ASR performed not only much worse overall and the performance degraded heavily with more noise (lower SNR). With the use of noise adaptation techniques the ASR performance improved considerably but was still ten times higher than for the human subjects. Lippmann (1997) reports several similar studies showing that human perception is minimally affected by noise and channel conditions. Even non-native listeners, although performing perceptually worse than natives under all conditions, seem not to decrease heavily in degraded conditions. The difference in performance between natives and non-native are thereby held constant for different SNR (Bradlow and Bent, 2002; Trimmis *et al.*, 2007). The robustness of human perception shows that there are again some important cues missing in ASR that humans make use of. For the moment, ASR continues to be very sensitive to noise although noise robustness techniques have made ASR a bit more solid.

2.4.1.2 Crosstalk and disfluencies

A specifically hard task is when the background noise is other people talking, i.e. speech, or even worse when the speaker suddenly does not address the system but someone else, so called *crosstalk*. Raux *et al.* (2005) report crosstalk as one of the real environmental factors that affect their Let's Go Public system. Ideally, crosstalk should be ignored by an ASR system. In reality, ASR systems often try to recognize crosstalk. When confidence scoring works well such input will be scored low and can be rejected by the dialogue system using the same approach as for misrecognitions. However, it is hard to apply different strategies in a dialogue system for crosstalk and misrecognitions such as ignoring the former and rejecting the latter using only confidence scores. Renders *et al.* (2005) addressed this issue showing that machine learning (support vector machines (SVMs)) is more suitable for this classification task. The problem of identifying crosstalk was also approached in Gabsdil and Lemon (2004). A classifier was trained using machine learning with both acoustic and pragmatic features to classify whether user utterances should be accepted, rejected or ignored. Crosstalk was supposed to be ignored. The classifier obtained 70% classification accuracy of crosstalk. Such an approach is interesting. Apart from trying to cope with bad conditions we also need to detect bad signal quality or detect bad conditions. If signal quality is poor we could inform the user about this (or for crosstalk ignore it) just as we do in human human telephone conversations instead of trying to recognize poor audio. We will address the issue of crosstalk detection in Chapter 8.

Non-stationary noise in speech is frequently non-verbal vocalizations such as smacking of the lips, laughter, breathing, and throat clearing and also disfluencies such as filled pauses (e.g. "uhm"). These sounds, rather than disturbing the speech signal, normally match a random word in the ASR vocabulary leading to misrecognitions. A common approach to the handling of filled pauses has therefore often been on the acoustical and language modelling level by considering them as words and modelling them explicitly (Ward, 1991). The problem is to train the occurrences of these statistically as they are not as context-restricted as words. On the other hand, they do not seem to appear randomly just anywhere

either. According to a study by Eklund and Shriberg (1998) filled pauses are, for example, more common sentence initially than when a sentence has begun. Stolcke and Shriberg (1996) show how filled pauses actually are a good predictor of following words. When it comes to disfluencies in general, such as filled pauses, repetitions, repairs, fragments and false starts, attempts have also focussed on detecting these to extract the disfluent parts to simplify subsequent processing. Such classifiers make a decision on each word boundary whether it is disfluent or not. In Stolcke and Shriberg (1996) a disfluency language model (aka cleanup model) was used to detect disfluencies and then edit the affected phrase. In this way they could predict following words based on an edited fluent previous context. Benefit was positive but scarce. Goldwater *et al.* (2008) report that disfluencies heavily impact error rates. They also show that the type of a disfluency and its position affect recognition of surrounding words in different manners. For instance, non-final repetitions and words next to fragments are more affected than words following or preceding repetitions (Goldwater *et al.*, 2008). Coping with crosstalk, non-verbal vocalization and disfluencies continues to be a big challenge in ASR and especially in SDS where they are more common.

2.4.2 Improving digital signal processing

On the digital signal processing level, knowledge about human auditory perception has improved ASR considerably and is currently used to greater or lesser extents in different ASR systems. The assumption is that as speech is intended for human hearing, which is limited, we should not consider what humans do not hear but focus on the perceived parts. Unfortunately, we do not have a deep insight into human auditory processing. As perception is an internal human process it is hard to study and the knowledge that researchers have, has been drawn from experiments with human subjects.

2.4.2.1 Inspiration from human auditory perception

In ASR we do not need to expect sounds that are impossible for humans to articulate. In the same way, it is not necessary to handle sounds that are impossible for humans to perceive. The most successful way of using knowledge about human auditory perception in ASR has been by using techniques that are inspired by the non-linear human perception of frequency bands as explained in Section 2.1.1. In Cepstral Analysis, features are transferred to a Mel scale which corresponds better with the human perception of frequencies. Mel frequency cepstral coefficients (MFCC) is nowadays the most common feature representation in ASR.

Perceptual linear prediction (PLP) is a feature extraction technique with a more direct relation to human hearing with non-linear frequency scale, equal loudness curve etc. (Hermansky, 1998). This has been shown to give a system more robustness to noise and channel distortions as well as speaker differences (e.g. adult vs child speech). In this way, PLP has been a way to obtain acoustical features that are more resistant to variation.

Experiments indicate that human perception does not portray the signal in fine detail (Hermansky, 1998). Feature extraction may consequently suffice by focusing on the coarse picture. As Hermansky (1998) points out the problem is not only that there may be

information missing from the use of current feature extraction techniques but there is also superfluous information due to the original purpose of current techniques: speech coding. We do not need to extract features that do not help in the decoding of the linguistic message. Hermansky (1998) argues that future feature extraction techniques should only focus on extracting features relevant for word recognition and disregard non-linguistic or speaker dependent features. Work on new feature techniques better suited for the purpose seems encouraging and will hopefully contribute to better and more robust ASR performance.

2.4.2.2 Auditory perception and articulatory production

From experiments on human subjects it has been shown that human auditory perception and human articulatory production seem to be somehow intertwined. For example, Wilson *et al.* (2004) showed how the articulatory part of the brain is activated when listening to speech. Another example comes from listening experiments with speech synthesis where subjects feel a sensation of exhaustion when TTS systems speak faster than humans are able to do due to breathing constraints.

As discussed in the introduction, human articulatory production seems to focus on phonetic contrasts rather than absolute phonetic targets. Perception seems to have been adapted to the limitations of the speech apparatus and to focus on contrasts. There seems to be a trade-off between production and perception where speakers try to minimize the energy consumption while maintaining the spoken signal perceptual distinctive on demands of the listener. This leads to the broad variation of acoustic realizations of speech sounds.

As explained in Section 2.4.3 ASR is based on the concept of phones where the task is recognizing phones (and later words) based on observed acoustical features. These acoustical features may vary considerably for the same phone. Currently, no consideration is taken of the production of the sound but only its acoustic result. Spectrograms are visual representations of acoustic signals used in phonetics to study speech. Human spectrogram readers do not match the spectral information in spectrograms directly to phones but rather identify certain patterns as belonging to certain articulatory features. Conclusions about phones are drawn based on that information together with contextual information. Identifying a segment as a vowel is for example much easier than identifying which vowel it is. This intermediate level of classification is not used in ASR. A recent interest of incorporating speech production knowledge in ASR by the use of articulatory features has shown encouraging results (Frankel *et al.*, 2007; Livescu *et al.*, 2003; Soltau *et al.*, 2002). In these studies, articulatory features, such as nasality or place of articulation, are recognized conjointly with phones. The first preliminary experiments have led to improved recognition performance and indicate the possibility of integrating such knowledge into the statistical framework.

In speech technology, ASR and TTS are two distinct fields with quite different approaches to the treatment of speech. Recently, speech recognition technologies have started to be applied in TTS which have had an important impact on the TTS field (Ostendorf and Bulyko, 2002). Still there have been fewer attempts in the opposite direction: to

incorporate knowledge from the TTS field into ASR. TTS techniques such as grapheme to phoneme models for automatic pronunciation lexicons have been used as well as text normalization techniques on corpora to be used for SLMs. Still, the handling of speech is very different in these two areas.

In a similar manner, research in human speech recognition (HSR) and automatic speech recognition (ASR) have approached speech with quite different aims and techniques. As Scharenborg (2007); Moore and Cutler (2001) point out a crossfertilisation of human and automatic speech recognition research is needed to improve both fields. Techniques from ASR can be used to learn more about HSR and the knowledge achieved can perhaps be used to improve ASR. However, to be able to apply theories to current ASR systems the techniques and representations need to be compatible with the HMM framework which makes application less direct. As Hermansky (1998) points out any imitation of human processing will not necessarily be successful. We need to discover the properties of human processing that are relevant for processing speech to a linguistic message and therefore likely to improve ASR performance.

2.4.2.3 Prosody: a missing cue?

Human perception seems likely to be cue-based just like ASR. This means that different cues on different levels are used to map a sound signal to a word sequence (and a meaning). However, humans use many more cues than current ASR systems do. One important cue that is not used in ASR is prosody. In contrast to the written media, in speech we have the possibility of using prosody to group words together, put stress on certain words or to indicate the function of a whole phrase by intonation. We also use prosody for word recognition to distinguish one word from another. In addition, prosody is used to convey emotions and attitudes. Prosodic features are currently not extracted during DSP for ASR. In spite of that, several researchers have shown the importance of the use of prosody for SDSs for several tasks. In these studies, prosodic features have been extracted independently of the ASR system. Prosodic cues have been shown to contribute in the prediction of recognition errors (Litman *et al.*, 2000; Hirschberg *et al.*, 2000) and user emotions (Ang *et al.*, 2002). Extensive research of the role of prosody for classification of dialogue turns into dialogue acts has been carried out (Shriberg *et al.*, 1998; Venkataraman *et al.*, 2003; Rangarajan *et al.*, 2007; Taylor *et al.*, 1998). In these studies, prosodic cues have been shown to be of significant help for dialogue act classification. Consider the use of the phrase "Okay" in different intonational contexts: "Okay!", "Okay.", "Okay?". To distinguish whether such a phrase is a statement or a question, a backchannel or an acknowledgement, would be impossible without the use of prosody as a cue. Prosody has also been shown to be useful for tasks such as sentence segmentation, topic segmentation and disfluency detection (Shriberg *et al.*, 2000; Liu, 2003). Still, efforts on showing the benefit of prosody for word recognition *per se* have been scarce (Shriberg and Stolcke, 2004; Ostendorf *et al.*, 2003). Whether this indicates that prosody will not result in a helpful cue in ASR or rather depends on unsuccessful attempts either of representing prosodic features or incorporating them into the statistical framework is hard to say. At least in

human-human spoken dialogue prosody plays an important role.

2.4.3 Improving acoustic modelling

In contrast to dictation systems, where the user trains the system to her voice, ASR for dialogue systems is often speaker independent and does not even take into account that it is the same speaker during the whole dialogue. The study by (Lippmann, 1997) shows how humans adapt their perception to the speaker, channel and speaking style using only short speech segments. So what we would want for recognizers are somehow dynamically adaptive acoustic models.

2.4.3.1 Adapting to the speaker

We have seen that speaker variation is one of the factors that complicates the recognition task. Recognition performance differs tremendously between different speakers. For some speakers, ASR just does not work properly whereas for others it works reasonably well. It is common to talk about speakers as "sheep" and "goats" where sheep are the good ones and goats the ones who perform badly (Doddington *et al.*, 1998). A desirable strategy for SDSs would be to take into account that it is the same speaker during the whole interaction and be able to adapt to the user. However, speech recognizers usually consider each utterance as an utterance from a new speaker. This is an issue that Young (1996) brings up and he points out that this could reduce error rates considerably, especially for atypical speakers (i.e. goats).

In some applications it is actually possible and necessary to know the identity of the user by telephone number recognition or speaker verification. In this case we should not neglect the enrollment techniques used in dictation systems but make use of user-adapted acoustic models. The two baseline systems in this thesis could well be used by one single person on their laptop and it would therefore be possible to train the speech recognizer on their voice. This would most certainly lead to improved recognition performance.

However, the focus of this thesis is speaker independent systems. The training procedure for dictation is often minutes long, the longer the better, and the system knows what the user is reading (supervised adaptation). For speaker independent dialogue systems we would need much faster, unsupervised adaptation techniques that permit recognition at the same time as adaptation. Researchers have therefore investigated adaptation techniques that can adapt the acoustic modelling to a speaker with only short speech segments. As reported in Lippmann (1997), humans seem to adapt to a new speaker after hearing only three syllables. The current state of the art, in automatic adaptation techniques, still needs as much as 10 seconds of speech. What these techniques adjust are the GMMs (see page 16) by adapting the mean values in the GMMs to the speaker's voice. Unfortunately, the time this adaptation currently takes is too long for many commercial dialogue systems where perhaps the total time for the user turns are expected to be around 10 seconds of speech (around three sentences).

Another strategy is to identify speakers as belonging to certain speaker groups and apply acoustic models trained on this speaker group to the speaker. A common binary grouping is by gender as female and male voices are quite distinct. The system detects gender and applies gender specific models. This has proved to lower error rates by 10% (Huang *et al.*, 2001) and is commonly used in ASR systems.

A different approach is *speaker normalization*, where you transform the incoming speech to a more normalized way with generic features. A common normalization is vocal tract length normalization (VTLN). Speakers with long vocal tracts normally produce much lower formants. This information can be used in the recognition process to detect vocal tract length and then normalize the incoming speech feature values before recognition starts.

2.4.3.2 Speaking style

We have already mentioned that the recognition performance degrades heavily in SDSs in real situations. As discussed, one of the reasons is background noise and disturbance to the acoustic signal. However, if we take the Let's Go Public system as example again, it was reported that even when taking away utterances with crosstalk and background noise the WER was still 60% as compared to 17% in laboratory settings (Raux *et al.*, 2005). This indicates that there must be other factors involved.

In an experimental study presented in Weintraub *et al.* (1996), recognition of spontaneous dialogue was compared to read speech. The channel, the speakers and the words spoken were held invariant by having subjects interacting spontaneously and then ask them in a subsequent experiment to read the transcriptions of their own spontaneous utterances. The difference in recognition performance for these two tasks was huge with a 53% WER for the spontaneous utterances whereas only 29% when read. Even, when subjects were asked in a third task to read in a conversational style the error rate was much lower (38%) than for the spontaneous task. This indicates that there must be something about the way of speaking in dialogue that complicates the recognition task. As discussed in Chapter 1, speakers vary in speaking style depending on the task, the situation and the acoustic environment. Lindblom (1990) defines speaking style as going from hyperspeech to hypospeech. Spontaneous dialogue would be more on the range towards hypospeech whereas read speech would be closer to hyperspeech. Hypospeech is characterised by a high speaking rate, less careful speech that leads to more reductions and more coarticulation. This makes the pronunciations of words more diverse. It has been shown that especially highly predictable words vary in their pronunciations as there is no real need for the speaker to articulate these well. In the Switchboard corpus, it was found that the pronunciation of words was extremely varied. There were for instance 100 different ways of pronouncing the word "the" (Godfrey *et al.*, 1992). One approach would be to add pronunciation variants to the HMM lexicon. In most ASR systems the developer can add and modify pronunciations of words. However, adding more pronunciations may lead to more ambiguity and a deterioration of the search process (Soltau and Waibel, 2000).

It seems that the spoken data used to train acoustic models is sometimes quite distinct

from the way people will speak in a real spoken human-machine dialogue. For example, it is quite common in spoken dialogue with machines that people get frustrated when things are not working correctly, especially when they are not understood by the system. What people often do when miscommunication occurs is to hyperarticulate, speak louder or show frustration. Raux *et al.* (2005) report figures for their Let's Go Public system with 10% hyperarticulations, 11% loud utterances and 11.5% frustrations. Although, the intention of speakers when hyperarticulating is to recover from previous recognition errors the result is often the contrary. As acoustic models have not been trained on this type of spoken data the speech recognition performance can actually get worse and worse when a speaker uses common human-to-human strategies like these for error recovery.

In Section 2.4.1.1 we discussed noise robustness techniques. These have as assumption that noise is independent of speech, i.e. if we can subtract the noise from the speech the recognition will work better. In reality, speakers actually adapt to noisy situations and change their speaking behaviour. This is the so called *Lombard Effect* where in an attempt to compensate for the noisy condition speakers speak more clearly and loudly, i.e. they go towards a speaking style close to hyperarticulation. Hyperarticulated speech can therefore occur in SDSs both in miscommunication situations and in noisy conditions. Hyperarticulated speech is characterized by a slower speaking rate, longer phone durations, more careful speech, fewer disfluencies, more pauses and differences in pitch contour and fundamental frequency (Soltau and Waibel, 2000; Oviatt *et al.*, 1998). For humans, hyperarticulated speech is more intelligible than normal speech whereas ASR systems degrade in recognition performance (Oviatt *et al.*, 1998). Interestingly, a study by Bradlow and Bent (2002), shows that the effect of clear speech in noisy conditions is less beneficial for non-native listeners than for native listeners. The reason why non-native listeners do not profit very much from this intended help from the speaker is unclear.

To handle hyperarticulated speech in ASR we would either need to train models that can handle or adapt to both normal and hyperarticulated speech or train separate models for each speaking style. The latter approach was chosen by Soltau and Waibel (2000) who trained acoustic models on hyperarticulated German speech. Performance improved significantly for hyperarticulated speech when using these specific models (23% error reduction). However, to be able to use acoustic models for hyperarticulated speech and other models for normal speech we need to predict hyperarticulated speech. Soltau and Waibel (2000) also investigated some cues that could be used for detecting hyperarticulation such as phone duration. Although the selection criterion used was not perfectly accurate it was good enough to give an overall improvement in recognition performance when choosing which model to use automatically. In Soltau *et al.* (2002) a different approach to compensate for hyperarticulated speech was investigated by the incorporation of articulatory features (see Section 2.4.2.2). This led to an important error reduction (25%) and the use of articulatory features also improved the recognition of normal speech.

In other studies the focus has been on predicting error-prone situations or detect situations where the user is in a delicate situation. Levow (1998) trained a classifier using decision trees to predict error correcting utterances. These are utterances where the user is trying to correct previous recognition errors. This approach achieved a classification accu-

racy of 77% when applying prosodic features. The most valuable predictors were shown to be duration measures. Hirschberg *et al.* (2000) predicted recognition errors using acoustic confidence scores in combination with prosodic features with 89% accuracy. In Ang *et al.* (2002) prosodic features were used together with language model features to detect annoyance and frustration in users. This is of interest in automatic customer service systems in order to be able to transfer annoyed or frustrated users directly to a human operator. Again, prosodic features were shown to contribute to a better performance. Longer durations and slower speaking rates were shown to be associated with frustration. Rotaru and Litman (2005) showed that the use of pitch features on the word level rather than on the turn level can improve the detection of emotional utterances in English spoken dialogues. This indicates that utterances are not equally affected by speaking style change or emotions but some words seem to be more affected than others.

2.4.4 Improving language modelling

Jelinek (1991) pointed out years ago that after decades of progress in speech recognition SLMs, or specifically trigrams, were still much the same. Although the weaknesses of the trigram models were known, improvements on them had come up short. As discussed in the introduction (Chapter 1), Jelinek was not alone in proposing the search for more sophisticated language modelling techniques. Brill *et al.* (1998); Moore (1999); Glass (1999); Rosenfeld (2000a) all proposed the use of more linguistic knowledge in SLMs.

Attempts at alternative ways of modelling language other than with SLMs have been scarce. One of the few alternatives uses *artificial neural nets* (ANN) to build language models (Xu and Rudnicky, 2000a). Xu and Rudnicky's experiments show that ANNs can learn to model language with comparable performance to SLMs but with a much higher computational cost. A more successful but limited alternative in SDSs have been the use of SRGs for smaller tasks and in cases where training data has not been accessible. Today, almost two decades after the publication of "Up from Trigrams" (Jelinek, 1991), we are still mostly using trigrams. According to a survey among speech scientists by Moore (2005) SLMs are expected to persist.

Statistical language modelling and human language modelling actually seem to have some things in common. SLMs are built on word frequencies just as human lexical access seems to be based on frequency. Similarly to SLMs, humans seem to make predictions of coming words based on previous words. In speech, humans even shorten more predictable words (reductions) while putting longer duration on infrequent words. Both humans and SLMs also seem to process multiple words in parallel (Jurafsky and Martin, 2008). However, statistical language modelling suffers from several problems:

1. data sparseness
2. restriction to very local word context
3. difficulty of adding or detecting new vocabulary

4. difficulty of adapting to different language contexts
5. static frequencies which do not rely on the bigger context

Humans on the other hand:

1. are "trained" on much more data (Moore, 2003)
2. use larger and more complex contexts
3. easily learn and detect new words
4. easily adapt to any topic
5. adapt word frequencies to context and topics

Comparison of humans' language modelling capabilities to the performance of trigram SLMs has shown that humans outclass SLMs (Lippmann, 1997) and can easily improve on output from speech recognizers (Brill *et al.*, 1998). Our intuition tells us that SLMs are too simplistic and too unstructured in comparison to the way humans seem to model language. Humans have the ability to make use of many more cues on a higher level such as grammatical, semantic and pragmatic knowledge as well as world knowledge and common sense. Therefore, we need to find better ways of producing models with much lower perplexities, that are more structured and that make use of more knowledge to be able to achieve a comparable performance to humans. In the following sections, I will describe some attempts by researchers to approach the problems of language modelling in ASR, listed above.

2.4.4.1 Data sparseness

SLMs suffer from data sparseness when there is not enough appropriate data to be able to estimate good probabilities of words and word co-occurrences. This is very common in SDSs where in-domain data is seldom available and spoken corpora are rare. When data is sparse an SLM will obtain low estimates for many word occurrences and will most probably not have been exposed during training to many of the words that it will encounter when used. All SLMs used in ASR therefore apply some sort of smoothing technique (see Section 2.1.3) to overcome low-frequency counts and *zero counts*. The most commonly used smoothing techniques (or discounting algorithms) are *Good-Turing*, *Witten-Bell* and *Kneser-Ney* (Stolcke, 2002). In addition, techniques for combining higher and lower ordered n-grams are used such as *Katz-Backoff* and *deleted interpolation*. These are applied to be able to rely on lower order n-grams (e.g. bigrams) when a higher ordered n-gram (e.g. a trigram) is not encountered in the model to be able to estimate the probability of the higher ordered n-gram. The difference between these last two techniques is that deleted interpolation also rely on lower ordered n-grams for non-zero counts whereas Katz-Backoff only use the information from lower ordered n-grams for zero counts (Huang *et al.*, 2001).

Extensions or improvements to smoothing techniques have played an important role in achieving better SLMs. Chen and Goodman (1999) is a good source for an overview and comparison of smoothing techniques. In most SLM toolkits it is possible to apply most of these smoothing techniques.

A different approach to alleviate data sparseness and decrease perplexity is to build *class-based models* (Brown *et al.*, 1992; Rosenfeld, 2000a). A class is considered to be a group of words that usually appears in the same word contexts. The occurrences of words belonging to a certain class are seen as occurrences of that class and therefore counted jointly. All words in a class will therefore share frequency estimates. In this way, although some of the members of a class have not been seen frequently in the corpus they will obtain a good estimate by the overall class estimate. Classes can be part-of-speech (POS) classes trained on a POS-tagged text. However, an approach that has been much more successful is to group words into semantic classes (Ward and Issar, 1996; Popovici and Baggia, 1997a; Galescu *et al.*, 1998; Solsona *et al.*, 2002). An example from our previous language model example (see example 1 on page 17) would be to group all the sense verbs (hear, speak etc) into the same class. Such classes then function as single words in a word-based model. This approach is very common in SDSs and has lead to significant improvements. Classes can also be modelled by CFG grammars. An example would be an SLM that models when date expressions appear by using a date class which then points to rules describing date expressions. Embedding grammars into class-based SLMs to model such phenomena is much more adequate.

As Moore (2003) points out, after comparing the amount of training data we use in ASR to the amount of spoken data humans are exposed to, it is probably impossible to collect enough data to achieve human like WERs using today's methods. At any rate, the amount of data needed to build a reasonably good SLM differs depending on the task to be performed but also dependent on the language in use. For commercial SDSs in English, with a medium-sized vocabulary, a rough rule of thumb is to collect 20 000 transcribed in-domain utterances (Cohen *et al.*, 2004; Nuance, 2006; Hockey *et al.*, 2008).

Highly inflected languages, compound languages or languages with freer word order normally need more data and much larger vocabularies to capture the more varied word forms and varied order (Jelinek, 1991). *Out-of-vocabulary* (OOV) rates (see Section 2.4.4.3) for these languages are often much higher (Carter *et al.*, 1996). One approach for compounding languages, such as Swedish or German, is to split compound words to get more accurate estimates of the parts as well as a reduced vocabulary (Berton *et al.*, 1996; Carter *et al.*, 1996). For highly inflected languages, inflected forms of nouns can for example be grouped into classes. A recent approach is to work on the morpheme level and recognize morphemes rather than words (Creutz *et al.*, 2007). As the focus has mainly been on English, which is not a morphologically rich language, there is still much to explore in adjusting statistical language modelling to other language types.

2.4.4.2 Long distances

At least for languages with more restricted word order, trigram SLMs seem to capture, if trained on a large corpus, both syntactic, semantic and pragmatic information (Jelinek, 1991). However, language is much more complex than three-word sequences. In human speech perception we exploit relations between the meanings of words in order to be able to prime future occurrences of words in a given context. For example, when introducing the word "pasta" the more rarely used word "al dente" becomes more likely. In a similar way when uttering the word "either" the word "or" is expected to follow in some word position after. These dependencies usually span more than two words. SLMs cannot capture such dependencies. One approach to deal with longer distances is to use higher order n-grams, e.g. 4-grams or 5-grams. With a large amount of corpus material this can give better success. However, for SDSs, where training data normally is scarce, higher order n-grams lead to even more severe sparseness problems as many n-grams will not appear often enough, if at all. Trigrams are therefore still the most commonly used form. In dictation, there have been attempts to model conceptual relations among words by using Conceptual Networks, such as ConceptNet[2], to favour related words such as "brake" over "break" when the word "bike" has previously occurred. Such a strategy was shown to avoid errors and increase dictation speed in Lieberman *et al.* (2005).

One statistical technique to be able to capture correlations between content words is *latent semantic analysis* (LSA), a.k.a. Latent Semantic Indexing (Zhang and Rudnicky, 2002). This technique is widely used in the information retrieval community in an attempt to structure the relationships among words by reducing dimensionality. A matrix of word co-occurrences is built up. To reduce the dimensions of such a matrix an algorithm called Singular Value decomposition (SVD) is used (Bellegarda, 1998). The use of LSA in ASR in combination with n-grams have led to reduction both in perplexity and WER when compared to n-gram models on the WSJ corpus (Rosenfeld, 2000b; Bellegarda, 1998; Zhang and Rudnicky, 2002). Gorrell (2006) shows how LSA does not depend on SVD but can be used with a different algorithm: *Generalized Hebbian Algorithm* (GHA) (Gorrell and Webb, 2005). SVD and GHA were used in Gorrell (2007) to show the value of decomposition in statistical language modelling. It was shown to be hard to obtain a good performance with language models with reduced dimensionality alone. However, when interpolating them with standard n-grams an important reduction in perplexity could be shown. For large domains there is a tractability issue as these models are computationally expensive to produce. However, as Gorrell (2007) points out for smaller domains such as in SDSs this approach could be of interest although it has not yet been examined.

Other researchers have tried to improve SLMs by incorporating syntactic structure to complement the locality of trigram models (Chelba and Jelinek, 1999; Wang *et al.*, 2004). In Jelinek and Chelba (1999) a language model that uses grammatical analysis to predict the next word, a so called *Structured Language Model*, is described. A modest decrease in perplexity and WER could be shown in Jelinek and Chelba (1999) by using such a model but almost a decade later this model does not seem to have "prospered". In a

[2]http://conceptnet.media.mit.edu

similar manner Wang *et al.* (2004) generated a more structured SLM based on a Context Dependency parsed corpus which yielded a slight reduction in perplexity and WER when tested on read speech. However, more structured models somehow need to be trained on parse trees, for example from treebanks, which is seldom available at least for spoken language (and in different languages).

As opposed to SLMs, SRGs can model longer distances and take into account non-local syntactic structure. This is one of the advantages of SRGs. The disadvantage is the restrictedness of grammars that often suffer from insufficient rule coverage. Also, in SRGs all rules and words are equally probable which makes large grammars hard to process as the dimensions are too broad. To put some probability estimates into SRGs *probabilistic context-free grammars* (PCFGs; also Stochastic CFGs, SCFGs)) are used to estimate probabilities for rules and words in a grammar by the use of a corpus (Jurafsky *et al.*, 1995; Moore, 1999; Hacioglu and Ward, 2001). The introduction of statistics in grammars has lead to improved recognition performance for SRGs. However, it does not solve the coverage issue.

Another phenomenon in language perception is that humans seem to access words more quickly if the words have been heard recently. One attempt to model this word autocorrelation is by the use of a *cache language model* (Kuhn and Mori, 1990; Jelinek, 1997). Smaller improvements in recognition performance have been reported when using caches in language models. It has especially been used for dictation tasks. However, as pointed out in Jurafsky and Martin (2008), cache modelling in speech is influenced by the uncertainty of speech recognition as we cannot be sure that previously recognized words are correct. This makes errors persist and cache models less appropriate for speech.

2.4.4.3 New vocabulary

The vocabulary in SDSs will probably never fully cover a user's needs. Unknown words, or so called OOVs, usually appear even if a large corpus has been used and developers have struggled hard to predict user vocabulary. In fact, users are very creative or rather language is rich which means that previously unseen words will most probably appear. The number of unseen words in a test set is measured as the *OOV rate*. The OOV rate affects the recognition performance significantly.

With a bigger vocabulary we have more chance of covering more of the user's vocabulary. However, the size of the vocabulary also affects recognition performance. The bigger the vocabulary the bigger the search space and the more room for ambiguity and failure as there will be more words acoustically similar to confuse the input word with. Also if there are more words in the vocabulary a bigger corpus will be needed to get good estimates of all these words in different contexts. A vocabulary which is too large may slow down the recognition process and actually lead to more errors. The choice of vocabulary and its size is therefore very important.

Unknown words are hard to tackle for recognizers whereas humans seem to have little problem detecting them and are often also able to recognize them (at least as long as they follow the phonotactics in our language). Although the automatic recognition of novel

words is desirable the most critical point in ASR is to detect OOVs correctly as they lead to misrecognitions. When users make use of words unknown to the ASR system it will try to match these to words in its predefined vocabulary. As language models are built on the probability of word occurrences such an incorrect recognition may therefore also affect the recognition of surrounding words. There has been extensive research in detecting OOVs whereas the attempts to model novel words have been more limited.

For a recognizer to be able to recognize new words it would need to work rather on the sub-word level than on the word level. Phone-based models as well as syllable-based models have been proposed (Young and Ward, 1993a; Kemp and Jusek, 1996; Hazen and Bazzi, 2001). A phone-based model works without word restrictions and can therefore recognize arbitrary phone sequences. A way to constrain the search space is to model the phonotactics of a language by estimating probabilities of phone (or rather triphone) sequences with the use of an n-gram model (Young and Ward, 1993a; Hazen and Bazzi, 2001). One of the most extensive studies of the problem of new vocabulary was carried out at CMU by Young and Ward (1993a, 1995). They used a phone-model in parallel with a word-based model for decoding with the purpose of both detecting and recognizing OOVs. The detection was made by estimating a normalized acoustic score based on the results from both decoders. This score was then used to estimate the reliability of the correctness of words. A low score suggested a misrecognition. However to detect new words they also needed to determine the cause of a misrecognition. The use of a novel word (an OOV) as the cause of a misrecognition was predicted if the acoustic match was very poor but the SLM probability was good. On the other hand, a good acoustic match but a very poor SLM probability was seen as an indication of the use of a known word in a new sense. When an OOV was detected the result from the phone-based model was used to get a recognition of the word in the form of a phone sequence. The next problem that they addressed was then how to add words to the recognition model.

It is perfectly possible to add words to an SLM off-line and then to regenerate an extended SLM. However, without any knowledge about the contexts where the word usually appears it is impossible to obtain any statistical estimates for the new word. The introduction of class-based SLMs opens up a solution to this problem. Since a class already has a well estimated probability words added to a class will not run into this problem. Consider our example in Section 2.4.4.1, the sense verb class. To this class we could add the word "feel". It would then get the same probability estimates as for example the word "hear" and in this way we would be able to recognize new phrases such as "Can you feel". Young and Ward (1993a) used this approach by adding the detected novel words to classes in a class-based SLM. However, an additional problem that needs to be solved when adding new words introduced by users is to identify their meaning. Young and Ward (1993a) proposed the use of higher level knowledge (semantic and pragmatic) to derive potential meanings of words. The proposed meaning of a word was then related to a class in the class-based SLM to which the word was added.

Class-based language models also enable the possibility of populating a class dynamically during dialogue. Gruenstein *et al.* (2005) created *context-sensitive dynamic classes* by populating classes with words dependent on the context. For example if the user has

been presented a list of possible flight times, these flight times will become part of the dynamic "time" class and their probability will thereby be boosted. Gruenstein *et al.* (2005) reported a significant error reduction using such an approach.

One approach to recognizing OOVs conjointly with known words is to model OOVs by adding an OOV class to an SLM and estimate its probability with the use of unseen data (Gallwitz *et al.*, 1996). Since OOVs are not equally probable in all word contexts this is a way to also estimate when they are more likely to occur. The OOV class in the SLM can then point to a phone-based model that can recognize any phone sequences following the phonotactics of the language (Hazen and Bazzi, 2001). This approach has primarily been used to detect OOVs during recognition by marking them as OOVs to avoid them interfering with the recognition of known words. The OOV model (the phone-based model) as such has, however, not been frequently used to obtain a result, that is a proposal of how the new word sounded. This indicates that the performance of such models is still brittle.

A common approach to OOV detection is to make the detection in a post-process stage by relying on word confidence scores. A low score of a word would indicate a possible OOV. However, as Hazen and Bazzi (2001) show, this method is not as accurate as using an OOV model mainly because confidence scoring is not focussed on detecting OOVs but on misrecognitions in general. The estimation and use of confidence scores (both on the word and the utterance level) is further discussed in Section 2.4.6. Hazen and Bazzi (2001) showed that combining an OOV model and word confidence scoring can be fruitful.

An additional problem of adding new words to a system is that we must give them a pronunciation. If the new word is added by the system dynamically we must derive a pronunciation automatically. For some languages this is an easy task whereas for others it is more cumbersome. Future recognition of the new word will only work if its pronunciation is correctly generated. If the new word comes from an OOV model as a phone sequence this representation will possibly be close enough to be used (Young and Ward, 1995). However, it will not be possible to get any spelling of the new word if necessary.

Another possibility for grammar-based recognition (SRGs) is to use an SLM with broader coverage as a back-off recognizer to capture OOVs. In this way it is possible to identify vocabulary out of the scope of the grammar quite accurately as long as the words exist in the SLM. This technique was applied in Gorrell (2003).

There have only been a few research experiments on the recognition on new words. This makes it important for dialogue system developers to build vocabularies appropriate in size and coverage. At least it seems that in speech the use of the most common words are more common and the vocabulary used is smaller than the one used in writing (Allwood, 1998). Most importantly for SDSs is that even if we could recognize new words we would need to find good ways of deriving their meaning. Although the study by Young and Ward (1993a) describes an interesting approach it only applies to a smaller set of predefined semantic classes. Firstly, for the automatic acquisition of novel words in dialogue systems we would need to be able to parse sentences with unknown words (Purver, 2002). Identification of syntactic category (for example by POS-tagging) can give evidence of the type of word. Sometimes the syntactic information in the utterance together with the dialogue context will be enough to derive a meaning. Oftentimes the meaning of a novel word will need to be

clarified with the user as proposed by Purver (2002). Related recent work has focussed on the coordination of word meaning (*semantic coordination*) in corrective feedback (Larsson and Cooper, 2009; Cooper and Larsson, 2009). More research on the acquisition of new vocabulary by dialogue systems is urgently needed.

2.4.4.4 Developing language models for new domains

SLMs are unfortunately very bound to the training data and very sensitive to new types of data. It is therefore hard to reuse SLMs from one domain to another or adapt them to a new purpose. The mismatch can either be in style or in content. A mismatch in speaking style could be for example using newspaper text to build a model for a broadcast news recognition task. A mismatch in content could be to use transcriptions from spoken interactions in a travel domain for a tax office domain. Dialogue system developers are often confronted by the dilemma of a small amount of in-domain corpus material and large amounts of other corpus material.

However, somehow there should be something generic, domain-independent, in all the amount of text we have that we could reuse. As an example some phrases such as "I want to" seem to be quite common in many spoken dialogue system domains. Researchers have therefore attempted to create language models based on a mix of topics that are expected to model what is generic in a language and does not vary from one application to another (Solsona *et al.*, 2002; Gao *et al.*, 2000). The idea is that such models can then be adapted to different domains and tasks by combining them with domain data. Adaptation is often done with *Linear Interpolation* which is a technique for combining (or interpolating) two (or more) SLMs with weights (Xu and Rudnicky, 2000b; Manning and Schütze, 1999) as in Equation 2.4.

$$\boldsymbol{P}_{AB}(\boldsymbol{W}) = \boldsymbol{\lambda} * \boldsymbol{P}_{A}(\boldsymbol{W}) + (\mathbf{1} - \boldsymbol{\lambda}) * \boldsymbol{P}_{B}(\boldsymbol{W}) \tag{2.4}$$

Unfortunately, the scarceness of resources of spoken dialogue corpora have often prevented the use of generic corpora that match in style. Another way of generalizing has been to group the more domain-specific words into classes in class-based SLMs and then change the vocabularies for these classes as in Galescu *et al.* (1998). However, this is only possible when the domains are closely related.

Apart from the elusive generic language in corpora there must be something specific either in style or content that we could extract from appropriate sources for reuse. Attempts to reuse material have mainly been by using large amounts of news data, as this has been the most common kind of data available. Unfortunately, this has not been very successful for SDSs primarily due to the mismatch in style but probably also in content (Rosenfeld, 2000b). Some parts of such data will lead to improvements whereas other parts will introduce noise. It is therefore necessary to pick out the information relevant in content or style in some way. The most common measure of relevance have been by using perplexity (Galescu *et al.*, 1998).

Innovative approaches in recent years have used the web to collect suitable data for new domains (Zhu and Rosenfeld, 2001; Bulyko and Ostendorf, 2003; Sarikaya *et al.*, 2005).

Most of them use a small domain SLM as a base to seed the data. The most simplistic approach has been to reestimate n-gram counts in an SLM with those found on the web as presented in Zhu and Rosenfeld (2001). This can give more accurate estimates. Another way of using the web is to query a search engine with appropriate n-grams. These n-grams can either be appropriate in style, taken from a spoken dialogue corpus (Bulyko and Ostendorf, 2003), or in content, for example domain keywords collected from domain documents (Misu and Kawahara, 2006). In this way texts with n-grams close in style or content are collected. Sentences can then be selected from these texts based on their perplexity given a limited domain-specific SLM. Such a perplexity filtering selects sentences with perplexities lower than a certain threshold, expecting these to be closer in style and content. Selected sentences can then be used to build a new SLM. Approaches using the web as a resource for SLM development have been many and varied and are mostly based on a limited amount of domain material (Wee, 2004; Ng *et al.*, 2005; Sethy *et al.*, 2005; Akbacak *et al.*, 2005). Simple approaches to domain-adaptation of SLMs by using external corpora will be further discussed and explored in Chapter 4.

The difficulty of reusing SLMs often puts dialogue system developers in a situation where they need to start from scratch when building a language model for a new domain. Therefore there is a need for better ways of getting started with a good initial model. A common way is to set up a *Wizard of Oz* (WOz) experiment where training data is collected by a simulated experiment. Human subjects are given tasks to carry out with a fictitious simulated system, normally believing that it is a real system. However, this is costly and tedious as data needs to be both collected and transcribed. Furthermore, the WOz experiment needs to simulate the system well to evoke realistic user behaviour.

For commercial dialogue systems the focus has primarily been on grammar-based approaches (Rayner *et al.*, 2001; Knight *et al.*, 2001) especially for those built with the W3C standard *VoiceXML*. This probably depends on the time-consuming work of collecting corpora for training SLMs compared with the more rapid and straightforward development of handwritten SRGs. In addition, many commercial speech recognition suppliers, such as Nuance or SpeechWorks, did not support SLMs until recently (around 2001). Another benefit with SRGs is that they can be written to include semantic interpretation. With the use of SLMs it is necessary to develop an additional interpretation model. Indeed if only a tiny corpus is available a grammar based on that corpus will generalize better than an SLM (Rayner and Hockey, 2003; Wang *et al.*, 2002b). Although as the corpus gets bigger an SLM will give a better overall performance (Rayner and Hockey, 2003). SLMs do not constrain users in the same way as SRGs as they are more robust and able to deal better with unseen data. This make them work better for inexperienced users and more spontaneous speech which is common in commercial telephone applications (Knight *et al.*, 2001). On the other hand when users know what to say and keep their language in coverage of the grammar, an SRG can perform much better than an SLM (Knight *et al.*, 2001; Hacioglu and Ward, 2001). Hybrid approaches that combine SRGs and SLMs have been proposed by for example Gorrell *et al.* (2002). The idea is to rely on a more robust SLM when the grammar fails. However, this would imply developing both.

Often the choice of model, SLM or SRG, is based on whether it is possible to obtain

appropriate training data in a reasonable amount of time and on the knowledge of grammar writing available. The issue is how to produce good models quickly to get around the chicken and egg dilemma. In the end many of today's language models (both SRGs and SLMs) are not developed by speech scientists or linguists but by application developers at commercial companies. Therefore, we need simple and efficient ways to make use of their knowledge, their intuition about the domain and available corpora. In Chapter 4 I will investigate further on how to get a better start in language modelling when no or little data is available.

2.4.4.5 Context specific language modelling

With an SLM we try to model the probability of words and sequences of words in a particular language. However, in dialogue the probability of words and expressions are not static during the course of a dialogue but depends on the dialogue context. For example a particular question will make some expressions and words more expected and probable than others. In contrast to dictation systems speech recognizers for dialogue systems actually have access to contextual information, for example information about the state of the dialogue and the dialogue history, which of course should not be neglected in the recognition process.

In dialogue systems with a directed dialogue, where users are guided from state to state, it is possible to use different language models in each state. An approach in many commercial systems to constrain ASR and thereby improve the accuracy is to use *state-specific models*. Such models will only be able to recognize a restricted set of utterances and words specific for the current dialogue state. Changing state-specific models on the fly improves recognition accuracy as they will better model what is expected in that state. For example, if we are asking for a telephone number we can use a grammar specifically developed for that recognition task. Another typical case is to use a yes/no grammar after a yes/no question. This has been a common approach in the VoiceXML framework where small grammars are fine-tuned for different dialogue states. However, using state-specific SRGs enforces restriction on what users can say and does not make an application very flexible. Also, writing and maintaining state-specific grammars is cumbersome.

Dialogue state dependency is also applicable to SLMs. State-specific SLMs are built by partitioning the training data collected with the application into the dialogue states where the utterances occurred. An SLM for each dialogue state can then be built by using only the utterances collected in that state (Eckert *et al.*, 1996; Popovici and Baggia, 1997a). However, such models can lead to even more severe sparseness problems as some states may have very little data to build on. In less directed dialogue, where we cannot rely on explicit states, such as mixed-initiative dialogues, models are chosen by determining the dialogue context (Baggia *et al.*, 1997). A common way is to take the preceding system utterances into account. Although state-specific (or *context-specific*) language modelling improves recognition by constraining the search space it also restricts the users and impedes flexibility in the dialogue. To allow the user more flexibility many research systems have instead used one big general model at each point in the dialogue. Although this enables

less restrictive dialogue the ASR performance degrades significantly and thereby also the dialogue system performance. A more sophisticated approach is to adapt the generic SLM, built from all the training data, to the different states or context by interpolation (Riccardi *et al.*, 1998; Xu and Rudnicky, 2000b; Hacioglu and Ward, 2001). This makes all utterances possible but the context-specific ones more plausible. Dialogue state adaptation of the SLMs was used in the CMU communicator system and gave a WER reduction of more than 10% (Xu and Rudnicky, 2000b). The combinatorial strategy of SRGs and SLMs mentioned earlier could be used to overcome the problem of specificity, by having a state specific grammar for recognition with a general SLM to back off to in case the grammar fails. This would allow the user to move outside the grammar's bounds. Solsona *et al.* (2002) report a 12% relative reduction in WER using such an approach. A slightly different approach for state-specific grammars is to use one single grammar but change the weighting of the rules dependent on the context (Fügen *et al.*, 2004). Recent research has shown other ways to handle this trade-off on ASR restrictiveness and dialogue permissiveness by using approaches that primes context-specificity but does not disallow actions falling outside the local context. Context specific language modelling will be the focus of Chapter 5. The issue of predicting what models to use based on dialogue context will be explored in Chapter 6.

2.4.5 Improving ASR hypotheses selection

A straightforward approach to testing new techniques or additional knowledge sources in ASR has been to apply them in a post-process step on the output from the speech recognizer, e.g. on the N-Best lists (see Section 2.1.4.1). In this way there is no need to integrate proposed techniques, for example more sophisticated language models, into the internal recognition process to be able to evaluate them. Techniques are evaluated by their success in selecting the best possible hypothesis from N-Best lists when *re-ranking* (also reordering or rescoring) the hypotheses in N-Best lists. The meaning of the "best" possible hypothesis can either be the hypothesis that would minimize the WER (best word sequence match) or the hypothesis that best captures the user's intention (minimizing the CER). As we saw in Example 2, page 21, the recognizer's top choice is sometimes not the most accurate option but hypotheses that have been rated lower by the recognizer can be more accurate. In the corpus used in Quesada *et al.* (2002) it was estimated that 12% of the time the correct recognition of the utterance was included in the N-Best list but not as the top ranked item. In Chotimongkol and Rudnicky (2001) it is stated that on the *Communicator* corpus, a human-machine spoken dialogue corpus, a 37% relative improvement in WER would be possible if an *Oracle* method existed to pick the best hypothesis from 25-best lists. For the Switchboard corpus a 26% relative improvement in WER is reported as the upper bound (the Oracle rate) (Brill *et al.*, 1998). A 59% relative improvement in WER was reported as possible on 10-best lists from the ATIS corpus using a bigram model in (Rayner *et al.*, 1994). These figures indicate that if we could identify the correct alternatives in N-Best lists we would be able to make a significant improvement in recognition performance.

To investigate the limits and possibilities of improving recognition with the use of N-

Best lists researchers have given humans the task of re-ranking the outcome of speech recognizers (Brill *et al.*, 1998; Chotimongkol and Rudnicky, 2001). In Brill *et al.* (1998) human subjects were given the task of selecting hypotheses that they thought would have the lowest WER from 10-best lists for three different speech recognition tasks (Switchboard, Broadcast News and Wall Street Journal). The purpose of the study was to explore what linguistic knowledge humans make use of when carrying out such a task as well as to estimate the possible gain. The subjects were also allowed to edit the hypotheses. For each N-Best list they were asked to determine what knowledge or information they had used for their decision. Human subjects were indeed able to improve on the output of all three recognizers. Taking into account the possibility of editing the improvement was even better. The most complicated task was shown to be the spoken dialogue task, Switchboard, where the gain was lower. This was probably because the higher error rate of the recognizer for this task which did not leave enough cues to work on in the hypotheses (Brill *et al.*, 1998). According to the subjects the most common knowledge/information that they had used (for the spoken dialogue task) was the choice of words in closed classes (e.g. "that" vs "than") and open classes and the completeness of the sentence. For the Broadcast news and Wall Street Journal tasks the choice of determiners and prepositions had an important influence. Apart from linguistic knowledge the subjects also stated that they had made use of world knowledge in their selections.

Chotimongkol and Rudnicky (2001) performed a similar study on the Communicator corpus by giving human subjects 5-best lists together with the previous system prompt. The task was to select the most appropriate hypothesis from each list. Similarly to Brill *et al.* (1998) subjects were also able to edit the hypotheses. Again, human subjects were able to improve WER by selecting better hypotheses than the recognizer. The knowledge that subjects reported using was syntax, match in topic with the system prompt and the naturalness of the response. Native subjects performed much better than non-native subjects. In fact native subjects were able, when editing, to perform better than the Oracle rate regarding CER. This seems to indicate that humans make their selections based on a conceptual rather than lexical optimization. These studies with human subjects indicate clearly that speech recognition could well profit from the use of additional knowledge sources.

By demonstrating improvement of automatized hypothesis selection over simple recognizers researchers have evaluated techniques and the use of additional knowledge sources. Normally these re-ranking processes have been restrained to operate on the upper part of the N-Best lists, such as the 10 to 15 first hypotheses (Rayner *et al.*, 1994; Hacioglu and Ward, 2001; Gabsdil and Lemon, 2004; Balakrishna *et al.*, 2006). This is because most of the potential improvement lies in this upper part and the possibility of introducing more errors grows with the depth of the lists (Chotimongkol and Rudnicky, 2001; Hacioglu and Ward, 2001; Bousquet-Vernhettes and Vigouroux, 2003; Balakrishna *et al.*, 2006). Some researchers have operated on fewer hypotheses and some on variable amounts (Chotimongkol and Rudnicky, 2001; Wai *et al.*, 2001; Gurevych and Porzel, 2003; Bousquet-Vernhettes and Vigouroux, 2003).

The simplest approach to N-Best re-ranking is what Rayner *et al.* (1994) call the *highest-*

in-coverage method which uses syntactic or semantic knowledge in the form of a parser to select a hypothesis. This is done by starting at the top of the list and attempting to parse each hypothesis until a useful parse is successfully computed. Rayner *et al.* (1994) report a 7% reduction in WER when applying this simplistic approach using the Core Language Engine for linguistic analysis. However, in Chotimongkol and Rudnicky (2001) this strategy did not yield any improvement at all but performed 2% relatively worse than the baseline. Although this simple strategy did not lead to any reduction in WER in a study by Chung and Seneff (1998) it did indeed give a significant reduction in CER. In Bousquet-Vernhettes and Vigouroux (2003) a 5-best list was sent to the natural language understanding (NLU) model, a stochastic conceptual model, which then selected the most appropriate hypothesis. Feeding the NLU model with the top-5 hypotheses to work on instead of only the top-choice gave a 17% relative improvement in CER.

A more sophisticated technique is to compute a weighted sum of the score from additional knowledge sources (such as a parser) with the acoustic score from the ASR. In Rayner *et al.* (1994) the score from the linguistic analysis took into account for example if a hypothesis was in-coverage, an unlikely grammar construction, the grammar rules used and semantic triples appearing in the hypotheses. The scores from these linguistic knowledge sources were combined with the score from the recognizer (a bigram model) to compute a new score for each hypothesis. This new score was used to re-rank the N-Best lists to select a hypothesis and led to a proportional decrease of 13% in WER (as opposed to only 7% reduction on the same data for the highest-in-coverage method above). An analysis of the remaining failures showed that some alternatives were impossible to choose between without additional knowledge sources such as prosody or dialogue context (referred to as intersentential context). Other researchers have taken into account even more knowledge sources and applied a variety of analysis methods to make a more accurate choice from N-Best lists. A modest relative improvement in WER of 4% was achieved in the Communicator dialogue system (Chotimongkol and Rudnicky, 2001) by re-ranking the top-5 hypotheses with a linear regression model that combined syntactic, semantic, pragmatic and acoustic features. This re-ranking model performed equivalently in WER to the human subjects on the same task (see above). Syntactic knowledge consisted of information about the quality of the parse given from the robust Phoenix parser. As a semantic knowledge source a bigram of concepts (topic slots) was used to estimate the probability of co-occurring concepts in an utterance. To take into account the correlation between a user utterance and its previous system utterance a pragmatic feature was estimated by the probability of the "topic slots" (the semantic concepts) of an utterance ocurring in the current dialogue state. It was only by combining all these knowledge sources that a satisfactory result was achieved. Another study within the context of the Communicator Dialogue System used the N-Best re-ranking approach to evaluate the use of a mixture of language models (Hacioglu and Ward, 2001). They used a concept model that estimated the probability of a sequence of concepts given a dialogue context (a dialogue state). A syntactic model, a PCFG, parsed the word sequence into concepts and estimated the probability of the utterance. The re-ranking model took these knowledge sources into account to select the best hypothesis from 10-best list yielding a 6.3% relative improvement in

WER. In a study by Gurevych and Porzel (2003) ASR hypotheses were selected by the use of three knowledge sources: the ASR, the parser and domain knowledge. Hypotheses were converted into a conceptual (semantic) representation using a domain lexicon. The domain knowledge was used to score ASR hypotheses (represented as concepts) based on their semantic coherence to a domain-based ontology. In addition, they also took into account how well the hypotheses, as a conceptual representation, matched the *conceptual discourse context*. The conceptual discourse context was in fact the conceptual representation of the previous user utterance. Success in selecting the best hypotheses was compared between the use of each of these knowledge sources. It was shown that the use of the parser improved the success rate considerably in comparison to the ASR and that the use of the domain knowledge module even gave a sligthly better result than the parser. Combining the three knowledge sources through majority decision gave the best result.

The use of other knowledge sources has also been proposed. In the SIRIDUS project (Quesada *et al.*, 2002) it was proposed to use the information held in the information state (see Chapter 3 for an introduction to information states) to select N-Best hypotheses. For example, in the Linguamatics House Simulator, the system might prefer "switch on" to "switch off" when knowing that the device in question is currently off. It could also be used to rule out sentences which do not have a likely referent in the context (using pronoun and reference resolution). However, no implementation or evaluation was carried out to investigate the possible gain of such an approach. In a commercial system for flight information it was shown that the recognition of flight numbers could be significantly improved if a re-ranking module took into account dynamic call information such as the location of the caller (Wai *et al.*, 2001).

Instead of working with re-ranking of N-Best lists, some researchers have chosen to work with *word lattice parsing* to select more optimal hypotheses using syntactic and semantic knowledge (Quesada *et al.*, 2002; Chelba and Jelinek, 1999; Baggia *et al.*, 1991). A word lattice, as described in Gold and Morgan (2000), is a graph of possible word sequences, with associated probabilities from the on-line acoustic and language models. Some recognizers provide such lattices and make it possible to, instead of waiting for the recognizer to generate an N-Best list, compute on the lattice and extract paths from it. In this way it is possible to find candidates that are not part of the N-Best list and it is possible to extract those using linguistic knowledge in an earlier process step. Selecting a hypothesis from a word lattice is more complex than working with N-Best lists due to the challenging search problem. Mangu *et al.* (1999) describes an early approach to linguistic parsing of word lattices. Noord *et al.* (1999) were not able to show any benefit by using syntactic knowledge to choose the best path in lattices. In Quesada *et al.* (2002) word lattices were converted into Directed Acyclic Graphs and parsed with context free parsing to find the best optimal path. Unfortunately, only a very small evaluation was conducted. This indicated that a reduction in SER could be possible. Using word lattices as opposed to working directly on N-Best lists has proved to give little extra benefit and to be computationally more expensive (Mangu *et al.*, 1999; Wang *et al.*, 2002a; Balakrishna *et al.*, 2006) which may be one of the reasons why work on N-Best reranking have been much more extensive.

Several methodologies have been proposed that show how recognition rates can be

improved by letting external processes do the selection of ASR hypotheses with the use of higher level knowledge. The way that seems most appealing for SDSs is to make use of all the knowledge in a dialogue system, such as the dialogue context, to re-rank N-Best lists. The main concern of Chapter 8 in this thesis will be to explore the contribution of such knowledge sources in a post-processing recognition step.

2.4.6 Improving confidence annotation

Confidence scores (see Section 2.1.4.1) measure the reliability of the correctness of recognition results. The output from ASR systems is undoubtedly uncertain and error-prone. ASR systems output the most likely word sequence among its possible word sequences but does not tell us how well that word sequence matches what the user actually said. Confidence scoring concerns estimating the extent to which the words in a hypothesis match what was actually said by giving a score of reliability to each word in a hypothesis. Such *word confidence scores* are also often used to estimate an utterance score to reflect the reliability of the whole hypothesis (the utterance). As reflected in the previous section improvement on confidence scoring have sometimes been related to N-Best hypothesis selection as re-estimated confidence scores have been used to *rescore* (or rerank) the lists. In this way better confidence measures can also lead to better hypotheses selection. Confidence scores have also been applied to detect OOV as discussed in Section 2.4.4.3.

Furthermore, good confidence annotation is essential for the usefulness of speech in dialogue systems. If confidence scoring is not reliable, for example high scores are given to misrecognized utterances (or words), a SDS will be incapable of dealing with both correct and incorrect utterances. Knowledge of the reliability of a hypothesis is crucial in dialogue systems to be able to properly decide what to do with a hypothesis. The most evident decision-making in SDSs is the binary decision of accepting correctly recognized hypotheses and rejecting wrongly recognized hypotheses (Pao *et al.*, 1998; Hazen *et al.*, 2002). In dialogue systems we want to avoid the rejection of correct recognitions, *False Rejections* (FRs), as well as avoid the acceptance of misrecognitions (*False Acceptances* (FAs)). The most common approach when using ASR confidence scores is to set a threshold and accept hypotheses with a confidence score above that threshold and reject hypotheses below it. However, it is very hard to choose an optimal threshold as current confidence scoring is brittle (Bohus and Rudnicky, 2005a). A too low threshold will increase FAs and a too high threshold will increase FRs. Also, the impact FAs and FRs have on dialogue behaviour is not equivalent but FAs are often more severe and harder to recover from (Pradhan and Ward, 2002; Bohus and Rudnicky, 2005a; Renders *et al.*, 2005). In addition to rejecting and accepting hypotheses it is also very common to confirm (or verify) doubtful hypotheses with the user (Pao *et al.*, 1998; Guillevic *et al.*, 2002; San-Segundo *et al.*, 2001a). The verbal strategies to convey the success or failure of perception and understanding in dialogue systems are called *grounding strategies* (Clark and Brennan, 1991; Traum, 1995). Well applied grounding strategies can improve the robustness of the dialogue and perhaps the user's impression of the recognition process. It could also lead to better recognition of subsequent utterances as the user would be more aware of how the conversation is

going. Grounding strategies are further introduced in Section 3.3.5. However, proper use of grounding strategies is dependent on solid confidence annotation.

Driven by the importance of confidence when ASR is applied to dialogue systems there has been extensive research on how to obtain better confidence estimations. A comprehensive survey is given by Jiang (2005). As discussed in Section 2.4.4.3, Young and Ward (1993a) used a phone-based decoder to obtain an independent acoustic score to calculate a normalized acoustic score for each word and this was used to estimate word confidence. However, most confidence annotation models are limited to the information from the decoder. Confidence measures are normally based on comparing competing hypotheses (Jiang, 2005). If a hypothesis clearly surpasses competitive hypotheses the confidence will be high whereas if there are many equivalent competing hypotheses the confidence score will be low. The most prevalent features to estimate word confidence scores have been by using the information available from the ASR system. This makes the confidence scoring highly dependent on the ASR system concerned. Zhang and Rudnicky (2001) distinguish three commonly used types of ASR features: acoustic, language model and word lattice or N-Best list features. Word lattice and N-Best list features have proved to be more important than acoustic features (Hazen *et al.*, 2002; Zhang and Rudnicky, 2001). These features are extracted by comparing successive hypotheses in N-Best lists or by comparing competing paths in a word lattice. They represent properties such as repetitive patterns or singularity, density (number of alternative paths), number of hypotheses etc. However, the most important contribution has proved to be the features extracted from language models (Zhang and Rudnicky, 2001; San-Segundo *et al.*, 2001b). The success of the language model features suggests that the use of higher level linguistic knowledge could contribute to improved confidence scoring. These results and the brittle performance of confidence scoring based on and dependent only on the ASR system have encouraged the exploration of additional independent knowledge sources. Although some researchers have attempted to improve the low level acoustic scores by for example comparing competing sounds (Hernandez-Abrego and Marino, 2000) the use of additional higher level knowledge sources for predicting or estimating recognition performance have attracted more researchers.

Young and Ward (1993b) were probably among the first to argue for the use of the information available in SDSs for the task of confidence scoring. The most evident additional knowledge source that people have used is the parser. The addition of syntactic (or grammatical) features such as if a word or utterance is parseable and if possible also the likelihood of a parse have shown to be fruitful (Pao *et al.*, 1998; Zhang and Rudnicky, 2001; San-Segundo *et al.*, 2001b; Carpenter *et al.*, 2001; Guillevic *et al.*, 2002). Confidence scoring is normally optimized for word errors and not concept errors. Some word errors will evidently not lead to an error on a conceptual level and will therefore have little impact on the dialogue. The issue of estimating a confidence score more related to semantic reliability has been explored in several studies by introducing semantic knowledge (San-Segundo *et al.*, 2001b; Guillevic *et al.*, 2002; Pradhan and Ward, 2002). Semantic features are normally obtained by first interpreting the hypotheses in N-Best lists and then extracting features in a similar way as for N-Best list features but on the semantic

level, e.g. how often a concept appears in the list, the number of concepts or coherence of concepts in a hypothesis. These studies have revealed the contribution of semantics and in a study by Guillevic *et al.* (2002) the semantic features were actually shown to be the most discriminative features. There has been little use of knowledge from the dialogue or discourse context but it has been shown to be promising. Pradhan and Ward (2002) conditioned their language model features (both word and slot-based) on the previous system prompt. Carpenter *et al.* (2001) took into account dialogue expectation by adding a feature representing whether a semantic slot was expected in the current dialogue state or not. A dialogue state was again represented only by the previous system prompt. This *expected slot feature* was shown to be one of the most informative features in their model. A more recent study by Higashinaka *et al.* (2006) takes into account discourse features to improve the estimation of confidence for slots. The 12 discourse features that they use indicates violation of or conformity to the Gricean maxims. The feature shown to be most informative attempts to model the maxim of quantity by giving a lower value to concepts in the ASR hypothesis which are the same as the ones the system is trying to confirm. In this way their model would give less reliability to a hypothesis of the user saying "Barcelona" to confirm a system utterance such as "Barcelona, is that correct?". Litman *et al.* (2000) show how prosodic features can be an informative knowledge source in the task of detecting misrecognitions. A related approach to research on confidence annotation is the study by Walker *et al.* (2000b) where machine learning is used to predict spoken language understanding errors in dialogue systems. They classified semantic representations as correct, partially correct or as mismatches yielding a 86% classification accuracy. The features used were taken from the acoustic (from ASR), syntactic, semantic and dialogue context level. It should be noted that they did not have access to the ASR confidence score. The most prominent features were shown to be the syntactic and semantic features which outperformed the acoustic (ASR) features. The dialogue context features added to the performance but were not a determining factor.

Although many studies have compared different features and their contribution to the task of estimating confidence of recognition results no single feature has been prominent enough for its use alone. It has been by combining different features that the best results have been achieved. As an example Zhang and Rudnicky (2001) obtain the best performance by combining parsing and language model features. With this in mind a variety of techniques have been applied to the combinatorial use of features such as decision trees (Pao *et al.*, 1998; Pradhan and Ward, 2002), neural nets (San-Segundo *et al.*, 2001b; Guillevic *et al.*, 2002), linear discrimination (Pao *et al.*, 1998; Hazen *et al.*, 2002) and machine learning (Zhang and Rudnicky, 2001; Litman *et al.*, 2000). Carpenter *et al.* (2001) even make a comparison of different techniques for estimating confidence on the same data.

The studies discussed above indicate that confidence annotation can improve considerably by applying additional linguistic knowledge sources that are independent of the ASR. Even though many of these studies show a significant improvement in the task of estimating confidence for ASR hypotheses when introducing additional features the results are often inadequate with accuracy rates around only 60%. The FA and/or FR rates would be too high to achieve a reliable behaviour. The most successful studies report accuracies of

around 80% (Pao *et al.*, 1998; Zhang and Rudnicky, 2001; Carpenter *et al.*, 2001) which is much more reasonable. However, this still leaves the dialogue manager with a great deal of uncertainty and too much opportunity for making decisive mistakes especially since the FA rates seem to be higher than the FR rates. Unfortunately, currently available confidence scoring is not good enough for good practical use. It is not sufficiently reliable for optimal decision-making and nothing tells us the causes of the unreliability (OOVs, OOGs, noise, crosstalk, ambiguity) so that a dialogue system can react properly. In addition, it is optimized on giving reliability on a perceptual level and not on an understanding level which is not appropriate for dialogue systems. These reasons have encouraged me to explore further the possible benefit of higher level knowledge for better confidence annotation (see Chapter 8).

2.4.7 Improvement on other levels

Work on other levels in SDSs can also improve on the robustness of ASR. A tighter integration of SLU and ASR can help avoid ASR errors by the use of syntactic and semantic knowledge at an earlier stage. In addition, SLU could probably improve by taking into account dialogue context as shown in Meza-Ruiz and Lemon (2005).

Sophisticated dialogue management techniques can both avoid, predict, detect and recover from ASR errors. One way of avoiding many ASR errors would be to develop better turn-taking models (in addition to better end-pointers) as many ASR errors are due to misinterpretations of barge-ins and turn boundaries (Ward *et al.*, 2005; Raux *et al.*, 2006). Relevant help messages and well designed system responses can also avoid chain effects of ASR errors (Gorrell *et al.*, 2002). Predicting ASR errors or problematic dialogues have been dealt with in studies by Litman *et al.* (1999) and Walker *et al.* (2000a) and can be used to apply more cautious dialogue strategies in these situations. A very early study by Young *et al.* (1989) showed how their MINDS system predicted concepts that were likely to be used in the next user utterance to constrain the ASR search space by applying a semantic grammar and lexicon in coherence with these predictions. A dialogue manager that can anticipate what is most likely to happen will be much better prepared. Prediction of dialogue moves is the main concern of Chapter 6.

Much research effort is also put into error-handling. Since the occurrence of ASR errors seems inevitable it is important for dialogue managers to approach them in a proper way. A comprehensive overview of error-handling can be found in Skantze (2007). Skantze (2005a) shows that by studying human error recovery and error detection strategies we can learn how to model more efficient ways of coping with error-prone situations in SDSs. However, one of the biggest drawbacks today for error-handling is that dialogue managers do not have any information about what caused an error. With such information a dialogue manager would be in a better position to get the dialogue back on track.

It may seem far-fetched that the system's output can improve upon recognition but as Glass (1999) points out "the precise wording of the response can have a large impact on the user response". Studies have shown that people tend to build up a shared terminology during interaction. This phenomenon is called *lexical entrainment* (Brennan, 1996). It is

therefore also important to choose the words of the system utterances carefully, and ensure that the system can recognize these words.

2.5 Conclusions

This chapter has given the reader an introduction to speech recognition and spoken dialogue systems from a non-technical point of view. It has also introduced the evaluation metrics that will be used to estimate the experimental results presented in this thesis. The final part of this chapter presented a survey of attempts to enhance ASR performance. Hopefully this survey has given a clearer picture of the wide amount of research that has been conducted during the last few decades in order to try to combat the deficiencies of ASR. By means of considering how human speech recognition works or by incorporating additional linguistic knowledge sources researchers have been able to show significant improvements on recognition performance. Improvements have been achieved on many different levels.

The purpose of this survey was also to exemplify the difficulties that arise when using speech in dialogue systems. Many of these difficulties have also been encountered in the experiments in this thesis. Although the dialogue system interactions recorded with the systems in this thesis have not been carried out in a real environment they were conducted outside the laboratory. Subjects used a headset and a laptop and included both experienced and inexperienced dialogue system users. All of them were informed about how to use the headset, to not speak too low or to mumble too much. This was to avoid worst case scenarios. However, as the reader will see in the reports of the experimental data, recordings include noise, crosstalk and disfluencies and are thus far from clean speech. This means we also had to cope with these problems in some way. Even though we have not used prosodic information as a knowledge source in our work as we have limited ourselves to the knowledge available in the ASR system and in the dialogue system the experiments in Chapters 6 and 8 could well integrate prosodic information as an additional cue. In work for this thesis we have not been able to adapt acoustic models but have been limited to the predefined acoustic models for Swedish and English that the ASR system provides. We have, however, been able to modify the pronunciation lexicon. Some changes and additions were in fact done for Swedish resulting in considerable improvement. In this thesis we also make use of many of the techniques in language modelling mentioned above, such as smoothing techniques, class-based models, use of external resources for domain-adaptation and context-specific models.

Although the survey shows many interesting areas with the need of more research the focus in this thesis is primarily on the problems discussed in Sections 2.4.1.2, 2.4.4.4, 2.4.4.5, 2.4.5 and 2.4.6. This choice is not only based on the fact that these are the most appealing issues to the author but also because they are the most tractable from a dialogue system developer point of view. The purpose of the work in the following chapters is to further explore how higher level knowledge can contribute to these tasks.

Chapter 3

Baseline systems

This chapter will briefly introduce the reader to the *information state update* approach and in particular to the TrindiKit toolkit and the GoDiS dialogue system which apply this approach to dialogue modelling. The two baseline systems, built as GoDiS applications, that are used for experimenting throughout this thesis will also be presented. Additionally, we will give a short description of the two machine learning toolkits that have been used for some of the experiments in this thesis.

3.1 The information state update approach

The introduction in Section 2.2 to spoken dialogue systems pointed out the limitations of the finite state and form-based approaches to dialogue modelling and briefly mentioned the more sophisticated *Information State Update (ISU)* approach proposed by Traum and Larsson (2003). The ISU approach to dialogue modelling aims at a more flexible, generic, reconfigurable and theoretically founded way of building advanced dialogue systems. The key concept of the approach is the *Information State (IS)* based on the notion of *dialogue game-board* introduced by Ginzburg (1996). In theory an information state could be seen as a dialogue participant's internal view of the dialogue at each particular moment and which in practice in dialogue system development is a rich formal representation of the dialogue context.

The theoretical basis for modelling dialogue in the ISU approach can be described in analogy to a chess game. Dialogue participants perform *dialogue moves* when saying something in the *dialogue game*. In resemblance to a chess game the participants are considered to share a game board. In contrast to chess, in spoken dialogue the dialogue participants do not share the game board visually and therefore need to model this dialogue game board conceptually. In the ISU approach the dialogue participants' shared view of this dialogue game board is modelled as part of the information state. Apart from this, the information state also models the dialogue system's private mental state which in a chess game would correspond to, for example, a participant's game strategies. As the dialogue progresses and the dialogue participants perform their dialogue moves the

dialogue game board will change just as in a chess game. Dialogue moves performed by the dialogue participants and other actions result in *updates* that will alter the information state. These updates are governed by predefined *update rules* and a control algorithm that decides which update rules to apply at each point in a dialogue. The information state is not only used to store the information about the dialogue (the dialogue context) but is also used to interpret the user's contributions and to decide what actions to take and what to say. *Selection rules* are used to take decisions concerning system reaction based on the current information state.

There are several toolkits for facilitating the development of information state based dialogue managers such as Midiki (Burke *et al.*, 2003), DIPPER (Bos *et al.*, 2003) and TrindiKit (Larsson *et al.*, 2004). The ISU approach is applied in different ways in several research dialogue systems and applications such as GoDiS (Larsson, 2002), Witas (Lemon *et al.*, 2001b,a), SAMMIE (Becker *et al.*, 2006b), Radiobot-CFF (Roque *et al.*, 2006), Beetle (Dzikovska *et al.*, 2007) and Daisie (Ross and Bateman, 2009). Due to its rich representation of context the ISU approach has also been exploited in the investigation of automatic learning of dialogue strategies and for user simulation (Georgila *et al.*, 2005; Henderson *et al.*, 2005). This is also the main reason why the ISU approach is of interest for the work in this thesis. We will exploit the ISU approach as implemented in the TrindiKit dialogue manager GoDiS.

3.2 The TrindiKit platform for dialogue system development

TrindiKit (Larsson *et al.*, 2004) is a toolkit for building and experimenting with dialogue move engines and information states based on the ISU approach to dialogue management. It was originally developed in the EC-funded TRINDI project[1] in Prolog but has been further developed in the EC-funded SIRIDUS[2] and TALK[3] projects. The Midiki toolkit (Burke *et al.*, 2003) is based on an early version of TrindiKit while the DIPPER toolkit (Bos *et al.*, 2003) is based on the core ideas of TrindiKit. An *information state* (IS) in TrindiKit is specified as a data structure to store the dialogue system's internal information of the course of a dialogue. A *dialogue move engine (DME)* updates the IS on the basis of observed dialogue moves and selects appropriate moves to be performed. TrindiKit includes rule languages and data structures to define update and select rules, dialogue moves, algorithms and information states. In this way, it is possible to develop your own generic dialogue system based on a specific dialogue theory. The GoDiS system is such a dialogue system developed with TrindiKit at Gothenburg University.

The system architecture in TrindiKit differs from the classical pipeline structure in SDSs presented earlier in Figure 2.3 (see page 24) by centralising the IS and requiring

[1] http://www.ling.gu.se/projekt/trindi
[2] http://www.ling.gu.se/projekt/siridus
[3] http://www.talk-project.org

all module calls to be written and read through it. The information shared by modules is held in *module interface variables (MIVs)*. In such an architecture the modules can access information in a less restricted manner to enable asynchronous dialogue and more complex and flexible dialogue processing. In order to control the flow and all the modules, a control algorithm needs to be specified. Apart from proposing a general system architecture and approach to dialogue modelling, TRINDIKIT also provides modules for interpretation, generation, input and output as well as interfaces to ASR and TTS engines. The latest version of the TRINDIKIT toolkit, TRINDIKIT 4, was delivered as a result of the TALK project with new modules, a new architecture using the Open Agent Architecture (OAA) (Martin *et al.*, 1999) and a logging format. For a more thorough description of the latest version of TRINDIKIT see Becker *et al.* (2006a).

3.3 The GoDiS dialogue system

GODIS (Gothenburg Dialogue System) is a dialogue system that implements a theory of issue-based dialogue management as proposed by Larsson (2002). It has been formalised according to the ISU approach and implemented in TRINDIKIT in Prolog. We will here give a short introduction to the GODIS system with focus on the parts that are essential for a complete understanding of the experiments in this thesis. For a more extensive description of the GODIS system I refer the reader to Larsson (2002).

The modularity of GODIS with generic dialogue management enables the development of many different applications without altering and adapting the dialogue manager. Since most dialogue management mechanisms, such as grounding, feedback, belief revision, clarification, information sharing between tasks and question and task accommodation, have been implemented from a general, domain-independent theory, the application builder need not worry about how to obtain a flexible dialogue behaviour. GODIS provides a library of *update* and *select* rules that implements this standard GODIS dialogue behaviour. The domain-specific resources in the form of *dialogue plans* that need to be scripted and an ontology that needs to be structured are usually quite straightforward to implement.

3.3.1 GoDiS information state

The GODIS information state refers to the information stored internally by the dialogue system about the state of the dialogue. It is represented as a record holding information that is considered to be shared by the dialogue participants and a private part holding the dialogue system's internal view of the dialogue at each moment. The original GODIS information state was specified in Larsson (2002). Since then it has been further developed and we will focus on the current information state used for action-oriented dialogue represented as a record of the type shown in Figure 3.1.

The foundation of the GODIS information state is its division into two parts: PRIVATE and SHARED. The shared part holds information which has been established or assumed to be grounded during the course of the dialogue. The private part of the information state

$$
\left[\begin{array}{lll}
\text{PRIVATE} & : & \left[\begin{array}{lll}
\text{AGENDA} & : & \text{OpenQueue(Action)} \\
\text{PLAN} & : & \text{OpenStack(PlanConstruct)} \\
\text{BEL} & : & \text{Set(Prop)} \\
\text{TMP} & : & \left[\begin{array}{lll} \text{USR} & : & Tmp \\ \text{SYS} & : & Tmp \end{array}\right] \\
\text{NIM} & : & \text{OpenQueue(Pair(DP, Move))}
\end{array}\right] \\
\text{SHARED} & : & \left[\begin{array}{lll}
\text{COM} & : & \text{Set(Prop)} \\
\text{ISSUES} & : & \text{OpenStack(Question)} \\
\text{ACTIONS} & : & \text{OpenStack(Action)} \\
\text{QUD} & : & \text{OpenStack(Question)} \\
\text{PM} & : & \text{OpenQueue(Move)} \\
\text{LU} & : & \left[\begin{array}{lll} \text{SPEAKER} & : & \text{Participant} \\ \text{MOVES} & : & \text{OpenQueue(Move)} \end{array}\right]
\end{array}\right]
\end{array}\right]
$$

Figure 3.1: GoDiS information state

illustrates GoDiS's internal state.

Consider the shared part of Figure 3.1. It consists of the six fields COM, ISSUES, ACTIONS, QUD, PM and LU with the corresponding datatypes defined. COM is a set of propositions which corresponds to the commitments the dialogue participants have made by agreeing on these propositions. QUD holds questions that are currently discussed in the dialogue and which the user can address. It is based on the notion of questions under discussion (QUD) (Ginzburg, Forth). The ISSUES field on the other hand holds all questions that have been raised but not yet resolved, i.e. all pending issues. The stack of shared ACTIONS contains all requested actions that have not still been executed. To keep hold of what moves have been performed (and grounded) in the previous utterance the information state stores previous moves in the PM field. The latest utterance is represented as the record LU holding the speaker of the utterance and the dialogue moves performed by the utterance.

The information state's private part consists of the five fields: AGENDA, PLAN, BEL, TMP and NIM. The AGENDA queue holds the dialogue system's current agenda which the selection rules take into account to determine what the system should do next. The PLAN field holds the current domain plan which the update rules work through to determine what system actions to execute and how to update the AGENDA. The BEL field holds the system's private beliefs which is mainly answers to database searches that the system has carried out. The NIM field holds non-integrated moves and is GoDiS's way of keeping track of which moves have been integrated and not. An integrated move is a move that has been understood both on a semantic and pragmatic level and that is assumed to be grounded. Integrated moves will appear in the shared part in the /SHARED/LU/MOVES field. The TMP part is a temporary storage of previous information of some parts of the information state. We will look more closely at this part in Section 3.3.5.

In a TrindiKit dialogue system the information state (IS) is part of what is called the *Total Information State (TIS)*. The TIS additionally includes resource interface variables (RIVs) and module interface variables (MIVs). The MIVs are the variables that modules

read and write from in order to pass information. The RIVs are the available resources such as the domain plans, ontology, devices or grammar resources hooked up to the IS.

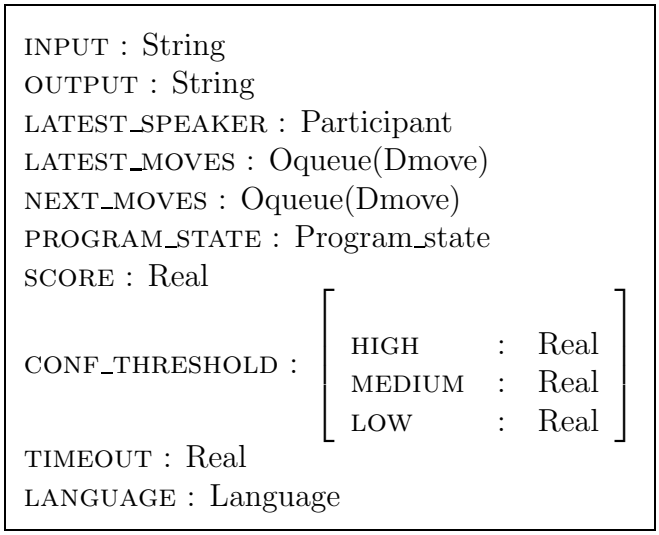

Figure 3.2: Module interface variables in GoDiS

Consider Figure 3.2 that shows the current set of MIVs in GoDiS. First of all we find some variables that permit modules to transfer information in a non-pipeline way by enabling modules to read and write to the same MIVs. The (user) input variable holds the output of the ASR (or text input module if non-spoken dialogue) in the form of a string. This is what the **interpretation** module makes use of to give a dialogue move representation of the input string. The latest_speaker variable holds as a value either the user or the system. The latest_moves data structure contains the dialogue moves performed in the latest turn either by the user or the system. This is where the output of the **interpretation** module is written. The next_moves variable contains the moves that GoDiS plans to perform next as decided by the DME. This is used by the **generation** module in order to determine how to express these moves verbally. The (system) output variable holds the system utterance the **generation** module have selected. The **output** module takes this string to for example send it to a TTS. To keep hold of the state of the GoDiS system (whether the system is running or should stop) the TIS uses the program_state variable. The timeout variable specifies the amount of time the system should wait until taking the initiative when the user is not responding. In the score variable the system keeps hold of the ASR confidence score of the last utterance. The conf_threshold record consists of three confidence thresholds that are set at start up. These thresholds and the score value are used to determine GoDiS's grounding behaviour as described in Section 3.3.5. The MIV language is needed to handle multilinguality and keeps hold of the current language used in the dialogue.

In summary, a GoDiS information state holds among other things information about what dialogue moves have just been performed and by whom, what questions are under discussion, what things the participants have agreed on, what the dialogue system plans to

do next and what internal beliefs the system has. This information is continually updated as update and select rules are executed or module calls are made. This means that a dialogue will consist of numerous information state instances where one instance of the information state is rarely identical to any other instance in the same dialogue. As we are interested in logging these information state instances it should be mentioned that the developer can decide in the control algorithm at which points the information state should be logged (see Section 3.5). This means that the logs normally only hold part of all possible IS instances. The principal reason of using GoDiS in this thesis is this rich, compact and easily extendable representation of dialogue context in the form of the GoDiS information state.

3.3.2 GoDiS dialogue moves

In the ISU approach user turns are interpreted into *dialogue moves (DMs)*. As discussed in Section 2.2.1 the notion of dialogue moves has its origin in Speech Act theory and the purpose is to classify utterances with their communicative function or the intention of the user. In addition to this, dialogue moves also capture the semantic content. Different ISU systems apply different taxonomies and different dialogue move representations.

In GoDiS, dialogue moves are activity related and exist in different types: request moves, answer moves, ask moves (i.e. questions), confirm moves, greet moves, quit moves and ICM:s. ICM:s are moves used for interactive communication management such as feedback and sequencing moves. We will discuss them further in Section 3.3.5. In the GoDiS dialogue move taxonomy an answer move is any utterance that provides information that is relevant to a question in some domain plan even if that question has not yet been raised. answer moves also include yes and no answers. As yes and no answers are a quite specific type of answer we have considered these as a specific class (yn answers) for the experiments in this thesis. In the application AgendaTalk (see Section 3.4.2) we have also introduced the social dialogue move type to represent utterances that have to do with social interaction such as thanking, flattering, insulting. These have a similar structure to ICMs.

A dialogue move is built up of several components where the *dialogue move type* represents the communicative function of the dialogue move. Update rules in GoDiS operate on the dialogue move type level. The *dialogue move types* in GoDiS can be combined with *propositional content* to form the dialogue moves. If we consider the utterance example from the Background chapter (Section 2.2.1) its representation as GoDiS dialogue moves would be as in (5).

(5) Usr> *I want to order broadband*
Usr DM: request(order) answer(order_type(broadband))

The example is represented as two dialogue moves. The first dialogue move is built up from the *dialogue move type* request representing that the utterance was a request (its communicative function) and the *propositional content* **order** representing that the request

was about ordering. This type of move relates to a task, i.e. a plan in the GoDiS framework. The propositional content of a dialogue move can also hold values (or individual constants) as exemplified in the second dialogue move. In this case the dialogue move type is an answer move, the propositional content is the **order type** with the *value* BROADBAND. That "broadband" is here interpreted as a separate answer dialogue move is one of the peculiarities of the GoDiS semantics. answer moves play a very central role in GoDiS due to their direct relation to the the domain plans as will be described in the following section. If we compare this semantic representation to the *frame and slot semantics* discussed in Section 2.2.1 we can see that the GoDiS semantics captures something related to the frame (the content of the first dialogue move) and something similar to a slot-value pair (the content of the second dialogue move) but adds in the user intention in the form of the dialogue move types. In this way it is possible to distinguish the request in (5) from the question "How do I order broadband?".

In this thesis we will often focus only on the *dialogue move type* to group utterances together according to their communicative function and not the semantic content they carry. In some of those cases we will use the term *dialogue move* rather than *dialogue move type* in accordance with the use of the terms "speech acts" and "dialogue acts" in the literature to refer to types of acts. In other cases we will represent dialogue moves both with the dialogue move type and the propositional content but discard the values in order to generalize over data.

The interpretation of utterances into dialogue moves can be carried out by any appropriate parser. GoDiS provides a simple Prolog-based key-phrase spotter where the developer can define how phrases should be matched to dialogue moves. TrindiKit also provides facilities for using grammars written with the Grammatical Framework (GF) (Ranta, 2004). We have made use of both in this thesis.

3.3.3 GoDiS plans and accommodation

In GoDiS *domain plans* are used to drive the dialogue forward towards the domain-oriented goal. A plan resembles a form in VoiceXML in the way that it specifies the pieces of information that are necessary to retrieve in order to carry out a specific task. In contrast to a VoiceXML form a GoDiS plan is a much more compact and specified representation that does not include any dialogue management, system prompts or grammar references. In this way a dialogue plan is language independent. Like VoiceXML a GoDiS plan does not define the procedural details, i.e. how to act, only what actions are needed for the system to reach the task goal. A GoDiS plan consists of a sequence of actions which is loaded to the PLAN field in the information state when activated. A principal component of a GoDiS plan is the findout construct which specifies information that the system needs to retrieve. Findout constructs correspond to the action of resolving (i.e. finding the answer to) alternative questions, yn-questions or wh-questions. The findout constructs in a plan can be resolved in any order and multiple findouts can be executed in one single dialogue turn. More complex plans can also include other constructs such as conditionals (if-then-else rules), calls to external resources (e.g. databases and devices) or actions that

manipulate the information state.

If we return to our telephone and broadband provider example the following simplified plan illustrates how the task of ordering could be represented:

(6) ACTION : order
PLAN: [findout(?x.**order_type**(x))
findout(?x.**address**(x))
findout(?x.**name**(x))
dev_do(order_device, 'Order')]
POSTCOND : **done('Order')**

Whenever the user expresses her intention to place an order this plan will be loaded to the information state and GoDiS would execute the actions in the plan starting by asking what type of order the user refers to, for example broadband or phone line.

To allow for a more flexible and cooperative dialogue behaviour with more possibility of initiative from the user GoDiS applies *accommodation* (Larsson *et al.*, 2000). Accommodation enables GoDiS to find out what the user refers to when she introduces information (i.e. performs answer moves) that has not yet been asked for. This is called *question accommodation*. What GoDiS actually does when exposed to an unrequested answer with an established issue is to accommodate by searching if the answer might be relevant to any planned but still not realized question. Accommodation works even when no concrete issue has been established and is then called *task accommodation*. If no issue has been raised GoDiS will accommodate by looking if the unrequested and apparently irrelevant answer might be relevant to a question in any of the predefined domain plans. If such a plan exists GoDiS will then assume that the user refers to that task (or plan) which holds the related question. In the case of ambiguity, where an answer matches to more than one question (in different plans), GoDiS will make use of clarification.

If we assume that the current domain includes the previous dialogue plan order as well as a dialogue plan for the task of asking how to go about ordering and both these plans include a findout action **order_type** then GoDiS would use task and question accommodation to carry out the following type of dialogue:

(7) Usr> Broadband
Sys> Do you want to order or know how to order broadband?
Usr> I want to order
Usr> Can you please give me your address?

As "broadband" interpreted into the move answer(order_type(broadband)) would then resolve a findout in two different plans GoDiS would first clarify before executing task accommodation. After the user's clarification GoDiS would select the order plan and add it to the PLAN field in the IS. GoDiS would thereafter apply question accommodation to resolve the first findout action (i.e. findout(?x.**order_type**(x))) of that plan. It would then proceed by executing the next action in the plan that has not been resolved, in this case the action of finding out the address.

3.3.4 GoDiS rule format

Update and select rules in GoDiS are built up as *preconditions* that need to be fulfilled in order to apply the rule and *effects* that are the results of the application of the rule. The GoDiS rule library consists of about 80 update rules and 20 select rules. The rules are grouped into classes in order to control their application in algorithms more easily. An important class of rules is the integration rules that update the information state with the dialogue moves performed by the user or the system as soon as these are considered understood and grounded. To exemplify the GoDiS rule format that will be used in Chapter 9 to create some additional rules we will here show a simple example rule for the integration of the quit move:

RULE: **integrateUsrQuit**
CLASS: integrate
PRE:
    not empty($/private/nim)
    $/private/nim/fst = DPM
    DPM/fst = usr
    DPM/snd = quit
EFF:
    pop(/private/nim)
    add(/shared/lu/moves, DPM/snd)
    push(/private/agenda, quit)

Figure 3.3: The GoDiS update rule **integrateUsrQuit**

The rule in Figure 3.3 tells us that the following conditions need to be fulfilled: the field non-integrated moves NIM should not be empty (i.e. it should hold some moves that still have not been integrated), the first element of the NIM queue should consist of a structure where the speaker is the user and the dialogue move performed a quit move. If this is fulfilled, the IS should be altered by popping the NIM queue and adding to the latest utterance (LU) the quit move, i.e. integrating this dialogue move into the IS. Finally, it should add the action quit to its agenda. In this way, using update and select rules, the information state is continually updated according to the rules applied as a dialogue proceeds. The accommodation behaviour described in the previous section is implemented in GoDiS by seven update rules of the class accommodate. The advanced developer also has the possibility of revising and adding update and select rules to customize the dialogue behaviour for her needs.

3.3.5 GoDiS grounding behaviour

To take into account that communication is imperfect and that mishearings and misunderstandings may occur a dialogue system needs ways of setting a common ground for what has been said and understood. The verbal strategies to convey the success or failure of perception and understanding in dialogue systems are called *grounding* (Clark and Schaefer, 1989; Clark and Brennan, 1991; Traum, 1995; Larsson, 2002). The use of speech recognition in dialogue systems introduces an even more urgent need for grounding behaviour as the dialogue manager will never be absolutely sure of what the user said as it may have been corrupted by the speech channel. It could well also be that the spoken language understanding has failed.

Most current dialogue systems, both commercial and research systems, therefore use some *grounding strategy* to verify with the user what has been understood and not. This is sometimes done explicitly:

(8) SYS> *The fourth of August, is that correct?*

or implicitly to drive the dialogue ahead meanwhile confirming:

(9) SYS> *The fourth of August. What time?*

A dialogue system needs minimally to know whether a hypothesis is worth grounding at all or if it should be directly rejected. As introduced in Section 2.4.6 this binary Accept/Reject decision is normally determined by comparing the ASR confidence score of a hypothesis against a predefined threshold. Accepted hypotheses are then either implicitly or explicitly grounded or directly accepted without any confirmation. Implicit and explicit confirmation have complementary strengths. With explicit confirmation it is easy for the user to correct the system's mistakes but at the same time it is tedious to continuously confirm everything. Implicit confirmation is less cumbersome as the user only interferes when the system gets something wrong. However, corrections are less straightforward and have shown to be more error-prone. Many dialogue systems introduce several confidence thresholds in order to explore different grounding strategies and to be more adaptive. It is common to distinguish between three levels: ACCEPT, REJECT and CONFIRM. Some systems use a four-tiered level of confidence where an utterance is either rejected, explicitly confirmed, implicitly confirmed or accepted.

The GODIS system uses a similarily fine-grained scale of grounding levels but the grounding behaviour in GODIS is not limited to the perception level but different strategies are also chosen dependent on semantic and pragmatic understanding of the user input (see Larsson (2002)). This section will present the current GODIS grounding behaviour which is an essential background section for the complete understanding of Chapter 8 and 9. GODIS implements a cautious strategy to grounding in that it grounds information optimistically but enables the possibility of reconsidering grounded material and undoing updates. To enable this backtracking the information state includes a temporary storage in the private part (TMP) that we only mentioned shortly in Section 3.3.1. This temporary storage holds a copy of relevant parts of the previous information state. In this way, the system can

always backtrack and recover previous information if negative feedback to grounded parts is given.

3.3.5.1 ICM moves

Feedback and sequencing moves in GoDiS are called ICMs which stand for Interactive Communication Management moves which is a concept introduced by Allwood (2000). We will here focus on feedback moves. A feedback ICM is represented as follows:

(10) icm:ACT:POL

Where ACT is some of the action levels: contact, perception, semantic understanding, pragmatic understanding or acceptance. This represents the communicative function of the ICM. An ICM is also defined by its polarity (POL) being either positive, interrogative or negative. In addition an ICM can have arguments such as strings or moves. On the pragmatic understanding level ICMs hold the argument dialogue participant (DP) and the propositional content (Cont). What is of interest for this thesis is primarily the system feedback moves used to ground the hypothesis from the speech recognizer as exemplified below:

(11) icm:con*neg –> SYS> *Are you there?*
icm:per*neg –> SYS> *Sorry, I did not hear you.*
icm:per*pos:meeting –> SYS> *I heard: meeting.*
icm:und*int:usr*event(meeting) –> SYS> *meeting, is that correct?*
icm:und*pos:usr*event(meeting) –> SYS> *meeting.*
icm:acc*pos –> SYS> *OK.*

However, GoDiS also handles feedback on the semantic and pragmatic level. If a user utterance is not interpretable the system will produce a positive ICM on the perception level (e.g. "I heard: at ten") but a negative ICM on the semantic level (e.g. "I do not understand"). For answer moves GoDiS will also check their relevance to the domain. If a user utterance holding some answer move is not successfully integrated but semantically understood the system will choose a positive ICM on the semantic understanding level (e.g. "at ten o'clock") but a negative ICM on the pragmatic understanding level (e.g. "I do not quite understand").

3.3.5.2 Grounding strategies

In GoDiS, grounding strategies are chosen based on the confidence score from the recognizer and the dialogue move type. The GoDiS developer will set three confidence thresholds: T1, T2 and the recognition rejection confidence threshold T3 as MIVs and where T1 > T2 > T3. The integration of dialogue moves and the choice of which ICM to use for system feedback depends on how the value of the confidence score relates to these thresholds. The integration of dialogue moves assumes that the user utterances have been

semantically understood (interpreted as dialogue moves) and will check for acceptability and relevance on the pragmatical level. If this holds the choice of ICM relies only on the confidence score from the speech recognizer. Table 3.1 shows how different feedback moves and grounding strategies are chosen dependent on the confidence score value. We have chosen to name the grounding strategies according to the most commonly used names in research, as discussed earlier, and not as defined in (Larsson, 2002). As long as the confidence score is higher than T2, GoDiS will opt for an optimistic grounding strategy and integrate, i.e. ground the dialogue move and generate a positive ICM of acceptance. In the second case, shown in Table 3.1, GoDiS will in addition perform an implicit confirmation by conveying to the user the propositional content that has been understood. In the third case, GoDiS will apply a pessimistic grounding strategy and confirm the dialogue move explicitly with the user before integration. A score lower than the third threshold (T3) will trigger a rejection strategy and the choice of a negative ICM of perception (e.g. "Sorry, I did not hear you.").

Table 3.1: Confidence score based grounding strategies in GoDiS

Confidence score	Feedback moves	Grounding strategy
Score > T1	icm:acc*pos	Optimistic acceptance
T2 < Score > T1	icm:acc*pos + icm:und*pos:DP*Cont	Implicit confirmation
T3 < Score < T2	icm:und*int:DP*Cont	Explicit confirmation
Score < T3	icm:per*neg	Rejection

To handle channel problems GoDiS also takes into account the possibility that the recognizer did not get any input at all. This will trigger the feedback move icm:con*neg to convey negative feedback on the contact level (e.g. "Are you there?").

The confidence thresholds are set by the developer in a configuration file. In its original version presented in Larsson (2002) T1 and T2 were arbitrarily set quite high (and the confidence threshold T3 probably to the ASR's default rejection score). This resulted in a too cautious grounding strategy with a lot of confirmations. In the baseline applications used in this thesis the thresholds have been set much lower based on experience from interactions. The choice of thresholds is a trade-off between relying on the recognizer to get a more fluid dialogue with less explicit feedback moves but also a higher occurrence of falsely accepted recognition errors or distrusting the recognizer, falsely rejecting some correctly recognized utterances and disrupting the fluency of the dialogue by explicitly confirming a great deal of the user utterances. Confidence thresholds need to be optimized in order to find the optimal point between *False Acceptances (FAs)* and *False Rejections (FRs)*. In Section 8.4.3 we will carry out such an optimization. As discussed in Section 2.4.6 reliable and solid confidence annotation is essential for the usefulness of speech in dialogue systems and for the proper use of grounding strategies. Section 2.4.6 pointed out the brittleness of current confidence scoring and the problem of finding an optimal threshold. In Chapter 8 we will therefore show a new way of choosing confirmation strategies that is not bound to the recognition confidence score.

When we go away from the more simplistic Accept/Reject decision considering just FAs and FRs come up short. The notion of FAs and FRs are originally borrowed from speaker verification where only one threshold exists (accept a speaker or reject a speaker). When we introduce several thresholds in dialogue systems the classification into FAs and FRs is more troublesome. For some levels, such as for explicit confirmation, it is not apparent what misrecognitions and correct recognitions should be classified as. You could consider that misrecognitions that are explicitly confirmed are FAs but you could also consider correct recognitions that are explicitly confirmed as FRs. However, what is not taken into account either is that FAs and FRs on different levels will have more or less severe consequences and will affect the dialogue efficiency differently. This issue has been brought up by several researchers, who argue that the cost of confirming an incorrect hypothesis is higher than confirming a correct one as the user will need to correct the former (Pradhan and Ward, 2002; Bohus and Rudnicky, 2005a; Renders *et al.*, 2005). In addition, it may also leave a worse impression of the system's understanding capabilities.

In this thesis we will therefore not count FAs and FRs on all levels but introduce the additional *False Confirmation* (FC) (as well as True Confirmation (TC)). As shown in Table 3.2 FAs would then be incorrect dialogue moves where optimistic acceptance or implicit confirmation have been applied. FRs would be correct dialogue moves that are either explicitly or implicitly rejected. A FC is an explicit confirmation of an incorrect dialogue move. This categorization is based on the ease of error-handling where it is easier for the user with explicit confirmation to either confirm a correct dialogue move or to reject an incorrect one. When it comes to implicit confirmation, although the user would actually become aware of the system's false assumption and be able to adjust, it would be more cumbersome to correct the system. For optimistic acceptance and rejection the user would not be aware of the false assumption immediately and FAs and FRs on these levels are therefore even more critical.

Table 3.2: Correctness for different grounding strategies

Grounding strategy	Correct DM	Incorrect DM
Optimistic acceptance	TA	FA
Implicit confirmation	TA	FA
Explicit confirmation	TC	FC
Rejection	FR	TR

What we want to achieve in dialogue systems is of course to minimize both the FA and the FR rate. In particular, we want to avoid FAs on the most critical level (optimistic acceptance). In addition, we want an acceptable distribution over the levels. If we manage to lower the FA and the FR rates by explicitly confirming everything this will lead to a sluggish dialogue. We therefore want to minimize the FAs and the FRs while maintaining the TC and the FC rate at a reasonable level. We will therefore take this categorization into account when we evaluate our work with confidence scoring in this thesis.

3.4 The baseline GoDiS applications

Several research applications have been built with the GoDiS system, among others a menu-based cell phone application (Olsson and Villing, 2005), a tram information service (Ericsson *et al.*, 2006a), a travel agent (Larsson, 2002), a speech-enabled MP3 player called DJ-GoDiS (Hjelm *et al.*, 2005), the talking calendar AgendaTalk (Jonson, 2000) and Dico a Volvo Driving information system (Villing and Larsson, 2006). I have used two of these GoDiS applications, DJ-GoDiS and AgendaTalk, for the experiments presented in this thesis in order to investigate domain-independent and multilingual strategies to improve speech recognition. To illustrate the domain-independence most of the experiments have been carried out both in the MP3 domain and the Calendar domain. This is also to explore if we can find a tendency in the results or if they were only coincidental. In this section we will familiarize the reader with the functionality of these two applications. They have both been built with the TrindiKit toolkit as multimodal and multilingual GoDiS applications in the EC-funded project TALK.

3.4.1 DJ-GoDiS: The MP3 player

DJ-GoDiS works as a voice interface to a graphical MP3 player[4]. The user can among other things change settings, choose songs to play or create playlists. DJ-GoDiS is multimodal and multilingual and can be controlled with speech and/or graphical input and works in both English and Swedish. For the interested reader, the MP3 OAA player application is more thoroughly described in Ericsson *et al.* (2006a).

3.4.1.1 Infrastructure

DJ-GoDiS was built with TrindiKit4 and GoDiS as a multimodal and multilingual application. It consists of a collection of OAA agents (see Martin *et al.* (1999)). The ASR agent controls the Nuance recognizer and the TTS agent controls the Realspeak synthesizer. The interpretation and generation is carried out by a GF OAA agent using grammars written with GF. The MP3 GUI agent is the graphical interface of the MP3 player and displays the current playlist and available songs. It allows the user to interact with it using graphical input by pointing at the graphics. The MP3 player agent controls the actual music playing.

3.4.1.2 Functionality

MP3 players are normally controlled via some sort of menu-based interface. The DJ-GoDiS system accepts both graphical and spoken input and enables the user to traverse these menus and use the preferred modality or a combination of both. As specified in Ericsson *et al.* (2006a) the functionality supported by DJ-GoDiS is given in 12.

(12) Add a song to the playlist

[4]DJ-GoDiS was developed mainly by David Hjelm and co-workers at the Gothenburg Dialogue Lab

Delete a song from the playlist
Clear the playlist
Shuffle the playlist
Play the current song
Stop playing
Play a specific song
Control volume
Ask about available songs and artists

3.4.1.3 Multilinguality and multimodality

The DJ-GoDiS system works in both English and Swedish with a double set of interpretation, generation and recognition grammars generated from GF. The user can switch language at any time during the course of the dialogue. The DJ-GoDiS system accepts user input by speech, through the graphical interface by clicking or pointing (dependent on the possibilities of the computer screen) or in a multimodal fashion by combining speech and graphical input in one single turn. The system outputs everything in parallel in both modalities as speech, graphics and/or text. In this thesis we have focussed on the spoken abilities of the system.

3.4.1.4 DJ-GoDiS GF grammar

The DJ-GoDiS grammar written with GF is in reality several grammars; a Swedish and English system grammar and a Swedish and English user grammar[5]. All grammars have a common language-independent structure inherited from the GoDiS resource grammar. In this thesis we have only made use of an early version of the Swedish user grammar that formalizes possible user utterances in the MP3 domain. For a thorough description of the GF GoDiS grammar resources and grammars I point the reader to Ericsson *et al.* (2006a) and Bringert *et al.* (2005). The grammar is written on the phrase level, accepting spoken language utterances such as e.g. "next, please". The resulting corpus when generating all possible Swedish user utterances that the grammar covers consisted of about 300 000 unique utterances.

3.4.1.5 DJ-GoDiS information state

The DJ-GoDiS system uses an information state, based on the one specified for action oriented dialogue (see figure 3.1, page 68), which has been extended in the TALK project to support multimodality. Simple utterance and move representations have been substituted for a representation in the form of records holding several fields including a field for the modality used to perform the dialogue move and a field for the ASR confidence score (or a GUI score for graphical input). This has required the data structure of several MIVs and

[5]The grammars were written by Peter Ljunglöf and Håkan Burden.

information state fields to be changed. The following example shows the new representation of the LATEST_MOVES variable:

$$(13)\quad \text{LATEST\_MOVES} : \text{Oqueue}(\begin{bmatrix} \text{MOVE} & : & \text{Move} \\ \text{MODALITIES} & : & \text{Set(Modality)} \\ \text{SCORE} & : & \text{Real} \end{bmatrix})$$

In order to handle asynchronity and multimodality, the TIS has been extended with several additional MIVs among others the INPUT_BUFFER that is shared among modalities and a holder for graphical output.

3.4.1.6 Dialogue example

The example dialogue in (14) has been borrowed from Ericsson *et al.* (2006a) to illustrate how an interaction with DJ-GoDiS can proceed:

(14)

SYS> Hello.

SYS> Let's see. What can I do for you?

SYS> Do you want to control playback or manage the playlist? *(DynGUI displays two buttons, one with the text "control playback" and the other with the text "manage the playlist")*

USR> manage the playlist

SYS> Okay. Let's see. Do you want to add a song to the playlist, delete a song from the playlist, clear the playlist or shuffle the playlist? *(DynGUI displays four buttons, one for each alternative)*

USR> add a song to the playlist

SYS> Okay. Let's see. Which song do you want to add? *(DynGUI displays the text "Which song do you want to add?" and a text input field)*

USR> London calling by the clash

SYS> Okay. I have added a song to the playlist. *(The song London Calling by the Clash appears on the MP3 Gui playlist)*

3.4.2 AgendaTalk : The talking calendar

AGENDATALK is a multimodal and multilingual dialogue application built with the GODIS dialogue manager and the TRINDIKIT toolkit as a spoken interface to a freely available schedule management software called the BORG Calendar.[6] The user can ask AGENDATALK about items noted in the calendar, e.g. "What time is the meeting?", as well as instruct the system to take down notes, e.g. "Add a meeting the 6th of October at 17". The calendar can also be accessed through the graphical interface like in a standard desktop calendar application in the in-home environment. AGENDATALK was originally built as a master's thesis project by the author (Jonson, 2000). It has then been further

[6] http://borg.mbcsoft.com as in 2006

developed and adapted to the latest GoDiS version and TrindiKit4 by the author in the TALK project. A brief description of the AgendaTalk system is given below. For a more thorough description I point the reader to Ericsson *et al.* (2006a).

3.4.2.1 Infrastructure

AgendaTalk is built with TrindiKit4 and uses Nuance for ASR and Vocalizer and Realspeak for TTS[7]. Figure 3.4 shows the AgendaTalk architecture. The database resource (Calendar DB) is a MySQL calendar database connecting AgendaTalk with the graphical calendar application BORG through the database server by sharing the same calendar information. An OAA wrapper for the BORG Calendar, the BORG Agent, has been built to be able to communicate directly with the graphical interface to enable multimodal output[8].

The **generation** module that AgendaTalk uses is a modified version of the TrindiKit generation module to be able to handle context-specific generation. This module was developed by the author for the original version of AgendaTalk to permit the **generation** module read access to the information state in order to check what propositions are held on com.

3.4.2.2 Functionality

The AgendaTalk application supports the functionality enumerated in (15) which makes it possible for the user to both change her schedule, check her bookings or navigate the graphical calendar.

(15) Add a booking
Reschedule a booking (date and/or time)
Delete a booking
Delete a whole day's bookings
Ask for the time of a booking
Ask if booked a certain time or date
Ask about bookings a certain date
Ask for today's date
Ask for the date of a booking
Ask for bookings on a certain part of the day
Go to a specific date in the calendar
Go to a specific month in the calendar
Change language of the dialogue and the Calendar

[7]The speech integration has been developed by David Hjelm.

[8]The adaptation of the BORG Calendar was carried out by Johan Bockgård based on a specification from the author.

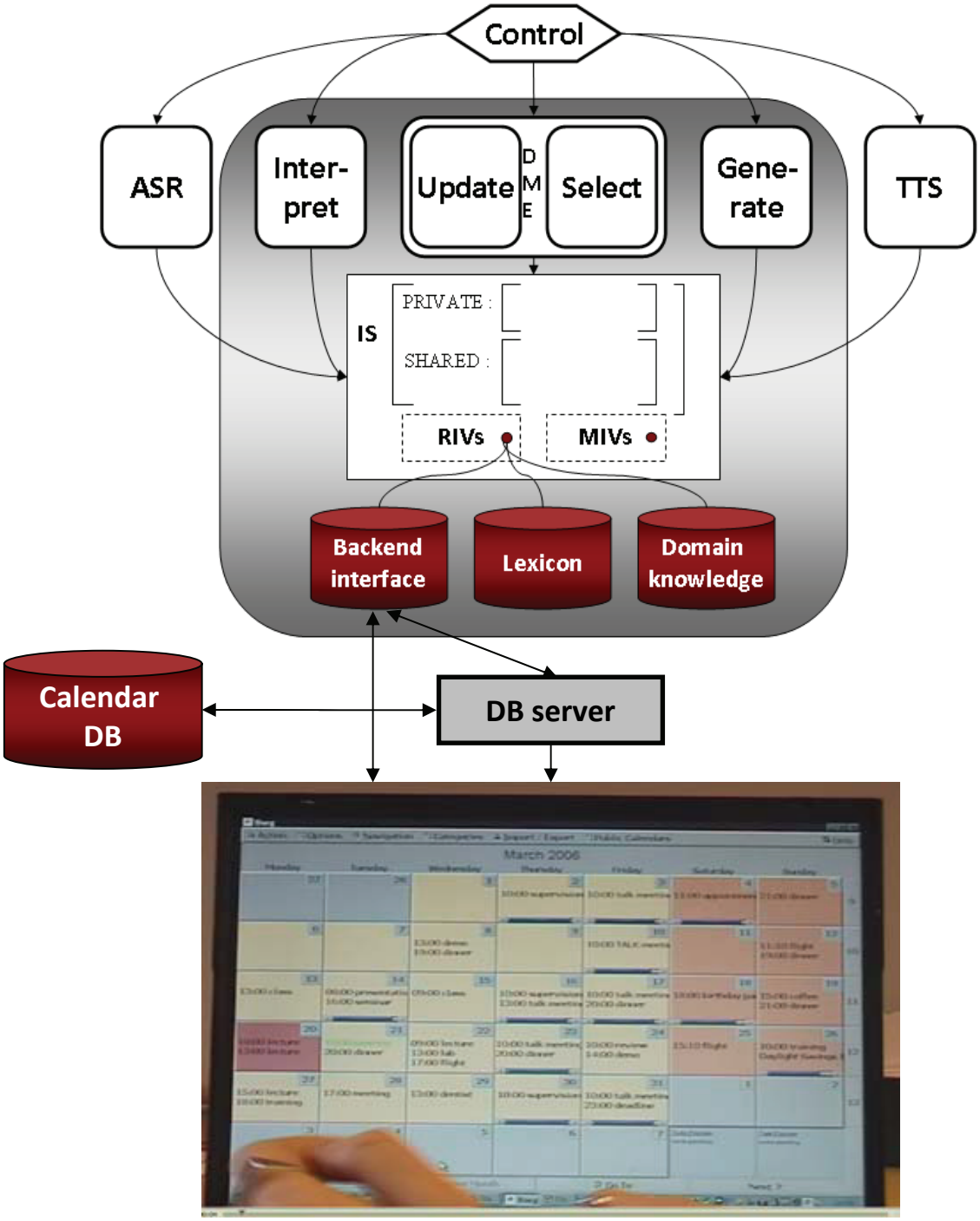

Figure 3.4: The AgendaTalk architecture

3.4.2.3 Multilinguality and multimodality

AGENDATALK works in Swedish and English and the user can switch between these languages on the fly whenever she wants, by giving a switch language command. The system will then switch grammars, language models, ASR, TTS and change the language of the BORG Calendar. However, the calendar information will be kept as it is. After a language switch the system will continue in the same state of the dialogue. This is possible due to the modularity of GODIS which keeps all dialogue management parts language independent. The only language dependent parts are the actual grammars and language models for ASR. This means that as all internal processes and the dialogue manager work in the same way for different languages the methods explored in this thesis can be used for both languages although the main focus will be on Swedish.

Although, in this thesis the focus is on the spoken input modality, AGENDATALK offers the user the opportunity to choose input modality. There is also an AGENDATALK version that handles multimodal fission that uses some additional modules[9]. These are described in detail in Ericsson *et al.* (2006b). AGENDATALK can be run either using advanced control of system output (multimodal fission), or without this due to the modularity of the GODIS system. The data collection used in this thesis was carried out in both of these modes and the collected logs include a variety of information.

3.4.2.4 AgendaTalk GF grammar

The AGENDATALK grammar was written with GF with the purpose of covering possible ways of expression of AGENDATALK users. It is based on a language independent structure built using the GF GODIS resource grammar and has been implemented in two languages; English and Swedish. The English and Swedish grammars share the same structure and thereby cover the same GF functions. The grammar consists of 500 GF functions (rules) where 220 are domain-specific and 280 inherited from the domain-independent grammar (the GODIS resource)[10]. A more extensive description of the AGENDATALK grammar and the GF GODIS resource grammar can be found in Ericsson *et al.* (2006a). The grammar has primarily been used to generate corpora artificially. The generation of all possible user utterances from the grammar resulted in an English corpus of over 4 million utterances and a Swedish corpus of 2 million utterances. These corpora have been used for the experiments in this thesis. In addition, AGENDATALK has a lexicon resource written in Prolog that has been used for parsing at run-time.

3.4.2.5 AgendaTalk information state

The information state used in AGENDATALK is an extended version of the information state used in the DJ-GODIS system. First of all, in AGENDATALK we additionally keep track of the dialogue history. This was mainly implemented for the purpose of the experiments

[9]These multimodal fission modules were built by Stina Ericsson.

[10]The domain-independent grammar was written by Peter Ljunglöf. The domain-specific part was written by Ann-Charlotte Forslund based on a specification by the author.

in this thesis to be able to collect logs holding an information state feature that has been widely used in related research. The dialogue history is part of the private information state and is a stackset of records in a similar format to the LU variable in the IS (see Figure 3.1) as shown in (16) below. It holds all dialogue moves that have been performed during the dialogue in the form of records representing a turn. Each record holds the dialogue moves performed, the confidence score of each dialogue move, the modalities used to perform them and the dialogue participant who made the contribution. The topmost element of the stackset is the representation of the most recent contribution.

(16) stackset(record([speaker : participant,
modality : set(modality),
turn_cont : set(record([move : dmove,
score : real]))]))

The GoDiS system does not incorporate any reference resolution which was a required dialogue behaviour for the AgendaTalk system to be able to handle dialogues such as the one in example 17.

(17) Usr> When is the meeting on Friday?
Sys> The meeting is at five pm.
Usr> Delete it.
Sys> Deletion done.

A simple reference resolution mechanism based on recency was therefore implemented by the author by extending the information state, revising some update rules and creating some additional rules. Although implemented to treat some simple reference phenomena in AgendaTalk the solution is domain-independent and can be used by any GoDiS application. The dref stackset (see 18 below) was added to the shared part of information state to hold possible discourse referents in the form of their attributes.

(18) dref: stackset(proposition)

For the AgendaTalk version implementing multimodal fission the core information state looks the same but the TIS makes use of several additional MIVs.

3.4.2.6 Dialogue example

An interaction with AgendaTalk is exemplified in (19) to show some functionality of the application as well as how multimodal input can be used in the dialogue and the user can switch languages on the fly.

(19) Sys> What do you want to do?
Usr> i want to add a booking
Sys> What kind of booking?

USR> a meeting
SYS> OK. What date?
USR> [clicks on a date (Dec 10)]
SYS> What time is the meeting?
USR> am i booked at ten am?
SYS> you have no bookings at that time
SYS> So. What time is the meeting?
USR> Switch to Swedish
System switches language and switches the language of the calendar interface.
SYS> OK. Vilken tid är mötet?
Eng. OK. What time is the meeting?
USR> Klockan 9 på morgonen
Eng. At 9 am
SYS> Vill du lägga till: möte den tionde december klockan 9?
Eng. Do you want to add: meeting the tenth of december at 9 am?
USR> jajamen
Eng. yeah
SYS> Bokat
Eng. Scheduled
The booking appears in the calendar in the GUI.
USR> Byt språk till engelska
Eng. Change language to English
System switches language and switches the language of the calendar interface.
SYS> OK. What do you want to do?
USR> Delete the meeting [clicks on the 10th of December]
The system deletes the meeting the specified day by the click and it disappears from the calendar GUI.
SYS> Deletion done

3.5 The TrindiKit logging format

This section will briefly describe the current TRINDIKIT logging format to give the reader a hint of how the dialogue logs that have been used for many of the experiments in this thesis are represented. The TRINDIKIT logging tool provides the developer with the possibility of logging the parts of the dialogue flow that she is interested in and specifying the frequency of the log stamps.[11] The logging tool is able to log the information that the ASR and TTS

[11]The logging tool was specified and developed by David Hjelm.

systems send (i.e. what the system thinks the users say and what the system says), the *update* and *select* rules that are used and additionally the system's information state at different points in the dialogue. Each new dialogue is logged in a unique file named with a time stamp and each file starts with a session construct:

(20) session('tkit_log-2007-2-1-15-6-39.pl').

The information is logged as Prolog predicates (or facts). The resulting knowledge base that the logs make up makes it possible to write Prolog programs to extract the information the developer is interested in, in the fashion she wants. The predicate state logs the TIS at a particular moment, i.e. the Information State, the MIVs and the RIVs. The logging format is independent of the information state structure and can thereby be used by different applications using different information states. Update and select rules are logged as the relation m with the time stamp and the rule that was applied. The predicate event logs ASR and TTS events. The following Prolog fact shows how a TTS event has been logged with a time stamp, what was sent to the TTS and the corresponding audio file:

(21) event(1169210681318,tts('Hi! This is AgendaTalk, your personal talking calendar. ','sys1169210681318.wav').

An ASR event is logged in a similar way holding all information sent from the speech recognizer. If the speech recognizer is running in a mode delivering N-Best hypotheses this means these will also be logged. In multimodal applications, events from the graphical interface will be logged as GUI events.

The logging can be turned on and off in the configuration file where the developer can also specify what information to log (e.g. omit rule applications).

3.6 Machine learning toolkits

In some of the experiments presented in this thesis we have made use of machine learning to train classifiers to solve different tasks. Machine learning deals with computer programs that learn to solve a specific task automatically based on some training material (experience). The memory-based learner TiMBL (Daelemans *et al.*, 2001) and the rule induction learner JRip provided with Weka (Witten *et al.*, 1999) have both been used for the experiments in this thesis. I have opted for these two learners as they represent two different learning techniques, are freely available and easy to use. The training material (or experience) given to these learners has been extracted from the TrindiKit dialogue logs described in Section (3.5) and converted into a format that these learners expect.

3.6.1 TiMBL

TiMBL is a memory-based learner developed at Tilburg University. The approach of memory-based learning, also called instance-based learning, is to store all instances seen

(all training examples) rather than trying to generalize from them. This kind of lazy learning postpones all reasoning until the moment of classification. A classifier trained with a memory-based learning method will do the reasoning when encountering a new instance that needs to be classified. It will then try to find similarities between the new instance and the memorized training instances (its experience). Normally some kind of distance measure, such as for example Euclidean distance, is used to find the most similar (the closest) stored instance. The new instance will then be classified in accordance with the stored one. For a more thorough description of instance-based learning I refer the reader to Mitchell (1997). The TiMBL software is an excellent tool for training memory-based classifiers. It requires that you feed the learner with training material in a specific feature vector format where each feature vector needs to be labelled with a classification. The TiMBL software provides tools that enables the developer to train different classifiers using different feature sets based on the same training file. The developer also has tools for testing and evaluation available with the possibility of different evaluation metrics.

3.6.2 JRip

The machine learner RIPPER (Repeated Incremental Pruning to Produce Error Reduction) is a rule induction learner proposed by William W. Cohen. This type of machine learner, as opposed to the memory-based approach, tries to generalize from the training examples and create an explicit model of the task in the form of if-then rules. The rules show what generalization the classifier has learnt. In contrast to memory-based learners the model learnt is thereby much more transparent and interpretable for the human developer and more interesting from a theoretical perspective. The rules to use are chosen based on how well they cover the training examples. However, it is not guaranteed that the learner will find the smallest or best set of rules. For the interested reader, Mitchell (1997) describes how rule set learning works in depth. We have used the freely available version of the RIPPER learner, JRip, as provided by Weka. JRip learns if-then rules from labelled training examples in the AIFF format. The induced rules are presented in the following form:

(22) (Weekday = monday) and (Holiday = false) => Class=workday (16.0/2.0)

This example rule says that if the weekday is a Monday and it is not a holiday then it is a workday. The numbers in brackets after the rule show the number of positive training examples in the training data (rule coverage) and the number of negative training examples (errors). In this example, the rule covered 16 training instances while misclassifying 2 instances.

Part II

Enhancing a dialogue system's ASR performance

Chapter 4

Generating SLMs from GF grammars

This chapter addresses the chicken and egg dilemma of creating initial spoken language models for speech recognition in spoken dialogue systems. It explores a bootstrapping approach to statistical language modelling by generating corpora from interpretation grammars to be used for creating initial statistical language models (SLMs). We will investigate if it is possible to boost the recognition and understanding performance when applying such grammar-generated SLMs in comparison to using their corresponding speech recognition grammars (SRGs). It is also investigated if such models can be further improved by the use of external corpora. The chapter also discusses the advantages and disadvantages of rule-based and statistical language models and evaluates the proposed approach with experiments in the two baseline domains: DJ-GoDiS and AgendaTalk.

4.1 Introduction

Ideally, when building spoken dialogue systems we would like to use a corpus of transcribed dialogues, corresponding to the specific task of the dialogue system, in order to build an appropriate SLM. However, it is rarely the case that such a corpus exists when starting the development of a dialogue system and collecting it and transcribing it is very time consuming, delaying the building of the actual dialogue system. Speech recognition for dialogue systems is therefore often caught in the trap of the sparse data problem which excludes the possibility of using SLMs. A common approach is to write a grammar for the domain, either as an SRG or as an interpretation grammar which can then be compiled into an SRG using some grammar development platform such as Gemini, Regulus or GF (Rayner *et al.*, 2000, 2006; Ranta, 2004). The latter approach will assure that the linguistic coverage of the speech recognizer and the interpretation module are kept in sync. In the TALK project the Grammatical Framework (GF) was extended with such a facility to compile GF grammars into SRGs in different formats including the GSL format (Bringert, 2008).

In the Background chapter (see Section 2.4.4.4) we pointed out that many commercial systems have adopted the rule-based approach despite the superiority in performance of

SLMs. There are several reasons for this but a key factor is the chicken and egg dilemma at startup. When no training data is available it is less time-consuming to write an SRG than collect data. In addition, it is possible to extend the same SRG to use for interpretation. However, as discussed in Section 2.4.4.4, SLMs are more robust, can handle out-of-coverage output, perform better in difficult conditions and seem to work better for naive users while SRGs are limited by their coverage and how well grammar writers succeed in predicting what users may say. Nevertheless, as SRGs only output phrases that can be interpreted by the grammar their output makes the interpretation task easier than it is with the unpredictable output from an SLM (especially if the SRG has been compiled from the interpretation grammar or the SRG is used also for interpretation). In fact, the rule-based approach in the experiments reported in Knight *et al.* (2001) outperforms the statistical approach on semantic error rate on in-coverage data.

Different dialogue system applications will have a different distribution of users. Some systems will always have a large number of naive or less experienced users who will use more out-of-coverage utterances and more OOVs whereas users of other applications will have the opportunity to obtain considerable experience which will allow them to adapt to the system, in particular to its grammar and vocabulary. Therefore, rule-based approaches will be preferable in conditions where we can put requirements on the users whereas in other cases the statistical approach should be chosen. In Rayner *et al.* (2006) some examples of applications suited for rule-based approaches are presented. However, it should be noted that even experienced users may go beyond the coverage of a grammar and outside its vocabulary due to the spontaneity of speech and the difficulty of learning a spoken controlled language. SLMs might therefore be preferable even in these situations.

An approach, used in both dialogue systems and dictation applications, to overcome the startup problem with SLMs is to write a grammar for the particular domain and generate an artificial corpus from the grammar to be used as training corpus for SLMs (Popovici and Baggia, 1997a; Galescu *et al.*, 1998; Pakhomov *et al.*, 2001; Raux *et al.*, 2003; Bangalore and Johnston, 2004). The idea has its origin in the 90s when several studies used probabilistic grammars to generate additional bigrams (word pairs) to enforce bigram SLMs (Zue *et al.*, 1991; Jurafsky *et al.*, 1995; Kellner, 1998). In later attempts corpora of sentences are generated in order to build trigrams. These grammar-based SLMs will not be as accurate as the ones built from real data as the estimates are artificial, lacking a realistic distribution. However, what we want to investigate in this thesis is if these grammar-based SLMs can obtain a much more robust behaviour than their corresponding grammars. A study made by Bangalore and Johnston (2004) indicates that this might be the case. They created a class-based SLM based on a corpus of 10 000 sentences randomly sampled from a hand-crafted grammar. The resulting SLM outperformed the hand-crafted grammar in recognition performance. The aim is to find a way of compromising between the ease of grammar writing and the robustness of SLMs. With such a methodology we could use the knowledge and intuition we have about the domain and include it in our first SLM and perhaps get a much more robust behaviour than with a grammar. It is in other words a quick way to get a dialogue system working with an SLM. When the system is up and running it would thereafter be easy to collect real data and improve the SLM by

incorporating the collected data.

What we also want to explore is if it is possible to improve grammar-based SLMs by the use of existing corpora. Domain adaptation of SLMs is an interesting issue for dialogue systems which involves re-using a successful language model by adapting it to a new domain, i.e. a new application (see Section 2.4.4.4 for an introduction to the topic). If a large corpus is not available for the specific domain but a large corpus with a mixture of topics is available we can use this corpus and adapt the resulting language model to the domain. One may assume that the resulting SLM should be able to capture at least a part of general language use that does not vary from one domain to another. I will address this issue by using the Gothenburg Spoken Language Corpus (GSLC) (Allwood, 1999) and a newspaper corpus and adapt these to our baseline domains.

In this chapter, we will further explore the grammar-based approach to statistical language modelling. In earlier attempts the corpora obtained from the grammars was often based on ungrammatical, non-meaningful, fragmentary or unnatural sentences or only on n-grams (Zue *et al.*, 1991; Popovici and Baggia, 1997a; Galescu *et al.*, 1998; Kellner, 1998; Raux *et al.*, 2003). A difference in our approach is therefore that we will focus on generating only complete, meaningful and grammatical sentences corresponding to dialogue moves from interpretation grammars. We will consider several different language models based on corpora generated from GF interpretation grammars and combine them with other corpora. We will compare their recognition performance against the baseline: an SRG compiled from the same GF interpretation grammar that will be used for generation.

4.2 First experiment: Grammar-based SLMs for the MP3 domain

In this first experiment we will explore the grammar-based statistical language modelling approach for the MP3 domain. We will introduce the methodology by showing how SLMs can be generated from a GF grammar and then present the experimental results for the different language models on a collection of user data.[1]

4.2.1 Description of corpora

We will first introduce the three different corpora that we have made use of in this experiment.

4.2.1.1 The MP3 corpus

The domain that we are considering in this study is the domain of an MP3 player application: the DJ-GoDiS baseline system (described in Section 3.4.1). The English and Swedish interpretation and generation grammars for the DJ-GoDiS application were written with

[1] An earlier version of this material was published as Jonson (2006b).

the GF grammar formalism[2]. GF offers the facility of generating SRGs from interpretation grammars as well as the possibility of generating corpora from the grammars. We applied both techniques to the same grammar to be able to compare our approach to the compiled SRG.

The interpretation grammar for the domain, written in GF, translates user utterances to dialogue moves and thereby holds all possible interpretations of user utterances. We used GF's facilities and generated a corpus in Swedish representing most of the content in the grammar consisting of all possible utterances to a certain depth of analysis. We chose to work with Swedish as that was the first grammar developed and we had access to a Swedish data collection. The grammar used was in an early stage of development missing some relevant linguistic structures. The grammar is written on the phrase level, accepting spoken language utterances such as e.g. "next, please". The corpus of possible user utterances resulted in around 320 000 user utterances (about 3 million words) corresponding to a vocabulary of only 301 words. The database of songs and artists in this early version of the application was limited to 60 Swedish songs, 60 Swedish artists, 3 albums and 3 radio stations. The vocabulary may seem small if you consider the number of songs and artists included, but the small size is due to a huge overlap of words in songs and artists as pronouns (such as *Jag* (*I*) and *Du* (*You*)) and articles (such as *Det* (*The*)) are very common. This corpus is very domain specific as it includes many artist names, songs and radio stations that often consist of rare words. It is also very repetitive covering all combinations of songs and artists in utterances such as "I want to listen to Mamma Mia with Abba". However, all utterances in the corpus occur exactly once.

4.2.1.2 The GSLC corpus

The Gothenburg Spoken Language (`GSLC`) corpus consists of transcribed Swedish spoken language from different social activities such as auctions, phone calls, meetings, lectures and task-oriented dialogue (Allwood, 1999). The corpus is composed of about 1,300,000 words and is turn-based which gives it long utterances, including e.g. transcribed disfluencies. To be able to use the GSLC corpus for language modelling a pre-processing was carried out stripping the corpus of annotations and taking away all non-alphabetic characters. Additionally, we substituted some spelling variants and assured that the spelling chosen in the GSLC corpus coincided with our MP3 vocabulary (e.g. *jajamen* and *jajamän* (Eng. *sure*). The final GSLC corpus consisted of a corpus of about 1,300,000 words with a vocabulary of almost 50,000 words.

4.2.1.3 The newspaper corpus

We have also used a corpus consisting of a collection of Swedish newspaper texts (`GNC`) of 397 million words[3]. This corpus is part of a collection of written texts that has been

[2]The development of the grammars was carried out in the TALK project by Ann-Charlotte Forslund, Peter Ljunglöf and David Hjelm.

[3]Made available by Leif Grönqvist, Dept. of Linguistics, Gothenburg

collected at the Department of Linguistics at Göteborg University. The corpus consists of newspaper text from several Swedish newspapers (including Göteborgsposten (GP)) collected mainly during the second half of the 90s.

Rosenfeld (2000b) argues that a little more domain corpus is always better than a lot more training data outside the domain. The GNC corpus gives us quantity but the arguments of Rosenfeld about quality have led to the idea of extracting domain relevant data from the bigger corpora. This section describes a very simple way of selecting domain relevant data from the Swedish newspaper corpus.

To create a domain relevant corpus, sentences with domain related words were extracted from the Swedish newspaper corpus. We started by creating a domain relevant vocabulary taking the existing MP3 application vocabulary and adding missing domain related words (e.g. music, mp3-player, songs etc.). The resulting vocabulary was a vocabulary without highly frequent words such as functional words and pronouns. We also excluded ambiguous words, e.g. *låt* (Eng. *song* or *let*). In this way it only consisted of typical domain words. We used this domain vocabulary to extract all sentences where these domain words occurred from the Swedish newspaper corpus. The corpus we obtained consisted of about 15 million words, i.e. 4% of the larger news corpus.

4.2.2 Test data

To collect a test set we asked students in Computational Linguistics to describe how they would address a speech-enabled MP3 player by writing SRGs that would cover the domain and its functionality. Another group of students evaluated these grammars by recording utterances they thought they would say to an MP3 player. One of the SRGs was used to create a development test set by generating a corpus of 1500 utterances from it. The corpus generated from another grammar written by some other students was used as evaluation test set. Added to the evaluation test set were the transcriptions of the recordings made by the third group of students that evaluated both grammars. This resulted in a evaluation test set of 1700 utterances.

The recording test set was made up partly of the students' recordings. Additional recordings were carried out by letting people at the department record randomly chosen utterances from the evaluation test set. We also had a demo running for a short time to collect user interactions at a demo session that we included in the test set. The final test set included 500 recorded utterances from 26 persons. The average utterance length was 4.3 words. This test set has been used to compare recognition performance between the different models under consideration.

The recording test set are just an approximation to the real task and conditions as the students only capture how they think they would act in an MP3 task. Their actual interaction in a real dialogue situation may differ considerably so ideally, we would want more recordings from dialogue system interactions which is now only a fifth of the test set.

In addition to the recorded evaluation test set a second set of recordings was created covering only in-grammar utterances by randomly generating a test set of 300 utterances from the GF grammar. These were recorded by 8 persons. This test set was used to

contrast with a comparison of in-grammar (in-coverage) recognition performance.

4.2.3 Language modelling

To generate the different trigram language models (SLMs) we used the SRI language modelling toolkit (Stolcke, 2002) with Good-Turing discounting.

The first model was generated directly from the MP3 corpus we got from the GF grammar. This simple SLM (named `MP3GFSLM`) has the same vocabulary as the SRG (named `MP3GFSRG`) and models the same language as the GF grammar. This model was chosen to see if we could increase flexibility and robustness in such a simple way while maintaining in-coverage performance.

We also created two other simple SLMs: a class-based one (with the classes `Song`, `Artist` and `Radiostation`) and a model based on a variant of the MP3 corpus where the utterances in which songs and artists co-occur would only match real artist-song pairs (i.e. including some music knowledge in the model).

These three SLMs were the three basic MP3 models considered although we are only reporting the results for the `MP3GFSLM` (the class-based model gave slightly worse results and the other slightly better than the `MP3GFSLM` model).

Apart from this we used our general corpora to produce three different models: `GSLCSLM` from the GSLC corpus, `NewsSLM` from the newspaper corpus and `DomNewsSLM` from the domain-adapted newspaper corpus.

4.2.3.1 Interpolation of the MP3 corpus and the general corpora

A technique used in language modelling to combine different SLMs is *linear interpolation* as introduced in Section 2.4.4.4. This is often used when the domain corpus is too small and a bigger corpus is available. There have been many experiments combining domain corpora with news corpora, as those are often the biggest type of corpus available. These have led to slightly improved models (Janiszek *et al.*, 2001; Rosenfeld, 2000b). In this work we are going to interpolate our domain model (`MP3GFSLM`) with a spoken language corpus, the GSLC, to see if this improves perplexity and recognition rates. As the MP3 corpus is generated from a grammar without probabilities this is potentially a way of obtaining better estimates on words and word sequences from real spoken expressions that even include disfluencies. Ideally, what we would like to capture from the GSLC corpus is language that is invariant from domain to domain. However, Rosenfeld (2000a) is quite pessimistic about this, arguing that it is not possible with current interpolation methods. The GSLC corpus is also quite small.

The interpolation was carried out with the SRILM toolkit (Stolcke, 2002) based on Equation 4.1.

$$\text{MixGSLC SLM} = \boldsymbol{\lambda} * \text{MP3GFSLM} + \mathbf{(1 - \lambda)} * \text{GSLCSLM} \tag{4.1}$$

The optimal lambda weight was estimated to 0.65 with the SRILM toolkit using the development test set.

In a similar manner we also created two models in the same way as above by interpolating the two variants of the news corpus with our simplest model.

$$\text{MixNews SLM} = \boldsymbol{\lambda} * \text{MP3GFSLM} + \mathbf{(1 - \lambda)} * \text{NewsSLM} \tag{4.2}$$

$$\text{MixDomNews SLM} = \boldsymbol{\lambda} * \text{MP3GFSLM} + \mathbf{(1 - \lambda)} * \text{DomNewsSLM} \tag{4.3}$$

Apart from these models we created a model where we tried to interpolate both the GSLC model and the domain adapted newspaper model with the `MP3GFSLM`. This model, based on three different sources, will be called `Triple`.

Choice of vocabulary The resulting mixed models has a huge vocabulary as the GSLC corpus and the newspaper corpus include thousands of words. This is not a convenient size for recognition as it will affect accuracy and speed. Therefore we tried to find an optimal vocabulary combining the small MP3 vocabulary of around 300 words with a smaller part of the GSLC vocabulary and the newspaper vocabulary.

In a first experiment we used the CMU toolkit (Clarkson and Rosenfeld, 1997) to obtain the most frequent words of the GSLC corpus. We selected three different collections of most frequent words respectively 300, 500 and 750 words. These different vocabularies were merged with the MP3 vocabulary resulting in three mixed vocabularies of 500, 700 and 900 words. The overlap was quite low (73 words for the smallest vocabulary) showing the peculiarity of the MP3 domain. We used these vocabularies to generate three new versions of the `MixGSLC` model. After testing we decided on the 500 word vocabulary. Thereafter we created a vocabulary that was a mixture of the most frequent words in the GSLC corpus, the most frequent ones in the newspaper corpus, the vocabulary used for extracting domain data and the small MP3 vocabulary. This resulted in a vocabulary of 1153 words. The mixed models have all used this mixed vocabulary in the tests. This vocabulary even included some Swedish disfluencies, for example, “öh”, “eh” and “aha”, as these were frequent in the GSLC corpus.

4.2.4 Experimental results

This section presents the results of the introduced models in perplexity and recognition performance. For an introduction to the evaluation metrics used in this chapter see Section 2.3 (page 30).

4.2.4.1 Perplexity measures

The 8 SLMs were evaluated by measuring perplexity (PPL) on the evaluation test set of 1700 utterances.

In Table 4.1 we can see a dramatic perplexity reduction with the mixed models compared to the simplest of our models the `MP3GFSLM`. Surprisingly, the `GSLCSLM` models the

Table 4.1: Perplexity for the different SLMs.

Language Model	PPL
MP3GFSLM	587
GSLCSLM	492
NewsSLM	386
DomNewsSLM	321
MixGSLC SLM	61
MixNews SLM	78
MixDomNews SLM	75
Triple SLM	129

test set better than the `MP3GFSLM` which indicates that our MP3 grammar is too restricted and differs considerably from the students' grammars.

Lower perplexity does not necessary mean lower WERs and the relation between these two measures is not very clear. One of the reasons that language model perplexity does not measure the recognition task complexity is that language models do not take into account acoustic confusability (Huang *et al.*, 2001; Jelinek, 1997). According to Rosenfeld (2000b), a perplexity reduction of 5% is usually practically not significant, 10-20% is noteworthy and a perplexity reduction of 30% or more is quite significant. The above results of the mixed models could then mean an improvement in WER over the baseline model, the `MP3GFSLM`, which we will test in our next experiment. Apart from this we want to test if we can reduce WER using our simple SLM opposed to the SRG (`MP3GFSRG`) which is our recognition baseline.

4.2.4.2 Recognition rates

The 8 SLMs under consideration were converted with the SRILM toolkit into a format that the Nuance 8.5 speech recognizer (Nuance, 2006) accepts and then compiled into recognition packages. The models were batch evaluated on the recorded evaluation test set of 500 utterances. Table 4.2 presents WERs and SERs (as well as N-Best error rates) for the grammar-based SLM and the SRG.

Table 4.2: Recognition performance for the recording test set

Language Model	WER (N-Best WER)	SER (N-Best SER)
MP3GFSRG	59.37 (53.19)	81.96 (79.96)
MP3GFSLM	37.11 (29.48)	64.73 (56.71)

As seen, our simple SLM, the `MP3GFSLM`, improves recognition performance considerably compared with the SRG baseline (`MP3GFSRG`) showing a much more robust behaviour in correspondence to the data. Remember that these two models have the same vocabulary

and are both derived from the same GF interpretation grammar. However the flexibility of the SLM gives a relative improvement of 37% over the SRG (significant at the $p<0.0001$ level).

Table 4.3: Recognition performance for the recording test set

Language Model	WER (N-Best WER)	SER (N-Best SER)
GSLCSLM	83.04 (71.51)	92.30 (88.38)
NewsSLM	61.62 (49.53)	80.76 (71.54)
DomNewsSLM	45.03 (31.58)	75.55 (59.52)
MixGSLC SLM	34.58 (22.68)	64.13 (47.49)
MixNews SLM	38.00 (27.37)	67.13 (54.13)
MixDomNews SLM	34.07 (22.07)	65.33 (47.90)
TripleSLM	33.97 (22.02)	63.53 (46.49)

Table 4.3 shows the results for the other models under consideration. The SLMs built up uniquely from any of the external corpora without any interpolation with the `MP3GFSLM` perform substantially worse than the others. It is interesting to see that the simple way we used to create a domain specific newspaper corpus gives a model that better fits our data than the original much larger newspaper corpus. The models giving the best results are the models interpolated with the GSLC corpus and the domain-adapted news corpus in different ways which at best gives a relative reduction in WER of 8% (significant at the $p<0.05$ level) in comparison with the `MP3GFSLM` and 43% in comparison with the baseline (`MP3GFSRG`). The best model is the model that is built up based on both GSLC and the domain news corpus (`Triple` model).

4.2.4.3 In-coverage recognition rates

To contrast the WER performance with in-grammar utterances, i.e. utterances that the original GF interpretation grammar covers, we carried out a second evaluation with the in-grammar recordings. We parsed the evaluation test set to extract the utterances that were in-grammar. These few recordings (5%) were added to the in-coverage test set. The results of the second recognition experiment are reported in Table 4.4 and Table 4.5.

Table 4.4: Recognition performance for in-coverage test set: SRG vs Grammar-based SLM

Language Model	WER (N-Best)	SER (N-Best)
MP3GFSRG	3.69 (1.49)	6.78 (2.36)
MP3GFSLM	4.95 (2.04)	10.67 (3.54)

The in-coverage results reveal an increase in WER for all the SLMs in comparison to the baseline grammar `MP3GFSRG`. However, the simplest model (`MP3GFSLM`), modelling the language of the grammar, does not show any important degradation in performance and

Table 4.5: Recognition performance for the in-coverage test set: Mixed SLMs

Language Model	WER (N-Best)	SER (N-Best)
MixGSLC SLM	14.23 (6,29)	24.78 (13.57)
MixNews SLM	18.63 (10.22)	30.38 (17.99)
MixDomNews SLM	15.57 (6.13)	27.43 (14.16)
TripleSLM	15.17 (6.05)	26.55 (14.45)

the difference in performance between (`MP3GFSLM`) and `MP3GFSRG` is not significant. The one of the mixed models that seems to best adapt to the in-grammar domain language is the `MixGSLC` model that was built up from the GSLC corpus.

4.2.5 Discussion of results

The WERs obtained for the best models (see Table 4.2 and 4.3) show a relative improvement over the SRG of 40%. The most interesting is that the simplest of our models (`MP3GFSLM`), modelling the same language as the SRG, gives such an important gain in performance that it lowers the WER 22 percentage points. We used the Chi-square (χ^2) test of significance to statistically evaluate the results. It was shown that the differences in WER of the grammar-based model `MP3GFSLM` and the baseline the grammar `MP3GFSRG` is significant on the $p<0.0001$ significance level. Furthermore, the χ^2 test points out that the difference of WER for in-grammar utterances of the SRG and the `MP3GFSLM` is not significant. This means that the SLM generated from the grammar significantly outperforms the baseline, i.e. the SRG. The more accurate performance of the Triple model in comparison to the `MP3GFSLM` on the evaluation test set is also shown to be significant (on the $p<0.05$ level) which shows that it is possible to improve a grammar-based SLM further. However, we see an important and significant degradation in in-coverage performance for the mixed models (see Table 4.5).

Despite just a slight decrease in perplexity (see Table 4.1) we get an important decrease in WER (27 % relative improvement) for our domain adapted model, `DomNewsSLM`, in comparison with the `NewsSLM` model. However, the domain adapted model does not outclass the handcrafted `MP3GFSLM` model which seems better suited to the domain. It is only by using the mixed model `MixDomNews` that we find a model better suited for the domain. This shows that our simple approach of extracting domain relevant data from a bigger corpus and integrating this data into our model can improve our original model. However, just adding any corpora and integrating this with our model does not necessarily mean an improvement, e.g. the `MixNews` model performs worse than the simple `MP3GFSLM` model. The GSLC corpus, although of very small size, but consisting of transcribed spoken language, seemed more suitable for the language style giving a small improvement when integrated with our `MP3GFSLM`. This leads us to think that if the GSLC corpus had been larger our simple way of extracting domain relevant data would perhaps have given us an even better model. The GSLC corpus is based on activity types (see (Allwood, 1999))

and consists of several different texts collected from different activities such as church sermons, auctions, task-oriented dialogues and meetings. We have carried out some similar experiments with the GSLC corpus by choosing the most appropriate activities and creating a language model from these. We take "appropriate" to mean activities giving low perplexities when we tested our baseline model `MP3GFSLM` on each activity text. The SLM created from the church sermon transcripts gave much higher perplexity than, for instance, the SLM created from transcripts of bus driver and passengers dialogues. This shows how some activities from the GSLC corpus would be more fruitful for language modelling in dialogue systems than others. Unfortunately, the GSLC corpus is very small and domain related corpora extracted from GSLC are too small to obtain a reliable language model.

As the reader may have noticed, the WERs are quite high, which is partly due to the very small original GF grammar and a totally independent test set with an important amount of OOV words (4-10% OOVs depending on the vocabulary used) indicating that domain language grammar writing is very subjective. The students have captured a quite different language for the same domain and functionality. This shows the risk of a hand-tailored domain grammar and the difficulty of predicting what users may say. In addition, a fair test of the model would be to measure concept error rate or more specifically dialogue move error rate (DMER) (i.e. both 'yes' and 'yeah' correspond to the same dialogue move `answer(yes)`). A closer look at the `MP3GFSLM` results suggests that in many cases the transcription reference and the recognition hypothesis have the same semantic content in the domain (e.g. confusing the Swedish prepositions 'i' (into) and 'till' (to) which are both used when referring to the playlist). It was manually estimated that 47% of the recognition hypotheses could be considered as incorrect in this way opposed to the 65% SER that the automatic evaluation gave. This implies that the evaluation carried out is not strictly fair considering the possible task improvement.

The N-Best results indicate that it could be worth putting effort on re-ranking the N-Best lists as both WER and SER of the N-Best candidates are considerably lower. This could ideally give us a reduction in SER of 10% and considering error rates on a conceptual level perhaps even more. As presented in Section 2.4.5 more or less advanced post-process methods have been used to analyze and decide on the best choice from the N-Best list. We will propose an information state based technique to this task in Chapter 8.

4.3 Second experiment: Grammar-based SLMs for the calendar domain

We will evaluate the approach presented in the preceding experiment in a different domain using a GF grammar written for the AgendaTalk application (described in Section 3.4.2). In this way, we hope to find further indications on the possible performance gain when applying the grammar-based approach to statistical language modelling. The AgendaTalk GF grammar is much more extensive and elaborate than the DJ-GoDiS grammar which suggests we will start with a better baseline. Also, we will have a much more extensive test

set collected through interactions with the AgendaTalk application which gives us a more realistic setting than in the preceding experiment where we had to rely on an approximation. Models were obtained by generating all possible utterances from the AgendaTalk GF grammar, building SLMs from the grammar-based corpus and compiling them into recognition packages. To build these trigram language models we have again used the SRI language modelling toolkit with Good-Turing discounting. For comparison we have also compiled the GF grammar directly into an SRG using the GF compiler. In this experiment we will additionally evaluate and contrast the performance of the grammar-based SLM and the SRG on test sets with naive and experienced users.[4]

4.3.1 Description of corpora

The GF grammar written for the calendar domain consists of 500 GF functions (rules) where 220 are domain-specific and 280 inherited from a domain-independent grammar[5]. It consists of two equivalent language versions that share the same GF functions: English and Swedish. We have used GF's facilities to generate a corpus from the Swedish version consisting of all possible meaningful utterances generated by the grammar to a certain depth of the analysis trees in GF's abstract syntax. The grammar is written on the phrase level accepting spoken language utterances such as e.g. "add a booking please". The resulting corpus consists of 1.7 million utterances and 19 million words with a vocabulary of only 183 words. All utterances in the corpus occur exactly once. This is more than five times the size of the MP3 corpus. However, all grammar rules are not expanded which leaves us with a class-tagged corpus, without e.g. all variants of date expressions but with the class `date`. What we get in the end is therefore a class-based SLM that we compile into a recognition package together with a rule-based description of these classes. The SLM has three different classes: `time`, `date` and `event` and the domain vocabulary when including all distinct words in these classes make up almost 500 words. This can be compared to the smaller MP3 vocabulary of only 300 words.

4.3.1.1 Adding real speech corpora

In the previous experiment we saw that the use of real corpora in interpolation with our artificial corpus was only valuable as long as the real corpora approximated the language of use. The big news corpus we had available did not give any significant improvement but the transcribed Swedish speech corpus we used (GSLC) was much more helpful. In this study we have therefore once again used the GLSC corpus to improve our word occurrence estimates by interpolating it with our grammar-based SLM. From this corpus we have built an SLM which we have interpolated with our grammar-based SLM keeping our domain vocabulary. This means we are just considering those n-grams in the GSLC corpus which

[4]Part of this material was published previously in Jonson (2007).

[5]The domain-independent grammar has been written by Peter Ljunglöf. The domain-specific part was written by Ann-Charlotte Forslund based on a specification by the author.

match the domain vocabulary to hopefully get a more realistic probability distribution for these. We will call this model our `Extended grammar-based SLM`.

4.3.2 Test data

The collection of test data was carried out by having people interacting with the AgendaTalk system. The test group included both naive users with no experience of the system whatsoever and users that had previous experience with the system to varying extents. We have classified the latter group as expert users although the expertise varies considerably. All users were given a printed copy of a calendar month with scheduled bookings and some question marks and were assigned the task of altering the voice-based calendar so that the graphical calendar would look the same as the printed copy except for the question marks which they were to find values for by querying the system. This would mean that they would have to add, delete and alter bookings as well as find out information about their schedule, e.g. the time of an event. The tasks could be carried out in any order and there were many different ways to complete the schedule.

The data collection gave us a recording test set of 1000 recorded utterances from 15 persons (all native, 8 female, 7 male). This unrestricted test set was used to compare recognition performance between the different models under consideration. We also partitioned the test set in various ways to explore different features. The test set was parsed to get all in-coverage utterances that the original GF grammar covers to create an in-coverage test set from these. In addition, we partitioned the data by users with a test set with the naive user utterances and another test set from the expert users. In this way we could explore how our models performed under different conditions. The average utterance length was 3.1 words which is shorter than for the MP3 domain. The average utterance length for expert users (3.4) where longer than for naive users (2.9).

The recordings for the unrestricted test set have an OOV rate of 6% when using our domain vocabulary. The naive test set makes up 529 of these recordings with an OOV rate of 8% whereas the expert test set of 471 recordings has a lower OOV rate of 4%. The in-coverage test set consists of 626 utterances leaving us with an in-coverage rate of 62.6% for the unrestricted test set. This shows the need for a more robust way of recognition and interpretation if we expect to expose the system to less experienced users. Almost all of the utterances in the unrestricted test set is relevant to the domain which means this test set represents well what should be expected and accepted by the system.

4.3.3 Experimental results

To evaluate the recognition performance of our different types of models we ran several experiments on the different test sets. As in the preceding experiment we report results both on WER and SER. In addition, we will also report on a semantic level by reporting what we call dialogue move error rate (DMER) (see Section 2.3 for an introduction to the metrics). The DMER was obtained by parsing the recognized utterances and comparing these to a parsed version of the transcriptions, calculating the rate of correctly parsed

Table 4.6: Results on unrestricted vs in-coverage test set

Model	Unrestricted			In-coverage		
	WER	SER	DMER	WER	SER	DMER
Grammar	39.0%	47.6%	43.2%	10.7%	16.3%	10.3%
Grammar-based SLM	29.0%	39.7%	33.0%	14.8%	18.3%	13.7%
Extended SLM	24.0%	35.2%	25.8%	11.5%	15.8%	10.4%

Table 4.7: Results for naive vs expert users

Model	Naive users			Expert users		
	WER	SER	DMER	WER	SER	DMER
Grammar	46.6%	50.3%	54.7%	31.7%	44.4%	33.2%
Grammar-based SLM	34.4%	42.9%	41.3%	23.8%	35.9%	25.8%
Extended SLM	27.6%	38.2%	29.5%	20.7%	31.8%	22.7%

dialogue moves. For parsing we have used a phrase-spotting grammar written in Prolog that pattern-matches phrases to GoDiS dialogue moves (see Chapter 3). We could have used the original GF interpretation grammar for parsing but that would have restricted the parsing to the coverage of the grammar which is not an optimal choice together with SLMs. We have investigated how the grammar-based SLMs perform in comparison to the SRG under different conditions to see how recognition and understanding performance varies. All models have the same domain vocabulary and the OOV figures presented earlier thereby apply for all of them.

4.3.3.1 Grammar-based SLMs vs. grammars

Table 4.6 shows the results for the different language models on our unrestricted test set of 1000 utterances as well as for the part of this test set which is in-coverage. As expected they all perform much better on the in-coverage test set with the lowest WER obtained with our grammar. On the unrestricted test set we can see that the use of the grammar-based SLM reduces WER from 39% to 29% (26% relative) and DMER from 43% to 33% (24% relative) which indicates the robustness of the grammar-based SLM to new user input. The performance differences in SER are significant at the $p<0.0005$ level. The Extended grammar-based SLM shows an even more important reduction in WER (38% relative) and DMER (40% relative). Although not reported in the table the N-Best figures indicate that there is room for improvement with an N-Best SER of 31% for the grammar-based SLM and 25% for the Extended grammar-based SLM. The best performance achieved on the in-coverage test set is with the SRG (Grammar) which performs significantly better than the grammar-based SLM (on the $p<0.001$ level). However, the Extended grammar-based SLM does not perform significantly worse than the grammar on WER and actually has lower SER and a similar DMER.

In Table 4.7 we can see how the performance of all our models are better for the expert

users with a relative WER reduction from 25% to 32% in comparison to the results for the naive test set. The same pattern is seen on the semantic level with important reduction in DMER. The result is expected as the expert users have greater knowledge of the language of the system. This is consistent with the results reported in Knight *et al.* (2001). It is also reflected in the OOV figures discussed earlier where the naive users seem to have used many more unknown words than the expert users. The most robust model is the Extended grammar-based SLM with less performance degradation for the naive users and significantly better performance than the grammar and the grammar-based SLM both for naive users and expert users.

4.3.4 Discussion of results

The results show that the models perform very differently depending on the types of users and how much they keep to the coverage of the grammar. Our grammar-based SLM gives us a much more robust behaviour which is good when we expect less experienced users. However, we can see that we get a degradation in in-coverage performance which would be critical if we are to use the model in a system where we expect that the users will achieve a certain proficiency. The `Extended Grammar-based SLM` seems to perform well in all situations and if we look at DMER there is no significant difference in performance between this model and our grammar when it comes to in-coverage input. In most systems we will probably have a range of users with different amounts of experience and even experienced users will fail to follow the grammar in spontaneous speech. This points towards the advisability of using an SLM as it is more robust and if it does not degrade too much on in-coverage user input like the `Extended Grammar-based SLM` it would be an optimal choice. From the results it seems that we have found a correlation between the DMER and WER in our system which indicates that if we manage to lower WER we will also achieve better understanding performance with our simple robust parser. This is good news as it means that we will not only capture more words with our SLMs but also more of the message the user is trying to convey in the sense of capturing more dialogue moves. This will definitely result in a better dialogue system performance overall. Interestingly, we have been able to obtain this just by converting our grammar into an SLM.

The experimental results for the Calendar domain strengthen the view that grammar-based SLMs can give an important reduction in both WER and DMER and goes in accordance with the results presented in the preceding section. This time we reach a relative improvement of 26% and a further 17% if we interpolate our grammar-based SLM with real speech data.

4.4 Summary and conclusions

A first observation is that grammar-based SLMs give a much more robust recognition performance in both domains than the grammars they model. The results show an important improvement in WER (37% and 26% relative) over the baseline, i.e. the SRG compiled

from the same GF interpretation grammar used for generating the artificial corpus. The results presented in the second experiment also show that this improvement in WER seems to propagate well to the understanding performance measured in DMER (24% relative). However, the use of grammar-based SLMs also implies a slight falling off in in-coverage performance. Although it does not degrade its performance to a great extent the difference is significant. The use of SLMs are therefore more suited to systems where users are not expected to be experts and aware of the grammar and vocabulary coverage. However, in systems where less demand is put on the users and there is a lack of domain corpora it seems promising to use grammar-based SLMs in a first version of the system with the possibility of improving the models when logs from system interactions have been collected.

It was also shown that the use of appropriate external corpora in interpolation with the grammar-based SLMs could further improve the SLMs. It seems apparent from the tests that the quality of the data is more important than the quantity making extraction of domain data from larger corpora an important issue. In both experiments the Gothenburg spoken language corpus (GSLC) seemed to be a valuable resource of real estimates of spoken language patterns that helped to improve our models significantly. This should encourage the collection of real spoken language corpora.

There seems to be a tendency in the performance gain we can obtain by converting a grammar into its corresponding grammar-based SLM. The results seem comparable with those obtained by Bangalore and Johnston (2004) using random generation to produce an SLM from an interpretation grammar. Similarily, a study, carried out in parallel with the present one in the TALK project, by Weilhammer *et al.* (2006b) but for a different domain and with a different speech recognizer indicates that such a relation exists. They present a performance gain of 29% relative for their grammar-based SLM versus their SRG. Furtermore, their grammar-based SLM performed similar to an SLM built up from an extense WOz collection.

This means we have found a good way of compromising between the ease of grammar writing and the robustness of SLMs in the first stage of dialogue system development. In this way we can use the knowledge and intuition we have about the domain and include it in our first language model to assure some minimal coverage just as with SRGs but at the same time get a much more robust behaviour than with an SRG. From this starting point we can then collect more data with our first prototype of the system and easily integrate it with our SLM to improve performance. Such an approach will avoid labour-intensive WOz collections.

Another strategy would be to alter the use of the two types of models, using the SRG or the SLM, depending on the confidence that the hypothesis is in-grammar or not. Solsona *et al.* (2002) use such a strategy switching between a state-independent n-gram and state-dependent finite state grammar depending on the acoustic confidence scores obtaining 12% relative word error reduction for certain dialogue states. Gorrell *et al.* (2002) also combine SRGs and SLMs, by for example relying on a more robust SLM when the SRG fails. However, in our case the grammar-based SLMs could probably be used alone as in-coverage performance does not degrade to a great extent and thereby the system architecture is simplified. The grammar-based SLMs presented in this chapter have therefore been used

for many of the data collections performed with DJ-GoDiS and AgendaTalk in the TALK project and as part of the work of this thesis.

Chapter 5

Dialogue move specific SLMs

The preceding chapter showed how we can combine the art of grammar writing with the power of statistics by bootstrapping statistical language models (SLMs) for Dialogue Systems from grammars written using the Grammatical Framework (GF). To take into account that the probability of a user's dialogue moves is not static during a dialogue this chapter will explore how the same methodology can be used to generate SLMs where certain dialogue moves are more probable than others. These models can be used at different points of a dialogue depending on contextual constraints. Two experiments have been performed to see how much recognition and understanding performance can be improved by using these context-specific models that we will call Dialogue Move Specific SLMs. In the first study we present a small evaluation of three models in the MP3 domain. The second experiment is a more extensive evaluation of this type of model in the Calendar domain.[1]

5.1 Introduction

As introduced in Chapter 2 (see Section 2.4.4.5) context-specific models have shown important recognition performance gain (Baggia *et al.*, 1997; Riccardi *et al.*, 1998; Xu and Rudnicky, 2000b; Lemon and Gruenstein, 2004). Context-specific models have usually been of two types: created as state-specific SRGs or SLMs built from collected data partitioned according to dialogue states. Both methods have their disadvantages. In the first case, we constrain the user heavily which makes them unsuitable for use in a more flexible system such as an information state based system. This can be solved by having a back-off method but leaves us with extra processing as Lemon and Gruenstein (2004) point out. In the latter case, we are presented with an even more severe sparse data problem than when creating a general SLM as we need enough data to get a good distribution of data over dialogue states. In an information state based system where the user is not restricted to only a few dialogue states this problem gets even worse. In addition, why we chose to work with grammar-based SLMs in the first place was because data is seldom available in the first stage of dialogue system development. This leaves us with the requirement of an

[1] This chapter contains material previously published in Jonson (2007).

SLM that although being context-specific does not constrain the user and which assures a minimal coverage of expressions for a certain context.

In this chapter we will explore the use of the same methodology as in the preceding chapter to create context-specific SLMs from grammars that match these criteria. Context-specific models and specifically grammars for different contexts have as we noted above been explored earlier but generating corpora for such language models artificially from an interpretation grammar by choosing which moves to combine seems to be a new direction.

5.2 Introducing dialogue move specific SLMs

SLMs capture the lexical context statistics of specific language uses. However, the statistical distribution in a dialogue is not static but varies by boosting and lowering probabilities for different words and expressions depending on contextual appropriateness. It is not only words and expressions that vary their distribution but on a semantic level different conceptual messages will be more or less probable as the interpretation of a user utterance at different points of the dialogue. This means that certain dialogue moves will have a higher degree of expectancy at a specific point of the dialogue. To capture this phenomenon, we want to build models that raise the probability of certain dialogue moves in certain contexts by giving a higher probability for utterances expressing these dialogue moves.

Following the methodology in Chapter 4 we can generate a corpus from a GF grammar where each utterance is annotated with the correspondent dialogue moves. We can then partition the grammar-based corpus by dialogue moves to create SLMs based only on utterances for certain dialogue moves.

In order to avoid the restrictedness of context-specific models we will thereafter interpolate these "pure" models with a general grammar-based SLM as per Equation 2.4 on page 52. We will call the resulting models *Dialogue Move Specific SLMs* or shortly *DMSLMs*. A similar term has actually been proposed in earlier work by Taylor *et al.* (1998) who called their models "Move specific models". These were created by dividing the training set into 12 move sets and training a bigram model from each set. DMSLMs are models where utterances corresponding to a certain dialogue move are more salient, for example, a model where all ways of answering yes or no are more plausible than other utterances. In this way, we get SLMs where certain dialogue moves are more probable than others and where minimally all possible expressions for these, which the grammar describes, are covered. By interpolating with a more general SLM, covering the whole domain, we put no hard constraints on the expected dialogue move. In this way the user can in fact say anything at any point in the dialogue despite the raised expectancy for certain dialogue moves. We just boost the expected probability of certain dialogue moves and their possible expressions. By using contextual constraints in the information state we could then predict which model to use and switch SLMs on the fly so that we obtain a recognizer that takes account of expected user input. In Chapter 6 we will investigate a strategy for dialogue move prediction.

5.3 First experiment: DMSLMs for the MP3 domain

In this preliminary study we created three different models that we tested on parts of the test set described in Section 4.2.2 to evaluate the viability of the dialogue move specific modelling approach.

5.3.1 Dialogue moves in the MP3 domain

Although the number of dialogue moves in our system is quite small the possibility of combining these in the same turn makes the possible classes of move sets per turn, encountered in dialogue logs, reasonably large. In the automatically generated logs we looked at from interactions with the MP3 player application (DJ-GoDiS), there were 40 different move combinations associated with turns. Section 3.3.2 in the Baseline chapter introduced the dialogue move types in GoDiS. Examples of user dialogue moves in DJ-GoDiS are shown in Table 5.1.

Table 5.1: Dialogue moves used in the MP3 domain

Dialogue move	Utterance example
greet	Hi!
quit	Bye!
help	Help.
answer(song(dancing queen))	Dancing Queen
answer(station(rant radio))	Rant Radio
answer(index(3))	number three
answer(group(Abba))	Abba
ask(X^current_song(X)))	what song is this
request(pause)	pause the music
request(clear)	clear the playlist
request(next_song)	i want to listen to the next song
answer(yes)	yeah
icm:acc*pos	OK

5.3.2 Building dialogue move specific SLMs

To decide which models to create we examined common combinations of moves from automatically generated logs from interactions with the MP3 player. It was very frequent to give either the name of a song or a group or a combination of these in the same dialogue situations. We therefore created an SLM, `AnsGrSong`, based on these `answer` moves and combinations of these by extracting all utterances corresponding to these moves from our GF grammar. This corresponds to utterances such as:

```
Dancing Queen
Abba
Dancing Queen with Abba
Abba with Dancing Queen
```

Another SLM (`Request`) created for our tests was a model including all requests the user makes in order to control the MP3 player such as lowering the volume, pausing the music, skipping to next song, etc. The utterances making up these requests were generated from the GF grammar. It should be noted that there exist a lot of other requests in the domain that are not included in this model as they correspond to other dialogue situations (such as altering the playlist). The third model, `YN`, included all yes and no answers. Yes and no models have been very common context-specific models to evaluate.

Applying context-specific models in DJ-GoDiS is not straightforward as the dialogues are so direct and shallow. However, these three models seem to be useful in the current setting as they are quite predictable. More models could of course be considered based on other moves and move combinations but for this preliminary experiment it will be sufficient with these three dialogue move types.

We interpolated each of these *Pure DMSLMs* with the general grammar-based SLM (`MP3GFSLM`) to get the less restrictive DMSLMs we are looking for. The interpolation was carried out with the SRILM toolkit based on Equation 5.1.

$$\boldsymbol{DMSLM = \lambda} * \text{Grammar-based SLM} + \boldsymbol{(1-\lambda)} * \text{Pure DMSLM} \tag{5.1}$$

We tested several weights and found an optimal interpolation weight of 85 for the mixed models. The tests presented in the following section compare the models only including a specific dialogue move (the pure DMSLMs), the general grammar-based SLM (`MP3GFSLM`) from Chapter 4 and the less restrictive DMSLMs in which a specific dialogue move has been boosted.

5.3.3 Experimental results

We evaluated the three models on test sets for each model that included only utterances that corresponded to the dialogue moves in the model. It should be mentioned that this implies that the test sets may include utterances not covered by the GF grammar although considered as belonging to some of the moves (e.g. a different wording for the same move). The transcriptions were partitioned manually into dialogue move sets by the author and one additional annotator. Figure 5.1 shows the performance in WER and N-Best WER for the `Request SLM`, `MP3GFSLM` and the `Request DMSLM` on the test set of Requests. We can see that the DMSLM outperforms the other models. The pure model (`Request SLM`) seems to be too restricted to perform well on an independent test set whereas the general model is too general.

In Figure 5.2 we show the WER results when testing the same models on a test set of in-coverage Requests. Here, we can see that the pure model `Request` has the best performance

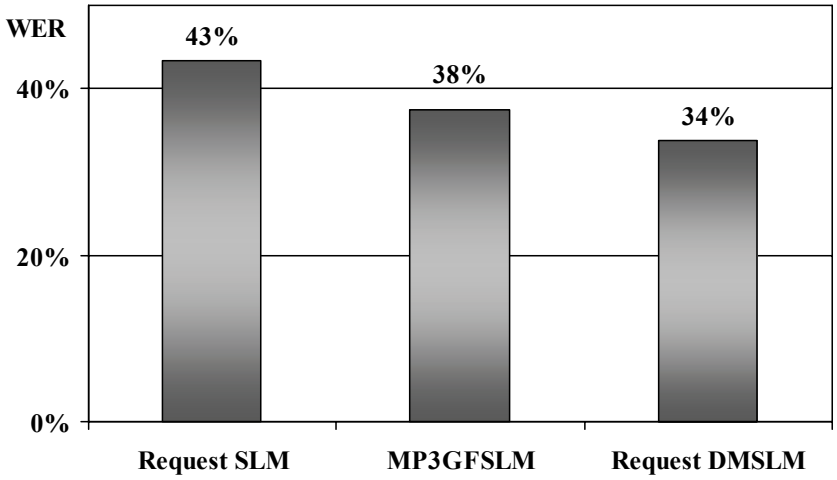

Figure 5.1: WER on Request test set

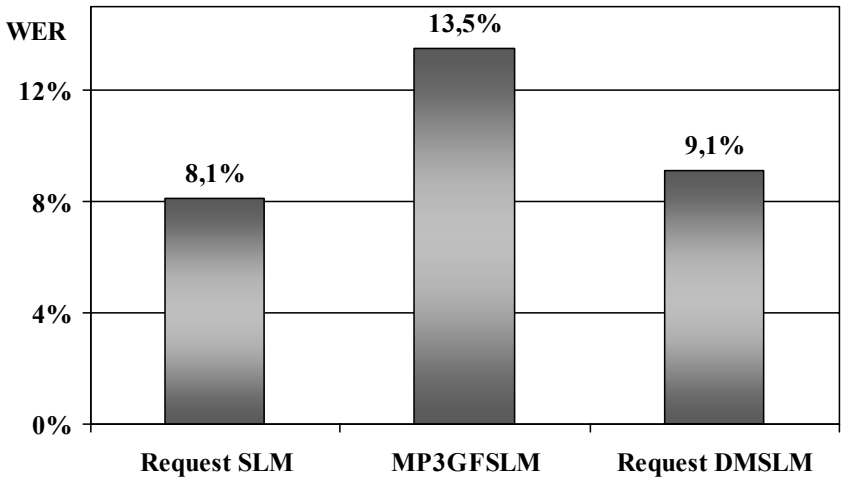

Figure 5.2: WER on in-coverage Request test set

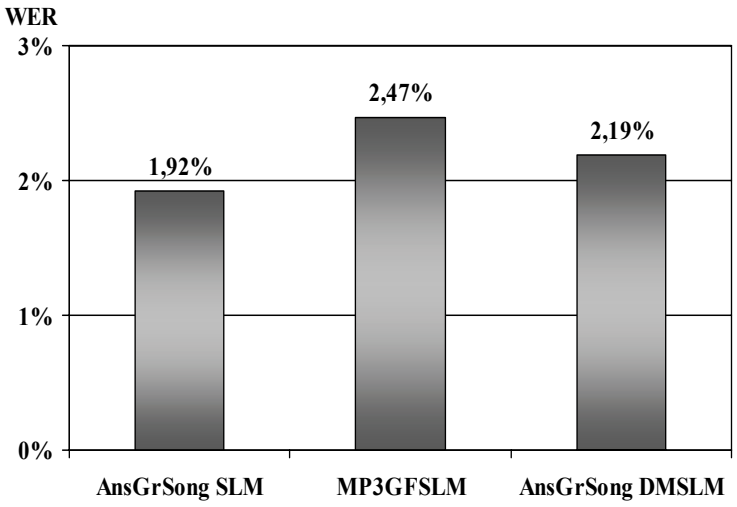

Figure 5.3: WER on Answer test set

although the difference with the `Request DMSLM` is minimal. The DMSLM thereby has the best performance overall.

In the same way, we tested the DMSLM boosting answers of group and songs against a test set with this type of answers. In this case all test utterances were in-coverage so we have only tested the `MP3GFSLM`, the pure DMSLM with only group and song answers and the DMSLM once (see Figure 5.3). In this case, the best performance is achieved with the pure DMSLM (`AnsGrSong`). However, we can also see that the DMSLMs profits from the boosting of answer moves and performs better than the general SLM.

In 5.4 and 5.5 we show the performance of the `YNDMSLM` and the `MP3GFSLM` on test sets with only yn answers. In this case we have not included the pure `YN` DMSLM in the test suite as it was built up from such a small corpus that the resulting SLM was not reliable. The results show us how we benefit from the boost in probability of the YN moves with an important reduction in WER for the `YNDMSLM` both on the general YN test set and the in-coverage YN test set.

5.3.4 Discussion of results

In agreement with previous research on context-specific language models as presented in Section 2.4.4.5 these first results show that by context specifying models we can achieve more accurate recognition. The proposed DMSLMs, which boost certain dialogue moves specific to a dialogue context, give an important gain in recognition performance. The improvement varies depending on the dialogue moves we are modelling giving us about 10% relative improvement for the DMSLMs modelling requests and answers while as much as 27% for the YN DMSLM. The generation method we use makes it easier to obtain context-specific language models and ensures that we have at least the same coverage as the original interpretation grammar. Collecting corpora for context-specific language

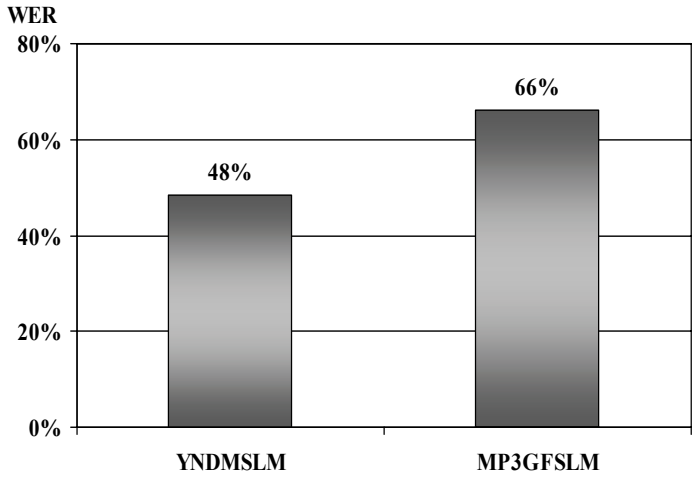

Figure 5.4: WER on YN test set

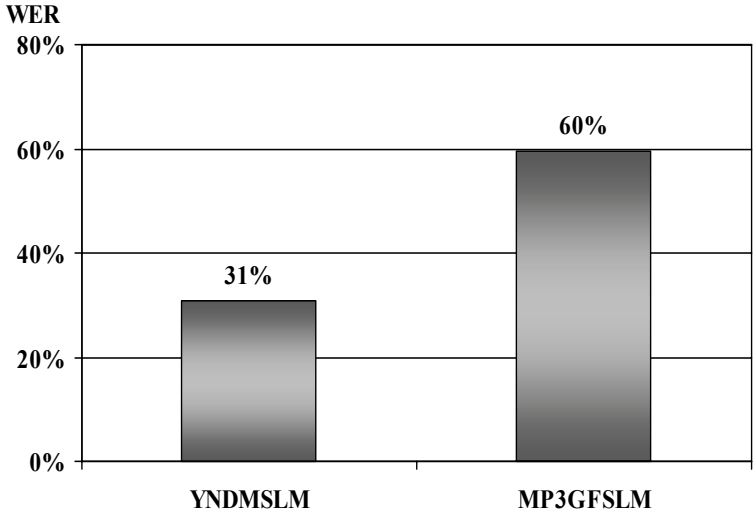

Figure 5.5: WER for the YN model on in-coverage test set

models has the drawback that sparse data turns into an even bigger problem and writing grammars for every state or move is tedious work. The first experiments seem promising but the dialogue move specific test sets are too small to draw any conclusions.

Ideally we would like to have our DMSLMs generated and interpolated on the fly. However, another solution would be to have certain models chosen from the beginning and change between these during dialogue interaction. In either case we need a way to be able to choose between them, i.e. choose which model suits best the current information state.

5.4 Second experiment: DMSLMs for the calendar domain

In this second experiment we investigate the proposed approach of generating context-specific SLMs from grammars more thoroughly by evaluating several DMSLMs created for the Calendar domain.

5.4.1 Dialogue moves in the calendar domain

As pointed out in Section 3.3.2, in GoDiS dialogue moves are activity related and exist in several different types. We have chosen to focus on four of the dialogue move types to build up our DMSLMs: the answer, ask, yn and request moves.

The decision to build these four DMSLMs was based on the distribution of dialogue moves in our data where these moves are the most common ones and the most critical for achievement of the dialogue tasks. As we only focus on the communicative function of the dialogue moves (the *dialogue move type* as introduced in Section 3.3.2) and exclude the propositional content we can treat them abstractly as domain-independent moves. This makes it possible to use a domain-independent prediction of the dialogue move types and thereby the language models. However, the content of the SLMs (i.e. the word occurrences) would be different in different domains.

5.4.2 Building dialogue move specific SLMs

GF was used to generate a corpus with all possible dialogue moves and their combinations with their corresponding expressions.[2] Our GF grammar defines 268 different dialogue move combinations. From this corpus we have extracted all utterances that can be interpreted as an answer move or a sequence of answer moves, all expressions for specification of a request, all ways of expressing questions in our grammars (i.e. ask moves) and all possible yn answers. This leaves us with four new sets of training data.

For each set of dialogue move specific training data we created an SLM that only captures ways of expressing a specific dialogue move: a "pure" DMSLM. However, we are looking for less constrained models which just alter the probability of certain dialogue

[2]This corpus was achieved thanks to Peter Ljunglöf.

moves. By interpolating the SLMs built on dialogue move specific corpora with the general grammar-based SLM we achieve models with contextual probabilities but that generalize to avoid constraining the user input.

Interpolation was carried out in the same way as in the previous experiment (see Section 5.3.2). The optimal lambda weight was estimated to 0.85 for all models with the SRILM toolkit using held-out data. We ended up with four new DMSLMs, in which either the probability of answer, ask, request or yn moves were boosted.

5.4.3 Test data

We used test data obtained with the data collection in Chapter 4 where test subjects interacted with the AgendaTalk system using the grammar-based SLM. The test group included both naive users with no experience of the system whatsoever and users that had previous experience with the system to varying extents. The data collection consists of a recording test set of 1000 recorded utterances from 15 persons (all native, 8 female, 7 male).

For the evaluation of the DMSLMs we partitioned the test data by dialogue moves. The utterances corresponding with the four dialogue moves chosen for our DMSLMs were divided into four test sets. The utterances left were used to create a fifth test set where none of our four DMSLMs would apply but where we would need to use the general model. Recall that the average utterance length was over three words. It is interesting to see how the average utterance length varies considerably over the different dialogue move sets from 1.6 words/utterance for the yn dialogue moves to 5.2 words/utterance for the ask moves. If we look at the distribution of the test data considering dialogue moves we find that 75.4% of the test data fall into our four dialogue move categories and that only 24.6% of the data would require the general model. This part of the test data includes dialogue moves such as greetings, quit moves and dialogue move sequences with combinations of different moves. The most common dialogue move in our data is an answer move or a sequence of answer moves resulting in common utterances such as: "A meeting on friday" as answer to system questions such as "What booking do you want to add?".

5.4.4 Experimental results

The recognition performance of the DMSLMs in contrast to the general grammar-based SLMs was evaluated on the dialogue move test sets as well as on a general test set. As in earlier experiments we report results on WER, SER and DMER (see Section 2.3 for introduction to the metrics). The DMER was obtained by parsing the recognized utterances and comparing these to a parsed version of the transcriptions, calculating the rate of correctly parsed dialogue moves. For parsing we have used the same phrase-spotting grammar as for the second experiment in Chapter 4 (see Section 4.3.3). The original GF interpretation grammar was not used for parsing as it would have restricted the parsing to the coverage of the grammar which is not an optimal choice together with SLMs. Ideally, we would like to use a robust parsing method with the GF grammar. Attempts to do this

have been carried out by the author in the TALK project for the MP3 domain by training a dialogue move tagger on the same type of corpus that was used for the DMSLMs where dialogue moves occur together with their corresponding utterances. This strategy is described in Chapter 7. Other methods of relaxing the constraints of the GF parser are also under consideration but out of the scope of this thesis. Meanwhile, we are using this simple but robust phrase spotting parser.

In the following sections we will report how our DMSLMs perform in comparison to the Grammar-based SLM and the Extended Grammar-based SLM from the second experiment in Chapter 4.

5.4.4.1 Dialogue move specific SLMs vs general SLMs

In a similar manner to the previous experiment we have evaluated our DMSLMs on test sets which include only utterances that correspond to the dialogue moves boosted in the model. These test sets may include utterances not covered by the original GF grammar, e.g. a different wording for the same move. The results for each DMSLM on its specific test set and the performance of the Grammar-based SLM and the Extended Grammar-based SLM (Extended SLM) are reported in Tables 5.2, 5.3, 5.4 and 5.5.

Table 5.2: Performance on Ask test set

Model	WER	SER	DMER
Grammar-based SLM	39.2%	68.4%	51.8%
Ask DMSLM	31.8%	68.9%	48.7%
Extended SLM	30.1%	58.0%	44.6%

Table 5.3: Performance on Answer test set

Model	WER	SER	DMER
Grammar-based SLM	17.3%	22.0%	16.3%
Answer DMSLM	15.7%	20.1%	14.1%
Extended SLM	18.2%	22.0%	16.7%

Table 5.4: Performance on Request test set

Model	WER	SER	DMER
Grammar-based SLM	29.1%	44.3%	27.0%
Request DMSLM	17.0%	36.1%	14.7%
Extended SLM	26.3%	42.6%	22.1%

We can see that the gain we get in recognition performance varies for the different models and that relative improvement in WER goes from 9% for the `answer` DMSLM to 42%

Table 5.5: Performance on YN test set

Model	WER	SER	DMER
Grammar-based SLM	37.3%	27.3%	22.7%
YN DMSLM	21.5%	16.5%	11.9%
Extended SLM	25.0%	18.2%	12.5%

for the yn DMSLMs. The reduction in WER when using the ask DMSLM from 39% to 32% is significant on the $p<0.001$ level. The gain in performance when using the answer DMSLM is relatively small (from 17% to 16%) and is not significant. Using the request DMSLM on the request test set gives a more important gain in performance and lowers WER from 29% to 17% (significant on the $p<0.0001$ level). The use of the yn DMSLM gives a reduction from 37% to 21% (significant on the $p<0.005$ level). As seen in the tables it is not only WER that is reduced when using our DMSLMs but SER and DMER are also affected. According to the results there seems to be a correlation of DMER and WER just as in the experiments in Chapter 4. Whenever we manage to lower the WER the DMER also goes down proportionally. In Table 5.3, 5.4 and 5.5 we can actually see that the DMER is actually lower than the WER which means that the recognizer seems to make some errors that are not semantically relevant and would therefore not affect the performance of a dialogue system. This was also true for the experiment on in-coverage utterances in the previous chapter.

We can see that our models have most problems with ask moves and yn answers. In the case of ask moves this seems to be because the original GF grammar is missing a lot of syntactic constructions of question expressions. This would then explain why the Extended grammar-based SLM gets a much better figure here. The GSLC corpus does capture more of this expressive variation of questions. In other words we seem to have failed to capture and predict the linguistic usage with our hand-tailored grammar. In the case of yn answers the result reveals that our grammar-based SLM does not have a realistic distribution of these expressions at all. This seems to be something the GSLC corpus contribute, considering the good results for the Extended SLM. However, we can see that we can achieve the same effect by boosting the probability of yes and no answers in our DMSLM.

Apart from these four dialogue moves our test data includes a lot of different dialogue moves and dialogue move combinations that we have not considered. As we have no specific model for these we would need to use a general model in these cases. This means that apart from predicting the four dialogue moves we have considered we would also need to predict when none of these are expected and use the general model for these situations. In Table 5.6 we can see how our general models perform on the rest of the test set. This shows that they seem to handle this part of the test data quite well.

Table 5.6: Performance on remaining test data

Model	WER	SER	DMER
Grammar-based SLM	22.2%	42.7%	31.7%
Extended SLM	19.6%	39.8%	26.0%

5.4.5 Discussion of results

If we look at the overall achievement in recognition performance, using our DMSLMs when appropriate and in other cases the general SLM, the average WER of 22% (27% DMER) is considerably lower than when using the general model for the same test data (29% WER, 33% DMER). If we had an optimal method for predicting what language model to use we would be able to decrease WER by 24% relative. If we chose to use the `Extended grammar-based SLM` in the cases our DMSLMs do not cover we could get an even greater reduction.

We have also tested how well our DMSLMs perform on the general test set (i.e. all 1000 utterances) to see how bad the performance would be if we chose the wrong model. In Table 5.7 we can see that this approach yields an average WER of 30% which is a minimal and insignificant degradation in comparison to the general grammar-based SLM. On the contrary, some of our models actually perform better than our general grammar-based SLM or very similarly. This implies that there is no substantial risk for recognition performance if our prediction model would fail. This means that we could obtain very good results with important recognition improvement even with an imperfect prediction accuracy. We have a relative improvement of 24% to gain with only a minimal loss.

Table 5.7: Performance of different DMSLMs on general test set

Model	WER	SER
Answer DMSLM	34.7%	55.6%
Ask DMSLM	28.2%	46.2%
Request DMSLM	26.5%	43.2%
YN DMSLM	29.8%	44.0%

5.5 Summary and conclusions

The experimental results presented in this chapter show that dialogue move specific SLMs (DMSLMs) give an important reduction in both WER (24% relative) and DMER (18% relative). This reaffirms earlier work on context-specific language models, see for example Baggia *et al.* (1997); Xu and Rudnicky (2000b); Lemon and Gruenstein (2004); Gruenstein *et al.* (2005), that by taking into account the statistical language variation during the course of a dialogue we can achieve more accurate speech recognition. The method we use

here has the advantage that we can build statistical context-specific models even when no data is available by generating annotated data from interpretation grammars and therefore assuring a minimal coverage. In addition, by interpolation with our general grammar-based SLMs from Chapter 4 we do not constrain the user input unduly.

To be able to choose which DMSLM suits the current information state best we need a way to predict dialogue moves. Optimally, we want a prediction model that we can use in different GoDiS domains to be able to generate new DMSLMs from our domain-specific GF grammar for the dialogue moves we have considered here. Such an approach will be considered in the following chapter (Chapter 6). The language model switch will be triggered by changing a variable in our information state: the predicted dialogue move.

If we take into account the experimental results for the AgendaTalk application from the preceding chapter (Chapter 4) we can estimate an overall reduction in WER of 46% and 40% in DMER if we were able to choose the best suited DMSLM instead of the SRG compiled from the GF grammar. Converting the GF grammar into a grammar-based SLM gave a WER of 29%. The SRG compiled from the same grammar had a much worse recognition performance with a WER of 39%. The results presented in this chapter shows that with an optimal method of choosing DMSLMs the WER falls to 22%. This is an absolute difference of 17%. The results also indicate that this drop in WER will propagate to the understanding performance. Naturally, we would have to take into account dialogue move prediction accuracy to get a more realistic figure. This will be considered in the following chapter. However, our experiments also show that the effect on performance if we failed to use the correct model would not be too harmful. This means we have much more to gain than to lose even if the dialogue move prediction is not perfect. This makes this approach a very interesting option in dialogue system development.

Chapter 6

Dialogue move prediction

"It is hard to predict ... especially the future"
Niels Bohr

If we could predict what the user of a dialogue system may do in his/her next turn we would give the dialogue system a chance to prepare itself for what is next to come. In this chapter we will explore *dialogue move prediction* and analyze what information may be important for the prediction of user dialogue moves. The ultimate goal of dialogue move prediction in this work is to use it to switch between the dialogue move specific language models (DMSLMs) presented in the preceding chapter but it could also be used for other tasks. We have used machine learning to predict user dialogue moves from information states. The training data has been extracted from dialogue logs collected with our two experimental dialogue applications. The first pilot experiment was carried out with DJ-GoDiS data and for the second experiment we have used AgendaTalk data.

6.1 Introduction

In pragmatic theory (Levinson, 1983) conversation is considered not to be random but to have a certain order which means that dialogue moves (acts) have some expectedness. However, Levinson points out that conversation flow is not restricted to adjacency pairs as dialogues can be quite complicated and nested so rather than taking for granted that an answer will follow a question we should commit to the view that an answer is relevant and expected after a question but that other dialogue acts can also occur. If we assume that the user of a dialogue system is cooperative and says things which are relevant in order to accomplish a task then we can experiment with ways to predict what it is most likely that a user will do in each turn, e.g. reply to a question, give some missing information or correct something that has been misunderstood. We will call this speculation about the near future: *dialogue move prediction*.

6.1.1 Related work

Dialogue act tagging has aroused great interest mostly in order to be able to annotate dialogue corpora (both human-human and human-machine) with dialogue acts automatically (Samuel *et al.*, 1998) but also in some cases to predict or decode the dialogue act of the user's last utterance in a dialogue system as shown in Stolcke *et al.* (2000).

The term dialogue move (act) prediction has sometimes been used in the literature in reference to the latter. In this thesis we are using the term *dialogue move prediction* to refer to the task of predicting what the user may do in her next turn without using any evidence at all from what actually happens in that turn. It is not a way to give a move assignment to utterances but a way to put some expectation into a dialogue system. The system should in this way have some idea of what might happen next and prepare itself for this.

The ultimate goal of dialogue move prediction in this work is to use the predictions to choose between the DMSLMs in Chapter 5. A survey of related work on context-specific modelling shows the sparseness of predictors used to choose models and the often hand-crafted rules that are employed. In finite state based dialogue systems the choice of context-specific language model or grammar is based on the current dialogue state. This approach is not possible in form-based or information state based dialogue systems where dialogue states are not explicitly defined. The most common approach is therefore to use the previous dialogue act as indicator of state, based on the assumption of adjacency pairs. In Popovici and Baggia (1997b) the prediction of SLM is dependent only on the system's last dialogue act and the range of possible user acts corresponding to each system act seems to have been chosen manually beforehand. In Lemon (2004) prediction of which SRG to use is made by checking what is called *the most active node* which corresponds to the last system move. Also in this case the appropriate grammar for each system move has been assigned manually. A study carried out by Holzapfel and Waibel (2006) drives the latter approach further by extending the categorization of system utterances and by using additional context information such as the information requested from the user (called the target) and information about dialogue goals to generate a list of semantically expected user contributions. Based on this list of expectations, rules in an SRG are weighted. This context dependent weighting of rules gives a considerable improvement in both WER and CER but the expectation model as well as the weighting are based on hand-crafted rules. There is no evaluation of the prediction accuracy alone but the improved ASR accuracy suggests that the expectation model is somehow acting properly. Probably one of the earliest attempts of prediction in the context of dialogue systems and speech recognition is presented in Young (1989). In the MINDS system (Young, 1989) predictions of conceptual concepts likely to occur in the next user utterance are generated based on contextual dialogue knowledge. These concepts are then extended to possible word sequences to produce semantic grammars and lexicons dynamically for use by the SPHINX recognizer. In this way, they manage to constrain the ASR search space and improve ASR accuracy. The prediction model takes into account objects in focus, anaphoric referents, a dialogue goal tree, coming plan steps in the goal tree, domain knowledge and even a user model.

The predictions are very elaborate and detailed. The system produces layers of predictions, ranging from specific to general. In this way the system does not restrict the user and has a more robust behaviour. When a specific prediction fails the system backs off to a less specific one. This means the system will first try to recognize the most specific predicted concept but if this fails it will go to the next level. The approach is very interesting and goes beyond the assumption of adjacency pairs but the elaboration of the rules seem to be very labour intensive and hard to port to other domains. Unfortunately, none of the above studies report any figures on how well their expectation or prediction model is performing apart from its effect on ASR accuracy.

We want to explore an approach that avoids hand-crafted solutions by using machine learning to train a dialogue move prediction model to be used to select appropriate context-specific SLMs. We want to train and evaluate this approach on real human-machine data and investigate possible predictors apart from the previous system move.

Related work on dialogue move prediction is scarce and machine learning does not seem to have been used for the task. We will therefore first take a look at related work in the area of dialogue act tagging. Transformation based machine learning (Samuel *et al.*, 1998; Lager and Zinovjeva, 1999), memory-based learning (Rosset and Tribout, 2005) and support vector machines (Surendran and Levow, 2006) have been used for dialogue act tagging. Other techniques used for dialogue act tagging have been statistical n-gram modelling (Stolcke *et al.*, 2000; Webb *et al.*, 2005; Reithinger and Klesen, 1997), HMMs (Venkataraman *et al.*, 2003), neural networks (Wright, 1998), maximum entropy models (Wright *et al.*, 1999; Ang *et al.*, 2005; Rangarajan *et al.*, 2007) and more recently graphical models (Ji and Bilmes, 2005). The best dialogue act tagging models have obtained an accuracy around 65-75% (Taylor *et al.*, 1998; Wright *et al.*, 1999; Stolcke *et al.*, 2000; Samuel *et al.*, 1998) but does not seem to improve much further than that providing evidence of the difficulty of the task of classifying utterances into dialogue acts. In Rosset and Tribout (2005) where the knowledge of utterance unit boundaries for each intention is assumed the tagging accuracy reaches as much as 80% but such an approach requires an automatic segmenter. Figures on human dialogue act labelling accuracy are given in Stolcke *et al.* (2000) reaching 84% using 42 distinct dialogue act labels on transcribed corpora (Switchboard). One of the best results reported for the transcribed Switchboard corpus uses lexical, syntactic and prosodic features in a framework using maximum entropy modelling and reaches the human inter-labeler agreement (84%) (Rangarajan *et al.*, 2007).

The results for different dialogue act tagging methods are very hard to compare as different dialogue act taxonomies have been used varying in both size and specificity and also because the taggers have been trained on very different kind and size of corpora. In the experiments carried out on the Verbmobil Corpus (Reithinger *et al.*, 1996) 18 high level and 43 specific dialogue acts are identified. The Switchboard corpus (Stolcke *et al.*, 2000) distinguishes 42 dialogue acts with the DAMSL taxonomy and the MapTask corpus (Taylor *et al.*, 1998) only 12. However, although having a smaller taxonomy the MapTask corpus seems to be trickier and the tagging accuracy results reported are much lower ($> 10\%$) than for the Switchboard corpus when applying the same approach (Rangarajan *et al.*, 2007). In some cases manually transcribed human-human corpora have been used

and in other cases models have been trained on automatically recognised human-machine corpora. Studies show that accuracy goes down by around 10% when applying an approach to ASR output instead of transcribed speech (Ang *et al.*, 2005). In Reithinger and Klesen (1997) it is shown that dialogue act tagging performance is language-dependent and that the approach they use perform better for English with a more strict word order than for German. They also show how the tagging accuracy differs for different dialogue acts and that dialogue acts such as greetings are easily detected whereas more uncommon dialogue acts are harder to recognize. Another issue when comparing different approaches to dialogue act tagging is the type of evidence provided and used. This results in very different task complexity and it is hard to draw any conclusions from the experiments other than that the assignment of dialogue acts to utterances is a very complex task in general with the need of more information than just the utterance words as the function of an utterance is highly dependent on the dialogue context. We also get an indication that the task of dialogue move prediction will not be much easier.

To improve the figures for dialogue act tagging, researchers have tried to employ diverse knowledge sources apart from the lexical information in the user utterance. The most common approach has been to include prosodic information (Taylor *et al.*, 1998; Wright, 1998; Rangarajan *et al.*, 2007; Venkataraman *et al.*, 2003). Another approach has been to take advantage of expectations on what comes next in a conversation (Poesio and Mikheev, 1998). This seems to relate to what we want to do with dialogue move prediction. In the Verbmobil project (Reithinger *et al.*, 1996) the next dialogue act was predicted statistically by using n-gram models (trigrams) trained on a dialogue history of dialogue acts. They used manually annotated logs for training and were able to predict 18 distinct dialogue acts with an accuracy of around 40% based on previous seen dialogue acts. Their dialogue act tagging accuracy when including this dialogue act prediction model rose 3% reaching 75% (Reithinger and Klesen, 1997). One of the first approaches to statistical dialogue move prediction was probably Nagata and Morimoto (1993) who reported prediction rates of 40% for prediction of 15 dialogue acts (intention types) using an SLM trained on manually speech act annotated dialogues. When considering the top three predictions the accuracy was 62%. In a suite of related studies (Wright *et al.*, 1999; Wright, 1998; Taylor *et al.*, 1998; Poesio and Mikheev, 1998) on the Map Task corpus prediction features were included in the task of dialogue act classification. The SLM used for prediction was trained on manually annotated dialogues and inluded the identity of the current speaker, the speaker of the previous move and the last move of the previous speaker (Wright, 1998). In addition an intonation model was used taking into account information about the acoustics of the user utterance. This hybrid approach classified 12 distinct move types with an accuracy of 64%. The intonation model alone had an accuracy of 42%. In Poesio and Mikheev (1998) and Wright *et al.* (1999) it was shown that dialogue game information, i.e. knowing the type of dialogue game and the position in that game, could also be an important predictive feature. In Stolcke *et al.* (2000) the best model classified utterances into 42 distinct dialogue acts on the Switchboard corpus with 71% accuracy. This approach used n-gram models for the words in the utterance, for intonational features as well as for dialogue act sequences, i.e. taking into account the dialogue history. Interestingly both the work in Taylor *et al.*

(1998) and Stolcke *et al.* (2000) show that models for dialogue act classification can also be used to condition a speech recognizer.

In this work we will use machine learning to predict the next user move by using the information that we will have at runtime in the dialogue system before the user's possible turn. We will not use annotated data but automatically produced data and we will not take into account any clues from the realization of the user move (such as acoustic evidence). This will result in a data set with some noise but it will also make it possible to use much more information and not be restricted to the information seen in annotated data. We will be able to use the dialogue system's whole information state and not just the previous moves and speaker identity to explore what features play a significant role in dialogue move prediction. We have carried out a pilot experiment on the MP3 player domain and then continued our investigation with a more elaborate experiment in the Calendar domain.

6.2 First experiment: Predicting dialogue moves in the MP3 domain

The purpose of the first experiment was to achieve more knowledge of what type of information is critical for the task of dialogue move prediction and if we could obtain sufficient accuracy to be able to use dialogue move prediction to predict the proposed DMSLMs from Chapter 5. Also, in this first study we wanted to investigate how many different dialogue moves it would be feasible to distinguish and how the distribution of dialogue moves looks in our dialogue system. To carry out this investigation we used material collected with the DJ-GoDiS system (see Section 3.4.1). We used the memory-based learner TiMBL (see Chapter 3, page 86, for introduction) and experimented with different features and different algorithm parameter settings to investigate the optimization and variability of the accuracy of dialogue move prediction.

6.2.1 The data

The training experience the machine learner had was dialogue logs from interactions with the DJ-GoDiS dialogue application where the dialogue flow and each information state were automatically annotated. For an introduction to the TrindiKit log format and the GoDiS information state see Chapter 3.

These dialogue logs were converted into a format that TiMBL could read. The resulting data set consisted of 514 information states. Although the number of dialogue moves in our system is quite small the possibility of combining these in the same turn makes the possible classes of move sets per turn, encountered in the dialogue logs, reasonably large. In the original data set there were 40 different move combinations associated with turns. We distinguish the fifteen moves presented in Table 6.1 and combinations of these.

Table 6.1: Dialogue moves used in the MP3 domain

Dialogue move	Utterance example
Greet	Hi!
Quit	Bye!
Help	Help.
Answer(Song)	Dancing Queen
Answer(Station)	Rant Radio
Answer(Index)	number three
Answer(Group)	Abba
Ask	who wrote this
RequestControl	pause the music
RequestList	clear the playlist
RequestHandle	i want to manage the playlist
YN	yeah
Top	restart
ICM	OK
NM	i.e. no move

6.2.2 Feature selection

The features considered for the experiment were chosen from the information available in the information state in the dialogue logs. For this experiment we have focussed on the information in the shared part of the information state, i.e. information which has been established or grounded during the conversation (see Section 3.3.1 for a description of the GoDiS information state).

The features selected were the previous move (PM) (i.e. the move before the current system move), the information in shared commitments (SHCOM), the shared actions (SHACT), the current question under discussion (QUD) and the current system move (NxtDM) to be realized.

Other features that may be interesting for the task, which were not possible to obtain for this experiment, are the previous speaker before the current state, the number of the current turn or the position in the dialogue (e.g. a greet move being more likely in the beginning of the dialogue and a quit move after a number of turns), the confidence score of the previous user move and information about the state of the GUI and of the device (e.g. if the music is on or off). We could, of course, also have included the system utterances but that would have made our classifier dependent on the exact wording of the system utterances, which may undergo changes as the system is developed, and also on the language used. As it is now, we could use the same classifier for both the Swedish and English versions of our system and we could collect logs from both versions for our training and testing.

An example instance of a dialogue state represented by the five selected features and

the dialogue move combination performed by the user is shown below:

ReqList, =, Add, WhGroup, ICM@ICM@AskArtist, AnsGr@AnsSo.

This corresponds to a dialogue state where:

(23) PM: The previous move was a request concerning the playlist (ReqList)

SHCOM: There are no shared commitments (=)

SHACT: There is a shared action of adding something to the playlist (Add)

QUD: The question under discussion is what group to add (WhGroup)

NxtDM: The system move next in turn is a combined move of grounding moves (ICM) and a question about what artist is under consideration (AskArtist).

PredDM: The user move in this case was a combination of two answers: the name of a group and of a song.

6.2.3 Experimental results

We divided our data into a training set and a test set, where the test set made up a little more than 10% of the data. We created a baseline result by taking our test set and assigning the most frequent move to all information states. This gave us a baseline of 22 % accuracy used for later comparison.

6.2.3.1 Feature optimization

First, we tried to investigate which of the features we had chosen seemed to be the most informative for the task. Our first thought was that the system's move to be produced just before the user turn (NxtDM) would be the most prominent feature but this turned out to be a false assumption. This makes the experiment even more interesting as it shows that the information in the information state other than the current system dialogue move plays an important role in dialogue move prediction.

We ran TiMBL with the default settings (IB,GR,Overlap, k=1) changing the number of features by using TiMBL's facility for ignoring certain features. The original setting gave a dialogue move classification accuracy of 52.5%. The results for the different settings are shown in Table 6.2. This shows that the most important feature was the QUD and that the two first features of our feature set the PM (previous move) and the SHCOM did not give much benefit to the result and that taking these away gave us better results. The NxtDM feature did have an important impact on the result although not as much as the QUD. The best result obtained, with a feature setting where the PM and the SHCOM were excluded, was an accuracy of 56%.

Table 6.2: Prediction accuracy for different feature sets

PM	SHCOM	SHACT	QUD	NXTDM	ACCURACY
X	X	X	X	X	52.5
X					40.7
	X				22.0
		X			35.6
			X		54.2
				X	45.8
	X	X	X	X	54.2
X		X	X	X	52.5
X	X		X	X	52.5
X	X	X		X	49.2
X	X	X	X		52.5
		X	X	X	56.0
			X	X	56.0
	X		X	X	54.2
X			X	X	52.5

6.2.3.2 Parameter optimization

We also investigated how the accuracy was affected by using the different algorithms that TiMBL provides and got the following results with default parameter settings:

Table 6.3: Prediction accuracy for different algorithms

Algorithm	Orig. Features	Opt. Features
IB	52.5	56.0
IG	50.8	52.5
TRIBL	52.5	52.5
TRIBL2	50.8	52.5

Changing from Overlap to MVDM for the IB algorithm and a higher K (3) gave an improvement for the original feature selection (from 52.5% to 54.2%) but did not improve the best result we obtained with the optimized feature set (taking away feature 1 and 2). We tried with different distance measures but did not get any further improvement.

We also tried different weightings for the different algorithms. The best result was IG with shared variance but the default setting (IB, GR) did not change by changing the weightings as seen in Table 6.4. We did the same test with the optimized feature set getting similar results as reported for the algorithm optimization. No setting improved our 56% accuracy with the default setting and the optimized feature set.

Table 6.4: Prediction accuracy for different weighting methods

Algorithm	NoWeighting	GainRatio	InfoGain	ChiSquare	SharedVariance
IB	52.5	52.5	52.5	52.5	52.5
IG	47.5	50.8	49.1	49.1	54.2
TRIBL	52.5	52.5	52.5	52.5	52.5
TRIBL2	49.1	50.8	52.5	52.5	52.5

6.2.3.3 Optimizing the data representation

The results so far show that the optimization of parameter settings do not make much impact on accuracy but that the gain obtained was primarily from changing the feature selection. The last optimizing attempt consisted of reconsidering the representation of the data set to see if this could yield any performance gain. Therefore we have tried to optimize our data representation by correcting some errors and by merging some of the classes. However the results obtained from this new data set are not comparable with earlier results as the classification task will be different.

First of all, it should be noted that there is noise in the training and test data due to recognition failures and this causes failure to recognize the correct user move. This has given rise to some odd dialogue states. The most obvious ones (e.g. the same dialogue move repeated in the same turn) have already been corrected but there is still a lot of noise left. As the aim is to prevent this noise (i.e. recognition failures), by predicting the next dialogue user move and choosing an appropriate language model for the current dialogue state, the test situation is a little unfavourable to the task. Therefore we have tried to manually correct the test set by listening to what the users really said (and judging what move they performed) so that it corresponds to plausible dialogue situations.

To generalize better we have taken away the feedback moves (ICMs) in the cases where these were combined with other moves in the feature NxtDM. This gave us a slight improvement. We also merged some classes into one to see if a smaller set of classes would make a difference. We merged all answers of songs and groups and their combination into one single class. This is more appropriate to a real setting where these moves occur in the same dialogue situations. A final test was done after correcting the data set by classifying dialogue move sequences that had the same combination of moves but in a different order as identical classes. This gave us a final set of 28 classes and a final result of 55.69% accuracy on the original feature set with default parameter settings.

A final optimization was done by merging the three request classes into one as the requests seem to appear in the same kind of states and are often combined together. This gave us a class set of only 19 classes which our dialogue move predictor classified with an accuracy of 59.38% on our test set. We did the same parameter and feature optimization as with the original data set and got an optimal accuracy of 65.63% when feature 1 and 2 was excluded and MVDM was used with Inverse Linear weigthing (default k). The most important features here were also NxtDM and QUD with a slightly higher information gain

for the QUD feature. However, more experiments are needed to set on a definitive set of classes by also taking into account how many different language models we want to have and how these perform.

Apart from testing against the test set we chose to do a leave-one-out test in case the test set that had resulted was too "special". The results were much better with the leave-one-out test showing that our test set had disfavoured our results a little. We obtained an accuracy of 67.51% with default parameter and feature settings. Optimizing did not improve this result significantly leaving us with a best result of 67.51% accuracy in dialogue move prediction and an F-Score of 63.25 by using the information state and classifying 19 different classes.

Another thing to keep in mind when looking at the results is that there are no uniquely correct matches of a dialogue move and a state as a user can choose between several possible moves in each state. What we want is a classifier that can predict the most plausible move (or moves) to help us choose an appropriate language model. To get a better idea of how our classifier is working we need to look at its top choices and see if one of these corresponds to the user move which was actually realized. Using the TiMBL verbose option "db" gave us results where this could be investigated and from which we could calculate a more appropriate accuracy score for the task. By only looking at the 1-Best result we got an accuracy of 67% percent. However, by considering the two best choices the classifier gives us we get an accuracy of 75%. Looking at the classifier's three best choices to see if the correct class is among these gives us an accuracy of 81%.

6.2.4 Discussion of the results

The results show a considerable improvement in comparison with the baseline. However, the optimization of parameter settings did not give much impact on accuracy but the gain obtained was primarily from changing the feature selection or reconsidering the data representation. Another influence on the accuracy that we have not tested is the amount of training data. The amount of data used is very small, only 514 dialogue states, but more data could easily be obtained and transformed by running the dialogue system. It would be interesting to see what impact the amount of training data would have on the outcome.

It is hard to compare the results to earlier work as different types and different number of classes were used but it seems that our classifier is doing quite well given the extremely small training set. Reithinger *et al.* (1996) predicted 18 distinct dialogue acts with 40% accuracy and Nagata and Morimoto (1993) achieved the same accuracy for 15 dialogue acts. The dialogue move predictor presented here classifies 19 dialogue moves (acts) with 67% accuracy. With a majority baseline of 36% in Nagata and Morimoto (1993) the actual improvement in that study was quite low. For our data with a majority baseline of 32% the improvement is substantial. A difference is that our training set is much smaller and consists of automatically logged human machine dialogue including some noise while the other two studies used manually annotated dialogues.

It would be interesting to use position information as in Wright *et al.* (1999) by auto-

matically annotating position information in the information state (e.g. start, middle). As it is possible to come back to the starting information state during the dialogue it seems that position information would be a good way to separate these states (e.g. it is not very plausible that the user would make a greet move other than at the beginning of the dialogue).

It should also be remarked that the MP3 domain is not optimal for this kind of task as the dialogue is very direct and shallow and does not result in deep nested dialogues where prediction would be more useful. In the opening state the user could say almost anything and prediction is almost impossible. We can therefore expect our classifier to give very different results for different domains and that the features in the information state will play a different role. Still, the results for this domain are relatively good which seems to depend on the ease of predicting when the user is not going to perform any move (the No move class) and the ease of predicting some of the moves in specific states. As the classifier can give confidence scores on its guess we could use the dialogue move predictor already, even with the accuracy we get, relying on it when the confidence score is high and ignoring it otherwise. In the first case we would choose an appropriate specific language model for recognition and in the second case we would use the general language model.

6.3 Second experiment: Predicting dialogue moves in the calendar domain

The preceding experimental results show that it does not look impossible to tackle such a complicated problem as predicting the next dialogue move. However this would only be useful if we could generate DMSLMs on the fly combining dialogue moves in the way we like to create dynamic models where some dialogue move probabilities are boosted or by weighting rules in an SRG. In the following experiment we will embark upon a much more simple case: to predict one of the five DMSLMs that we developed in Chapter 5 for the AgendaTalk system. If we could manage to do this we would be able to integrate this dialogue move prediction model into our current system and switch between DMSLMs. The following sections will describe how we have trained our dialogue move prediction model for the AgendaTalk system.

6.3.1 The data

The data used for this experiment were collected with naive users as well as experienced users as described in Chapter 4 and Chapter 5. We extracted 1922 dialogue user turns from the data giving us 1922 user dialogue moves with adjacent information states. The data represent dialogues both in English and Swedish which are represented in a language independent way. The data were collected automatically and no manual processing has been done. This means that the data may be corrupted by speech recognition errors. The data collection was performed with the grammar-based SLM described in Chapter 4 which reported a WER of 30%. It is possible that word errors have in some cases propagated to

the dialogue move interpretation giving us incorrect dialogue moves. The dialogue excerpt below (24) exemplifies such a case from the data where the user herself seems to realize that the first utterance would be hard for the system to capture and thereby repeats the utterance in a more clear way. However, the system has already misrecognised the first attempt as a quit move and is on its way to finish the dialogue. USR> here stands for the real user utterance whereas ASR HYP> is the ASR's recognition of that utterance.

(24) SYS> Så, vad vill du göra? *Eng. So. What do you want to do?*
SYS DM: ask(X^action(X))
USR> Vad gör jag på torsdag? *Eng. What do i do on Thursday?*
ASR HYP> Avsluta på torsdag *Eng. Quit on Thursday*
DM HYP: quit + answer(date(thursday))
USR> Vad gör jag på torsdag? *Eng. What do i do on Thursday?*
ASR HYP> Vad gör jag på torsdag? *Eng. What do i do on Thursday?*
SYS> Ok. Ha en bra dag. *Eng. OK. Have a nice day!*
SYS DM: quit

Another noise factor is barge-in which will leave us with a dialogue move which perhaps is not clearly related to the current information state as GODIS may not have had time to make all necessary updates. A related issue is synchronization as it is not always the case that the system has had time to produce a dialogue move as a spoken utterance when the user starts talking or it could be the case that the user may not have listened to the system's complete contribution before starting to talk. This is due to the lack of a more advanced turn-taking algorithm. The following example (from the logs) shows such a mismatch where the user does not respond to the system's contribution in turn one until turn four (example 25).

(25) SYS> På eftermiddagen, är det korrekt? *Eng. In the afternoon, is that correct?*
USR> Börja om *Eng. restart*
SYS> Okej. Låt se. Vad vill du göra? *Eng. OK. Let's see. What do you want to do?*
USR> Nä *Eng. No*

This makes the logs corrupted as the system does not take into account whether the user heard or understood the system's dialogue move before making his/her contribution and therefore performed a dialogue move that may seem irrelevant to the logged information state. Apart from this, there is some noise and crosstalk in the data that has triggered the ASR and resulted in seemingly irrelevant dialogue moves. Also, in some cases the users did not collaborate much with the system to give missing information or confirm or correct things but instead of finding out what the system had understood, they repeated their contributions as soon as they did not get any response on the screen, i.e. graphical

feedback in the form of the system performing some action on the schedule. As barge-in was available this could mean that the user talked several turns in a row, holding the floor, without giving the turn back to the system. This is probably an effect of having a multimodal system where the main focus is a graphical interface. The following example (in 26) shows such a situation where the user first responds to the system's request for explicit confirmation by correcting the system but then also gives additional information which the ASR does not manage to capture very well. This makes the system confused and in the subsequent system turn we can see that the dialogue has not advanced any further as the ASR has again captured an incorrect time.

(26) SYS> Klockan nio nollnoll, är det korrekt? *Eng. at nine sharp, is that correct?*
SYS DM: icm:und*int:usr*start_time_to_store(900)
USR> nej nej nej öh klockan tolv blir bra *Eng. no no no uh twelve is fine*
ASR HYP> nej nej nej till klockan tolv tjugo *Eng. no no no to twelve twenty*
DM HYP: answer(no) + answer(no) + answer(no) + answer(time(1220))
USR> så första måndagen i månaden *Eng. so the first Monday of the month*
ASR HYP> nio torsdag till måndagen tjugo morgonen *Eng. nine Thursday to Monday twenty the morning*
DM HYP: answer(time(900)) + answer(date(thursday)) + answer(newdate(monday)) + answer(time(2000)) + answer(am_or_pm(pm))
SYS> Klockan nio nollnoll, är det korrekt? *Eng. at nine sharp, is that correct?*
SYS DM: icm:und*int:usr*start_time_to_store(900)

In addition, as some data was collected with an early version of the system there are some cases of mismatch between ASR and parsing coverage where the ASR has captured the word sequence but the parsing is not able to give a correct dialogue move to the sequence. This was especially the case for some question formulations where the parser did not identify the ask move but only gave back the identified slots as answer moves as in 27.

(27) ASR HYP> I want to ask about when the meeting is on Friday
DM HYP: answer(event_to_store(meeting)) + answer(date(friday))

In other cases it was the ASR that failed to capture all words in the question expression and the parser was not robust enough to be able to interpret it as a question (ask move) without the missing words. Due to these mismatches the percentage of ask moves is much lower than the figures given in the SLM experiment and this is also why the percentage of answer moves gets higher as shown in the following section. The example below (in 28) shows such a case where the first user utterance has been misrecognised by not capturing the personal pronoun.

(28) SYS> Okej. Låt se. Vill du fråga om tiden för en bokning, fråga om vilket datum en bokning ligger, fråga om vad som finns bokat eller fråga om dagens

datum? *Eng. OK. Let's see. Do you want to ask about the time of a booking, ask about the date of a booking, ask about what is scheduled or ask about today's date?*

USR> vad har jag bokat klockan ett *Eng. what do I have booked at one o'clock*

USR> på eftermiddagen *Eng. in the afternoon*

ASR HYP> vad har bokat klockan ett *Eng. what have booked at one*

ASR HYP> eftermiddagen *Eng. the afternoon*

DM HYP: answer(time(100)) + answer(am_or_pm(pm))

An interesting phenomena here is that the user is giving his/her contribution stepwise. The ASR has identified an endpoint after "ett" due to a long pause by the user but the ASR is triggered again when the user continues talking adding "eftermiddagen". The two ASR results are stored on an input buffer from which the interpretation module reads. This enables the interpretation module to output a dialogue move sequence based on both utterances. This barge-in behaviour in GODIS was developed in the TALK project and seems to have been something that the users in our study got used to very quickly and took advantage of. However, it also seemed to cause some problems to the system as some users kept taking the floor and did not give the system a chance to ground its understanding of the user's contributions. Anyway, what is shown with this example is that the system does not manage to identify the user's question but only the time expression.

Another problem with the logs was that the collection was carried out long before the dialogue move prediction experiment was planned and therefore the logging algorithm was not set for the task and did not always log the exact decision-making moment needed for dialogue move prediction, i.e. just before the system will perform its move. This means that the information state in the logs sometimes does not look exactly as it will look when we predict in the actual system. When extracting the features for the machine learning experiments we have tried to adjust for these cases.

To avoid our results becoming too corrupted by all these noise factors we extracted a test set from the data that we corrected manually based on the transcriptions in Chapter 4 of the collected audio files. This means that although we trained on the corrupted data we at least tested on data where we knew what the users actually did and not what the system thought they did. The test set was taken from a data collection with naive users.

6.3.1.1 Selected dialogue moves to predict

In accordance with the choice of DMSLMs in Chapter 5 we first chose to predict these five DMSLMs or rather the four dialogue moves (with combinations): Request, Answer, Ask and YN in addition to the general SLM which is any other move apart from these. However, when studying the dialogue flow in the logs we could see that the number of moves falling outside these classes was extremely low (4%) and the moves were not very homogeneous. Also, these moves did not seem of great interest for improving ASR, being dialogue moves such as for example greetings or ICMs for negative perception (e.g. I did not hear you!). However, the quit move could be very interesting to predict, i.e. being able to predict the

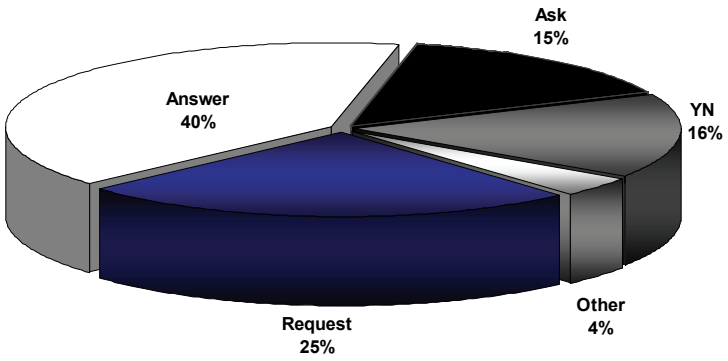

Figure 6.1: Dialogue move frequency

user's desire to end the dialogue. But it seemed really hard as the move appeared in very distinct dialogue situations. Apart from this, there seem to have been some misrecognitions of moves as quit moves when the user actually wanted to continue the dialogue (as shown in example (24)). In any case, the following experiments will show classification results both with and without these moves. The few ICMs for negative perception in the data were removed as it seemed impossible and not useful to try to predict when the user would for example not hear the system. If we could predict such a situation we would rather try to prevent it happening rather than trying to improve the recognition of the user's reaction to that situation.

In contrast to the experiment for the MP3 domain a dialogue move combination such as request(add) + answer(event(meeting)) would here be classified as a request. We do this to minimize the number of SLMs. In order to analyse our choice of dialogue move classes we will take a look at the dialogue move frequencies in our test data.

Figure 6.1 shows the distribution of dialogue moves in the data. The figures are not exactly comparable to the ones given in Chapter 5 as the dialogue moves there were manually transcribed and the data is only partially the same. We can see that the frequencies for yn answers are similar (16% vs 18%) but the ask moves, i.e. the questions, are much fewer (15% vs 21%). As mentioned earlier this is partially due to a parsing restriction in an early version of the system but also as was seen in the DMSLM experiment that our ASR model had a hard time recognizing questions so there are probably a lot of questions missing here that the users actually performed and the frequency of answers is therefore overestimated.

The most common dialogue move in our collected data is the answer move followed by the request move. 96% of the dialogue moves fall into one of our four basic categories which shows that with these four dialogue moves we manage to cover most of the data. It does not seem worthwile to take into account the different dialogue moves that fall outside these categories. However, a finer distinction of our classes, e.g. separating different kinds of answers, could have been fruitful as shown in the previous experiment for the DJ-GoDiS domain.

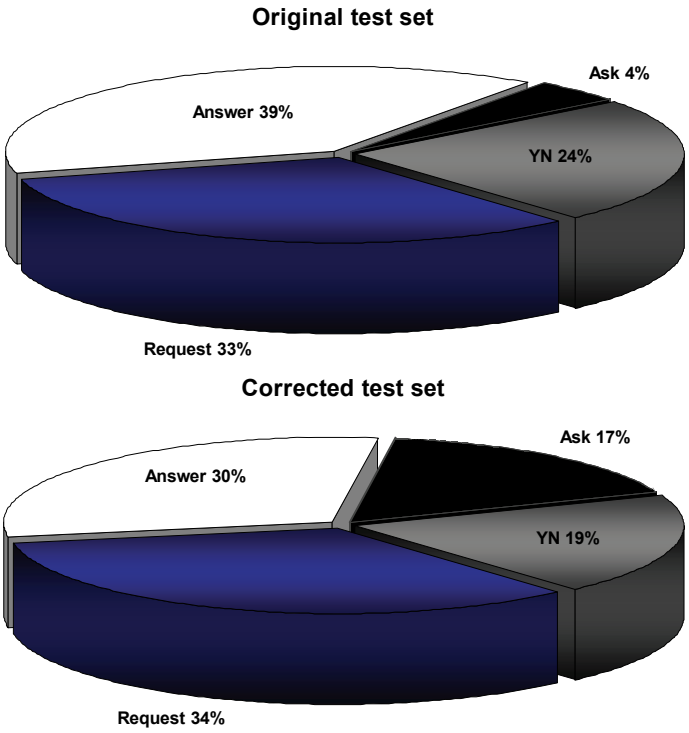

Figure 6.2: Comparison of dialogue move frequency

In order to get an idea of the impact of noise on our data we will compare the distribution of dialogue moves between the test set in its original form and after manual correction where user dialogue moves are based on what the users actually said.
The result of the comparison in Figure 6.2 shows us that our assumptions were correct and that it clearly seems that ask moves have been misidentified for answer moves and that the frequency of questions in our data is much higher than reported. This comparison thereby puts evidence to the noise factor showing that it has relevance and skews our data considerably. Unfortunately, this will affect our prediction accuracy as the prediction model will favour answer moves.

6.3.2 Feature selection

In this experiment we could take advantage of the fact that the logging algorithm had been further developed in the TALK project and had given us more detailed logs containing much more of the information that would be available at runtime. Also, the information state in AgendaTalk had been extended with new fields and variables in comparison to the DJ-GoDiS application such as a dialogue history, a holder for referential objects etc. (these changes were introduced in Section 3.4.2.5). The features selected and extracted or derived from the data represent the knowledge at hand at the moment the system has selected what dialogue moves to perform next. This is the moment when it would be

possible to change language model, just before the system decides on how to perform a dialogue move and before realizing it. It is in other words the perfect moment for dialogue move prediction. Originally, we started with fifteen different features obtained from logged information states. These are enumerated below:

(29)
LSP: Latest speaker
LM: Latest moves
PM: Previous moves
FSTCOM: First of shared commitments
SHCOM: Shared commitments
SHACT: Shared actions, i.e. previous actions that were agreed on in the last turn.
SHISS: Shared issues, i.e. previous issues that were agreed on in the last turn.
QUD: Questions under discussion
FSTPLAN: First findout in current plan
ConfScore: Confidence score
DiaLen: Dialogue length
DiaHis: Dialogue history
DREF: Referential objects
NXTDMs: Next system moves to be performed
LNXTDM: The last move of next system moves to be performed

Many features have been changed to a form better suited for machine learning and some have been transformed to a more abstract representation. An example is the shared commitments (SHCOM) where a proposition such as **time(1200)** would be represented by **time** as the importance is that the dialogue participants have grounded a time and not what time that was. In this way the data is clustered and grouped together to get less fine-grained feature values. This abstraction and transformation of the information state values was carried out with a Prolog program implemented specifically for this task that could easily be reused in the GoDiS system run-time to extract the same types of feature values.

Of these 15 features only 8 features seemed to be of importance for dialogue move prediction: LSP, LM, SHCOM, SHACT, SHISS, QUD, NXTDMs, LNXTDM. We will call these the selected features. The fact that the system's moves would be of importance was expected but it is interesting to see that again the information state features SHCOM, SHACT, SHISS and QUD make a contribution. Information about previous moves or about the dialogue move history did not seem to provide any benefit here meaning that the learner does not manage to see any repeated patterns of longer dialogue move sequences that impact the choice of user dialogue move. This could be due to the short plans which are typical for the domain where the user performs one task in a few turns and then starts a new task often independent of the previous task. It could also be due to the

shortage of training data which does not hold enough repeated patterns of longer dialogue act sequences. We also have to remember that the dialogue act sequences in the dialogue history may be skewed due to misrecognitions and misinterpretations. Therefore, we do not want to jump to the conclusion that information of turns several steps back are not helpful in dialogue move prediction although we will not make use of that information. Also, the features we discussed as potential feature candidates in the previous experiment (see Section 6.2) such as keeping track of the length of the dialogue or the confidence score did not give us any benefit.

An example of a training instance using the 8 selected features and classified with the prediction of an answer move is shown below to illustrate the form of the training data.

usr, request, olddate-event_to_store, change_event, newtime, [],
(icm:acc*pos)-((icm:loadplan)-(icm:und*int)), icm:und*int, answer.

This corresponds to a dialogue state where:

(30)
LSP: the latest speaker was the user
LM: the latest move was a request
SHCOM: the shared commitments are a **date** and an **event**
SHACT: the shared action is to change an event
SHISS: there is a shared issue of finding out the new time of that event
QUD: there is no current question under discussion ([])
NXTDMs: the next moves that the system is going to perform are some grounding moves (ICMs) including an interrogative ICM
LNXTDM: the last move of the next moves that the system will perform is an interrogative ICM corresponding to an expression such as "X, is that correct?"
USRDM: The user's contribution when the next moves had been realized by the system was an answer move

In this example we can see that we have an information state representing a dialogue where the user is trying to change the time of an event. The system and the user have managed to ground the type of event and also the current date of that event (olddate). In the previous move the user probably specified this request about changing the event. The current issue is to find out what new time the user wants for the event. It seems that the system has understood something (perhaps a time) with a low confidence score as it has chosen as it's next move to confirm this explicitly by producing an interrogative ICM. However, the user in this case does not respond with a yes or no question which could be expected but gives new information in the form of an answer. It could perhaps be a form of correction by the user by providing a different time expression than the one the system has proposed as understood.

Our data consisted of 1922 instances of this form divided into a training set of 1647 instances and a test set of 275 instances. The test set was taken only from logs from naive

users to make the test setting more realistic. The feature set chosen is independent of language and the training data was extracted both from Swedish and English logs with the hope of being able to train a prediction model that can be used both for the Swedish and English versions of the AgendaTalk system.

6.3.3 Experimental results

Based on the feature set extracted from the information states in the data collection and by knowing what dialogue moves the users actually performed in the next turn (according to the system's understanding) we will try to train a classifier that will predict the user's next dialogue move based on our selected features. This we do with a view to being able to load the corresponding DMSLM. It should be noted that the dialogue moves we have obtained from the data collection may be corrupted due to ASR errors and subsequent parsing errors. This means we have some noise in our data. However, in the test set we have tried to correct this. We have carried out several experiments with two different machine learning approaches: memory-based learning and rule-based learning. We will start by presenting the results for the memory-based learner.

6.3.3.1 Results with memory-based learning

We ran several experiments with the memory-based learner TiMBL to explore dialogue move prediction and find out what features in the information state play a role in this task. All results reported have used the default parameter setting with the IB algorithm (IB,GR,Overlap, k=1) as no noticeable improvement was found when running other settings. We are only reporting the most interesting results from a broad collection of experimental settings. For all cases we will report the dialogue move prediction accuracy for the majority baseline (i.e. if we chose to select the most common dialogue move in all cases), for all 15 features, for the single feature LNXTDM (the last move that the system performed before the user's turn) and for the selected features.

For our first classifier we considered the case to predict six distinct dialogue moves (ask, request, answer, yn, quit, greet). The classifier was trained on our 1647 training instances and tested against the test set (275 instances).
In Figure 6.3 we can see that the LNXTDM turns out to be a very important feature giving 54% accuracy by itself. That the previous system move would have such importance was relatively expected based on the results from the previous experiment but also based on results from related work. By using our selected features we get a small gain in accuracy which is not shown to be significant according to the χ^2test. We can also see that some features from the big feature set (all 15) seem to harm the classifier as the performance goes up when using the selected features or only the LNXTDM feature. If we compare to the majority baseline, i.e. always choosing the most common dialogue move (in this case an answer move) our classifier is performing much better by increasing prediction accuracy from 40% to 55.6% (significant on the $p<0.0001$ level). However a prediction accuracy of almost 56% is quite modest. If we look closer at the confusions made we can see that

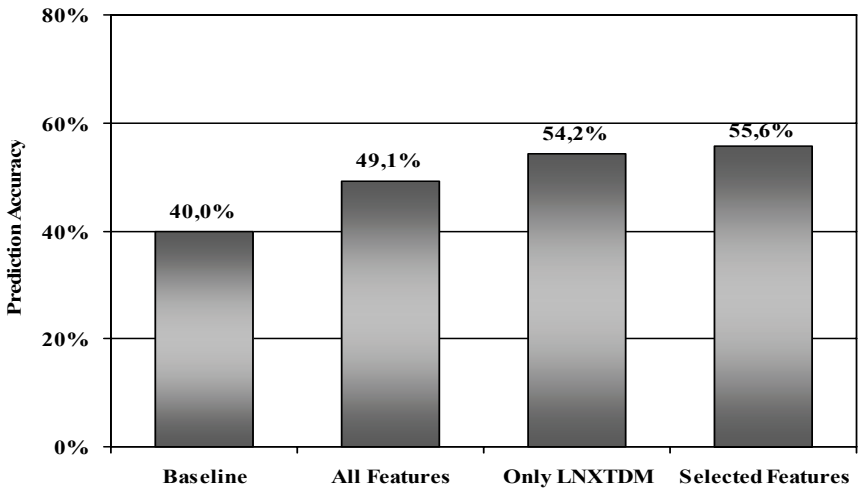

Figure 6.3: Dialogue move prediction accuracy (6 classes)

requests and ask moves are often confused as well as answers and requests. The precision for different dialogue moves change and our classifier is best on classifying yn (65% accuracy) and answer moves (64%) with most problem predicting the quit move (14%). The prediction accuracies of request (51%), ask(45%) and greet (41%) moves fall in between.

In Figure 6.4 we present the results when training our dialogue prediction model without greet and quit moves and only with the four moves: ask, request, answer and yn. In this case the amount of training and test data diminished to 1569 training instances and 266 test instances when removing the instances with greet and quit moves. Although this task is easier the accuracy goes up only minimally and we reach a prediction accuracy of 57%. This is signficantly better than the baseline of 44% (on the $p<0.0001$ level). However, we cannot show any significance of the differences in performance using different feature sets.

To show the impact of the noise in our data we show the results when testing the same model on the original test set before manually correcting the falsely recognized and interpreted user dialogue moves. As seen in 6.5 the accuracy is much lower as the misrecognitions and misinterpretations many times result in dialogue moves that do not seem to fit into the current dialogue state and thereby are not following the usual user patterns that the classifier has learnt. However, as the training data is noisy it is also possible that our classifier has learnt some false patterns that are rather system error patterns than user patterns which may influence on our prediction accuracy in the rest of the experiments.
The problem of recognizing and parsing ask moves seem to have propagated to the prediction accuracy of these moves. As the training data is corrupted and ask moves are not well represented our classifier does not seem to learn to classify this move correctly. Also, questions and requests seem to be highly confusable and occur in the same situations. They are also treated as alternatives in some sense in the system as many alternative questions use questions and requests (internally represented as issues and actions) as menu possibilities:

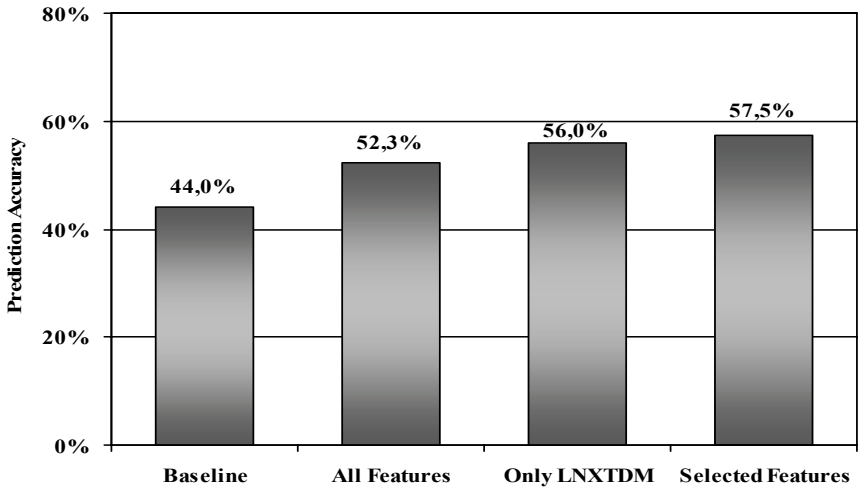

Figure 6.4: Dialogue move prediction accuracy (4 classes)

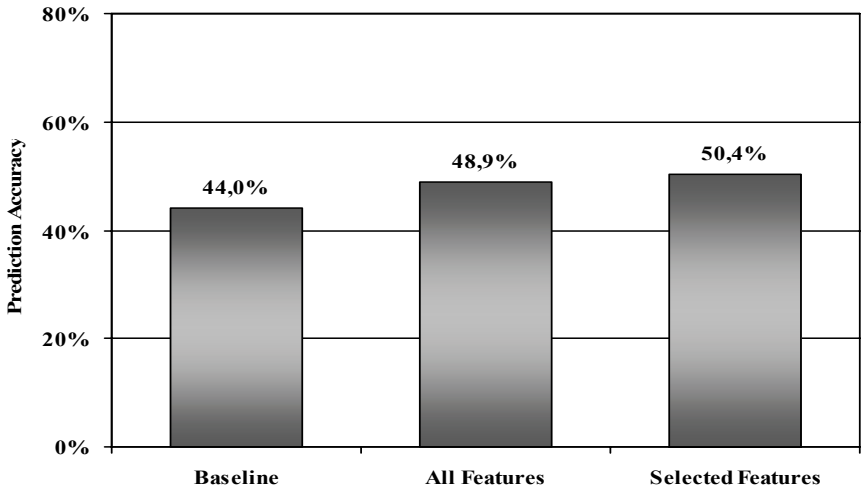

Figure 6.5: Dialogue move prediction accuracy (4 classes) on original test set

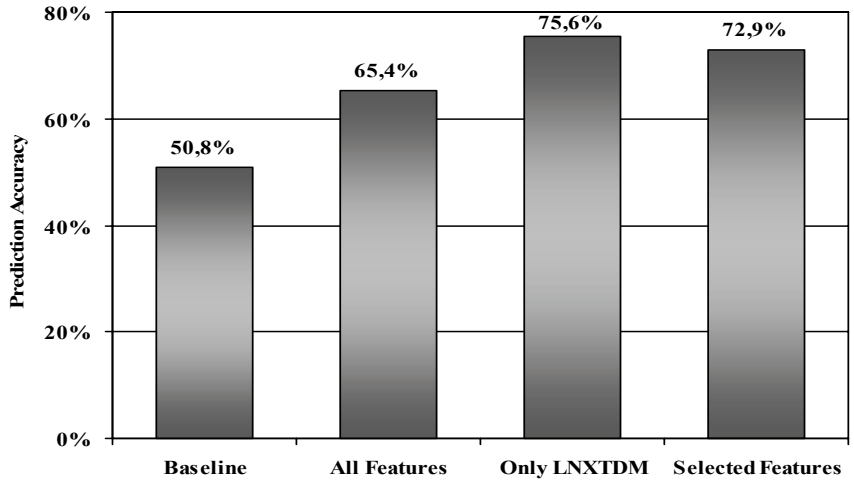

Figure 6.6: Dialogue move prediction accuracy (merging Ask and Request moves)

(31) SYS> Do you want to add a booking or ask about the time of a booking?

Where "add a booking" is seen as a request and "ask about the time" is considered an issue related to a question (ask move). This means it does not seem worthwhile trying to predict them separately. We have therefore carried out a second series of experiments where these two classes are merged to the AskReq class. Naturally, we would have to create a new DMSLM which boosts both request and ask moves.

Figure 6.6 shows the results for one of the experiments with our new three classes: AskReq (including ask and request moves), answer, yn. The classifier has been trained on 1569 instances and tested on the corrected test set.
As seen the prediction accuracy goes up considerably when the task gets easier, i.e. without the confusions between the ask and request moves. However, it is not sure whether a common SLM for these two dialogue moves would give as much ASR gain as the separate models. A prediction accuracy of 73%, although still moderate seems to be useful. However, the best result (76%), although not signicantly better than the selected features, is obtained by just using the feature LNXTDM, i.e. the last system move which shows that the classifier does not seem to learn much more than adjacency pairs. The accuracy for the common AskReq class now reaches 71%, the answer class 72% and the yn class 79%. The majority baseline in this case is the AskReq class which is the most common class in the test set. Our best prediction accuracy is substantitally more accurate than the baseline. When using our selected features we have a relative improvement of 45% relative (significant on the $p<0.0001$ level). In this case the performancy gain when using the selected features or only LNXTDM instead of all features is significant (on the $p<0.01$ level). However, we were not able to prove the importance of the selected features opposed to the LNXTDM.

Although we could take our TiMBL classifier and implement a prediction module of it that could be used in the GODIS system a much more direct way is to see if we can use

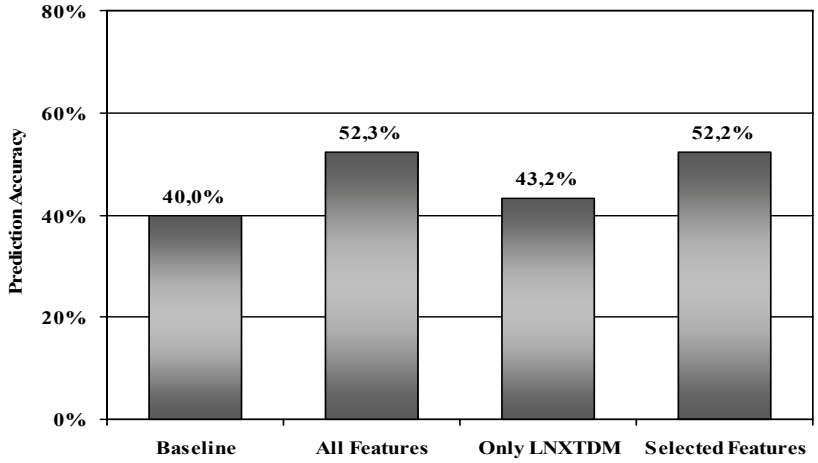

Figure 6.7: Dialogue move prediction accuracy with Ripper (6 classes)

machine learning to learn some rules that we can integrate into GoDiS straightaway.

6.3.3.2 Results with rule-based learning

In order to get a better view of what a machine learner is actually learning from our data we have run some experiments with the rule-based learner Ripper (the Weka version JRip presented in Section 3.6). In this way we will obtain some rules that we can use in the GoDiS **dme** to predict dialogue moves based on the information state. This is a more direct way to get our dialogue move predictor working. It should be noted that in these experiments we did not test against the corrected test set but ran 10-fold cross-validation instead (where the test set used in the previous TiMBL experiment was included as part of the training data). We have used the same training data as earlier converted into a format that Weka can read (AIFF). First, we trained a model to classify six categories. Our best model used only five features (LM, SHACT, QUD, NXTDMs, LNXTDM) to predict our 6 user dialogue moves. Figure 6.7 shows an accuracy of 52% which is worse than the result obtained with the TiMBL classifier. In contrast to the previous experiment the Ripper classifier does not profit as much by using the LNXTDM feature obtaining a significantly lower accuracy when the other information state features are taken away. The difference in performance when using all features or only the selected features does not show any significance.
This classifier learns 10 rules that are used for prediction which are shown in 32 (see Section 3.6 for an explanation of the rule format).

(32)
1 (lm = greet) and (shact = add_event) => dmslm= greet (11.0/4.0)
2 (qud = altq) and (lm = greet) => dmslm= greet (2.0/0.0)
3 (lm = quit) => dmslm= quit (6.0/0.0)

4 (nxtdms = (icm:reraise:top)-ask(action)) and (lm = (icm:loadplan)-report) => dmslm= quit (2.0/0.0)

5 (lnxtdm = ask(action)) and (lm = answer) and (qud = []) => dmslm= ask (201.0/94.0)

6 (lnxtdm = askyn) and (qud = []) => dmslm= yn (130.0/23.0)

7 (lnxtdm = icm:und*int) and (qud = []) => dmslm= yn (151.0/69.0)

8 (lnxtdm = icm:und*pos) and (qud = []) and (nxtdms = (icm:acc*pos)-(icm:und*pos)) and (lm = answer-answer) and (shact = add_event) => dmslm= yn (14.0/4.0)

9 (shact = top) and (qud = []) and (nxtdms = askalt) => dmslm= request (59.0/25.0)

10 => dmslm= answer (1346.0/676.0)

The first four rules predict greet and quit moves. Greet moves are expected when the latest move was a greeting. Rule 4 tells us that a quit move is expected when the latest move was a report from the system that an action had been carried out and the system is about to ask the user for the next action to perform. In this case it seems probable that the user may decide not to carry out any other task as the current task has been solved. Rules 5–9 seem to have captured some patterns that are familiar to us. Rule 5 tells us that if the system's next move is to ask the user what action is to be performed and the latest move was an answer and there is no question under discussion then it should predict that the user will provide information of what issue to consider by performing an ask move, i.e. a question. This is capturing the case when either ellipsis has been performed and the system is not able to accommodate the provided information as specific to any clear task (i.e. a plan) or where the system has misrecognised and has not managed to capture the issue but only a provided slot (i.e. an answer move). In this case the user will try to make clear to the system what the current issue should be. This rule may actually partly be a pattern based on the fact that it was very common that the system misunderstood questions and did not capture the issue but just managed to recognize and understand the informative slots. Here, the system seems to have learnt how to get around its own errors. However, in a new version of the system we hope to be able to capture and predict the ask move earlier to avoid getting to this point of the dialogue. It could also be the case that the user provides the information in this order by ellipsis trusting the system's ability of accommodation. Rules 6–8 capture three cases when it is probable that the user will perform a yn move. Our classifier predicts yes and no answers when the system is just about to perform a yes and no question, an interrogative ICM (such as "meeting, is that correct?") to explicitly confirm a dialogue move or an ICM of positive understanding to implicitly ground a dialogue move (such as "OK. meeting."). However, rule 8 also includes some other conditions such as that the current shared actions should be to add an event and the last moves should be answers. Rule 9 tells us that the classifier predicts request moves when: the shared action is top (i.e. no specific plan has yet been specified), there is no question under discussion and the system move to be performed is an alternative

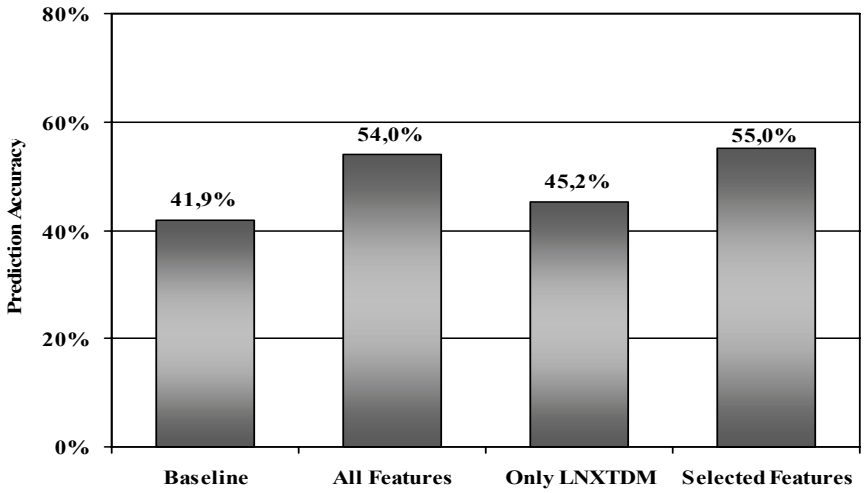

Figure 6.8: Dialogue move prediction accuracy with Ripper (4 classes)

question (probably about what action or issue to perform). Rules 5 and 9 seem correct although in both cases it seems intuitive that the user could perform either an ask move or a request move equally probably. Rule 10 does not tell us much more than that answer moves are the most common and thereby the classifier will predict these in any other case. It would have been interesting to see what kinds of rules could be learnt to predict answer moves.

Predicting our four basic classes gives a slightly better accuracy (55%) as seen in Figure 6.8 and a different set of rules. Our best accuracy was achieved by using the five selected features (LM, SHACT, QUD, NXTDMS and LNXTDM) to predict the user dialogue moves: ask, request, yn and ask. This is still a bit lower than the results from the TiMBL experiment which was expected as we are not testing against a corrected test set. Again, we can see that the Ripper learner indeed profits from using the information state features and performs significantly better when using these than the LNXTDM feature alone.
This classifier learns the following 8 rules where many are very similar to the previous ones.

(33)

1 (lnxtdm = ask(action)) and (lm = answer) => dmslm= ask (199.0/92.0)

2 (lnxtdm = askyn) and (qud = []) => dmslm= yn (130.0/22.0)

3 (lnxtdm = icm:und*int) and (qud = []) => dmslm= yn (149.0/67.0)

4 (lnxtdm = icm:und*pos) and (qud = []) and (nxtdms = (icm:acc*pos)-(icm:und*pos)) and (lm = answer-answer) and (shact = add_event) => dmslm= yn (13.0/3.0)

5 (shact = top) and (nxtdms = (icm:reraise:top)-ask(action)) and (qud = []) => dmslm= request (134.0/66.0)

6 (shact = top) and (lm = request) => dmslm= request (78.0/38.0)

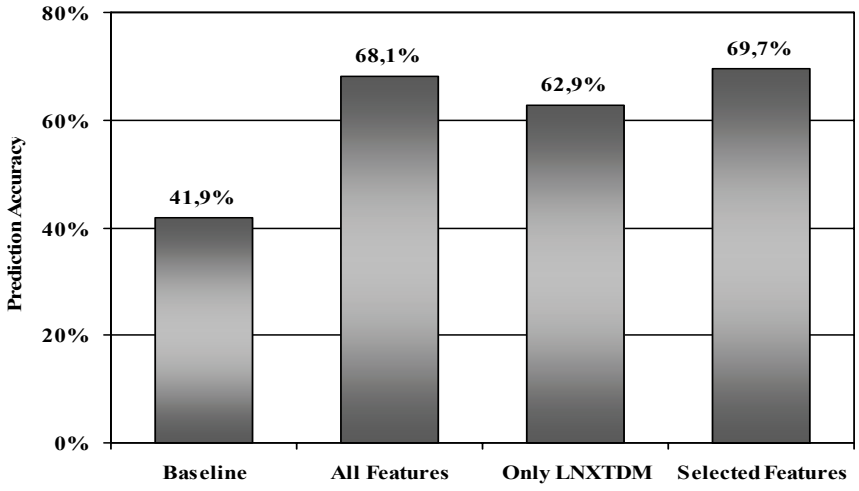

Figure 6.9: Dialogue move prediction accuracy with Ripper (3 classes)

7 (shact = top) and (nxtdms = askalt) and (qud = []) => dmslm= request (56.0/22.0)

8 => dmslm= answer (1076.0/466.0)

Again, some of the rules seem intuitive such as when the system's next last move is a yn question we should predict that the user performs a yn answer (Rule 2). We can see some variations in the way of predicting requests where this move type is expected not only when an alternative question is going to be performed but also when the system is back in the top node (shact = top), the QUD is empty and the system is going to ask the user what he/she wants to do next. Rule 6 is a bit of surprise as it says that if we are in the top node and the latest move was a request then we should predict that the user performs another request. This could be that the users actually repeat what they just said when they don't get an immediate reaction from the system on the request. Another explanation could be that the previous request was a request to restart and go back to the beginning of the dialogue.

The last experiment shows a classifier trained to predict the two dialogue moves yn and answer and the joint dialogue move class AskReq. The prediction accuracy goes up considerably as this is a much easier task. With the six features LM, SHCOM, SHACT, QUD, NXTDMs, LNXTDM we reach 70% prediction accuracy presented in Figure 6.9. This is a considerable improvement (48% relative) in comparison to the majority baseline (i.e. predicting answer moves constantly) of 42%. The prediction accuracy by class is the following: answer moves (69%), AskReq Class (71%) and yn (66%).
In this case the classifier model looks as following with seven rules (see 34).

(34) **1** (lnxtdm = askyn) and (qud = []) => dmslm= yn (130.0/22.0)

2 (lnxtdm = icm:und*int) and (qud = []) => dmslm= yn (149.0/67.0)

3 (lnxtdm = icm:und*pos) and (qud = []) and (abscom = am_or_pm-(start_time_to_store-(date-event_to_store)))=> dmslm= yn (14.0/3.0)
4 (shact = top) and (qud = []) => dmslm= askreq (727.0/208.0)
5 (lnxtdm = askalt) and (qud = []) => dmslm= askreq (42.0/15.0)
6 (qud = []) and (shact = get_info) => dmslm= askreq (22.0/5.0)
7 => dmslm= answer (751.0/226.0)

These rules are very similar to the rules for the other classifiers. We find once again three rules for yn prediction. The AskReq class seems to have inherited the rules from the request class in the previous experiments. All the rules presented in this section give us a hint about what a machine learner seems to learn from our data. We have used these rules to implement a dialogue move prediction model in the GoDiS system in combination with some human intuitions of how such rules should look. The implementation is presented in Chapter 9.

6.3.4 Discussion of the results

Although we said in the beginning of this chapter that conversation is not governed by strict adjacency pairs it seems that what our classifier actually managed to learn were in fact mainly adjacency pairs shown by the importance of the system move feature (LNXTDM) and the rule formulations. This may be due to the small amount of data where other sequences than the adjacency pair patterns are sparse. However, the data and the moderate prediction accuracy shows us that this is not enough even for human-machine conversation as the logged dialogues goes beyond this with other types of sequences such as the one in example (35) found in our logs where the user comes up with a counter-question.

(35) USR> Add a meeting on Friday
SYS> What time is the meeting?
USR> Am I booked at ten?
SYS> No, you have no booking at that time. Do you want to add the meeting on Friday at ten?
USR> Yes

The simplified prediction model that we use can thereby never obtain really high accuracy. More training data and a training set with less noise may lead to better accuracy. However, the ultimate goal is not perfect dialogue move prediction but rather overall improvement of ASR accuracy. We want our system to be able to prepare itself for the most expected user contributions and be able to recognize these with less WER. In the case of less expected contributions we hope to not decrease ASR accuracy heavily but be able to keep an accepted level of ASR accuracy. In the experiments in Chapter 5 it was shown that we would be able to achieve this as the DMSLMs performed well even on dialogue moves not in focus. This means that even if the prediction fails this does not necessarily mean that the ASR will break down. The good thing is that in the cases where the prediction is

correct (i.e. in 70% of the cases according to our last 3-fold model) the ASR WER will fall dramatically as shown in Chapter 5 and we will achieve the ASR enhancement we were looking for. If we take yn dialogue moves as an example we will be able to predict these by 79% with a TiMBL classifier and with 66% using our Ripper Model. If we recall the ASR figures for the yn DMSLM in Chapter 5 the WER fell from 37% to 21% when using the yn DMSLM instead of the General SLM. This corresponded to a reduction in WER of 42% and in DMER of 48%. Although we will not be able to predict all cases of yn answers and thereby not be able to obtain such a substantial ASR accuracy improvement at least the times we manage to predict yn answers correctly our system will be able to capture these moves more accurately. Using our TiMBL classifier we can imagine a possible reduction of 38% in DMER taking into account the prediction accuracy and with our Ripper model a 32% DMER reduction. This is still an important reduction making it worth considering the use of DMSLMs with dialogue move prediction in our system.

In reality, the best way would probably be to predict a set of possible dialogue moves in each state as some are inclusive and others, e.g. a greet move in the middle of the dialogue, are exclusive. This was also shown by the N-Best figures in the first experiment in this chapter. To use such a model we would need to be able to combine and generate SLMs on the fly. Another approach would be to weight rules in an SRG as in Holzapfel and Waibel (2006) or to generate grammar rules with a lexicon as in Young (1989). However the latter approach would go back to grammar-based ASR and its restrictedness which we are trying to avoid.

Something we expected would be even easier to predict were yn answers. Part of the low prediction accuracy for these could be due to the noise in the data but in fact it seems quite common in the data to respond to a yn question with something other than a yes or no answer. If we look at the confusions made there seem to be a lot of confusions between yn and answer moves. It is actually not very strange that a user tries to correct a misunderstanding by giving the correct piece of information instead of responding "no". The following example from the logs shows such a situation where the system implicitly tries to ground the date in question.

(36) SYS> Okej. idag. *Eng. OK. Today*
SYS DM: icm:und*pos:usr*date(today)
USR> på måndag *Eng. on Monday*
USR DM: answer(date(monday))

Rather than using a yn DMSLM in these situations the optimal choice would be a model where also the dialogue move type to ground is boosted, in this case all possible date expressions (corresponding to answer moves with the semantic content **date**). However, this would mean that we would either need to prepare DMSLMs for these different combinations beforehand or be able to generate DMSLMs on the fly combining the predicted dialogue moves. It should be noted that our DMSLMs accept any dialogue move so the ASR would be able to recognize the answer move in the preceding example even if the prediction model had chosen to use the yn DMSLMs for ASR.

6.4 A follow up experiment: Predicting DMSLMs in the MP3 domain

To be able to compare dialogue move prediction between domains we have carried out a third experiment similar to the previous experiment but in this case for the MP3 domain. The first experiment on the MP3 domain predicted 19 different dialogue moves. In practice, 19 different classes would mean preparing beforehand 19 different SLMs and load all these into the ASR to switch between them. With the current setup and recognizer we are not able to generate models dynamically, from e.g. GF grammars, which would be an optimal solution. Loading 19 SLMs may be a bit too much technically for a speech recognition platform at least when each SLM has a considerable size which is the case with DMSLMs as they all have the same vocabulary so as not to restrict the user. Therefore, we would like to consider the less technically heavy approach proposed in Section 6.3 where we only predict the most critical moves, the ones that drive the dialogue forward. Our goal is a domain independent prediction model that could be used in any GoDiS application and therefore we will carry out a comparative experiment using the same dialogue move classes as in the previous Calendar experiment.

6.4.1 The data

For this experiment we made use of logs collected with the DJ-GoDiS application. Some of these were used in the first experiment in Section 6.2) but we also had some additional logs available making up a total of 678 information states. As these logs had been collected with an earlier version of the logging format it was not possible to extract all the features that we had used in the Calendar experiment. But we managed to extract the features that had proven to be of most importance. These were extracted with the same Prolog program that was used in the Calendar experiment. We ended up with the following features:

(37)
LM: Latest moves
FSTCOM: First of shared commitments
SHCOM: Shared commitments
SHACT: Shared actions, i.e. previous actions that were agreed on in the last turn.
SHISS: Shared issues, i.e. previous issues that were agreed on in the last turn.
QUD: Questions under discussion
FSTPLAN: First goal in current plan
NXTDMs: Next system moves to be performed
LNXTDM: The last move of next system moves to be performed

To be able to compare the distribution of dialogue moves in the MP3 domain with the Calendar domain distribution presented in Table 6.1 and Table 6.2 we have summarized the MP3 data dialogue move frequencies in Table 6.10.

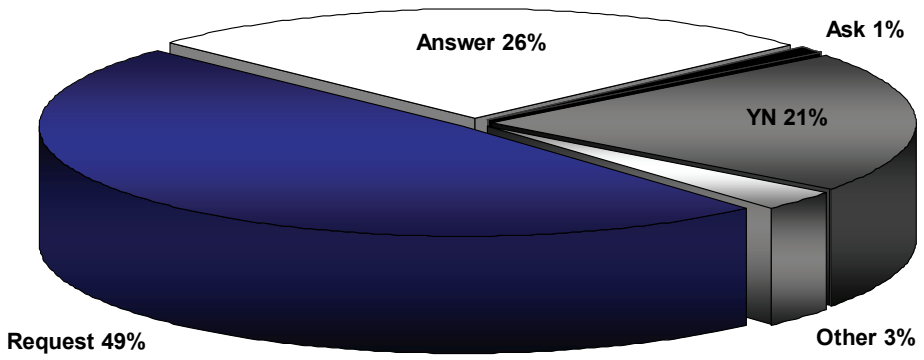

Figure 6.10: Dialogue move frequency (6 classes)

The dialogue move frequencies are in many ways similar to the Calendar domain. First of all we can again see that most of the user dialogue moves (97%) fall into one of the four dialogue move classes: Request, Answer, Ask and YN. The YN frequency are very similar in both domains which is to be expected as the confirmation strategy of GoDiS (which provokes most YN answers) is the same. This also shows the importance of a good recognition of this dialogue move as it is highly critical with the current grounding behaviour in GoDiS. A difference between the domains is the number of ask moves. Although low in both cases questions seem to be minimally used in the DJ-GoDiS application. This is not a surprise as the DJ-GoDiS application is much more action-oriented than AgendaTalk and does not encourage users to ask questions. AgendaTalk has a much wider functionality and handles issues as well as actions allowing users not only to request actions on their schedule but also to ask multiple questions about their schedule. We can therefore also see a difference in the number of requests which is much more common in the DJ-GoDiS application. Answer moves are much more common in the AgendaTalk application where the plans are deeper and thereby require more information from the users. In the DJ-GoDiS application it is much more common to instruct the MP3 player with short commands, e.g. "lower the volume". Such commmands require much less information than typical tasks in the Calendar domain, e.g. scheduling a booking, which often requires several answers to system questions.

It should be noted that as the DJ-GoDiS application is multimodal and the user can perform dialogue moves both by speech and pointing, the user dialogue moves in the training data may have come from either source. However, they are represented in the same way, i.e. as user dialogue moves in the data.

6.4.2 Experimental results

In a similar way to the described experiment in Section 6.3 we trained a 6-way, 4-way and 3-way dialogue move predictor both with memory-based and rule-based machine learning. The memory-based classifier was evaluated with a leave-one-out test whereas we carried out 10-fold evaluation on the Ripper classifier. The results are presented in the following

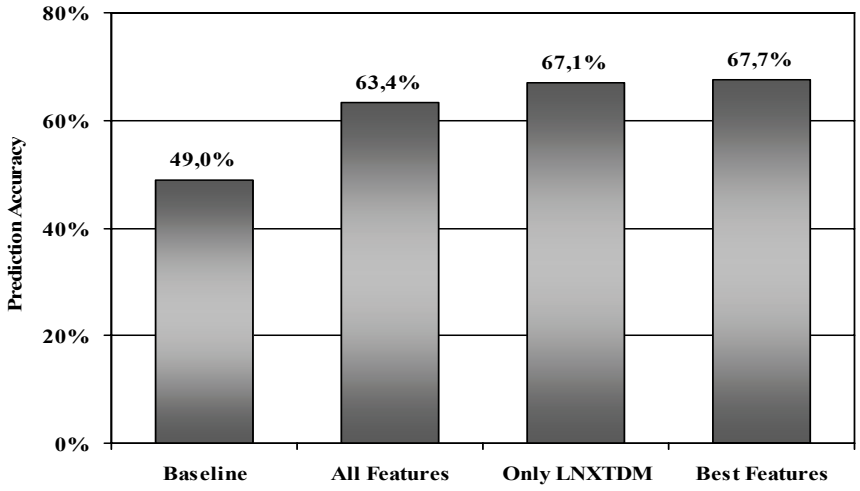

Figure 6.11: MP3 domain: DM prediction accuracy with TiMBL (6 classes)

sections.

6.4.2.1 Memory-based learning results

We trained several dialogue move predictors using the memory-based learner TiMBL for different feature sets and with more or less fine-grained dialogue move classes. First of all we trained a classifier that predicts the six dialogue moves: Request, Answer, Ask, YN, greet and quit. The prediction accuracies are presented in Figure 6.11 where we can see that our predictors outperform the Majority Baseline of 49% (i.e. assigning a request move to every state). Our best predictor was acheived by using combinations of the features: LNXTDM, SHCOM, QUD and FSTPLAN. However there is no significant difference than using only the LNXTDM feature. The accuracy for the dialogue moves greet and quit was minimal.

In our 4-way classifier we have excluded these moves which gives us a Majority Baseline of 50.6%. Using our information state features we outperform this baseline and reach a dialogue move prediction accuracy of 70% as shown in Figure 6.12. Again the most informative features are LNXTDM, SHCOM, QUD and FSTPLAN.

As the number of ask moves was minimal (1%) in the training data it was very hard for the classifier to learn any patterns. Therefore we carried out an experiment with a dialogue move set of only three moves: YN, request and answer. Again we can see an important improvement (27% relative) over the baseline using our best feature set (LNXTDM, SHCOM and QUD). Our best dialogue move predictor for the MP3 domain thereby reaches 71% dialogue move prediction accuracy (see Figure 6.13).

The three tests all show an important improvement over the baselines and we can see that the classifier performs better when using a smaller number of features. Although we get the best results when using our selected features as opposed to only using the LNXTDM feature we have not managed to show the significance of this result.

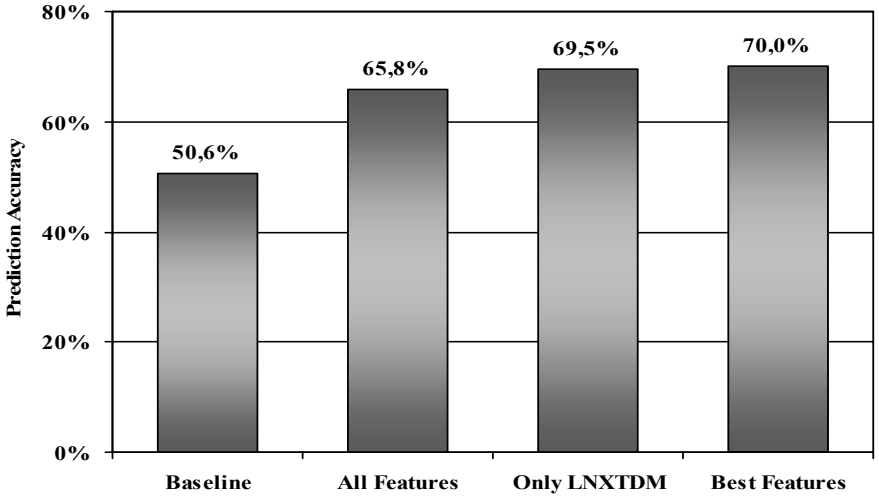

Figure 6.12: MP3 domain: DM prediction accuracy with TiMBL (4 classes)

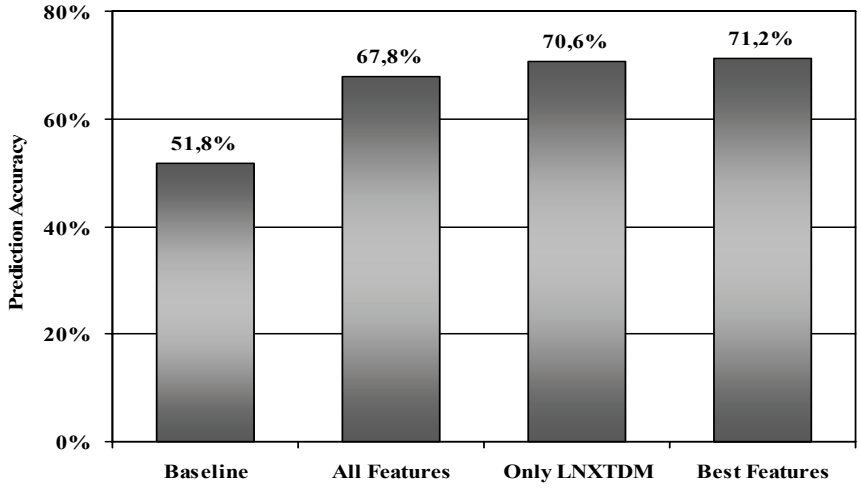

Figure 6.13: MP3 domain: DM prediction accuracy with TiMBL (3 classes)

6.4.2.2 Rule-based learning results

The experimental results for the dialogue move predictor trained with rule-based learning obtained similar prediction accuracy as the memory-based learnt classifier. The 3-way classifier obtained an accuracy of 71% using the following feature set: FSTCOM, QUD, FSTPLAN, NXTDMs and LNXTDM . However what is more interesting are the rules that were learnt.

(38)
1 (lnxtdm = icm:und*int) => dmslm= yn (102.0/9.0)
2 (lnxtdm = icm:und*pos) and (fstplan = findout(group)) and (fstcom = item) => dmslm= yn (6.0/1.0)
3 (lnxtdm = ask) and (fstplan = group) => dmslm= answer (92.0/16.0)
4 (lnxtdm = ask) => dmslm= answer (138.0/60.0)
5 => dmslm= request (673.0/205.0)

The first rule tells us that we should predict YN when the system is about to produce a interrogative ICM such as "Abba, is that correct?". The second rule tells that YN should also be predicted when the next system move is a positive ICM and the first part of the plan is to find out which group the user has in mind and the last shared commitment was an item of the playlist. Both the third and the fourth rule predicts answers after questions. The third rule also includes an additional condition which is that the last shared commitment should be a music group. In any other case the dialogue move predictor goes for the most frequent move, i.e. the request move.

6.4.3 Discussion of results

The results in this last experiment are very similar to the ones achieved for the Calendar domain. We get similar prediction accuracy figures and the same information state features seem to be important. In both domains the most informative feature seem to be the last of the system's next dialogue moves LNXTDM and the feature QUD also seem to provide an important information gain. In neither domains does it seems helpful to keep hold of a longer dialogue history. A difference in this experiment is that the feature FSTPLAN, i.e. the first item of the plan, seems to contribute to the task whereas in the Calendar domain the SHACT feature is more informative.

As our goal is to build a domain-independent dialogue move predictor it is very encouraging to find similarities between the rules learnt in both experiments. We can see that the rules for predicting YN moves are almost identical. The rules we missed in the first experiment, that is, how to predict answer moves, are clearly defined here. Similarily we can find the rules for predicting request moves in the first experiment as these are predicted by default here. It does not seem too far-fetched to build a prediction model based on the rules from both experiments that would work as a generic dialogue move predictor in the GoDiS dialogue system.

In a subsequent suite of experiments we made use of new training data to investigate if additional data could improve our prediction model. In this case we had 50% more

training data available summing up a total of 1028 feature vectors representing information states to which the user had reacted speech-wise. However it was shown neither that the distribution of dialogue moves changed nor that the prediction accuracy figures improved. It seems that the amount of training data used in the previous experiments is enough to get reliable figures. The downside is that it seems to be very hard to improve the accuracy figures even if more data becomes available.

6.5 Summary and conclusions

The ASR figures when using DMSLMs for recognition in Chapter 5 pointed to an opportunity to enhance ASR behaviour considerably in a dialogue system. In this chapter we have shown that predicting dialogue moves to be able to choose an accurate language model is a very complex task. We have tested the prediction of many different sets of dialogue moves from 28 dialogue move classes down to only 3 different dialogue move classes. It has been shown that even predicting three or four distinct dialogue moves is a much harder task than we expected. This may partly be due to the representation of the training data and the noise in the data but also to the fact that whatever classes of dialogue moves we choose it is rare that these will always be mutually exclusive in any dialogue state. This impedes obtaining optimal results as several dialogue moves are possible and plausible in the same dialogue state. If we have a dialogue system that strives after providing the user with flexibility this will imply that we have a user dialogue behaviour with less expectedness. Dialogue move prediction will naturally work better in a more restricted dialogue but our aim is not to restrict the user only to perform better when the user follows the most expected behaviour. Our assumption is that users perceive misrecognition of expected contributions as a worse system behaviour than misrecognition of unexpected contributions. We assume that a system having trouble with recognizing a yes and no answer after asking a yes and no question is more unacceptable than a system misrecognizing a question expression in the same situation. Therefore we think that using dialogue move prediction with DMSLMs may not only enhance the speech recognition but may improve the user experience.

The distribution of dialogue moves in both domains shows that the four classes we have chosen make up most of the user dialogue moves performed in the data collections. However, the first experiment shows that it would be possible to build a more fine-grained classifier. Our dialogue move prediction accuracy in the last two experiments exceeds 70% in both domains which seems to be an upper limit. This is an important improvement in comparison to the baselines and although being a moderate figure we consider it sufficient enough to outperform the current behaviour which has no expectation at all of what dialogue moves are to come. The results are very similar for both machine learners used. The rules obtained with the rule-based learner in the two domains show how the classifiers have learnt some expected adjacency pair patterns such that an answer often follows a question. We can find similarities between the rules learnt for the different domains and we can also see that they compensate each other which will facilitate the development of

a domain-independent prediction model.

One purpose of these experiments was to investigate possible predictors of user dialogue moves by using different contextual features. The results for the two different action-oriented applications DJ-GoDiS and AgendaTalk in the second experiment are consistent with each other. Apparently, the most predictive feature is the system move to be performed before the user turn (LNXTDM). Also, the move before that was shown to be an informative source. However, keeping hold of longer dialogue act sequences does not seem to be helpful. This may be due to the characteristics of the dialogue applications with short tasks or to the sparseness of training data which makes it hard to learn such patterns. It could also depend on the representation chosen. Interestingly, it was shown that information state features such as questions under discussion (QUD) and shared actions (SHACT) make an important contribution to the task of dialogue move prediction. In the first experiment QUD was in fact the most informative feature. This was also apparent in the Ripper experiments. However, although the results were almost constantly better when using some selected features instead of only the LNXTDM as predictor we did not manage to prove the significance of these results for the AgendaTalk domain.

So, one question you could ask when going from predicting 28 different dialogue moves and combinations to only three or four different dialogue moves is whether it is worth putting an effort into distinguishing three or four moves. The fact is that as the reduction in WER and DMER for DMSLMs is so high for the dialogue moves we have considered even a prediction accuracy of only 70% will leave us with a considerable improvement in ASR accuracy. We should bear in mind that the risk if we fail to predict the correct dialogue move did not seem insurmountable as the DMSLMs perform quite well even on other dialogue moves than the ones boosted in the model. This leads us to the conclusion that it is worth an attempt to integrate Dialogue move prediction into our system and use our DMSLMs for recognition. In Chapter 9 we show how this integration has been carried out. However, we leave it open for future work to make a more fine-grained classification and make use of even more specific language models such as predicting not only an answer move but also the type of answer move and loading an appropriate language model where utterances corresponding to this type are boosted.

We can also envision a dialogue move prediction model that would learn from interactions with users using some adaptive learning method. It would update the model with the performed user dialogue moves for each prediction state. In this way we may achieve a model that gets better and better. However, due to ASR and parsing errors it may also learn some error patterns. A way to get around some of these errors would be to only learn from confirmed patterns, i.e. where the user has confirmed that the ASR and interpretation have succeeded. We could for example apply the technique of implicit learning for spoken dialogue systems proposed by Bohus and Rudnicky (2007) by using user confirmations or lack of user corrections as an indicator of correctness.

Eventually, dialogue move prediction is not tied up with context-specific recognition but other tasks could profit from knowing what is expected to be the next user dialogue move. Dialogue move prediction has previously been used and improved the task of dialogue move (act) tagging (Reithinger and Klesen, 1997). Prediction of user dialogue moves could also

help out when re-ranking N-Best-list by giving more weight to hypotheses coinciding with the expected dialogue move or by giving more weight to certain rules in speech recognition or parsing grammars. Also, dialogue move prediction could be taken into account for the estimation of confidence in a dialogue system. To leave these options open we will integrate dialogue move prediction and the switch of DMSLMs as two separate parts of our system. Also, as we strive for a reconfigurable dialogue move prediction model we will as far as possible keep the dialogue move prediction language and domain independent.

Part III

Enhancing a dialogue system's use of ASR output

Chapter 7

Bootstrapping a dialogue move tagger

In Chapter 4 we showed how to bootstrap SLMs by generating training corpora from GF grammars. This resulted in an enhanced speech recognition performance. However, using unconstrained SLMs instead of restricted recognition grammars means that the output from the speech recognizer will also be unconstrained and thus the GF grammar will not always be able to parse the ASR output. Our preliminary strategy for handling this problem in the DJ-GoDiS and AgendaTalk applications was to use a simple rule-based parser written in Prolog that looks for keywords and phrases and maps them to dialogue moves (just as was done before the integration of GF and GoDiS, see Larsson (2002)). However, such a parser is hard to maintain and doubles the grammar work. In this chapter, we will therefore explore the possibility of bootstrapping a *dialogue move tagger* using the same methodology as in Chapter 4.[1]

7.1 Related work

As was discussed in Chapter 6 dialogue act (move) tagging has been of great interest mostly in order to be able to annotate corpora with dialogue acts automatically (Samuel *et al.*, 1998). Dialogue move tagging has been explored with different statistical and machine learning techniques (Samuel *et al.*, 1998; Stolcke *et al.*, 2000; Lendvai *et al.*, 2004). The best dialogue move tagging models have obtained an accuracy of 70% showing the difficulty of the task of classifying utterances to dialogue moves. As different studies use varying numbers and types of dialogue moves it is hard to draw any comparative conclusions from previous work. A difference to previous work on dialogue move tagging is that we do not want to tag only dialogue moves types such as answers or requests but dialogue moves with their propositional content and values, i.e. dialogue moves such as answer(group(abba)) or request(playlist_add) answer(song(dancing,queen)) (see Section 3.3.2 for an introduction to GoDiS dialogue moves). This is a much harder task.

[1]This is a revised version of material published in Weilhammer *et al.* (2006a)

In Section 2.2.1 we gave a very brief introduction to spoken language understanding and discussed the dominance of rule-based techniques for interpretation in dialogue systems. When an SLM is used instead of an SRG for speech recognition, then a lot of unexpected expressions will appear in the speech recognition output. Grammatically unconstrained strings are not well suited to be parsed by a grammar. A better alternative would therefore be to use a more robust parser instead of the restrictive GF parser for semantic decoding. At the current time there are no possibilities for robust parsing within GF which would be needed for an optimal performance when using SLMs. Implementing robust parsing in GF is beyond the scope of this thesis but at least we will propose some possibilities for achieving more robust parsing using GF grammars. Statistical semantic decoders provide a solution to the problem of parsing unconstrained output from SLMs. However, they need training data in the form of semantically annotated corpora. Following the methodology in Chapter 4 we here investigate how to bootstrap semantic decoders from interpretation grammars. We will use data generated from GF grammars which means that we need not put any effort on manual tagging to obtain annotated data. At the same time we can ensure that our data includes all possible dialogue moves that we want to be able to decode in our domain. Unfortunately, in this way we will not be able to use any additional features for the task of dialogue move tagging.

The purpose of the development of a dialogue move tagger in this thesis is mainly to be able to tag the N-Best lists more robustly in Chapter 8. However, in this chapter we will also investigate if this simple approach to developing a machine-learnt dialogue move tagger can give a more robust semantic decoder than the original GF grammar.

7.2 Training and test data

The main difference with our tagging approach in comparison to previous work is that we have trained our taggers on a corpus generated from a GF grammar, written for the domain, where all utterances appear together with the dialogue move(s) they should be interpreted as. In this experiment we have focussed on the DJ-GoDiS application and used the same GF generated corpus as for the experiment in Chapter 5.

The number of user dialogue move types in GoDiS is limited to requests, answers, ask moves, greetings, quit moves and icms as introduced in Section 3.3.2 (see page 70). In DJ-GoDiS we have six propositional contents used with answers (e.g. answer(song(X)) where X can be any of the songs in the MP3player), 3 different types of ask moves and 18 different requests. In GoDiS the pairing of a an utterance and a dialogue move is not necessarily one-to-one but an utterance can be interpreted as several dialogue moves, e.g. a request and an answer. This means that the number of possible dialogue move combinations gets very large. The GF grammar that we have used distinguishes 3873 dialogue move combinations and holds 55702 utterances representing these which means that we have 55702 training instances marked with dialogue moves in our training corpus. In this study we have focussed on the Swedish GF grammar and generated Swedish utterances. However, the English grammar would have given the same dialogue move combinations as the grammars

have a common abstract level. A corpus fragment generated with GF from an early version of the English grammar follows below to show the format of the original training data where all dialogue moves were generated together with the utterances the grammar covered for these moves.

(39) request(playlist_add) answer(item([the,final,countdown])) answer(group([europe]))

i want to add the final countdown with europe please
i would like to add the final countdown with europe please
i want to add the final countdown with europe
i would like to add the final countdown with europe
add the final countdown with europe please
add the final countdown with europe
i want to add europe with the final countdown please
i would like to add europe with the final countdown please
i want to add europe with the final countdown
i would like to add europe with the final countdown
add europe with the final countdown please
add europe with the final countdown

Our taggers have been tested on a test set of 263 transcribed and annotated Swedish user utterances including both unknown words and unknown constructions. These user utterances were collected with the DJ-GoDiS system and thus represent the type of input a semantic decoder for this domain could be exposed to. The utterances vary in length and range from simple one-word utterances (e.g. yes answers) to more complicated twelve word utterances. An excerpt from the test set with dialogue move tags is found below. The test set was tagged manually with dialogue moves by two annotators with an inter-annotator agreement of kappa 0.99 (Carletta, 1996).

(40) Usr> jag vill fråga om vilka låtar han har gjort *Eng. I want to ask about what songs he has done*
Usr DM: ask(X^songs_by_artist(X))

(41) Usr> lägg till sommaren är kort med tomas ledin *Eng. Add "sommaren är kort" with Tomas Ledin*
Usr DM: request(playlist_add) + answer(item([sommaren,är,kort]))
+ answer(group([tomas,ledin]))

(42) Usr> sommaren är kort *Eng. Song title: The summer is short*
Usr DM: answer(item([sommaren,är,kort]

7.3 Dialogue move tagging

We have built two different taggers to simulate a more robust way of parsing. Both were trained on the corpus generated from GF where all utterances appear together with the dialogue move(s) they should be interpreted as. The first tagger is utterance-based and built with the memory-based learner TiMBL. Although, this seemed to work successfully we opted for building a second tagger with the memory-based tagger generator MBT (Daelemans *et al.*, 2003), as it gave us a tagger we could use directly at run-time and that would make it able to give us confidence scores on the dialogue move level as explained below.

7.3.1 Utterance-based dialogue move classifier

The utterance-based tagger or rather dialogue move classifier was trained on 55702 utterances represented as bags of words (BoW) and additionally the length of the utterance which in total gives a vector of 237 features. The BoW is as big as the corpus vocabulary and holds a position for each word which will be marked with a number representing the occurrences of the word in the utterance. A feature vector example representing a request to play a specific song (vara vänner) by a certain artist (jakob hellman) follows below where the first number means that the utterance consists of ten words. These ten words ("jag skulle vilja starta vara vänner med jakob hellman tack" *Eng. I would like to start "vara vänner" by "Jakob Hellman"*) correspond to the positions in the BoW marked with 1 (only one occurrence) and should be interpreted as the dialogue move tags: request(start_specific) answer(item([vara vänner])) answer(group([jakob hellman])). The BoW format does not take into account the order of these ten words.

```
10,[börja,toppnivå,glömma,man,kan,ha,hjälp,få,avbryta,musiken,
stäng,stopp,stoppa,visa,bakåt,igen,spelningen,återuppta,allt,
listan,rensa,höj,höja,viss,speciell,start,början,från,paus,
pausa,ljudet,sänk,volymen,sänka,radiostation,välja,spelaren,
prata,framåt,spola,avsluta,sluta,hejdå,hörde,förlåt,va,sa,
ursäkta,jaha,visst,ok,okej,inte,hallå,tjena,hej,nu,spelas,
heter,japp,jajamen,ja,nepp,nä,nej,ettan,höger,vänster,skifta,
mitten,balansen,ändra,bort,ta,tredje,tionde,sjätte,sjunde,
andra,nionde,fjärde,första,femte,åttonde,föregående,nästa,
den,tre,tio,sex,sju,låt,nio,fyra,fem,nummer,lyssna,höra,1,
spela,radio,rant,stationen,gunfire,digital,lägg,spellistan,1,1,1,
till,lägga,låten,skrivit,de,gjort,han,fråga,någonting,låtar,
vilka,artisten,har,vad,ytan,under,moln,ett,segla,vingar,grader,
hundra,åtta,tro,ska,vindarna,diamanter,vill,göra,får,vet,vem,
här,var,två,tv,på,flickorna,tunga,kärlekens,kråkan,och,flickan,
hörnet,runt,himlen,du,som,precis,om,håll,landskap,öppna,finns,
det,vargar,jagad,hellre,blir,1,mig,ihåg,kom,rummet,i,ängeln,
```

```
sarah,kort,är,sommaren,1,1,hjärta,mitt,av,del,en,solglasögon,
lundell,ulf,leva,di,svenningsson,uno,lemarc,peter,jackson,
michael,nilsson,rickfors,madonna,wiehe,mikael,ryde,annelie,
lakejer,lustans,grön,ebba,1,1,ledin,tomas,tider,gyllene,imperiet,
freda,dahlgren,eva,eldkvarn,orup,kent,irma,död,docent,isaksson,
hhpatrik,ekdahl,lisa,winnerbäck,lars,orkester,kaspers,bo,1,undantag],
[request(start_specific),answer(item([vara,vänner])),
answer(group([jakob,hellman]))].
```

We tested the tagger against our test set of manually tagged user utterances from real DJ-GoDiS interactions. The tagger showed a 79% accuracy on the test set where 156 were exact matches (i.e. existed in the training corpus and likewise in our original GF grammar). These exact matches could be seen as the grammar coverage giving a 59% accuracy which means that we have been able to boost the performance getting a more robust interpreter by using the grammar corpus as training data. This means that we get 34% increase in tagging accuracy by using the bootstrapped tagger (significant at $p < .0001$) instead of the GF grammar. We get a more robust behaviour than with the grammar and are able to interpret unexpected expressions that are similar to the training data. A closer look at the tagging results shows that the tagger even manages to give a partial interpretation to utterances including unknown songs, i.e. an utterance such as "I want to add UNKNOWN" will get the tag request(playlist_add). This means that the dialogue manager will be able to take the dialogue a step forward. This would not be possible with the grammar which would fail in giving any semantic interpretation of the utterance at all.

However, this tagger does not take into account word order which means that "John saw Mary" will be tagged the same way as "Mary saw John". In our domain this order does not really matter for cases like "Abba with Dancing Queen" or "Dancing Queen with Abba" (both interpreted as answer(group(abba)) answer(song(dancing queen))) as long as we do not have artists or songs with the same name. However, in many other domains, of course, we need to be able to make this distinction. Simple cases can be solved by having an additional Bag of Bigrams (BoBi), where the bigram "John saw" would have a position and would be marked in the first case but not in the second case where "Mary saw" would be marked instead. For the moment the utterances in this domain are simple enough to do without this extension but a more advanced technique would be needed if you want to do more advanced parsing.

A TiMBL classifier does not only give a class (in this case a dialogue move or dialogue move combination) as output but can also give a confidence score for its choice. Our classifier could therefore be used to tag utterances together with a confidence score given from TiMBL for the choice of dialogue move tag. In this way we could just reject dialogue moves with a confidence score which is too low and in that way avoid some of the incorrect tags.

Additional training data could be obtained from dialogue system logs by running DJ-GoDiS with the simple Prolog parser. Using this material could improve the accuracy even further. However, in this case we used the existing logs as test data.

7.3.2 Word-based dialogue move tagger

The second tagger was generated with MBT. MBT is normally used to develop POS taggers. We have used it to be able to decide what dialogue move a word in an utterance belongs to. As training data we used the GF corpus converted into a format where each line holds a word and a dialogue move. As an example the utterance "lägg till abba på spellistan" (*Eng. add Abba to the playlist*) is represented as follows:

```
<utt>
lägg request(playlist_add)
till request(playlist_add)
abba answer(group(abba))
på request(playlist_add)
spellistan request(playlist_add)
</utt>
```

We generated a tagger that for known words takes into account two tags before the focus word to be tagged and two words after. For unknown words the tagger looks at the previous tag and at the first four letters of the focus word for clues. This means that the tagger can tag unknown words correctly by identifying a known lemma (e.g. "lägga" (*Eng. add* (infinitive)) if "lägg" (*Eng. add* (imperative)) is known). This contextual feature set was chosen in a development phase. Enlarging the context on either side when tagging known words did not give any improvement but two words back and two words ahead seemed to be optimal. For unknown words we also tested looking at suffixes, but although suffixes normally are useful for the task of POS-tagging it did not seem to be very useful for dialogue move tagging of Swedish words where the lexical meaning of the words is more important. In Swedish the endings of words do not usually bear the lexical meaning. At runtime the tagger can be fed with utterances followed by the sentence delimiter `<utt>`. The output of the Swedish phrase "jag vill lägga till Orup" (*Eng. I want to add Orup*) looks as follows:

```
jag/request(playlist_add)
vill/request(playlist_add)
lägga/request(playlist_add)
till/request(playlist_add)
orup/answer(group([orup]))
```

As seen, each word will get a dialogue move tag. Unknown words will also get a tag but will be indicated with // instead of /. The tagger has been tested on the manually dialogue move tagged test set of transcribed user utterances which included for the GF grammar both unknown words and unknown constructions. The tagger has a 79% tagging accuracy (84% for known words) on this test set of 263 utterances where 156 are exact matches (i.e. existed in the training corpus). These exact matches could again be seen as the grammar coverage which gives us a baseline of 59%. This means that we once again have been able

to boost the performance, getting a more robust interpreter, with a 34% increase in tagging accuracy. Interestingly, this tagger performs similarly to the previous tagger.

The word-based tagger seems to have a problem when common words occurring in song titles appear alone (such as *you*, *a* etc.) tagging them rather as belonging to a song title instead of the overall dialogue move. It seems that it has been over-trained on songs and groups. This could be solved by a post-process checking if the rest of the song title words really appear in the utterance. Another option is to retrain the tagger with songs and groups represented as whole entities (e.g. dancing_queen). A restriction of the word-based tagger is that it cannot identify the occurrence of more than one move of the same kind.

7.4 Dialogue move confidence scores

The word-based tagger also makes it possible to calculate what we call *dialogue move confidence scores* by taking the word confidence scores from the ASR for all words tagged with a specific dialogue move and calculate the mean confidence of these. This means that for the example above we would get two scores: one for the dialogue move request(playlist_add) based on the word confidences of four words and one for the move answer(group(orup)) based on the word confidence of the artist "orup". This is an interesting feature to consider for our experiments in Chapter 8.

With dialogue move confidence scores we would not need to rely on the ASR confidence score for the whole utterance when choosing grounding strategies. In GoDiS each dialogue move is actually grounded separately but the choice of grounding strategy is currently conditioned on the confidence score for the whole utterance (see Section 3.3.5). However, it is often the case that some parts of an utterance have a higher confidence rating than others. A better dialogue behaviour would then be to be able to confirm only the parts rated lower. This is easily done if we can obtain dialogue move confidence scores. Dialogue move confidence scores will be further discussed in Chapter 9.

7.5 Summary and conclusions

In this chapter we have shown that it is possible to bootstrap dialogue move taggers in the same way we bootstrapped SLMs in Chapter 4 by using artificial training corpora generated from GF grammars. Our dialogue move tagger performs better than our interpretation grammar just as the SLMs performed better than the SRGs. We have also pointed out that we would probably get a much better performance if we took into account dialogue context in the semantic decoding process just as was done for the task of predicting the next dialogue move in Chapter 6. This is something that would be easily done in an ISU-based framework by using the information state as an additional knowledge source when parsing the user input.

Investigations were conducted in Swedish in the MP3 domain using pattern matching techniques such as TiMBL and MBT. Although these taggers were not capable of cap-

turing deep semantic relationships they were sufficient for the semantics of the domain. Both methods worked well and yielded 79% accuracy. This means an important boost in performance in comparison to the more restricted parsing behaviour of the GF grammar. Although the semantic decoders obtained have not been integrated into the dialogue system they have been used successfully for other decoding tasks as will be shown in Chapter 8.

Chapter 8

Information state based confidence classification and re-ranking of ASR N-Best hypotheses

In Part II we investigated how we can obtain more accurate initial speech recognition in a dialogue system when little or no training data is available. This was done by generating SLMs from grammars. We have also seen how we can improve performance even further by predicting user dialogue moves and making use of grammar-based SLMs specified for the predicted dialogue move. Although we managed to show that speech recognition accuracy could be improved considerably in this way, our dialogue system would still be liable to a large number of misrecognitions. The N-Best WERs indicate that there is still room for improvement. Furthermore, the speech recognizer's confidence annotation model is still error-prone which means that our dialogue system will have a hard time knowing when the recognizer is doing well and when it is not. This will seriously affect our dialogue system and impair the grounding behaviour. In the current chapter we will therefore investigate how a dialogue system can make better use of the output from the ASR. In a first study, we will investigate if humans can improve the output of ASR by re-ranking N-Best lists. Specifically, we want to reveal the actual benefit they would have from dialogue context and linguistic knowledge. If humans are capable of re-ranking N-Best lists using diverse knowledge sources would it be possible to represent this computationally so that a machine can profit from the same knowledge sources? This will be the focus of the first experiment. In a second experiment we will tackle the problem of inaccurate confidence annotation and explore possibilities for using linguistic knowledge sources to obtain a confidence annotation model much better suited to dialogue systems. The aim is thereby not only to obtain more accurate speech recognition but also to obtain a more reliable model for determining the success of the speech recognition process. We will investigate if this can be achieved by taking into account dialogue context and other linguistic knowledge sources.

8.1 Introduction

That speech recognizers with their lack of linguistic knowledge and no track of the dialogue situation have a hard time getting a correct ranking of the recognition hypotheses is not surprising. Previous studies have shown that humans are able to improve recognition performance by re-ranking N-Best list from speech recognizers (Brill *et al.*, 1998; Chotimongkol and Rudnicky, 2001). As described in Section 2.4.5 human subjects seem to profit from higher level knowledge such as syntactic, semantic or world knowledge to select a more appropriate hypothesis than ASR systems. A more recent study by Skantze and Edlund (2004) on spoken dialogue corpora has reaffirmed that N-Best lists are useful for human subjects in the task of error detection. Both Chotimongkol and Rudnicky (2001) and Skantze and Edlund (2004) have shown that the immediate context (in the form of the previous system utterance) is indeed an important knowledge source in these experiments. Our intuition tells us that more contextual information and knowledge of the dialogue would be helpful to be able to select the hypothesis which is most appropriate in the dialogue situation. However, Skantze and Edlund (2004) were not able to prove such a benefit. One of the purposes of this chapter is to investigate this further by carrying out experiments with human subjects in an attempt to prove our intuition that dialogue context matters. We want to explore to what extent they can take advantage of dialogue context when charged with the task of re-ranking N-Best lists.

The survey in Section 2.4 describes attempts to make use of additional knowledge sources in order to improve recognition performance. Still it has been very hard to show important performance improvement when integrating such information sources into the actual speech recognizer as discussed in Section 1.2. A more straightforward application is therefore to use additional knowledge sources and more sophisticated methods in a post-process step. In this way there is no need to interfere with the main recognition process and the methods applied are not dependent on a specific ASR system. However, the ultimate goal would of course be to integrate successful methods into the statistical HMM framework to not only readjust the recognition afterwards but contribute to better recognition from the start.

This chapter will focus on the use of higher level knowledge in recognition by integrating such knowledge into rescoring and re-ranking of N-Best hypotheses. In related work more or less advanced automatic post-process methods have been used to analyse and decide on the best choice from an N-Best list using additional knowledge sources. An overview of previous research was given in Section 2.4.5. In this overview we could see that although many different types of knowledge had been used (syntactic, semantic and pragmatic) few studies had really exploited all the information available in a spoken dialogue system. In particular the use of dialogue context was little represented. More recent research where there is more integration with the dialogue system seems appealing. Gabsdil and Lemon (2004) used machine learning to predict the quality of N-Best hypotheses in an information state based dialogue system. Their machine-learnt classifier used a combination of acoustic, semantic and pragmatic features to decide whether to ACCEPT, REJECT or IGNORE a speech recognition hypothesis. Based on this classification a simple re-ranking

algorithm was used to select the best hypothesis from 10-best lists. The classifier made use of 20 features extracted from the ASR system, from the waveforms and from the dialogue manager. Among these features we want to highlight the use of both confidence and word confidence scores, prosodic features, semantic knowledge (in the form of the dialogue move associated with the hypothesis), pragmatic knowledge (in the form of coherence between the hypothesis dialogue move and the previous system dialogue move and the number of unresolved pronouns and NPs) as well as task knowledge by estimating the conflicts on task level for each hypothesis. The results were encouraging showing that such a classifier would improve dialogue system performance overall and that the dialogue features that were used made an important contribution to the overall performance. The benefit of the prosodic and task level features was less significant. The study presented in this chapter is inspired by the work of Gabsdil and Lemon (2004) and aims to explore further to what extent dialogue context, represented as an information state, can contribute to the task of ASR hypotheses selection and what role information state features play in contrast to other knowledge sources such as grammaticality and ASR information.

Eventually, the problem is not only choosing a hypothesis but also deciding how reliable a hypothesis is. We want to accept correctly recognized utterances, reject incorrect ones and clarify or confirm dubious hypotheses. Many current dialogue systems use different verification or grounding strategies basing their choice of system response on ASR confidence scores. An overview of confidence score estimation and grounding was given in Section 2.4.6 and Section 3.3.5. Confidence scores are known to be unreliable and a number of attempts have been made to improve them (Litman *et al.*, 1999; Hazen *et al.*, 2002; Skantze and Edlund, 2004; Gabsdil and Lemon, 2004). More reliable ASR hypothesis scoring would lead to better choice of grounding strategies and consequently better dialogue flow without unnecessary confirmations. It would also improve the system's ability to detect misrecognitions.

In this study, we will use machine learning to train several confidence classifiers based on different linguistic knowledge sources and based on different amounts of dialogue context. These will classify ASR hypotheses into confidence classes in accordance with the GoDiS system's grounding behaviour (see Section 3.3.5). We will thereafter investigate if this classification can be used to re-rank N-Best lists and by that achieve also a better recognition performance. The first experiment in Section 8.3 will explore the possible benefit of dialogue context in a confidence annotation and re-ranking task. For comparison, we will evaluate the re-ranking performance of human subjects to that of automatic re-ranking methods when exposed to dialogue context information. This first experiment is dedicated to the DJ-GoDiS domain. The second experiment presented in Section 8.4 will make use of data from the AgendaTalk domain to investigate the impact of different linguistic knowledge sources for the task of confidence annotation and re-ranking.

8.2 Confidence classes

The GoDiS system uses a more fine-grained scale of grounding levels than many other dialogue systems. This is described in Section 3.3.5. The grounding behaviour in GoDiS is not limited to the perception level but also chooses different strategies depending on semantic and pragmatic understanding of the user input. However, in this study we focus only on the perception level, i.e. how confident we are that we have perceived the user's message correctly. We will classify recognition hypotheses with a confidence label in accordance with GoDiS's grounding behaviour using the following five classes: Optimistic (OPT) (certainly correctly recognized), Positive (POS) (probably correctly recognized), Pessimistic (PESS) (possibly correctly recognized), Negative (NEG) (probably a misrecognition) and Ignore (IGN) (certainly a misrecognition).

These classes could, for example, be used instead of ASR confidence scores to determine system responses to a situation in the DJ-GoDiS application where the output of ASR was "Abba". For the 5 confidence levels possible system responses could be as follows:

(43) OPT: OK. What song do you want to play?
POS: OK. What song do you want to play with Abba?
PESS: Abba, is that correct?
NEG: I heard you say Abba. What group do you mean?
IGN: I did not catch that. What group do you mean?

In GoDiS, like in many other dialogue systems, the choice of grounding strategy is currently conditioned on the confidence scores from the recognizer as described in Section 3.3.5. In Section 8.4.3 we will show how the settings of confidence thresholds for such an approach can be optimized. However, the principal aim of the following two experiments is to explore the possibility of relying on more information sources than just the information from the ASR for the estimation of confidence used to choose grounding strategy. It is not a proposal about grounding and the grounding strategies to be used based on such a classification are therefore left open. What we want to explore is if it is possible to achieve a reliable automatic classification of ASR hypotheses into these fine-grained confidence classes.

8.3 First experiment: Dialogue context-based confidence classification and re-ranking in the MP3 domain

The aim of this experiment is to investigate how the use of dialogue context can contribute to the task of confidence classification and N-Best re-ranking[1]. The first part consists of a study with human subjects where we investigate their ability to rank and confidence

[1]This section contains material that has been previously published in Jonson (2006a)

classify ASR hypotheses using dialogue context. Based on the results of these experiments we have explored whether an automatic machine-learnt confidence classifier and ranker can profit from using dialogue context features. The data used for this experiment comes from the DJ-GoDiS domain.

8.3.1 The data

The experimental data we have used is limited and comes from logs generated automatically by DJ-GoDiS when interacting with users and holds 486 user utterances (i.e. 486 N-Best lists). The logs consist of the utterances made by the system and the users, the information that the ASR and TTS system send and additionally the system's information state during the dialogue (see Chapter 3 regarding the format of the logs and the GoDiS information state). The user utterances were transcribed manually. We took 40 N-Best lists from this data to use for testing. This test set was not chosen randomly but a high number of N-Best lists with possibilities for re-ranking were chosen. The same test set was used in the following human experiments as well as an evaluation test set for the machine learning experiment in Section 8.3.4.3. It has the following characteristics (see Section 2.3 for an introduction to the metrics):

(44) SER of 90% (WER of 34%)
N-Best SER of 22% (N-Best WER of 15%)
DMSeqER of 70% (kappa 0.85)
Oracle DMSeqER 0% (kappa 0.85)

The SER tells us that the proportion of correctly ranked N-Best lists was quite low which was desirable from the point of view of the experiment. The N-Best SER shows that only 31 of the 40 lists include the correct hypothesis (identical to the transcription) somewhere in the list. Consequently, although we have a lot of room for improvement the Oracle SA is 78%. However, a dialogue system does not always need to get an exact transcription of what was said. For that reason, we have also looked at *Dialogue Move Seqence Error Rate* (DMSeqER) as defined in Section 2.3.4. DMSeqER for the topmost ranked hypotheses of the 40 test lists is 70% (this was manually annotated by two annotators with an inter-annotator agreement of kappa 0.85 calculated according to Carletta (1996)). As shown by the Oracle DMSeqER of 0% above it is always the case that a hypothesis corresponding to the transcription's semantic interpretation into dialogue moves occurs in the N-Best test lists.

8.3.2 Human N-Best re-ranking using dialogue context

We have carried out two different experiments on human subjects to investigate on the one hand if dialogue context helps for the task of re-ranking and on the other hand if people are able to choose a hypothesis from an N-Best list and classify it into one of the 5 confidence classes we are proposing. We have used different groups of subjects for the two

Table 8.1: Experimental set-up

	Group 1	Group 2	Group 3	Group 4
List Set 1	Task 1	Task 4	Task 2	Task 3
List Set 2	Task 2	Task 3	Task 1	Task 4
List Set 3	Task 3	Task 2	Task 4	Task 1
List Set 4	Task 4	Task 1	Task 3	Task 2

tests all naive annotators but with a limited knowledge about the dialogue system, speech recognition and N-Best Lists. The N-Best lists vary in length from 6 to 10 hypotheses.

8.3.2.1 Ranking N-Best hypotheses with more or less context

This experiment explores to what extent humans take advantage of dialogue context when charged with the task of choosing the most plausible hypothesis in an N-Best list. This has been done by dividing 16 subjects into four groups where each group had to complete four different tasks. In each task they were given a set of 10 N-Best lists to re-rank with more or less context. Each group and subject re-ranked 40 lists altogether during the course of the four tasks. No group repeated lists through tasks and every group had different lists given for the same task. In this way we could compare how the same lists were re-ranked when given more or less context. The experimental set up is illustrated in Table 8.1.

The four tasks are described below:

1. **No context:** Choose the hypothesis that seems most plausible that a user may have said given only a ranked N-Best list with confidence scores. In this case there is no context available and the subjects can only draw their conclusion based on the information in the list itself.

2. **Immediate context:** Subjects were given a bit of dialogue context in the form of the previous system utterance in addition to the N-Best list. Otherwise the task was exactly the same as task 1.

3. **Close context:** Subjects were given two turns before the recognition output. This may include previous system utterances, user utterances (as recognized by the recognizer) or user clicks in the GUI.

4. **Dialogue context:** Larger portions of dialogue were given (at least five turns back) sufficient to place the N-Best list in its dialogue history. The dialogue history may consist of not only system and user utterances but also user clicks in the GUI.

8.3.2.2 Experimental results for human subjects

The results of the experiments (see Figure 8.1) show, like similar experiments in the literature (Brill *et al.*, 1998; Chotimongkol and Rudnicky, 2001), that humans are quite good

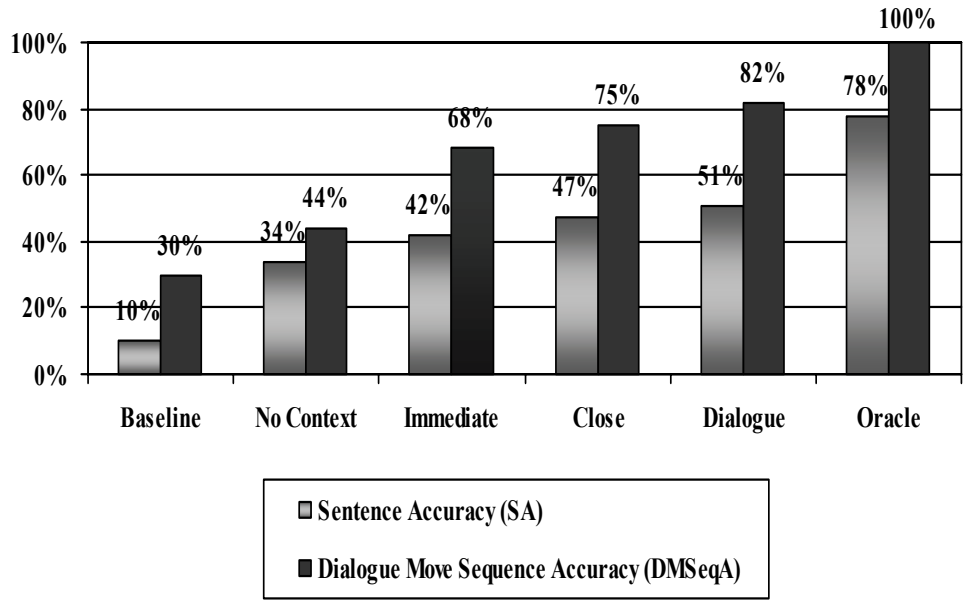

Figure 8.1: Human re-ranking results: Adding context

at the task of re-ranking N-Best lists and find better choices than the topmost one when they are available. In comparison with the baseline, the Sentence Accuracy (SA) increases 41 percentage points (significant at $p < .0001$) and the Dialogue Move Sequence Accuracy (DMSeqA) 52 percentage points already for task 1 (No context). The experiments also show that humans perform even better when the lists are presented in a dialogue context and that the performance improves as more context is made available. The subjects gain 86% in DMSeqA in task 4 in comparison to task 1 (significant at $p < .001$). It is clear that it is not only the immediate context that matters but the results from task 4 show that making the previous dialogue flow available seem to help humans to get a better idea of the dialogue situation and the plausibility of different user utterances in that context. The Oracle results show that we actually have a bit left to the upper limit which is understandable considering that the subjects do not have access to any acoustic information.

8.3.2.3 Classifying N-Best hypotheses into confidence classes

In the second experiment a new group of subjects (10 people) got the same 40 N-Best lists as in the previous experiment presented as in task 4, i.e. with a larger amount of dialogue context. These subjects were given the more complicated task of not only choosing the hypothesis from the list they thought was most plausible but they also needed to classify their choice with our confidence classes. To make the task a bit easier we gave them a 1-5 scale where 1 corresponds to the IGN category and 5 to optimistic grounding (OPT). The scale was presented for the subjects as in Section 8.2 and it was explained that the

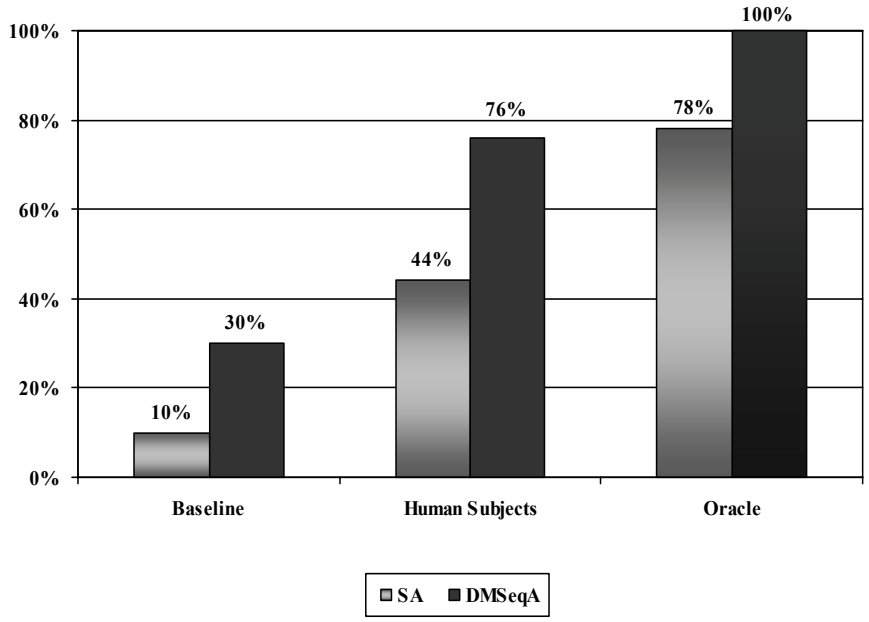

Figure 8.2: Human re-ranking results: Experiment 2 with dialogue context

classes should correspond to their confidence of their ranking choice. That means that if their choice probably was a misrecognition despite being the most plausible in the list they should mark this as 1 and if their choice seemed extremely plausible in the context they should mark it as 5.

8.3.2.4 Experimental results for human subjects

If we look at the task of re-ranking in experiment 2 alone (see Figure 8.2) we can see a slight degradation of performance in comparison to task 4 in experiment 1 which may be due to the fact that the subjects are actually performing two different tasks: re-ranking and classification. However they are still performing very well and we can also see a large variation between individual subjects with a SA varying from 35% to 55% and a DMSeqA varying from 65% to 90%. This shows that some individuals perform much better than others. The agreement on the ranking task is kappa 0.51 if we compare all pairs of rankers (calculated according to Carletta (1996)). This shows that the subjects are not ranking randomly and seem to agree quite often. A certain disagreement is expected for this task as there are several plausible and conceptually correct hypotheses in the lists to choose from.

When asked, all the subjects agreed that it was a difficult task but that they expected their ranking choices to outperform the recognizer's performance, which they apparently did. For the grounding task it was hard to find any inter-rater agreement at all with a kappa of only 0.15 which shows that the choice of grounding level is very subjective. The

classes we have, actually form a scale and they are not independent classes (e.g. NEG and IGN are much more alike than IGN and OPT). Inter-rater agreement is hard to get with such a scalar task. We also have dependency with the ranking task and the subjects are actually ranking their hypothesis choice which is not the same for all subjects. This shows that we probably need a different experimental setup to explore grounding classification. One possibility is to carry out the grounding task alone on already ranked lists. Another more natural setting could be as in Skantze (2005a) where the subjects used grounding interactively. Although the subjects did not agree on exactly what level to choose at least they seemed to agree on what part on the scale a hypothesis should go, i.e. from pessimistic to optimistic or from pessimistic to ignoring. To be able to calculate inter-rater agreement as if the task had been in terms of three classes we merged the NEG and IGN class into one and the positive and optimistic classes into another. This resulted in 3 levels: NEG, PESS and POS. This gave us an inter-rater agreement of kappa 0.44 which is still very low but shows that there exists an agreement over chance. The choices are very subjective although we cannot find any bias for any rater, i.e. someone using much lower or higher levels than the others. The low agreement may indicate that either the test setting, i.e. combining two tasks, or the instructions were deficient or that it is just an unnatural way to think of grounding for the subjects. Unfortunately, we will not be able to use these results as a gold standard as we had expected.

If we see the grounding levels as a scale we can calculate a mean score for all the subjects to see how cautious or optimistic they are. We get a confidence of 3.29% which shows that they would accept more hypotheses than they would reject. The mean grounding confidence of the test set for the GoDiS system is 3.25% with the thresholds we are considering. However it should be clear that the GoDiS system has a DMSeqA of only 30% compared to the human subjects with 76% which shows that either the system is too confident or the human subjects too doubtful about their choices. An interesting thing to see is that the NEG and IGN levels (i.e. the doubtful levels) are used 21% of the time by the raters which seems to match quite well with the concept accuracy figures so it seems that the human subjects are aware of their possibility of correctness.

8.3.3 Automatic N-Best re-ranking using dialogue context features

The outcome of the experiments with humans shows us that context seems to contribute to an improvement in re-ranking performance. However we can also see that the human subjects did not agree on the task of confidence classification although they seemed to agree on what side of the scale a chosen hypothesis would match to. To see if it would be possible to automate this task for application in dialogue systems we will apply the memory-based learner TiMBL (see Section 3.6) to confidence classify and thereafter re-rank ASR hypotheses. We have experimented with different features divided into groups representing the different tasks in the human experiments in an attempt to approximate the possible features the human subjects were using in the earlier experiments in each task.

We converted the hypotheses from the N-Best lists with adjacent dialogue logs into feature vectors with 21 dimensions (the features are presented in Section 8.3.3.2) which resulted in 2645 training hypotheses. Although the data is domain-dependent the features we use are domain-independent and available in all GoDiS applications so it would be easy to train a new classifier for a new domain if dialogue logs were available.

8.3.3.1 Tagging N-Best lists with dialogue moves

What we needed for the experiment, in order to prepare the training data, was a more robust way of parsing the N-Best hypotheses and the manual transcriptions of the user utterances. We used the GF grammar to get a grammaticality score but opted for the word-based tagger, presented in Chapter 7, for the tagging task. The test set used in the experiment in Chapter 7 was more extreme than the N-Best lists used here. In this task the words in the N-Best lists are all known to the tagger as the vocabulary for the SLM is the same as for the original GF grammar. We used the word-based tagger to tag the 2654 ASR hypotheses in our training data and the 391 transcriptions with dialogue move tags for each word. We then took the word dialogue move tags and eliminated all duplicates to get a dialogue move tag (or tags) for the whole utterance. This was used as one of the features for our machine classifier. Another feature that we were able to obtain was a list of dialogue move scores calculated from the dialogue move word tags and the word confidence scores as explained in Section 7.4. From the resulting dialogue moves in an N-Best list we extracted the most frequent dialogue move of the list as an additional feature.

In addition, the dialogue move tags were used to be able to compare each hypothesis with the transcription on the concept level and by that automatically label all hypotheses as being conceptually similar or not to the transcription.

8.3.3.2 Feature groups

The features considered for the machine learning experiment were extracted from the information available in the dialogue logs. We used five feature groups. The first four correspond to the four tasks given to the human subjects: **No context**, **Immediate context**, **Close context** and **Dialogue context**. The fifth group (**List**) is a subgroup of the first as it does not take into account any dialogue context. This group holds information about the list a hypothesis belongs to which the human subjects had indicated as useful information. As this group seems to contain interesting features that do not seem to have been explored in depth before we have chosen to treat it as a separate group. The hypotheses are represented as vectors with features that we have chosen in an attempt to approximate to the possible features the human subjects were using in the earlier experiments in each task. The five groups include the following features (where the number is the number of features per group):

1. **Utterance features (8):** This group consists of features available without context extracted from the recognizer and from the parser.

- Recognition: HYPConf, WordConf, HYPRank, HYPStdDev, HYPLength
- Parsing: Gram, HYPDM, DMScore

The recognition information extracted from the logs is the confidence score (HYPConf), the word confidence scores (WordConf) and the ranking of the hypothesis (HYPRank). Apart from this we calculate the standard deviation of the list of word confidence scores (HYPStdDev) and the word length of the hypothesis (HYPLength). The parsing information that we have available is a grammaticality score (Gram), i.e. whether the hypothesis is grammatical or not (when parsing it with the GF grammar), a dialogue move tag of the hypothesis (HYPDM) and a list of dialogue move scores (DMScore).

2. **Immediate context features (2):** This group represents the immediate context with the previous system move and the relevance of the hypothesis to that dialogue move.

   - SysDM: The system move performed immediately before the user turn.
   - QAMatch: Represented as *qamatch* if the hypothesis dialogue move is relevant to the system move, e.g. if the hypothesis move is a relevant answer to the system question. The relevance feature is already in use in GoDiS and we have by those means been able to use available code and the domain knowledge to extract this feature.

3. **Close context features (4):** Close context features are features extracted from the information state and include:

   - PrevDM: Dialogue move (or click in the GUI) performed before the last system move
   - QUD: Questions under discussion
   - SHCOM: Shared commitments
   - SHACT: Shared actions, i.e. previous actions that were agreed on in the last turn.

4. **Dialogue context features (3):** To this group belongs features extracted based on the entire dialogue.

   - DiaHist: Dialogue history as a sequence of all previous dialogue moves.
   - OnTrack: Number of recognizer rejections in the dialogue so far.
   - Action: Action performed in earlier turn. This will hold GUI and device information such as if the music has been turned on or off or if the playlist has been altered.

5. **List features (4):** This group considers features that take into consideration the whole N-Best list that a hypothesis belongs to, e.g. the standard deviation of the confidence scores in the list or whether the dialogue move of the hypothesis matches with the most frequent dialogue move in the list. In other words it represents the list context of a hypothesis.

    - NrofHyp: Number of hypotheses in the list.
    - ConfDev: The standard deviation of the confidence scores in the list.
    - Mean: The mean confidence score of the list.
    - FreqMatch: Represented as freq/infreq and dependent on if the HYPDM matches with the most frequent dialogue move in the list.

These groups give us a 21-dimensional feature vector for each hypothesis. An example instance of a hypothesis labeled with optimistic grounding is shown below:

```
5,request(start_specific) answer(item(flickorna på tv två)),5,42,
[38 50 48 34 80],[38.0 53.0],16,10,4,44,freq,ask(x^action(x)),
qamatch,0,[start,greet,ask(x^item(x)),icm:sem*neg,ask(x^item(x)),
confirm(playlist_add),icm:reraise:top,ask(x^action(x))],playlist_add,
icm:acc*pos,=,top,x^action(x),gram,opt.
```

We had even more features in mind from the beginning but these were the features we settled on as they were easily extracted directly from the logs or possible to infer from the information in the logs. It should be noted that all dialogue moves used are specific, i.e. including slots and values, and that we could have profited from considering only *dialogue move types* (as specified in Section 3.3.2) as it would have been easier for the classifier to generalize over dialogue move types. This is something we will consider in following experiments.

8.3.3.3 Hypothesis labelling

The training instances extracted from the logs were labelled with the confidence classes described earlier (see Section 8.2): optimistic (OPT), positive (POS), pessimistic (PESS), negative (NEG) and ignore (IGN). The labelling was carried out automatically while extracting the feature vectors from the logs by comparing each hypothesis with the manual transcription and its dialogue move interpretation following these criteria:

(45) OPT: Hypothesis identical to transcription (perfect recognition)

POS: Hypothesis grammatical and interpreted as same dialogue move as transcription (minor misrecognition)

PESS: Hypothesis and transcription interpreted as same dialogue move (misrecognition but same semantic content)

NEG: Dialogue move type agreement (e.g. the user turn was an answer of a group but not the group the user said, i.e. slot disagreement) or partial dialogue move agreement

IGN: None of the above cases apply (total misrecognition and no dialogue move agreement)

In this way we are always labelling hypotheses conceptually equivalent to a transcription with high confidence (worth grounding) and total misrecognitions as rejections. The labelling is quite strict in the sense that we prioritize perceptually equivalent hypotheses over semantically identical ones.

8.3.4 Experimental results

The classifier was trained on the 2645 21-dimensional feature vectors and tested both in a *leave-one-out* setup and on the test set representing the 40 N-Best lists used for the human experiments. We first evaluated the classifier's ability to choose confidence levels with the automatic labelling as a gold standard. We altered the number of features used by the classifier by adding the feature groups stepwise starting with group 1, which does not include any context features, to see how the classifier exploited context features. In addition, we evaluated the classifier's ranking ability by looking at the classifications it gave the test set's hypotheses to see if the highest confidence level in each list was given to the correct hypotheses. To perform ranking we used the same simple selection procedure used in Gabsdil and Lemon (2004) by choosing the hypothesis in the list that had the most confident classification with the preference ordering OPT>POS>PESS>NEG>IGN. In the cases where there were various equally confident hypotheses we chose the highest ranked one as ranked by the speech recognizer. The classifier was evaluated for SA and DMSeqA by comparing it both with the GoDiS baseline, which always chooses the top-ranked hypothesis, and with the human subjects' performance.

We ran several tests with TiMBL varying the number of features groups used and optimizing the settings. Due to the low inter-rater agreement on the 5-class task in the human experiments we chose to run the experiments also as a 3-way classification by merging the rejecting classes NEG and IGN into one class and the accepting classes POS and OPT into another withholding the intermediate PESS class as it was. We also carried out a small experiment with Weka's rule-based learner JRip for comparison (see Section 3.6 for an introduction to JRip).

8.3.4.1 Classification results

We ran TiMBL with default settings and chose the baseline to be the case if the most frequent confidence label, IGN, would be chosen in every case for the five-way classification which gave a majority class baseline of 44%. In the three-way classification task this corresponds to the case of choosing the combined NEG and IGN class which constitutes 67% of the cases.

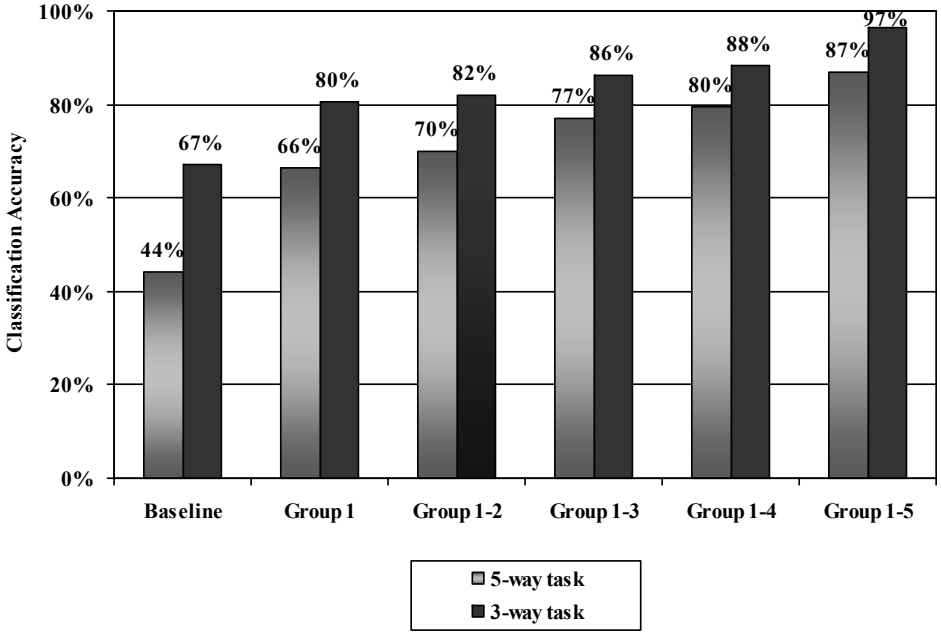

Figure 8.3: Results for 3-way and 5-way confidence classification: Adding context

Figure 8.3 shows the results for the classification of ASR hypotheses into confidence levels for the different feature groups. Just as for the human experiments we can see that the automatic classifier profits from the use of more and more context features and that the performance goes up for every group added (all improvements are significant at $p < 0.025$). We can also see that group 5 (considering the whole list as an information source) makes an important contribution to the overall performance (8% increase). TiMBLs Paramsearch tool was used to find an optimal parameter setting which gave the optimized result of 89% classification accuracy for the five-way task using all feature groups, i.e. a 34% increase in comparison to only using the non-context features in group 1. We also looked at the impact of different features on the result but could not find any improvement by excluding any of the features in any group. However, it was obvious that some features seemed to be more important than others. The FreqMatch from group 5 and the grammaticality feature from group 1 were the most prominent ones. Apart from these two, the dialogue move of the hypothesis HYPDM, the word confidence scores WordConf, the dialogue history DiaHist and SHCOM seemed to contribute an important information gain. However, the confidence score itself, which is the basis for choosing grounding levels in many dialogue systems and in GoDiS, did not seem to contribute significantly to the overall result. The confusions made by our classifier (see confusion matrix 8.2) do not seem to endanger the dialogue system behaviour as it is rarely the case that an incorrect concept is accepted and most confusion seems to be between the ignore and negative categories.
As we can see, the ignore and negative categories are prominent which is an expected distribution as there are a lot of misrecognized hypotheses in each list and much fewer

Table 8.2: Confusion matrix for the 5-way task

Category	ign	neg	pess	pos	opt
ign	1079	23	17	0	1
neg	40	580	16	0	0
pess	27	22	427	0	8
pos	0	0	0	84	66
opt	6	2	6	58	183

good options. However, whether the classifier confuses bad choices is irrelevant as long as the classifier manages to give the best hypothesis in the list the highest level of confidence of the list, i.e. manages to choose the best hypothesis.

8.3.4.2 Rule-based classification

To compare the results obtained with TiMBL with another machine learning algorithm we had Ripper (Weka's JRip) perform the same learning task on the same data. By using Ripper we would also be able to get some rules that could give us clues about what features might be interesting for this task. With default settings and by using all features for the five-way classification we reached a classification accuracy of 84% by constructing and using 34 rules. This is slightly lower than with with TiMBL. The three topmost rules are shown below:

(46) (Freqmatch=freq) and (Hyprank>=2) and (Gram=gram) => Conf= POS
(Freqmatch=freq) and (Nrofhyp <= 5) => Conf= OPT
(Freqmatch=freq) and (Gram=gram) => Conf= OPT

The rules show that just as we saw in the TiMBL experiment the FreqMatch feature and the grammaticality feature Gram are important features. This means that by parsing a hypothesis and getting a grammatical score for it and then seeing if the interpretation of the hypothesis corresponds to the most frequent interpretation in the N-Best list we can get a long way. However, the classifier uses many more features than these.

8.3.4.3 Re-ranking results

To be able to compare our classifier with the human experiments we tested our classifier on the same 40 N-Best lists that were used for the human experiments. The re-ranking is done by using the classification results to re-rank the lists with the selection procedure explained earlier. The 40 N-Best lists were converted into 391 training instances using the same 5 feature groups as before. Again we performed two different classification tasks (five-way and three-way classification) and estimated the baseline by choosing the most frequent move (NEG) in all cases. Figure 8.4 shows the confidence classification results.

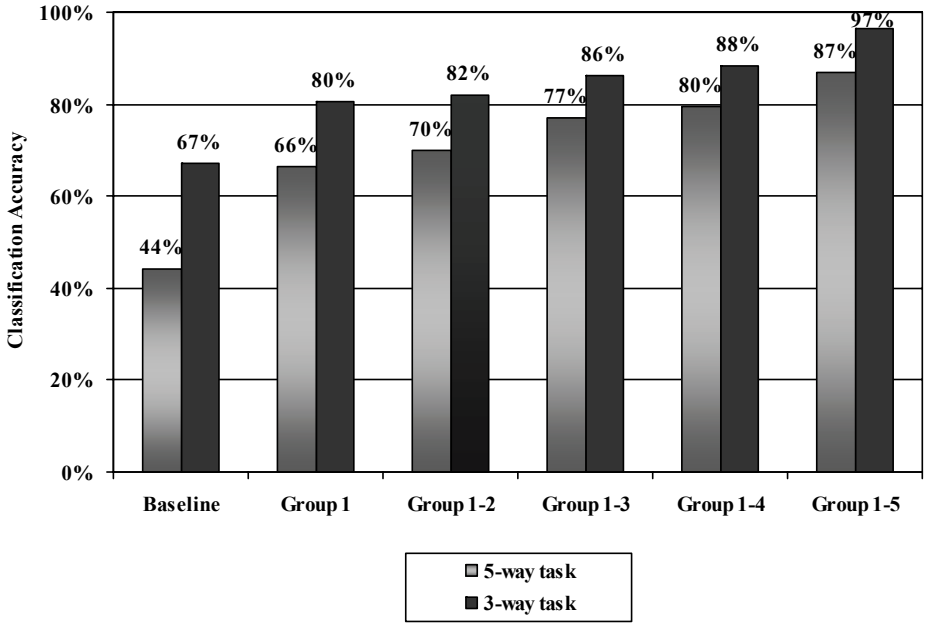

Figure 8.4: Classification results on the 40 N-Best lists: Adding context

We can again see how performance improves incrementally by adding context (i.e. group 2-4). It should be observed that once more group 5 makes an important contribution to the overall task. Once again, the classifier performs much better on the easier 3-class task which would also be a possible grounding scale to use.

To investigate the impact of the features that are normally used for re-ranking and grounding, i.e. confidence score and grammaticality, we tested these features alone. Results are shown in Figure 8.5. When using the grammaticality feature alone accuracy goes slightly up to 52% which shows the importance of parsing N-Best lists. This relatively positive result is expected as we actually use the grammaticality feature as a condition in the automatic labelling. However, the accuracy is far from the results obtained when using all features. The accuracy when using the confidence score feature alone goes down to 18% which again indicates that confidence scores from a speech recognizer are not very reliable for the task of more fine-grained confidence levels. For the 3-way task both classifiers perform slightly worse than the baseline.

It is hard to compare these results with the current GoDiS behaviour as GoDiS only considers the first hypothesis of the list. However, if we compare how GoDiS would label the first hypothesis in each list with the automatic labelling of these we see that GoDiS would get a labelling accuracy of 22.5%. We can also see that the most common label that GoDiS would choose based on the confidence scores we have for these hypotheses is the PESS label. 62.5% of the time GoDiS would have chosen a pessimistic grounding response for these hypotheses and would have tried to explicitly confirm the recognition. A user getting system responses such as "X, is that correct?" constantly would proba-

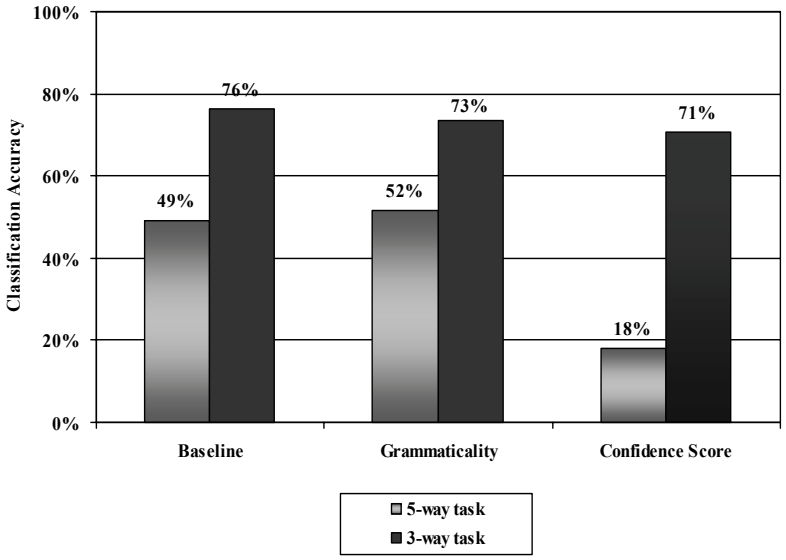

Figure 8.5: Classification results: Grammatical and confidence score features

bly feel that the dialogue flow is a little sluggish. Besides, when it is the case that the system only confirms a conceptually correct hypothesis 20% of the time and thus mostly confirms misrecognitions and misunderstanding (a False Confirmation (FC) rate of 80%) we can imagine the impression the user will get of the system. Our classifier is much less pessimistic, only 35% of the time, and most of the time (a FC rate of only 14%) it would try to confirm a correctly recognized dialogue move. With such accuracy, the classifier can permit itself to be a little more optimistic in its grounding behaviour.

Looking at the confusion matrix in 8.3 for the 5-way classification (when using all feature groups) we can see that the confusions do not seem to endanger the dialogue system behaviour as they appear on the same part of the scale. It is never the case that an incorrect concept is accepted and most confusion seems to be between the ignore and negative categories. Again, as expected, the ignore and negative categories are prominent. As pointed out earlier, whether the classifier confuses bad choices in a list is irrelevant as long as the classifier manages to give the best hypothesis in the list the highest level of confidence of the list.

We can use the classification results to re-rank the N-Best lists with the selection procedure explained earlier. This gives us re-ranking results for our five-way classifier that we can compare to the human results as both re-ranking experiments were carried out on the same test set. The results in Figure 8.6 show how the context-based classifier and ranker actually performs slightly better (although not significantly) than the human rankers and considerably better than the baseline: *the topmost chooser*. However, recall that there were some human subjects that reached 55% SA and 90% DMSeqA in the second experiment presented in Section 8.3.2.3. Although the three-way classifier outperformed the five-way classifier on the classification task it does not gain anything in the ranking task

Table 8.3: Confusion matrix for the 5-way classification

Category	ign	neg	pess	pos	opt
ign	45	19	0	0	0
neg	53	169	1	0	0
pess	8	2	48	0	0
pos	0	0	4	8	10
opt	0	2	1	1	21

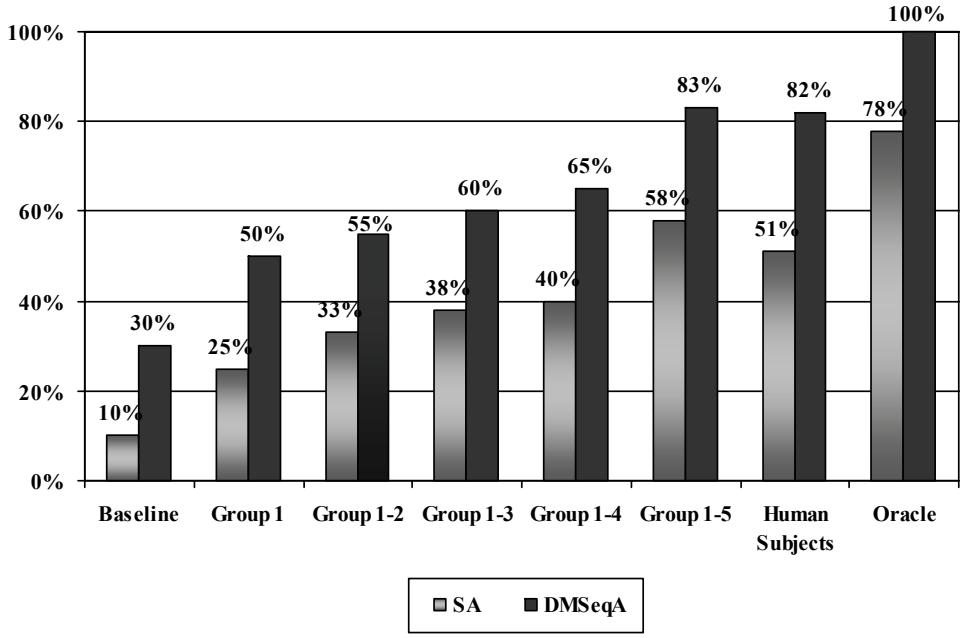

Figure 8.6: Re-ranking results for 5-way classifier: Adding context

but performs very similarly and is thereby not reported here. Using this five-way classifier instead of the topmost choice strategy, as is done in the current version of the GoDiS system, would boost the recognition performance considerably. We get an improvement of 48 percentage points in SA (significant at $p < 0.0001$) and are able to choose a hypothesis that matches the transcription conceptually 83% of the time in comparison with 30% of the time for the baseline. We can also see how both SA and DMSeqA increase for every feature group added which shows once again how the classifier profits from dialogue context features. A closer look at the cases where things go wrong (seven cases if we look at DMSeqER) shows that the classifier always chooses one of the lower scale confidence classes for incorrect cases, i.e. PESS, NEG or IGN, which means that the dialogue system need never accept such cases without first checking with the user.

8.3.5 Discussion of results

This section has presented results from experiments with human subjects where we have investigated their ability to rank ASR hypotheses using dialogue context. Based on the results of these experiments we have explored how an automatic machine-learnt ranker can profit from using dialogue context features. An evaluation of the ranking task shows that both the human subjects and the automatic classifier outperform the baseline (i.e. always choosing the topmost of an N-Best list) and that they perform better and better the more dialogue context is made available. Actually, the automatic classifier performs slightly better than the human subjects and reduces SER considerably in comparison to the baseline. The results show a considerable ASR performance improvement in comparison to the baseline and indicate that dialogue flow would become much smoother.

Although the data used for the experiment is heavily domain dependent the features we have considered are domain independent and exist in all GoDiS applications. This means that with a moderate amount of material from automatically generated logs available it would be easy to build a classifier and ranker for a new domain. However, what we would really like to aim for is to build this classifier into the GoDiS system as a domain-independent ranker that could be used for any domain. In the following experiment we will turn to the AgendaTalk domain to investigate the possibility of such a confidence classifier and ranker.

8.4 Second experiment: Confidence classification of ASR hypotheses using acoustic, lexical, semantic and pragmatic features

The encouraging results in the previous experiment open up for more investigation of the possible role of different knowledge sources in confidence annotation. As mentioned earlier, a speech recognizer's confidence accuracy is crucial to be able to use speech recognition successfully. If the ASR fails on knowing when it is doing well or not, then the dialogue system's behaviour gets very risky, even if the speech recognition accuracy is high, as it may well accept something falsely recognized.

In the previous experiment, the test set for re-ranking was minimal (40 lists) and the distribution biased towards difficult cases. The reason for this was that we wanted to use the same test set as for the human experiment where the focus had been on investigating how well they could re-rank "re-rankable" lists. In this experiment we hope to be able to show that these results were not misleading but that the approach will also give good results on a bigger and randomly selected test set. In this case, we have used data from interactions with the AgendaTalk system. We will also investigate further what features contribute to a good confidence classification behaviour. Can we profit from lexical, semantic or pragmatic knowledge sources? As in the previous experiment we will first build a confidence classifier using machine learning to give a recognition confidence to ASR hypotheses. Based on the

classification made on a test set we will then re-rank N-Best lists and estimate the possible recognition improvement.

8.4.1 The data

For this experiment, we have used logs (in the TrindiKit format described in Section 3.5) from 125 interactions with the AgendaTalk system. The main part of these dialogues was collected by having students of Computational Linguistics and employees of the Department of Linguistics (at the University of Gothenburg) interact in Swedish with the system. The logs collected in this way were also used in Chapter 6. A smaller part of the interactions is from members of the Dialogue Lab Group including some of the dialogue system developers. The grammar-based SLM from Chapter 4 was used for recognition. The logs include, apart from the information state, all the information sent from ASR during interaction not limited to the Top-1 ranked hypotheses (hereafter abbreviated as Hyps) but including up to 10-Best lists. A summary description of the data is given below:

(47) 125 dialogues

1752 audio files with transcribed user utterances

6% of the transcriptions are considered noise or crosstalk and thereby do not have any transcription

11926 ASR Hyps extracted from logs representing the 1752 audio files (1752 N-Best Lists).

10% of the N-Best lists were withheld for later testing (175 lists, 1166 Hyps).

By taking a closer look at the audio files there seems to be a very low agreement between the speech recognizer and the transcribers on what should be rejected in the data. The speech recognizer especially have problems with speech not directed to the system, so-called *crosstalk*. It hardly ever rejects crosstalk but instead attempts to recognize it. The confidence mean on crosstalk indicates that many times the recognizer is giving quite high confidence to speech that clearly should be rejected. The difficulty of detecting crosstalk in speech was discussed in Section 2.4.1.2. We have decided to include crosstalk in the data to investigate if we can achieve a more confident way of identifying crosstalk and thereby be able to treat it properly.

All transcriptions and hypotheses have been parsed with the AgendaTalk GF grammar. In addition to this we have used the Prolog Lexicon Resource to give a more robust dialogue move interpretation. We have thereafter compared the transcriptions and the hypotheses both on the word level and the semantic level. The following figures sum up the structure of the data.

(48) 55% of the ASR Hyps are parsed by the GF grammar

73% of the Top-1 ASR Hyps are parsed by the GF grammar

64% of the transcriptions are parsed by the GF grammar

3% of the Hyps does not match any dialogue move according to the Prolog Lexicon

993 Top-1 ASR Hyps match transcription (SER of 43%)

1202 ASR Hyps match transcriptions (N-Best SER of 31%)

The parsing figures show that the top-ranked hypotheses are more often grammatical (according to the GF grammar) than lower ranked hypotheses. What we also see is that many of the transcriptions are not parseable by our GF grammar. This is due to either OOV words, in-domain phrases or out of domain phrases not covered by the GF grammar. It was found that even if the GF grammar had been written for spoken language it was still too textual and restricted compared with the users' creative use of language. What we also can derive from the figures is that in 12% of the cases the correct recognition (exactly the same as the transcription) exists somewhere lower down in the list than in the top-ranked place. The N-Best SER of 31% is thereby the Oracle SER we could obtain if using an optimal re-ranking method. This means we have the possibility to improve the recognition on the sentence level considerably. As the logs only contained 10-Best lists we were limited to this N-Best length. However, we have tested running the same audio files in batch, setting the N-Best parameter to different lengths. Using 50-best lists would have given us an N-Best SER 2 percentage points lower. This means that the biggest gain is in the upper level of the lists so it does not seem that we are missing the possibility of a much greater gain by using only up to 10-Best lists. This is in agreement with other studies discussed in Section 2.4.5.

8.4.2 Hypothesis labelling

Confidence is normally estimated as a score on a scale. For the purpose of the experiment and for a tighter coupling with the actual use of confidence in a dialogue system we have chosen to assign confidence labels rather than scores. We will use the same confidence classes as in the previous experiment (see Section 8.2) which gives us a classification going from maximum confidence (OPT) to minimum confidence (IGN). To investigate the possibility of detecting crosstalk we introduce a new additional class: CROSS.

All hypotheses have been classified automatically with one of these confidence classes according to the following criteria:

(49) OPT: Hypothesis identical to transcription

POS: Hypothesis grammatical and interpreted as same dialogue move sequence as transcription

PESS: Hypothesis and transcription interpreted as same dialogue move sequence (wrong slot value is accepted)

NEG: Dialogue move type agreement or partial understanding

IGN: None of the above cases apply or transcription is considered noise

CROSS: The transcription marks that the audio is crosstalk

This labelling is similar to the one in Section 8.3.3.3 apart from the classification of PESS which here accepts a wrong slot value. Table 8.4 shows a labelled N-Best list of the user utterance "Vilket datum är presentationen klockan tio?" (*Eng. What date is the presentation at ten?*). We can see that the ASR has been able to capture the user message in Top-1 and even has the correct recognition as one of the hypotheses at rank 4. The former is labelled POS and the latter OPT. However, it also includes several hypotheses with some word confusions mainly about the time of the event. The automatic labeller labels these as PESS and the correct dialogue move sequences as POS. This N-Best List also illustrates the homogeneity of ASR N-Best lists (when recognition is working properly), both on the word level and the conceptual level, with repeating patterns throughout the list.

Table 8.4: Labelled N-Best list

Rank	Hypothesis	Dialogue Move Interpretation	Conf
1	vilket datum är presentation klockan tio	ask(X^date(X)) answer(event(presentation)) answer(time(1000))	pos
2	vilket datum är presentation vid klockan tio	ask(X^date(X)) answer(event(presentation)) answer(time(1000))	pos
3	vilket datum är presentation klockan nio	ask(X^date(X)) answer(event(presentation)) answer(time(900))	pess
4	vilket datum är presentationen klockan tio	ask(X^date(X)) answer(event(presentation)) answer(time(1000))	opt
5	vilket datum är presentation klockan tre nio	ask(X^date(X)) answer(event(presentation)) answer(time(309))	pess
6	vilket datum är presentation klockan ett tio	ask(X^date(X)) answer(event(presentation)) answer(time(110))	pess
7	vilket datum är presentation till klockan tio	ask(X^date(X)) answer(event(presentation)) answer(time(1000))	pess
8	vilket datum är presentation klockan tid tio	ask(X^date(X)) answer(event(presentation)) answer(time(1000))	pos
9	vilket datum är presentation klockan tio ett	ask(X^date(X)) answer(event(presentation)) answer(time(1001))	pess
10	vilket datum är presentation vid klockan tre nio	ask(X^date(X)) answer(event(presentation)) answer(time(1000))	pess

This transcription-based labelling gives us training data marked with an accurate labelling and correctly labelled test data to calculate accuracy. A hypothesis labelled as OPT according to this labelling means that the hypothesis has been correctly recognized and should then be given maximum confidence whereas a hypothesis labelled as POS means it was correctly understood. Figure 8.7 shows the confidence class distribution of the hypotheses in our data after this labelling. A large number of the hypotheses belong to either the negative, the ignore or the crosstalk class (65.5%). This means that a lot of the hypotheses in the lists do not have much in common with what the users actually said. The majority class is the negative class which gives us a majority baseline of 39% for classification accuracy. The distribution looks a bit different if we only consider the Top-1 hypotheses. In this case the most common class is the optimistic class (57%). We can see that the ASR is not doing too badly and ranks as topmost many correct hypotheses. 67% of the Top-1 hypotheses actually agree semantically with the transcription as they belong to OPT or POS class. This gives us a semantic baseline error rate (DMSeqER) of 33%.

The purpose of the following experiment is two-fold. We want to investigate if we can obtain a classifier that can classify ASR Hypotheses into the confidence classes above. The first classifier will be based only on confidence scores. We will then build a classifier that makes use of more knowledge sources to see if we can obtain an improved classification accuracy. What we aim for is a more accurate way of estimating confidence on what has been recognized. We will compare not only confidence classification accuracy but also measure the number of falsely accepted hypotheses (FAs) and the number of falsely rejected hypotheses (FRs) as those are the most critical errors. FAs and FRs were introduced in Section 2.4.6 and further discussed in Section 3.3.5. The categorization was illustrated in

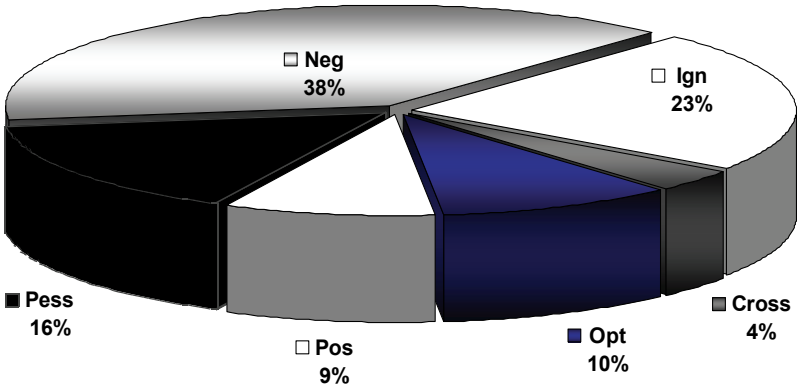

Figure 8.7: Confidence class distribution

Table 3.2. We consider FAs as hypotheses of class NEG, IGN or CROSS that are wrongly classified as OPT or POS. FRs are hypotheses that belong to the OPT or POS class which get wrongly classified as NEG, IGN or CROSS. If a wrong or correct hypothesis is classified as PESS this will not be considered a FA or FR as the confirmation strategy for the PESS confidence class should be to confirm explicitly with the user before proceeding with the dialogue. Although those errors if too common will interfere with the fluency in the dialogue, with a lot of explicit confirmations, they are not critical. Instead we consider them false confirmations (FCs) as introduced in Section 3.3.5.

The second purpose of the experiment is not only to confidence classify Top-1 hypotheses but whole N-Best lists. We hope to be able to classify the most accurate hypotheses in the list with the highest confidence class. In this way, we can then re-rank the lists based on the confidence classes the hypotheses are given. Finally, we will estimate the success of this re-ranking approach by measuring the possible decrease in SER and DMSeqER.

8.4.3 Confidence classification based only on ASR confidence scores

As described in Section 3.3.5 the current grounding behaviour in GoDiS is based on the confidence score given from the ASR for the Top-1 hypothesis. To apply the speech recognizer's confidence score against some threshold(s) is a very common approach in dialogue systems to decide grounding (confirmation) strategy. To investigate how successful this approach is we will study how the Top-1 hypotheses in our data are classified based

on different confidence score thresholds as opposed to the correct classification of these according to our transcription-based labelling.

The Top-1 hypotheses in the training data (1577 hypotheses) were first classified with the transcription-based labelling. Thereafter they were classified by comparing the confidence score of each hypothesis to four confidence thresholds. The most common confidence class of the Top-1 hypotheses (according to the transcription-based labelling) in this data is the OPT class (57%) which gives us a majority baseline. With help of the four confidence thresholds we scaled the confidence score range into our five original confidence classes (excluding CROSS) as illustrated in 8.5.

Table 8.5: Confidence score thresholds

Confidence score	Confidence Class
Score > T1	opt
T2 < Score > T1	pos
T3 < Score < T2	pess
T4 < Score < T3	neg
Score < T4	ign

Evidently, the success of this approach depends to a large extent on how the thresholds are set. We tried out different confidence thresholds by evaluating the confidence classes obtained against the transcription-based confidence classes. The best confidence classification accuracy we could obtain was 57% by setting all the confidence thresholds quite low. This accuracy is similar to the majority baseline and is probably obtained as most of the hypotheses belonging to the OPT class get a correct classification. However, this gives us an alarming FA rate of 18%. On the contrary, if we set the confidence scores high, to lower the FA rate, we get a very high FR rate (over 26%). By testing different ranges of confidence thresholds we tried to optimize the equal error rate by lowering both the FA rate and FR rate. This gave us a confidence classification accuracy of only 28% but with a FA rate of 2.8% and a FR rate of 9.2%. The confusion matrix in 8.6 shows that although the majority of the confusions are between close classes there are still too many confusions between other classes to obtain a good grounding behaviour. For example we can see that there is a huge number of hypotheses classified as the PESS class (542). This would mean that a third of the hypotheses would be confirmed explictly. This would make the dialogue flow sluggish, especially as many of the hypotheses to confirm would be completely wrong (27% of the time) and would be needed to be corrected somehow.

Apart from evaluating the current approach in GODIS this experiment has also been a way of investigating what confidence score thresholds are most optimal for the current grounding strategy. The choice of thresholds is based not only on recognition accuracy but also on concept accuracy as the transcription-based labelling takes into account dialogue move similarity. In this way we are able obtain an improved range of thresholds in comparison with the arbitrary choice in the current system as introduced in Section 3.3.5. These results were applied immediately to the GODIS system. However, as we can see, it

Table 8.6: Confusion matrix for GoDiS

Category	opt	pos	pess	neg	ign
opt	195	265	297	112	24
pos	15	30	53	9	1
pess	7	13	44	8	5
neg	3	30	96	81	35
ign	3	8	52	104	87

is hard to find an optimal choice of thresholds because when we try to optimize the FA rate the FR rate goes up and the other way around. Therefore, it will be interesting to see if we can obtain a more accurate confidence classification by using knowledge sources other than just the ASR confidence score. What we have pointed out is that it is not only the confidence accuracy that counts but also the distribution of errors. We are looking for a classification method that lowers both the FA and FR rate to approximate an equal error rate. At the same time we want to avoid too many explicit confirmations and maintain a low FC rate. We consider FAs more critical than FRs as it has been shown that these are harder to recover from (Pradhan and Ward, 2002; Bohus and Rudnicky, 2005a; Renders *et al.*, 2005).

8.4.4 Feature groups

For the machine learning experiment we will represent all hypotheses with features extracted from the dialogue logs. These features have been divided into four groups containing acoustic, lexical (and grammatical), semantical and pragmatical features. In this way we will be able to compare the possible contribution to the task of these different types of features. Some of the features are specific to a certain hypothesis whereas others will represent the N-Best List a hypothesis belongs to or make a comparison of the hypothesis (hyp) and the other hypotheses (hyps) in the list. This is a way to put the hypothesis into the context of the N-Best list it is part of. The groups are the following:

1. **Acoustic features (ASR)**
   - A: ASR features for the hyp
   - B: Comparison of acoustic features of the hyp and the hyps in the N-Best list
2. **Lexical and grammatical features (LEX)**
   - A: Lexical and grammatical features of the hyp
   - B: Word level comparison of the hyp vs the hyps in the N-Best list
3. **Semantic features (SEM)**
   - A: Semantic features of the hyp

- B: Semantical comparison of the hyp vs the hyps in the N-Best list

4. **Pragmatic features (PRAG)**
   - A: Pragmatical features of the hyp
   - B: Pragmatical features of the N-Best list

We have chosen these groups to investigate the contribution of different linguistic knowledge sources to the classification task from phonetic (acoustic) to pragmatical features. What we hope to achieve is an indication that the use of lexical, syntactic, semantic and pragmatic knowledge leads both to a better confidence model and better recognition accuracy.

In this section we will describe the features selected for the experiment. These have been extracted automatically from the logs. It should be noted that many more features were preliminary candidates but were discarded since they did not contribute to successful results. For the interesting reader these discarded features have been collected in Appendix A.

The following list enumerates the ten acoustic features obtained from the speech recognizer for each hyp. This constitutes the first group (**ASR**).

(50)
HYPRank: Ranking in N-Best list
HYPConf: Confidence score
HypConfStdDev: Standard deviation of word confidence scores
HypMinWordConf: Lowest word confidence score
HypProb: Probability score
HypConfDropFromTop: Difference of hyp confidence score with top score
HypProbDropFromMean: Difference of hyp probability with top probability
HypConfUpFromLast: Difference of hyp confidence score with minimum confidence score in N-Best list
DiaHisConf: Mean confidence score during dialogue
DiaHisConfStdDev: Standard deviation of confidence score during dialogue

The first five acoustic features represent the ASR's confidence concerning the hypothesis. Features six to eight represent the confidence of the hypothesis in comparison to the top-ranked hypothesis in the list and the lowest ranked hypothesis. The last two acoustic features represent the progress of the confidence score during the dialogue.

The second group (**LEX**) contains features on the syntactic and lexical level.

(51)
HypGram: Grammaticality of hyp according to GF grammar
HypWordLen: Length in words of hyp
MeanWordLen: Mean word length of hyps in N-Best list
StdDevWordLen: Standard deviation of word length of hyps in N-Best list
HypWLenComp: Hyp is longer, shorter or same as MeanWordLen

ListWordVariety: Word variety in N-Best list

WordPurity: How frequent are the words in the hyp in the rest of the hyps in the N-Best list?

HypWordUniqueness: How unique are the words in the hyp in comparison to the words in the rest of the N-Best list?

The HypGram feature represents whether the hypothesis is grammatical or not. The following four features represent word length. The sixth feature is an attempt to represent the *word variety* in the list, i.e. if the hypotheses in an N-Best list have a lot of words in common or not. This is given by the ratio between the number of unique words to the total number of words in the list. The variety is then represented on a scale from 0 to 1 where 1 is high variety. This metric has been used by Higashinaka *et al.* (2005) to represent what they called slot variety during a discourse. The WordPurity feature represents how frequent the words in the hypothesis are in the rest of the hypotheses in the same list. The term *purity* is borrowed from Hazen *et al.* (2002) where it is defined as "the N-Best purity of a word is the fraction of N-Best hypotheses in which that particular hypothesized word appears in the same location". In our case, for simplicity, we do no take into account the position of the word and we estimate purity a little differently. For each word in a hypothesis we count the frequency of that word in the N-Best list and sum up a frequency score which is divided by word length. The term purity has also been used by Higashinaka *et al.* (2005) to represent slot purity during a discourse, i.e. how often a slot has been used. The last lexical feature is a measure of the *uniqueness* of a hypothesis in comparison to the rest of the list. This is estimated by counting the number of words that are unique for the hypothesis, i.e. do not exist in any other hypothesis in the N-Best list, and thereafter we divide by word length. This will give us a scale from 0 to 1 where 1 is total uniqueness.

To investigate the role of semantics we have extracted the following ten semantic features from the dialogue move tagged N-Best lists to form the third feature group **SEM**.

(52)
HYPDM: Dialogue move interpretation of hyp

HypDMLen: Number of dialogue moves of interpretation

HypMinDMScore: Minimum dialogue move score

HypDMScoreStDev: Standard deviation of dialogue move confidence Scores

ListDMSeqVariety: Dialogue move sequence variety in N-Best list

ListDMVariety: Dialogue move variety in N-Best list

FreqMatch: HypDM matches most frequent dialogue move sequence?

InclMajDM: HypDM includes most frequent dialogue move?

DMPurity: Frequency of Hyp dialogue moves in the rest of the Hyps.

HypDMUniqueness: How unique are the dialogue moves in the Hyp in comparison to dialogue moves in the rest of the N-Best list?

Firstly, we represent the dialogue move sequence of the hypothesis and the number of dialogue moves. By using the word confidence scores from the ASR when parsing the

hypotheses into dialogue moves we obtain dialogue move confidence scores. We represent the standard deviation of these and the lowest dialogue move score. The variety feature from the previous group is here used at the semantic level to represent the variety of the dialogue move sequences in the list but also the variety of individual dialogue moves. These are calculated as the ratio of the number of dialogue moves (or dialogue move sequences) occurring only once in the N-Best list to the total number of dialogue moves (or dialogue move sequences). The feature FreqMatch represents whether the dialogue move sequence of the hypothesis matches the most frequent dialogue move sequence in the list. With the InclMajDM feature we capture whether the dialogue move interpretation of the hypothesis includes the most frequent dialogue move in the list. The last two features estimate purity and uniqueness on the semantic level by comparing the dialogue moves of the hypothesis to the interpretations of the other N-Best list hypotheses. This is calculated in the same way as for the corresponding lexical features. In our data we can see that hypotheses are more unique in words than in dialogue moves. Also, the percentage of hypotheses that include the most frequent dialogue move of the N-Best list it belongs to is reasonably high (82%). This means we have a high semantic homogeneity in N-Best lists with a lot of repeated dialogue move patterns.

The last group (**PRAG**) contains 17 pragmatical features extracted from the information state in the dialogue logs (see Chapter 3 for a description of the GoDiS information state).

(53)
SysDM: Last system dialogue move (e.g. ask(action))

LASTUSRDM: Last user dialogue move

ACT: Shared action

DMPredMatch: HYPDM matches predicted dialogue move

QAMatch: HYPDM makes up a valid adjacency pair with SysDM (e.g. answer matches question)

RelQUD: Relevance to QUD

RelISS: Relevance to Shared issues

SolvePlan: HYPDM solves part of current plan

ShortDiaHis: Last 5 dialogue moves

ShortUsrDiaHis: Last 5 user dialogue moves

DiaLen: Dialogue length

SysOnTrack: Number of negative perception, semantic and understanding ICMs.

UsrOnTrack: Number of negative contact ICMs, number of negative perception ICMs and number of rejections of propositions

SameAsLastUsrDM: Is HYPDM the same as last user dialogue move?

RepDM: Has HYPDM been performed earlier in dialogue?

PropRepeat: Does HYPDM include a proposition mentioned previously?

PropActive: Does HYPDM include a proposition that is active in shared commitments?

The features SysDM and LASTUSRDM contain the last system dialogue move and the last user dialogue move in an abbreviated form. ACT holds the last shared action that has been performed. The feature DMPredMatch takes into account our expectations on user dialogue moves from Chapter 6 by using the implemented dialogue move prediction rules from Chapter 9. We use these to check whether the head dialogue move type of the hypothesis corresponds to the predicted dialogue move type. Our assumption is that unexpected moves should be given lower confidence and dialogue moves that fit the context should get higher confidence. QAMatch is an attempt to represent whether the hypothesis makes up a valid adjacency pair with SysDM. From the data, we can see that 62% of the hypotheses match the predicted dialogue move and that 79% of these hypotheses would give a QA match. The features RelQUD and RelISS represent whether the hypothesis is relevant to QUD or Shared Issues. The SolvePlan feature gives a match if the hypothesis can solve part of the current plan. This is a way of representing whether the hypothesis would lead to question accommodation. We also represent short versions of the dialogue history as keeping the whole dialogue history as a sequence of moves was found not to be fruitful. Rather than keeping track of dialogue moves we only represent the dialogue move types (e.g. ask). In addition, we have kept the dialogue length as a feature. The SysOnTrack and UsrOnTrack features are an attempt to numerically represent how the interaction has gone so far both from the system's perspective and from the user's perspective by counting negative ICMs (see Section 3.3.5 for introduction to ICMs). The last four features contain information about whether the dialogue move interpretation and the possible propositions of that move sequence have been performed earlier by the user either in the previous turn or earlier in the dialogue.

The 45-dimensional feature vector below exemplifies the representation we give to hypotheses. This hypothesis has been labelled as OPT meaning that it is a correct recognition in spite of the fact that it has rank 3 and a low confidence score (31).

Acoustic features: 3, 31, 8, 34, -84, 0, -25, 17, 0, 0
Lexical and grammatical features: gram, 2, 2, 0, shorter, 0.5, 2, 0.5
Semantic features: request(add_event), 1, 42, 0, 0.9, 0.6, infreq, nomajdm, 1, 1
Pragmatic features: askalt, [ ], top, predicted, noqamatch, irrelevant, irrelevant, solves, greet, [ ], 3, 0, 0, diffdm, notrepeat, noprop, noprop
Confidence Class: OPT

8.4.5 Experimental results

We will use two machine-learners, TiMBL and Ripper (Weka's JRip), to train confidence annotators based on the selected features from the different linguistic knowledge groups.

We will evaluate the importance of individual features to choose the best possible feature set and investigate the contribution of each linguistic group. We will not only evaluate our 6-way classification but also compare to results when using less classes by merging some of the confidence classes. In addition, a comparison of the resulting confidence model with the more simplistic ASR confidence score model from Section 8.4.3 will be given. Finally, based on the results of the confidence classification, we will make an attempt to re-rank the N-Best lists in the test set and measure possible gain in recognition and concept accuracy.

8.4.5.1 Memory-based confidence classification

We started by dividing the training data into a training set and a development test set. The development test set was created by randomly extracting hypotheses from the training set. This development test set, unlike the test set we excluded at the outset, does not include ordered N-Best lists, but has a random distribution. It was used to choose the optimal feature set and to compare different confidence class sets. The original six confidence classes can be merged into fewer classes if the 6-way classification task turns out to be too complex. For the 5-way classification we classify crosstalk as IGN. In the case of 4-way classification we merge the IGN and NEG classes to a common REJECT class. For the 3-way classification, in addition, we do not distinguish between hypotheses of the OPT and POS class as they have the same conceptual meaning. We used TiMBL to train these confidence classifiers based on the training set of hypotheses represented as feature vectors and labelled automatically with our transcription-based labelling.

We evaluated the four confidence classifiers obtained (6-way, 5-way, 4-way and 3-way) on the development test set. First, we trained them with all the features from all the groups. Nevertheless, the best possible results were obtained by using an optimized feature set with 39 of the 45 selected features. The discarded features were: HypConfStdDev, StdDevWordLen, ListWordVariety, SameAsLastUsrDM, ShortDiaHis and ShortUsrDiaHis. The five most informative features were shown to be: HypGram, HypDM, FreqMatch, InclMajDM and HypDMUniqueness. As expected the grammaticality feature plays a determining role. This is of course due to the fact that grammaticality was used as a constraint in the transcription-based labelling. What seems to be clear is that comparing the hypotheses on a semantic level seems to be fruitful. Knowledge about whether the interpretation of a hypothesis corresponds to the interpretation of the other hypotheses in the list is shown to be crucial. Other informative features seemed to be HYPRank, RelQUD, RepDM, SolvePlan and ProbDrop. Although the rank of the hypothesis seemed to contribute to the task, the confidence score *per se* did not turn out to be a crucial feature. Relevance of the hypothesis to the information state appears to be of moderate importance.

If we consider Figure 8.8 we find that all our classifiers outperform the majority baseline. For the 6-way task the classification accuracy increases from 39 to 77% (significant at $p < .0001$). Also, by using the optimized feature set we get significantly better results than when using all the features (significant at $p < .0005$). We can also see that the less confidence classes the higher the confidence accuracy. This is not surprising as the task gets easier. Although, there is shown to be little difference between the 6-way and 5-way

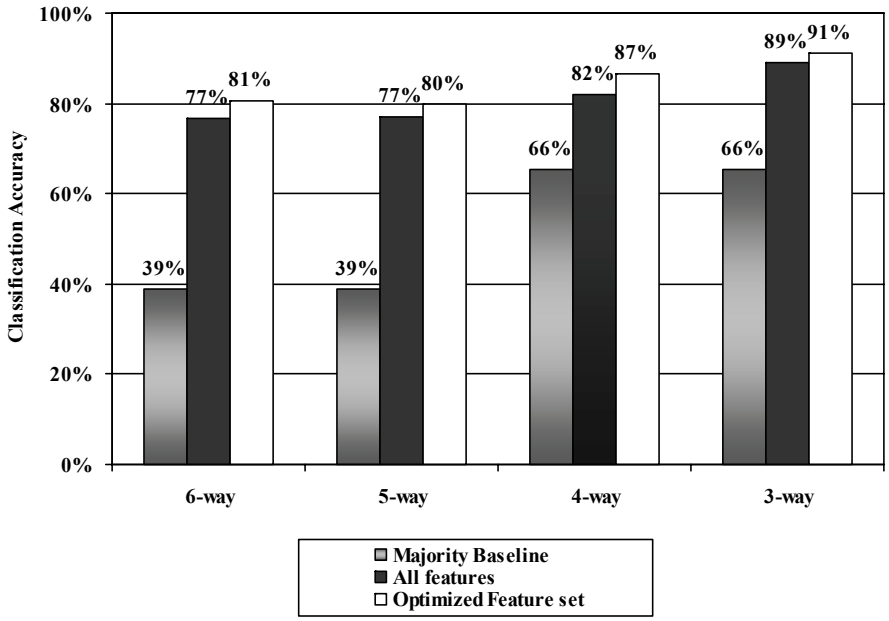

Figure 8.8: Classification with TiMBL: Different tasks

task. This means that the classification of the additional class CROSS is accurate enough not to disturb the classification results.

Consider the classification of crosstalk in the confusion matrix in Table 8.7 for the 6-way classification task. The classification of hypotheses as crosstalk is highly accurate (82% accuracy) which means we could well use this additional confidence class. In no case does crosstalk lead to a false acceptance. Furthermore, the confusion matrix shows how most

Table 8.7: Confusion matrix 6-way classification on random test set

Category	opt	pos	pess	neg	ign	cross
opt	104	55	20	10	5	0
pos	65	118	14	6	0	0
pess	34	16	242	27	8	0
neg	4	1	17	773	42	7
ign	3	3	6	54	426	4
cross	0	0	2	5	9	72

of the confusions are made between neighbouring classes (e.g. between OPT and POS). The FA rate is only 0.5% while the FR rate stays at 1.0%. These results are surprisingly promising and give us a much better confidence model than with confidence scores. From the matrix we can tell that the PESS class is only used 14% of the time and only 8% of the hypotheses classified as PESS are FCs (FC rate of 1%). This means we would not get an

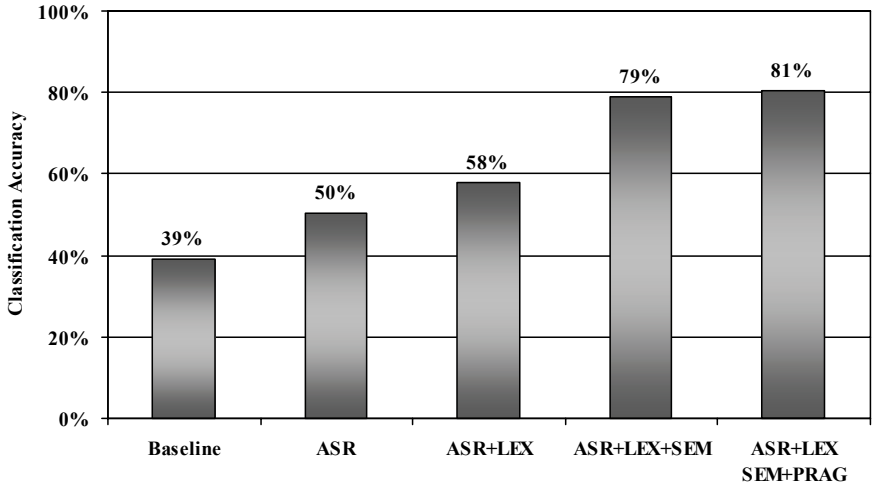

Figure 8.9: Classification with TiMBL: Adding linguistic knowledge

unnecessary pessimistic grounding behaviour with too many explicit confirmations as the classifier based only on the ASR confidence score in Section 8.4.3.

Consider Figure 8.9. This table shows stepwise addition of the feature groups representing more and more linguistic knowledge. By only using the acoustic features (the **ASR** group) we get an important improvement over the majority baseline (from 39 to 50%). However, the accuracy stays far from the result when using all features (81%). By adding grammatical and lexical features (the **LEX** group) the accuracy goes up considerably (from 50 to 58%). Including semantic features (the **SEM** group) improves the result even more (58 to 79%). By adding the last group of pragmatic features (**PRAG**) we get a slight improvement (79 to 81%) although not statistically significant. These results definitely show that the linguistic features matter even though we were not able to show the importance of the pragmatic features. For the 5-way, 4-way and 3-way classifiers the results were similar with an increasing accuracy as groups were added and with most improvement by the addition of group 3 (**SEM**), the semantic features. As the order of the additions of the groups may distort the result we made a second test. The assumption is that we will get more improvement in the beginning as we have few features than in the end as we already have almost all the features. In Figure 8.10 we can see the results when starting with group 4, the pragmatic features (**PRAG**), and thereafter adding semantic, lexical and acoustic features respectively. The first observation is that the pragmatic features give a much better starting point than the acoustic features. Again we can see that the semantic features seem to play an important role. The last addition, this time the acoustic features only improves the results minimally and is not statistically significant. For the 5-way, 4-way and 3-way classifiers the results were similar with the exception that for the 3-way task the accuracy actually went down when adding the acoustic features (group **ASR**).

It actually seems to be the case that the order has an impact on the results and we could easily be misled by the results when jumping to conclusions about the contribution

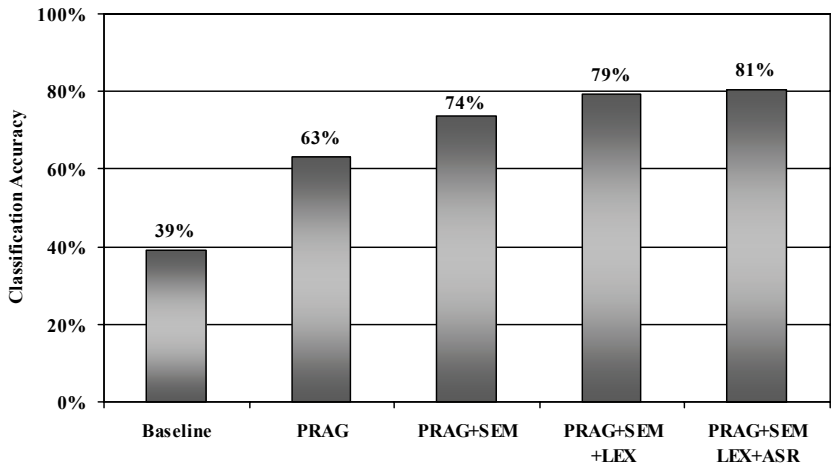

Figure 8.10: Classification with TiMBL: From pragmatic to acoustic knowledge

of each group. Consequently, we decided to build classifiers using all but one of the groups to show the impact of excluding a particular group of features. The results in Figure 8.11 show that all the linguistic knowledge features seem to contribute more to the task than the acoustic features. The pragmatic features do not have the same impact as the lexical and especially the semantic features. Neither the exclusion of acoustic features nor the exclusion of pragmatic features causes a significant change in classification accuracy. It is only when excluding the lexical and the semantic features that we get a significant negative impact on the classification accuracy. This tendency was the same also for the 5-way, 4-way and 3-way task. For the following experiments we have chosen to use the optimized 39-dimensional feature set which includes features from all of the feature groups.

8.4.5.2 Rule-based confidence classification

The memory-based learner gives us some indications of what features seem important for the task but it is hard to know what the classifier actually learns and in what way it profits from different features. For this reason, we have trained a classifier using Ripper (Weka's JRip) to see what kind of rules it learns from our feature vectors. This time, we used a randomized part of the training data and carried out a 10-fold cross validation.

In a first experiment we gradually added more and more linguistic features until using all the 39 features from the previous TiMBL experiment. Figure 8.12 shows how as in the previous experiment we achieve better and better performance the more linguistic knowledge is added. Again, the most prominent gain is when adding semantic features. Analyzing the classification accuracy per class shows that the classification of hypotheses such as CROSS, OPT and NEG is highly accurate even when only using the acoustic features (the **ASR** group). This indicates that these classes are well defined by acoustic features. Adding the lexical and syntactic features improves performance for IGN and POS consid-

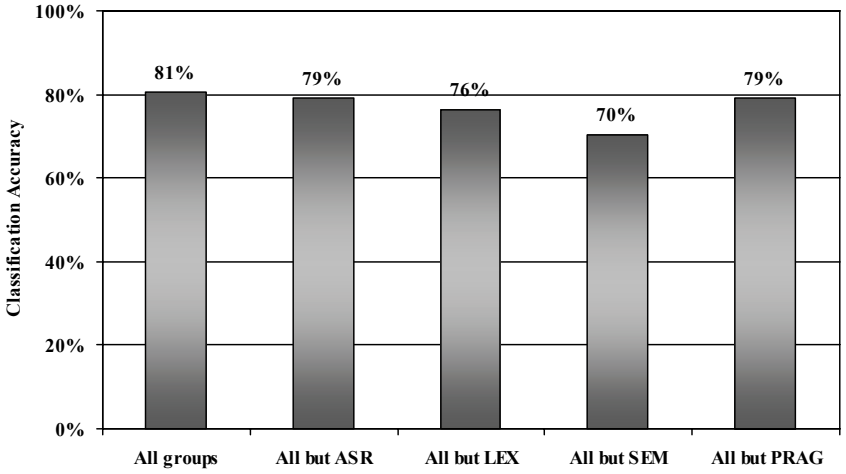

Figure 8.11: Classification with TiMBL: Excluding feature groups

erably. With the incorporation of semantic features we get an important improvement in the ability to detect hypotheses as PESS, that is, hypotheses that are perceptually different from the transcription but semantically similar. By incorporating pragmatic features the performance improves overall and the classifier gets better at classifying all types. In comparison to the memory-based classifier the pragmatic features here seem to play a much more important role and the contribution of the pragmatic features leads to a significant improvement (at $p < 0.00001$).

The confidence accuracy for the 6-way task with all feature groups was 76.2% which is slightly lower than with TiMBL (80.6%) in the previous experiment. Both the FA rate and FR rate are very low (1.8% and 2.1% respectively) although higher than with TiMBL. This means that most confusions are made between adjacent classes. To avoid cases where the patterns of hypotheses belonging to the same N-Best list would result in a learnt rule based on the common list patterns we set the minimum instance weight to learn a rule to 11. The resulting model when using all the 39 features consisted of 102 rules. Below we show ten example rules that this classifier learnt.[2] A description of the features was given in Section 8.4.4.

(54) **1** (HypProb <= -43) and (HypConf <= 41) and (HypDMLen >= 3) and (UsrOnTrack >= 6) and (DiaHisConfStdDev >= 15) and (ListDMSeqVariety >= 0.9) => ConfClass= CROSS

2 (HypRank <= 1) and (HypMinDMScore >= 56) => ConfClass= OPT

3 (QAMatch = qamatch) and (HypDMScoreStDev <= 0) and (HypRank <= 1) => ConfClass= OPT

4 (HypGram = gram) and (FreqMatch = freq) and (HypRank >= 2) and

[2]Redundancy in the rules that Ripper does not resolve has been removed for clarity.

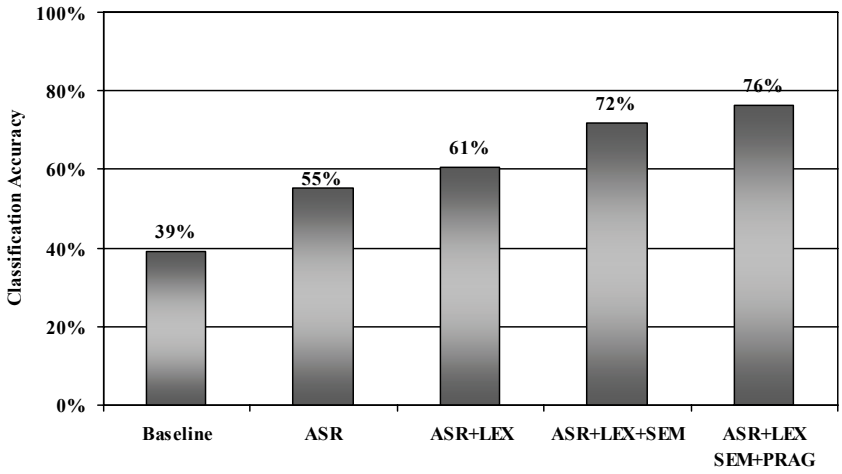

Figure 8.12: 6-way classification with JRip: Adding linguistic knowledge

(ListDMSeqVariety <= 0.3) => ConfClass= POS

5 (HypGram = gram) and (FreqMatch = freq) and (HypRank >= 2) and (HypMinDMScore >= 38) and (SolvePlan = solves) => ConfClass= POS

6 (HypGram = gram) and (HypDMUniqueness <= 0) and (DMPurity >= 8) and (HypDMLen <= 3) and (HypRank >= 3) and (HypMinDMScore >= 20) and (DMPredMatch = predicted) => ConfClass= POS

7 (HypMinDMScore >= 55) and (FreqMatch = freq) and (DMPredMatch = predicted) and (QAMatch = qamatch) => ConfClass= PESS

8 (HypMinDMScore >= 51) and (FreqMatch = freq) and (SolvePlan = solves) and (DMPurity >= 9) => ConfClass= PESS

9 (HypConf <= 44) and (InclMajDM = nomajdm) and (HypWordUniqueness >= 1) and (DiaHisConfStdDev >= 10) => ConfClass= IGN

10 (HypConf <= 53) and (HypMinDMScore >= 8) and (DiaLen <= 2) and (HypGram = ungram) and (ListDMSeqVariety >= 0.7) and (HypRank <= 8) and (DMPredMatch = unpredicted) => ConfClass= IGN

The first rule shows how hypotheses with a confidence and probability score below a certain range get classified as CROSS if the hypotheses consist of many dialogue moves, the user seems not to be on track (high number of negative feedback moves) and the dialogue move sequence variety is very high, i.e. the N-Best list holds many distinct hypotheses. The second rule classifies hypotheses as OPT if they are ranked as number 1 and the minimal dialogue move score is quite high. The third rule classifies top-ranked hypotheses as OPT whenever they make up a valid adjacency pair with the previous question and the standard deviation of the dialogue move scores in the hypothesis is minimal. The rules 4–6 exemplify how hypotheses get classified as POS. Rule 4 tells us that the hypothesis

should be grammatical, should match the most frequent dialogue move sequence in the N-Best list but does not need to be at the top of the list. It also has as condition that the variety of dialogue moves in the N-Best list should be rather low, i.e. high homogeneity. The first three conditions for rule 5 coincides with rule 4. In addition the rule conditions the HypMinDMScore and requires that the hypothesis solves part of the active plan. The sixth rule uses the grammatical condition once again and adds a requirement of optimal dialogue move purity and minimal uniqueness, i.e. the interpretation should be the same as the others in the list. Although the ranking can be rather low, the interpretation should not be longer than three dialogue moves, the minimal dialogue move score higher than 20, and the interpretation should match the predicted dialogue move. In summary, the rule tells us that if there is a high semantic homogeneity in the list and the hypothesis follows this we can be rather confident that the hypothesis is correct albeit a low rank. Rules 7 and 8 illustrate how to classify hypotheses as PESS. Both condition the value of the minimal dialogue move score and require that the hypothesis matches with the most frequent dialogue move sequence in the N-Best list. Rule 7 has as additional conditions that the hypothesis should match the predicted dialogue move and that it should be relevant to the previous question. The other example (rule 8) requires that the hypothesis solves part of the current plan and that the DMPurity is maximal. The last two rules detect hypotheses which should be considered as class IGN and therefore be rejected. Rule 9 says that if the confidence score is reasonably low, the hypothesis does not include the most common dialogue move in the list and the words recognized are totally unique and do not appear in the other hypotheses in the list the hypothesis should be rejected. The last rule shows that hypotheses with low confidence score, that are ungrammatical, does not match the predicted dialogue move, occurs in the beginning of the dialogue and that belongs to an N-Best list with high variety of dialogue moves should be classified as IGN.

We have only displayed ten of the 102 rules. If we consider all of them the most commonly used feature is the minimum dialogue move score (HypMinDMScore) which gives an indication that confidence on the semantic level is high. The next ten most used features are: HypConf, HypGram, HypProb, DiaHisConf, HypDMScoreStdDev, DMPurity, SolvePlan, DMSeqVariety, FreqMatch and DiaLen. It is interesting to see the repeated use of the semantic variants of variety, purity and uniqueness as conditional features. The same features on the lexical level seem to be much less informative. In accordance with the results from the experiment in the MP3 domain (see Section 8.3.4.2) and the previous TiMBL experiment the grammatical and the frequent match features are crucial. On the other hand we can see that novel features such as whether a hypothesis resolves part of the current plan is also of high importance.

As some of our feature values are domain-dependent and we in the end aim for a domain-independent confidence model we also built a classifier discarding the four domain-dependent features: HypDM, SysDM, LastUsrDM and ACT. This classifier performs only slightly worse with a 75.5% accuracy. A classifier with only domain-independent features therefore seems feasible. Interestingly, although the classification of IGN and POS worsen the classification of CROSS and PESS actually improves.

To show the real influence of the acoustic features we also built a confidence model

by only using the confidence score, the rank number and the minimum word confidence score. This model gave a quite poor accuracy of 49% using 14 rules and was not able to identify the PESS or CROSS class. It performed reasonably well for the OPT and NEG class. This is not surprising as the confidence scores are given by a model that has as purpose to either reject or accept in order to optimize word accuracy. This simple rule model classifies hypotheses as OPT whenever the ranking is 1 and the confidence score is higher than 64. Although the FA rate is reasonably low (2.4%) the FR rate reaches 9.6%. It is evident, that this model is too simple for the task and that the use of more knowledge clearly contributes to a better solution of the task of confidence classification. If we compare this classifier to the classifer with all our acoustic features we can see that that it is possible to improve confidence annotation even by only introducing novel acoustic features.

In contrast, we trained a classifier without using any of the acoustic features at all or any feature derived from ASR parameters, such as for example HypMinDMScore, to verify how much our other knowledge sources contribute to the task by themselves. A 69.9% accuracy indicates that the additional knowledge sources are not highly dependent on ASR features and that confidence can well be estimated without any acoustic information. This is in agreement with the previous results. It is especially the classification of the OPT class that is affected which could be explained by the fact that the only distinction between the OPT and POS is on the perceptual level. The detection of CROSSTALK also decreases considerably.

To explore the impact of the amount of training data we trained a classifier by adding in a fifth of the training data at each step. Figure 8.13 clearly shows how the performance improves as more training data becomes available even though the trend seems to be starting to stabilize. It is therefore unclear how much more gain we could get by having more training data. Surprisingly the classifier performs quite well even with only a smaller amount of data. According to these results it would be possible to achieve a reasonable confidence model with only around 2400 hypotheses (i.e. 240 10-Best lists).

In an additional experiment we explored the difficulty of the 6-way task by creating more simplified models that only distinguished between 5, 4, 3 or 2 confidence classes. Figure 8.14 shows the confidence accuracy for the different tasks. The 5-way task differs from the 6-way task by considering crosstalk as part of the IGN class. There is no significant improvement. This depends on the high accuracy which with our 6-way classifier detects crosstalk. This result is also consistent with previous TiMBL experiments (see Figure 8.8). In the 4-way task we have in addition merged the PESS and NEG class. This is a different approach than in the previous TiMBL experiment. The 3-way task makes the classical distinction of REJECT, ACCEPT and CONFIRM where the REJECT class includes CROSS and IGN, CONFIRM is given to hypotheses previously labelled as PESS and NEG and ACCEPT to hypotheses originally labelled as OPT and POS. In contrast to the TiMBL experiment we also introduced here the 2-way task where the classifier only accepts or rejects. OPT, POS and PESS are re-labelled as ACCEPT and the rest as REJECT. As seen in Figure 8.14, except for the 5-way task, the accuracy goes up as the classification task is simplified and for the binary task we achieve a very high accuracy. We also built a different binary classifier that only distinguished between crosstalk and non-crosstalk. This classifier reached 98.9%

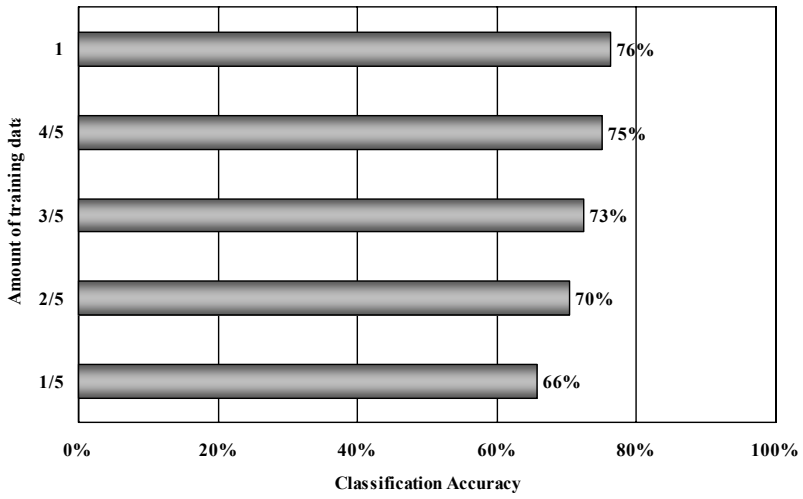

Figure 8.13: 6-way classification with JRip: Adding training data

accuracy proving that crosstalk is definitely possible to detect with the knowledge sources at hand.

Finally, we trained an optimized model by selecting the best features for JRip. The resulting model used only 25 features, all domain-independent, and consisted of 114 rules. It gave an accuracy of 78% with a FA and FR rate of 1.9% and 1.8% respectively. This shows it is possible to build an effective but simpler model. Although the rule patterns vary from the previous model when using different settings and different features there are some clear tendencies in the rules and the most important features stay the same.

8.4.5.3 Knowledge-based confidence modelling compared to confidence modelling based on ASR confidence score

To compare our new knowledge-based classifier with the ASR confidence score classifier we decided to evaluate them on the same test set. This time we used as test set the 10% of the data that had been taken away from the original data. This test set consisted of 175 N-Best lists (1166 hyps) represented by 1166 feature vectors of the hypotheses in these lists. As the confidence score classifier has no way of detecting crosstalk we used the 5-way classifier as our knowledge-based classifier. Figure 8.15 shows how the 5-way classifier performs on this test set. Once again, we see a progress as we add more linguistic knowledge. However, the addition of the acoustic features does not signficantly improve the classification accuracy. On the other hand, it was shown that the differences in classification accuracy when adding the linguistic knowledge groups were all statistically significant. When introducing the pragmatic features (the information state features) the accuracy goes up from 65% to 71% (significant at $p < .005$). However, the accuracy is overall lower on this smaller N-Best list based test set than on the randomized development test set.

We used the confidence score classifier with the best possible thresholds selected in the

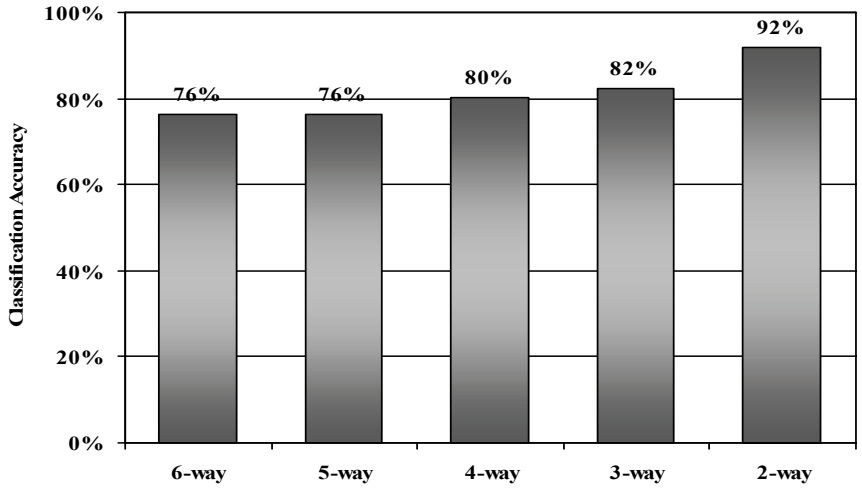

Figure 8.14: Classification with JRip: Different classification tasks

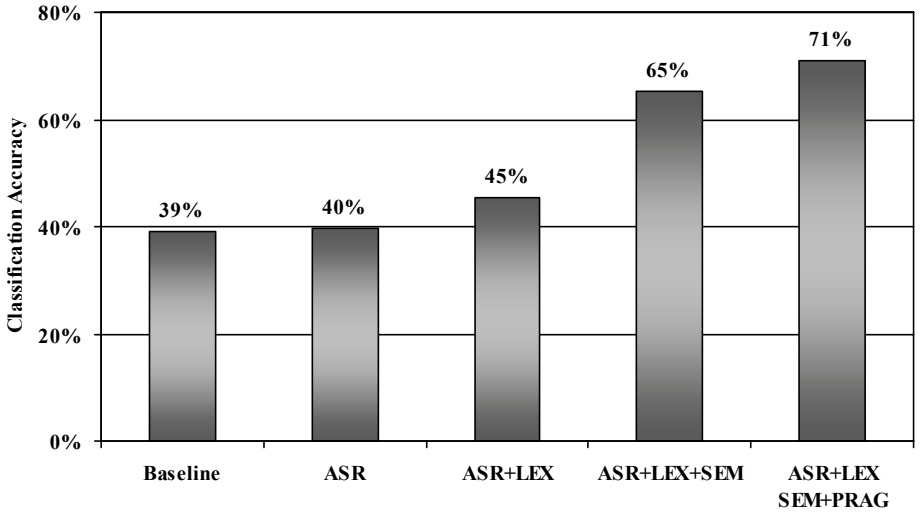

Figure 8.15: 5-way classification on held-out test set: Adding linguistic knowledge

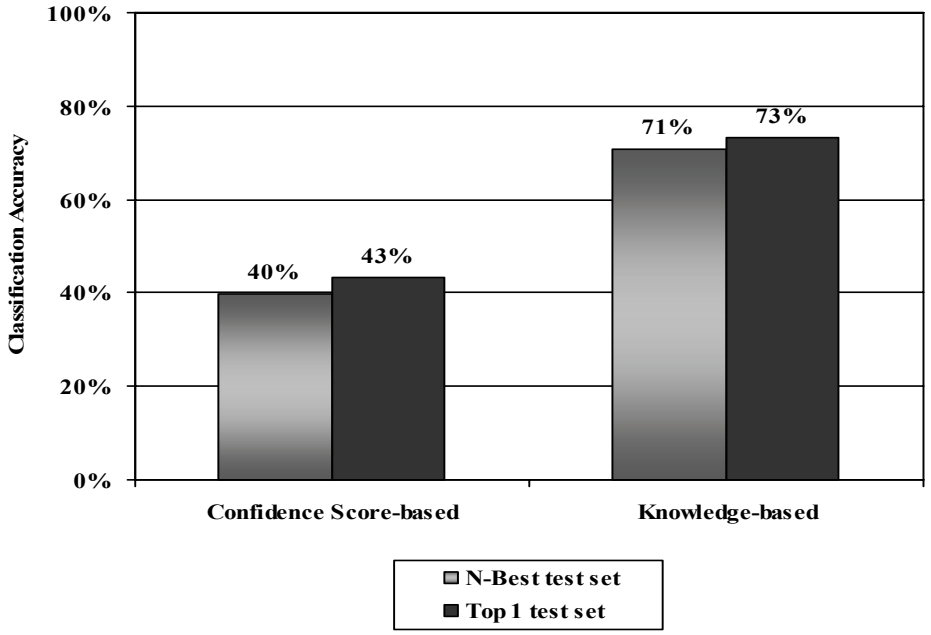

Figure 8.16: 5-way confidence classification: Comparing classifiers

experiment presented in Section 8.4.3 and the 5-way knowledge-based TiMBL classifier (based on the best possible feature set) from Section 8.4.5.1 to simulate the two approaches to confidence modelling. We used the two classifiers to label the test set with five confidence classes and then compared accuracy against the transcription-based labelling. The results in Figure 8.16 show confidence classification accuracy on the whole test set but also only for the Top-1 ranked hypotheses. In both cases the knowledge-based classifier outperforms the ASR confidence score classifier significantly. However, as stated earlier the importance is not only to confidence classify with much higher accuracy but to avoid making risky confusions.

If we compare the FA and FR rate for both approaches on all the test hypotheses it is clear that the higher accuracy of the knowledge-based approach also leads to less critical errors. Both FA and FR rate is only 1.4% whereas the confidence score classifier has a FA rate of 4.5% and a FR rate of 5.7%.

On the other hand, the confidence score classifier cannot be used for re-ranking as the confidence scores follow the ASR's ranking. This means that a fairer comparison of the classifiers' labelling should be made only on the performance for Top-1 hypotheses as these are the ones that would be sent to the dialogue system. Even though the knowledge-based classifier outperforms the confidence score classifier in accuracy also for Top-1 hypotheses, the confusions made by the classifiers make the picture less clear. If we take the confusion matrices illustrated in Tables 8.8 and 8.9 into consideration we can see that the FA rate of both classifiers actually turns out to be equally low (2.3%).

On the other hand, the FR rate of the knowledge-based classifier is much lower (2.9% vs

Table 8.8: Confusion matrix based on ASR confidence score for Top-1 hypotheses

Category	opt	pos	pess	neg	ign
opt	44	24	19	10	3
pos	2	4	4	4	0
pess	2	0	5	2	1
neg	2	2	9	7	3
ign	0	0	4	8	16

Table 8.9: Confusion matrix for knowledge-based classification for Top-1 hypotheses

Category	opt	pos	pess	neg	ign
opt	91	2	3	2	2
pos	5	7	1	1	0
pess	7	1	1	0	1
neg	2	0	3	10	8
ign	2	0	0	7	19

9.7%). This means that although earlier results indicated that we would get a much lower FA rate by using the knowledge-based classifier this did not turn out to be the case for the particular test set we used for the comparison. However, we get many fewer FRs which means that we would get a much better flow in the dialogue as we would not need to ask the user to repeat her input as frequently. We can also see that for the knowledge-based classifier we have many fewer hypotheses wrongly classified as the PESS class. This would result in a more optimistic grounding behaviour with less explicit confirmations, especially less explicit confirmations of misrecognitions. A measure for confirmations was introduced in Section 3.3.5 as false and correct confirmations (FCs and CCs). The knowledge-based classifier has a FC rate of only 2% whereas the FC rate for the confidence score classifier goes up to 7%. Moreover, we would also have less implicit confirmations of correct recognitions. In other words, we have achieved a confidence annotation model that is much better at knowing when the recognizer is doing right or wrong.

8.4.5.4 Re-ranking N-Best lists based on confidence classes

It seems that by using our knowledge-based confidence classifier we can obtain a more accurate confidence behaviour in a dialogue system than by using the ASR's confidence score. The remaining doubt is whether we are able, by classifying the hypotheses in N-Best lists, to identify the hypothesis that should be given the most confidence and thereby should be selected from the list independent of its ranking. If we in this way, based on the confidence classes, are able to select more accurate hypotheses will this also propagate to recognition accuracy? To clarify this we took the test set that had been classified with our 5-way knowledge-based classifier and re-ranked it in the following way:

"Return the first hypothesis in the classified N-Best list with the highest confidence class."

We chose the 5-way task as the CROSS class is not less or more confident than the other classes – just distinct. In this way the selection procedure will first try to find a hypothesis classified as OPT, thereafter POS and so on. Consider the N-Best list example from Section 8.4.2 (on page 190) now classified with our confidence model as shown in Table 8.10. The selected hypothesis from this N-Best list would be at rank 4 as it is the first and only hypothesis classified as OPT. This means our confidence annotation model considers that this hypothesis ranked by the ASR as number 4 matches the user utterance exactly and should be given maximum confidence. In this case, the classification coincides with the correct confidence label and our ranker is actually able to select the correct hypothesis. However, in this case as the top-ranked hypothesis is of class POS the gain is only on the word level and not on a conceptual level. The difference between hypothesis 1 and 4 is the definite marker (-en) of the word "presentation" which could make a difference in some domains. Although, our simple semantics does not distinguish these cases it seems that our confidence model is able to do so. We can also see that the confidence model has discarded some of the dubious hypotheses (of class PESS) in the N-Best list and classified them as NEG.

Table 8.10: Example N-Best list with classification

	Hypothesis	Dialogue Move Interpretation	Label	Conf
1	vilket datum är presentation klockan tio	ask(X^date(X)) answer(event(presentation)) answer(time(1000))	pos	pos
2	vilket datum är presentation vid klockan tio	ask(X^date(X)) answer(event(presentation)) answer(time(1000))	pos	pos
3	vilket datum är presentation klockan nio	ask(X^date(X)) answer(event(presentation)) answer(time(900))	pess	pess
4	vilket datum är presentationen klockan tio	ask(X^date(X)) answer(event(presentation)) answer(time(1000))	**opt**	**opt**
5	vilket datum är presentation klockan tre nio	ask(X^date(X)) answer(event(presentation)) answer(time(309))	pess	pess
6	vilket datum är presentation klockan ett tio	ask(X^date(X)) answer(event(presentation)) answer(time(110))	pess	pess
7	vilket datum är presentation till klockan tio	ask(X^date(X)) answer(event(presentation)) answer(time(1000))	pess	neg
8	vilket datum är presentation klockan tid tio	ask(X^date(X)) answer(event(presentation)) answer(time(1000))	pos	pos
9	vilket datum är presentation klockan tio ett	ask(X^date(X)) answer(event(presentation)) answer(time(1001))	pess	neg
10	vilket datum är presentation vid klockan tre nio	ask(X^date(X)) answer(event(presentation)) answer(time(309))	pess	pess

To give an example where the selection procedure also contributes on a conceptual level we will take a look at what happened with our feature vector example from Section 8.4.4 (on page 197). Table 8.11 shows our example (ranked by the ASR as number 3) and the other hypotheses in the same list. The hypothesis in rank 3 corresponds exactly with the user's utterance "lägga till" (*Eng. add*) whereas the other hypotheses are wildly wrong both on the word level and the conceptual level. Our confidence model has been able to detect this and assigned the correct confidence class to the correct hypothesis and discarded the rest. This means the ranker is also able to select the correct hypothesis among bad options. It is evident that the confidence model cannot have used only acoustic features to make this distinction and detect the correct hypothesis. If we consider the feature vectors representing these ten hypotheses there seem to be primarily two features on the pragmatic level that the third hypothesis fulfills and the others do not. This reveals what features the confidence model may have used. Unlike the other hypotheses the third one coincides with the predicted dialogue move type and also solves part of the current plan. In addition, a

third feature may have played a role. The minimum dialogue move score is higher than for the other hypotheses.

Table 8.11: Example N-Best list with classification

Rank	Hypothesis	Dialogue Move Interpretation	Label	Conf
1	mötet noll	answer(event(meeting)) answer(number(0))	ign	ign
2	mötet elva	answer(event(meeting)) answer(time(1100))	ign	neg
3	lägga till	request(add_event)	opt	**opt**
4	mötet tio	answer(event(meeting)) answer(time(1000))	ign	neg
5	basket noll	answer(number(0))	ign	ign
6	mötet tolv	answer(event(meeting)) answer(time(1200))	ign	neg
7	mötet imorrn	answer(event(meeting)) answer(date(tomorrow))	ign	neg
8	till mötet noll	answer(event(meeting)) answer(number(0))	ign	neg
9	mötet nio	answer(event(meeting)) answer(time(900))	ign	ign
10	match ett noll	answer(event(match)) answer(time(100))	ign	ign

We implemented a Prolog program that automatically re-ranked the whole test set based on the classification as well as based on the transcription-based labels. It calculated ranking accuracy by comparing the re-ranked N-Best lists based on the classification to the re-ranked N-Best lists based on the labels. A selected hypothesis was considered as correctly ranked either if the transcription-based re-ranker had made the exact same selection (same rank) or if it had selected a hypothesis with the same confidence class. In this way selecting a lower ranked hypothesis in an N-Best list over a higher ranked hypothesis with the exact same label is considered a correct selection. For example, it does not matter which IGN you choose in a list of only IGN-labelled hypotheses. As a baseline we have estimated ranking accuracy for the topmost chooser which always selects the topmost hypothesis. The baseline uses the confidence classification of our knowledge-based classifier. Table 8.12 compares the results after re-ranking with the baseline results without any re-ranking (Topmost selection). The baseline is quite high which shows that the ASR system manages to top-rank the most correct hypothesis in 77% of the cases.

Table 8.12: Re-ranking results

Classification method	Ranking Accuracy	Classification Accuracy	FA	FR
Topmost selection	77.1%	73.1%	2.3%	2.9%
Re-ranker	84.0%	66.3%	6.9%	1.7%

With the use of our simple re-ranking approach based on the knowledge-based confidence classification (our 5-way TiMBL classifier) we can actually improve the ranking accuracy considerably (from 77% to 84%) and find more suitable hypotheses deeper in the N-Best lists. As exemplified in Table 8.10 some selections do not matter on the conceptual level. Finding a hypothesis which has less word errors but is interpreted in the same way as the top choice will not improve the performance of the dialogue system. In a similar manner,

wrongly selecting a hypothesis with the label POS over a hypothesis labelled OPT will not affect the performance of a dialogue system either. Therefore, we also estimated the ranking accuracy when not considering such cases as ranking errors. The ranking accuracy then goes up to 87.7% which is considerably better than the topmost chooser (significant at $p < .01$). As seen, we have also estimated the confidence classification accuracy for the selected hypotheses for each approach. Although the re-ranker is able to find more suitable hypotheses in the N-Best lists it seems that the classifier misclassifies these more often than it misclassifies top choices. The reduction of the FR rate is encouraging as it shows that the ranker is able to find much better options when available further down in the N-Best lists which would have been discarded using top selection. Unfortunately, the FA rate goes up. Whether this is the result of an unfortunate test set or indicates that the classifier is too optimistic on its selections is unclear.

Our re-ranking approach makes critical mistakes in 11 cases. Four hypotheses that should be rejected (classified as NEG or IGN) are classified as POS and seven as OPT. Four of the critical cases are hypotheses of rank 1 and are thereby not confusions introduced by the re-ranking. In only one of these cases was there a useful hypothesis further down the list. If we take a look at the other 7 confusions we can see that in 4 cases the whole list should be rejected as there are no good options in the list at all. In two cases the correct hypothesis exists in the list but is not selected. If we look at the manual transcriptions, it seems that 5 of these 7 hypotheses were regarded as noise, out of domain or included cutoffs. It is hard to say what leads to the confusions but we can ponder on what impact they would have on the dialogue. In Table 8.13 we display the erroneously selected hypotheses' dialogue move interpretations together with the dialogue move interpretations of the transcription. In this way we can see how critical the 11 mistakes really are on a conceptual level. In 5 cases the selected hypothesis is wildly wrong. In the other 6 cases we have actually managed to capture an important part of the user's message. In many of those cases the ASR has introduced some errors such as an incorrect time or an incorrect date. These would not be too hard for the user to correct afterwards. The most critical error is found in the first example where the system reacted to noise and interpreted the noisy input as a request to delete some event. In this case the whole list should have been rejected. For some reason the classifier has selected this low-ranked hypothesis and given it high confidence. Evidently, the classifier is far from perfect. The four last hypotheses have been classified as POS. If applying the grounding strategy exemplified in Section 8.2 an implicit confirmation would be given for all these which would be an opportunity for the user to correct the misunderstandings. From this perspective, the increased FA rate does not seem so critical.

The ultimate goal of re-ranking in ASR is to determine if we can increase recognition performance by selecting the best possible hypothesis in an N-Best list. In dialogue systems the most important thing is not to recognize the exact wording better but to be able to capture the user's message accurately. We will consequently measure not only SER but also how the error rate on a semantical level degrades by re-ranking. We will measure DMSeqER, i.e. how many hypotheses include some incorrect dialogue move in its interpretation. In addition we will measure DMER by measuring the number of incorrect dialogue

Table 8.13: Dialogue move interpretation of selected FAs

Hypothesis Dialogue Moves	Transcription Dialogue Moves	Conf
request(delete_event)	No Move	OPT
answer(no) answer(event(yoga))	request(top)	OPT
answer(time(100)) answer(am_or_pm(pm))	ask(X^bookings(X)	OPT
request(change_event) answer(event(dinner)) answer(time(119)) answer(date(friday)) answer(time(100)) answer(am_or_pm(pm))	request(change_event) answer(event(dinner)) answer(time(1900)) answer(date(friday)) answer(am_or_pm(pm))	OPT
ask(X^bookings(X)) answer(date(friday)) answer(time(1100))	ask(X^bookings(X)) answer(date(friday))	OPT
ask(X^bookings(X)) answer(date([ninth,february]))	ask(X^bookings(X)) answer(date([tenth,february]))	OPT
request(add_event) answer(event(training)) answer(date(tuesday)) answer(time(106)) answer(am_or_pm(pm))	request(add_event) answer(event(training)) answer(date(tuesday)) answer(time(600)) answer(am_or_pm(pm))	OPT
answer(time(300)) answer(date(friday))	answer(no)	POS
answer(event(dinner))	No Move	POS
request(add_event) answer(event(meeting)) answer(time(200))answer(date(thursday))	request(add_event) answer(event(meeting)) answer(date(thursday)	POS
ask(X^bookings(X)) answer(time(1700)) answer(date(yesterday)) answer(am_or_pm(pm))	ask(X^bookings(X)) answer(time(1700)) answer(am_or_pm(pm))	POS

moves (see Section 2.3.4 on page 33 for an introduction to the metrics).
Figure 8.17 shows that our re-ranking approach actually leads to a reduction in recognition error rates. The proposed approach is able to select hypotheses from the N-Best lists that are more accurate. We get a relative improvement of 7.7% in SA. When there is a hypothesis in the list that can be interpreted as the same dialogue move sequence it turns out that our approach is quite good at finding it. We reduce DMSeqER from 31% to 27% which corresponds to a 13% relative improvement. The relative improvement in Dialogue Move accuracy (DMA) is even greater (24%) which means we do manage to select hypotheses that capture more of the message that the user tried to convey. Unfortunately, it was not possible to prove any statistical significance of the improvements on this minor test set. The error rates for the ranking based on the transcription-based labelling (the correct labels) give us an Oracle rate and show that it is possible to improve the performance even further by selecting the best possible hypotheses. Although when it comes to DMSeqER and DMER we are actually quite close.

The re-ranking was carried out both for the 6-way, 5-way, 4-way and 3-way task. Again, there were no measurable differences between the 6-way and 5-way task. On the contrary the 4-way and 3-way classifiers that performed much better in confidence classification accuracy turned out to give much worse results when measuring recognition performance gain. By using so few confidence classes it is hard to minimize SER as OPT and POS is the same. Even on a semantic level the error reduction was much less significant. This indicates that the selection of our six (or five) confidence classes is well-founded.

8.4.6 Discussion of results

The results leave no doubt that we can profit from additional linguistic knowledge when assigning confidence to ASR hypotheses. The confidence model we achieve by taking into account much more than just the ASR confidence score is a significantly more accurate model. Our confidence model was shown to be much better than the ASR confidence score approach on assigning accurate confidence to hypotheses. Most importantly, it was shown to make less critical mistakes with very low FA and FR rates. Also, a comparison of the

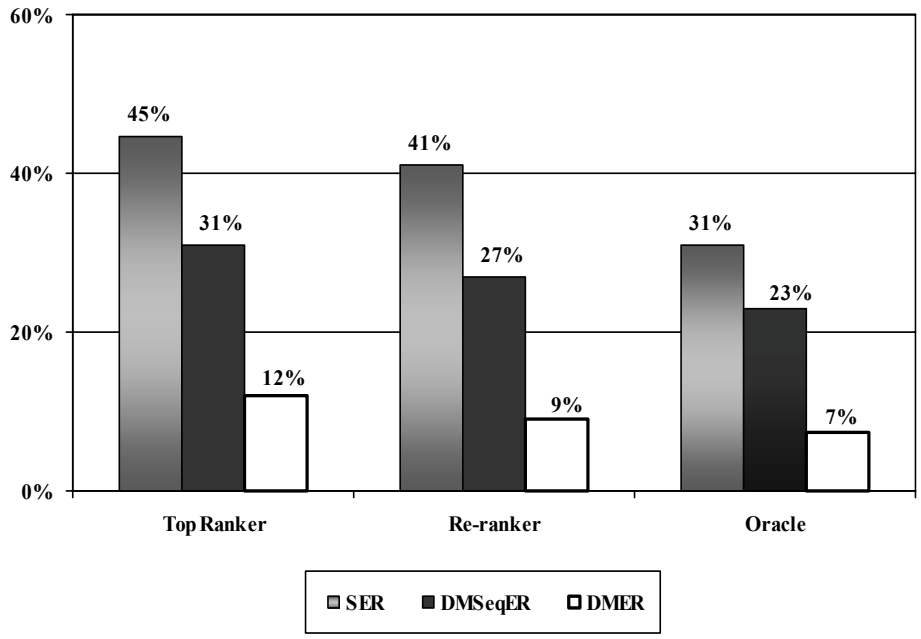

Figure 8.17: Recognition performance with re-ranking

hypotheses that would lead to confirmations in the discussed grounding model would be much less. With a confidence model that is much better on knowing when the ASR has recognized something correctly or has performed a misrecognition the grounding behaviour will improve significantly. With less doubts there will be less to confirm which will most certainly lead to a smoother dialogue behaviour.

The linguistic feature groups made an important contribution to the confidence classification task. We were able to improve classification accuracy by 61% relative when adding the lexical, semantic and pragmatic features to the acoustic features. The most important features were revealed to be on the semantic level. The results of the memory-based and rule-based classifier lead to very similar results with slightly higher results for the memory-based learner. Both classifiers have the grammatical feature (HypGram) and the frequent match (FreqMatch) feature among the most informative features. It is interesting to see how pragmatic features like DMPredMatch, QAMatch and SolvePlan contribute both to classifying hypotheses with high confidence as well as low. Most interestingly, it was shown that ASR information such as the confidence score is actually not indispensable for the task but that we can obtain good performance only with the linguistic features. It was also shown that it would be perfectly possible to obtain a classifier that does not rely on domain or language dependent features. In the JRIP experiment we could show that with more training data it might be possible to achieve slightly better results.

The results have shown that our confidence scale of 6 classes seems to be viable. Crosstalk proved to be easily classified and the CROSS class is thereby a possible additional class to our confidence scale. On the other hand, this is only useful if we want to be

able to detect crosstalk as something other than a rejection. We could for example consider using an interrogative contact ICM (icm:con*int) whenever crosstalk is detected and ask the user something like: "Are you talking to me?" The small increase in accuracy for the easier 5-way, 4-way and 3-way tasks shows that distinguishing between 6 confidence levels is actually not much more difficult than distinguishing between less.

The final ranking task based on the confidence classification shows that it is possible to automatically find more accurate hypotheses in the N-Best lists. The ranker prioritizes perceptually correct hypotheses over semantically correct ones. When discarding this distinction the ranking accuracy is even higher. The re-ranking actually leads to an improvement also in recognition and understanding performance (7.7% relative and 13% relative respectively). A survey of the critical errors showed that these actually were not that critical. In Chapter 9 we will discuss further how a dialogue system would react based on the selections and the classifications.

8.5 Summary and conclusions

In this chapter we have tried to prove with several experiments in two different domains (DJ-GoDiS and AgendaTalk) that linguistic knowledge matters to speech recognition. In the first experiment we showed how human subjects profit from the use of more dialogue context when re-ranking N-Best lists. It was shown that the more dialogue context was available the better the results. Although it is unclear what information the subjects actually made use of and how they structured it, we proposed a computational representation of it. It was thereafter shown that this computational representation actually contributed to the task of automatic N-Best list re-ranking. In fact, we achieved a similar result as for the human subjects and were able to prove how contextual features improved the performance. Both the human subjects and the automatic ranker were able to decrease SER and DMER considerably when using dialogue context as an additional information source. For the automatic task, hypotheses were represented as 21-dimensional feature vectors and classified by a machine learnt classifier into five confidence classes. These five classes: OPT, POS, PESS, NEG and IGN represent how much confidence a hypothesis should be given in order to choose grounding strategies. The N-Best lists were then re-ranked based on this confidence classification. It was shown that this confidence classification was highly accurate with an accuracy of 89% and that the FA and FR rate was extremely low (0% and 2% respectively). Such a confidence classifier would be much more reliable than one which relies on only taking into account information from the ASR.

In the second experiment we showed that with the use of linguistic knowledge, on the lexical, grammatical, semantic and pragmatic levels, it is possible to obtain a much more solid confidence annotator than by only taking into account the ASR confidence score and ranking. The experiment resembled the previous one by using machine learning to classify N-Best hypotheses into confidence classes and thereafter re-rank the classified N-Best lists. In this experiment an additional confidence class was added to represent crosstalk and it was shown that the classifier could detect crosstalk surprisingly accurately. This time each

hypothesis was represented by 45 features and divided into linguistic groups. It was shown that with more linguistic knowledge the classifier performed much better. A confidence accuracy of 81% was reached when using all linguistic features. The most important feature group was the one holding semantic features. It was also possible to show the importance of the features from the information state. Although some of the features, that were used in the experiment, are specific for information state based dialogue management and the GoDiS system, a great number of them could well be used by any dialogue system. We also showed that a domain-independent and language-independent confidence classifier is achievable by excluding some features. Nevertheless, we should bear in mind that both domains applied are quite small and that more extensive studies on larger domains and other languages are necessary. In both experiments we applied memory-based machine learning (TiMBL) and rule-based learning (Weka's JRIP). The results from both were very similar with slightly worse results for JRIP.

Once again, we analyzed the FA and FR rate to investigate the impact of the confusions. With a FA rate of 0.5% and FR rate of 1% the resulting confidence model seemed extraordinarily reliable. A comparison with a traditional classifier based only on ASR confidence score was carried out. It was shown that the accuracy and reliability of the proposed model outperformed the traditional approach and would lead to a much more subtle dialogue behaviour. Although we propose a very fine-grained confidence model with six levels the experiments have also evaluated more simpler models with as little as two confidence levels (accept and reject). These were shown to perform very well due to the simpler task. With a more accurate confidence model, the dialogue system would better know what hypotheses to discard immediately and which ones hold information useful for driving the dialogue forward. If the system is more confident there is less need for explicit confirmations and less possibility for confirmations of misrecognized input. This would lead to a smoother dialogue with fewer tedious confirmations and fewer correction dialogues. This leads to the assumption that we would also obtain a better dialogue behaviour with the use of a linguistically knowledge-based confidence annotator. Although we have discussed the experiments in relation to the grounding behaviour in GoDiS the proposed confidence model is totally independent of the grounding model.

In fact, what we might want to do is to ground dialogue moves separately rather than whole dialogue move sequences. Word confidence scores show that different parts of an utterance should be given more confidence than others and this seems to propagate also to the semantic level. This means that some dialogue moves from the same turn should be given more confidence than others. On the contrary, our approach gives confidence to the whole turn. This does not mean we have not considered how to combine the confidence from our annotator with dialogue move scores. In Chapter 9 we will show how this integration is planned to be carried out to be able to ground each dialogue move separately. We will also describe how we approach the possible implementation of our confidence annotator and re-ranker. Also, we do not discard that the approach presented in this chapter would also be applicable on a dialogue move level by confidence classifying single dialogue moves instead of whole dialogue move sequences.

Finally, we evaluated whether the confidence classification in the second experiment

would again lead to better recognition performance by being able to select any more accurate hypotheses lower down the N-Best lists. The re-ranking approach was indeed able to find more suitable hypotheses and it was possible to show a relative improvement of SA and DMSeqA of 8% and 13% respectively. This means that better confidence models are achievable by using linguistic knowledge sources and that they can be used also to re-rank N-Best lists and by that improve recognition accuracy.

Part IV

Integration and future work

Chapter 9

Integration

In Part II and Part III I have presented the results of various experiments related to speech recognition in dialogue systems. The experiments in Part II principally aimed at developing and predicting SLMs that were more appropriate to the dialogue context. In this first experimental part of the thesis additional knowledge sources were applied in a preprocessing stage in order to improve recognition performance. However, in Part III we acted on the output from the speech recognizer in order to enhance it. In these experiments higher level knowledge was applied in a post-processing stage to improve the selection of ASR hypotheses but above all to achieve a more reliable confidence annotation.

Throughout this thesis, the results from earlier experiments have served as basis for subsequent experiments. For instance, the methodology in Chapter 4 to develop grammar-based SLMs is used in Chapter 5 to develop dialogue move specific grammar-based SLMs (DMSLMs). The first experiment in Chapter 8 makes use of one of the dialogue move taggers from Chapter 7. In a similar manner, the last experiment in Part III applies the predicted dialogue move from Chapter 6 as an experimental feature.

The current chapter will describe and define the integration of the experimental results into the GoDiS system, described in Chapter 3. The first part of this chapter describes how the DMSLMs and the results from the dialogue move prediction experiment in Part II have been adapted, implemented and integrated into the GoDiS dialogue system. The subsequent part of the chapter will then define a proposal of how the confidence annotation model and re-ranking approach from Part III could be integrated into the GoDiS system.

9.1 Predicting and switching dialogue move specific SLMs

In Part II we showed how we could without training data develop SLMs that performed considerably better than an initial SRG. These grammar-based SLMs from Chapter 4 were thereafter used throughout this thesis in the subsequent recognition experiments as baseline models. They were also used in the AgendaTalk and DJ-GoDiS applications during the TALK project. In Chapter 5 we showed how the use of DMSLMs could improve

speech recognition performance further and therefore be an asset to a dialogue system. However, to be able to use DMSLMs we needed a way to predict which model to use at each particular point in the dialogue. Chapter 6 concerned this issue and proved that we could obtain a reasonable dialogue move predictor using machine learning. We used two different machine learning methods to train dialogue move predictors: memory-based and rule-based. The advantage with the rule-based learner was that we obtained explicit rules concerning what conditions needed to be fulfilled to predict a specific dialogue move. Based on these rules and our own intuitions about user dialogue behaviour in the AgendaTalk and DJ-GoDiS domains we have implemented a dialogue move predictor module in the GoDiS system and used this predictor to choose and switch DMSLMs. The following sections will describe how the prediction of DMSLMs and the switching of DMSLMs have been implemented.

9.1.1 Dialogue move prediction in the information state

First of all we needed to keep track of the predicted dialogue move by extending the information state (IS). We added a field to the IS called the PREDDM field. This field is a record holding two values: the predicted dialogue move and its confidence score. The confidence score part is for future use in case the predictor is changed to a statistical one or to one using our TiMBL classifier. We consider the predicted dialogue move as part of the private IS as this expectation will not be shared with the other dialogue participant. It is thus seen as GoDiS's internal expectation of succeeding user dialogue moves. In Figure 9.1 we can see how PREDDM has been integrated into the private IS.

$$\left[\begin{array}{lll} \text{PRIVATE} & : & \left[\begin{array}{lll} \text{AGENDA} & : & \text{Stack(Action)} \\ \text{PLAN} & : & \text{StackSet(Action)} \\ \text{BEL} & : & \text{Set(Proposition)} \\ \textbf{PREDDM} & : & \left[\begin{array}{lll} \text{MOVE} & : & \text{dmove} \\ \text{SCORE} & : & \text{real} \end{array}\right] \\ \text{DIAHIS} & : & \text{StackSet(LU)} \\ \text{TMP} & : & \left[\begin{array}{lll} \text{USR} & : & \text{TMP} \\ \text{SYS} & : & \text{TMP} \end{array}\right] \\ \text{NIM} & : & \text{NIM} \end{array}\right] \end{array}\right]$$

Figure 9.1: Private information state with dialogue move prediction

9.1.2 The moment of prediction

Before describing how the prediction model has been implemented we need to clarify when exactly dialogue move prediction would be triggered in our system. When do we create expectations about what the listener will say? After producing an utterance, while realizing it, or even before when we are preparing our contribution to the dialogue? If we think about the example of producing a yn question it is actually not only the case that we

expect the user to answer yes or no after we have realized the question but the actual purpose of selecting such a move in the dialogue game was possibly to retrieve a yn answer from the user. In this case, the expectation arises much earlier than at dialogue move production. In practice, people overlap with each other, for example, answering a question before the current speaker has finished. It could also be the case that the listener makes her contribution even earlier, as soon as her expectation has been fulfilled. In telephone-based menu systems where users choose alternatives by push-buttons it is not uncommon that the pushing is performed much before the system contribution has ended, especially when experienced users are involved. In this way they try to shorten the interaction and get through the menus as fast as possible by pushing as soon they know which state they are in and how to get to the next one. If they remember the alternatives and their numbers they do not need to listen to the system contribution. Overlaps or barge-ins are plausible even in spoken dialogue systems and we can imagine situations where the users learn the system strategy just as they do with push-button menus and will try to traverse the dialogue as fast as possible. This means that dialogue move prediction should happen as early as possible to leave the dialogue system with enough time to react to it. As the selected next system move seems to be such an important feature in dialogue move prediction, and as the user often reacts to the current system move, we need to predict as soon as possible after the next system move has been selected and before the system starts to produce it. This means that in the GoDiS system we need to integrate prediction somewhere in between dialogue move selection by the **selection** module and output generation by the **output** module. It could be placed either before the **generate** module or after it as it is not necessary to know how the selected system dialogue move is going to be realized, i.e. the actual wording of the system. To give the system some more time to prepare itself we will trigger the dialogue move prediction module before the **generation** module.

9.1.3 Dialogue move prediction rules

To be able to handle prediction we have created a new GoDiS module: the **predict** module. This module will trigger after the system has chosen what dialogue move to perform next and before calling the **generate** module to decide the wording of the next dialogue move. To obtain this behaviour we have made a change to the GoDiS control algorithm (see Figure 9.2) by adding a call to the **predict** module between the select call and the generate call.

The **predict** module is a dialogue move engine (DME) module using the DME_ADL language just as the **update** and **select** module do (see Chapter 3). It consists of two classes of rules: predict_move and select_model. The prediction algorithm is set to first predict a dialogue move and then select an appropriate language model. The rules for dialogue move prediction or rather for predicting one of the four classes of dialogue moves that our dialogue move predictor handles have been created based on the rules the rule-based learner Ripper learnt from our data in Chapter 6 (see Section 6.3.3.2). These "if-then rules" have been implemented as preconditions and effects according to the GoDiS rule format introduced in Section 3.3.4. We have implemented six distinct rules that set a

```
systurn([ repeat( [

select
test(not is_empty($next_moves))
predict
generate
test(is_empty($active _inputs) and is _empty($input_buffer))
output
....
...
])]).
```

Figure 9.2: The modified control algorithm including the **predict** module

dialogue move value to the PREDDM field in the IS depending on the conditions fulfilled.

The first rule, **predictRequest**, predicts that the user will perform a request. This rule presented in Figure 9.3 is based on several rules obtained with Ripper saying that request moves should be predicted if the system is about to perform an ask(X^action(X)) move or an ask(set(X)) move and the shared action is top. This means that requests are predicted when the dialogue system will realize system questions such as "What do you want to do?" or "Do you want to X, Y or Z?". Apart from this, the rule also captures another rule that specified that if the latest move was a request and the shared action was top (meaning that we are in the top plan) then we should predict another request move.

```
RULE: predictRequest
CLASS: predict
PRE:
      in($next_moves,ask(X^action(X)) or in($next_moves,ask(X^set(X)) or
      in($latest_moves,request(X^P(X)))
      in($/shared/actions,top)
EFF:
      set(/private/preddm/move,request )
      set(/private/preddm/score, 1 )
```

Figure 9.3: The GoDiS predict rule **predictRequest**

In this first implementation of dialogue move prediction we have not merged the ask and request moves but predict them separately. However, this does not stop us later using a common DMSLM for these two classes. The rule for predicting a user question is shown in Figure 9.4 saying that an ask move should be predicted when the system is about to perform an ask(X^action(X)) move and the latest move was an answer move. This rule

learnt by Ripper seems to be a learnt error-pattern due to the problem of perceiving and identifying user questions. It is therefore unclear if this rule is necessary and if we should have a distinction between ask and request moves on the prediction and language model level.

RULE: **predictAsk**
CLASS: predict
PRE:
in($next_moves,ask(X^action(X)))
in($latest_moves,answer(X))
EFF:
set(/private/preddm/move,ask)
set(/private/preddm/score,1)

Figure 9.4: The GoDiS predict rule **predictAsk**

Our classifier in the AgendaTalk experiment did not learn any explicit rules about how to predict answer moves. We have therefore relied on the last experiment with the DJ-GoDiS application where the classifier learnt that answers are expected after questions (see Section 6.4.2.2). However, to adapt this to our system we have, as seen in earlier rules, excluded questions about actions and issues as well as yn questions. The **predictAnswer** rule is shown in Figure 9.5.

RULE: **predictAnswer**
CLASS: predict
PRE:
in($next_moves,ask(Q))
Q = _^_
EFF:
set(/private/preddm/move,answer)
set(/private/preddm/score,1)

Figure 9.5: The GoDiS predict rule **predictAnswer**

To predict yn moves we have implemented two predict rules. The first rule (Figure 9.6) predicts a yn answer whenever next_moves (i.e. the next system move) includes an ICM of positive understanding (implicit grounding) or an ICM of interrogative understanding (explicit grounding). However, it also excludes the case when the ICM has a negative value of a meaning to confirm, i.e. an explicit rejection of a proposition. An example would be the system saying "not meeting". The second rule implements the expectation of a yn answer after a yn question (see Figure 9.7).

RULE: **predictYN1**
CLASS: predict
PRE:
in($next_moves,icm:und*pos:usr*Content1) or in($next_moves,icm:und*int:usr*Content2)
not in($next_moves,icm:und*pos:usr*(not P))
EFF:
set(/private/preddm/move,yn)
set(/private/preddm/score, 1)

Figure 9.6: The GoDiS predict rule **predictYN1**

RULE: **predictYN2**
CLASS: predict
PRE:
in($next_moves,ask(Q))
$Q \neq \_\hat{}\_$
EFF:
set(/private/preddm/move,yn)
set(/private/preddm/score, 1)

Figure 9.7: The GoDiS predict rule **predictYN2**

The sixth predict rule, shown in Figure 9.8, handles any case falling outside the previous predictions. Rather than predicting an answer move as our classifier did we will not specify any move in these cases but predict general, by which we mean that any move is allowed. We have opted for this strategy due to the boosted frequency of answer moves in our data which does not really prove that the best option is to predict an answer move when our specific prediction rules fail.

Although the distribution and flow of dialogue moves may look completely different in a different domain at least the rules learned with Ripper and used for implementing the prediction rules are domain-independent as the information used in the IS is generic for all GoDiS domains. To avoid domain-dependence the implementation is based on results from experiments with both AgendaTalk and DJ-GoDiS, all domain-specific feature information having been excluded. In this way we obtain a **predict** module that can be used by any GoDiS application. The prediction is also independent of language as there is no language-specific information used in the rules. Our dialogue move prediction rules were, as shown in Chapter 6, obtained by training on both English and Swedish data. To work properly, in the case the user switches language during current interaction, DMSLMs in all the languages that are being used need to be provided.

RULE: **predictOther**
CLASS: predict
PRE:
    in($next_moves,Move)
EFF:
    set(/private/preddm/move,general)
    set(/private/preddm/score, 0.5)

Figure 9.8: The GoDiS predict rule **predictOther**

Prediction is not necessarily used only to switch DMSLMs. Having a prediction of the next user move could be used, for example, to re-rank ASR hypotheses, choosing interpretation grammars or re-ranking parsing hypotheses. In addition, it could be used to calculate new confidence scores. Imagine, that the system has predicted that the user may utter a yn answer and the ASR indeeds returns a yn answer but with a low confidence score. In this case the system should assign greater confidence to this hypothesis than the actual ASR confidence score shows. However, if the ASR system in the same state returns a greeting the system could actual lower the ASR confidence for that hypothesis. To hold the possibility open for any use of dialogue move prediction we have therefore separated the prediction of dialogue moves and the selection of language model into two steps. Dialogue move prediction can therefore run separately.

9.1.4 Switching DMSLMs

The switch of ASR language models (both SLMs and grammars) in TrindiKit was developed in the TALK project to be able to switch language or domains. The switching control was therefore already implemented and with minor changes[1] the ASR OAA Java agent was changed to be triggered by our new asrmodel value instead of domain and/or language value.

9.1.4.1 DMSLMs in the information state

Apart from a change in the IS we needed to keep hold of what speech recognition model to load. This was done by adding the MIV asrmodel to the TIS. It holds the name of the language model to use for recognition and is composed of the domain name, the language name and the dialogue move specification as seen in Figure 9.9.

In this way, we will be able not only to switch dialogue move model but we will keep the domain and language switching capabilities. This actually means that if the system, for example, asks a yn question in Swedish and thereby predicts a yn move and sets the asrmodel to "agenda-svenska-yn" (where "svenska" stands for Swedish) and the user

[1]Thanks to David Hjelm.

[ASRMODEL : Domain-Language-Move]

Figure 9.9: The module interface variable: ASRMODEL

requests a switch to English this would mean it would (if the dialogue move prediction of yn has not changed) load the English yn model as the ASRMODEL would have been changed to "agenda-english-yn". As discussed in Chapter 6 we minimized the amount of different DMSLMs due to technical restrictions which means we will only need to predict 4 different moves apart from the general one and only load and switch between the 4 DMSLMs and the general SLM (per language). However, we could imagine predicting more dialogue moves and, for example, creating DMSLMs dynamically or weighting certain grammar rules on the fly.

9.1.4.2 Selecting and switching DMSLMs

The second part of our **predict** module selects the appropriate language model to use based on the predicted dialogue move. For the moment, this selection is merely an update of the ASRMODEL variable with the value in the PREDDM field, i.e. the prediction move. This have been captured in a generic rule shown in Figure 9.10.

RULE: **selectAny**
CLASS: predict
PRE:
    not empty($/private/preddm)
    $language = Language
    domain-Domain =$$underscore2dash($domain)
    $/private/preddm = Move
EFF:
    asrmodel := $$dash2underscore(Domain-Language-Move)

Figure 9.10: The GoDiS predict rule **selectAny**

The rule copies the current domain and language values and creates together with the predicted user dialogue move a new ASRMODEL value. If the user has requested a language switch this will still work as we would then get the new language value from the language holder to build up the ASRMODEL value. The dialogue move prediction and DMSLM switch is therefore language independent.

As soon as the ASRMODEL value has been changed it will trigger the OAA agent controlling the ASR to switch recognition package. This means that when the system starts producing the selected system move the ASR has already loaded the corresponding DMSLM where the probability of the expected user dialogue move is boosted. When the user

starts speaking the recognizer will use this selected SLM for recognition.

The select_model rules can be used not only for SLMs but also for switching SRGs if they follow the same naming principle. If the application does not have different SLMs or SRGs available but just a generic one the select_model rules can be excluded from the predict algorithm and the system will run the model set in the start file during the entire dialogue without producing any switch. However, you would still get the dialogue move prediction which could be logged and used for future evaluation. If prediction is to be used for other procedures than switching ASR models, for example, to switch parsing grammar, we would just need to add a new rule class and rules in the **predict** module (e.g. select_grammar).

9.1.5 Runtime dialogue move prediction and DMSLM switching

The following AgendaTalk dialogue example in (55) shows how the implementation of dialogue move prediction and DMSLM switching work together in practice. GoDiS predicts user dialogue moves and thereafter switches DMSLMs depending on the IS. In this way, the speech recognizer is better prepared for the most expected user dialogue moves without putting any restrictions on the possible moves that the user may perform.

(55)
Sys DM: greet
System predicts: general
Sys> Hi! This is AgendaTalk your talking calendar. Today's date is the eighth of October.
Recognizer has switched to SLM: agenda_english_general
Sys DM: icm:loadplan, ask(X^action(X))
System predicts: request
Sys> Let's see. What do you want to do?
Recognizer has switched to SLM: agenda_english_request
Usr> delete a booking
Usr DM: request(delete_event)
Sys DM: icm:acc*pos,icm:loadplan,ask(X^event_to_store(X))
System predicts: answer
Sys> OK. Let's see. What kind of booking?
Recognizer has switched to SLM: agenda_english_answer
Usr> a meeting
Usr DM: answer(event_to_store(meeting))
Sys DM: icm:acc*pos,icm:und*pos:usr*event_to_store(meeting)
System predicts: yn
Sys> OK. meeting.
Recognizer has switched to SLM: agenda_english_yn
Usr> yes

USR DM: answer(yes)

SYS DM: ask(X^date(X))

System predicts: answer

SYS> What date is the meeting?

Recognizer has switched to SLM: agenda_english_answer

USR> on Thursday

USR DM: answer(date(thursday))

SYS DM: icm:acc*pos,confirm(delete_event)

System predicts: general

SYS> OK. Booking deleted.

Recognizer has switched to SLM: agenda_english_general

SYS DM: icm:reraise:top,ask(X^action(X))

System predicts: request

SYS> So, what do you want to do?

Recognizer has switched to SLM: agenda_english_request

USR> add dinner on Friday

USR DM: request(add_event), answer(event_to_store(dinner)), answer(date(friday))

SYS DM: icm:acc*pos,icm:loadplan,ask(X^start_time_to_store(X))

System predicts: answer

SYS> OK. Let's see. What time is the dinner?

Recognizer has switched to SLM: agenda_english_answer

USR> at ten pm

USR DM: answer(time(1000)),answer(am_or_pm(pm))

SYS DM: icm:acc*pos,ask(take_down_event)

System predicts: yn

SYS> OK. Do you want to take down: dinner on Friday the nineteenth of October at ten pm?

Recognizer has switched to SLM: agenda_english_yn

USR> yes

USR DM: answer(yes)

SYS DM: icm:acc*pos,report(AddEvent,done)

System predicts: general

SYS> OK. Booked.

Recognizer has switched to SLM: agenda_english_general

SYS DM: icm:reraise:top,ask(X^action(X))

System predicts: request

SYS> So, what do you want to do?
Recognizer has switched to SLM: agenda_english_request
USR> bye
USR DM: quit
SYS DM: quit
System predicts: general
SYS> Have a nice day!

The implementation of dialogue move prediction was shown to be quite straightforward and did not cause any processing problems due to the constant switch of SLMs. With this prediction behaviour we have hopefully achieved a more context-aware speech recognition performance. However, to be able to evaluate the performance of our implemented dialogue move prediction and DMSLM switch we would need to carry out tests with real motivated users in a realistic system. This is hard to achieve with the current prototypes of the AGENDATALK and DJ-GODIS systems. It should be noted that dialogue move prediction can easily be used in our system without the DMSLM switching by not calling the selection rule. This opens up for alternative uses of dialogue move prediction. For example, in the second experiment in Chapter 8 our dialogue move prediction rules were used to add information about predicted dialogue moves to logs from interactions with earlier versions of the AGENDATALK system. The dialogue move prediction was used there as a feature for confidence annotation.

9.2 Information state based confidence annotation, N-Best hypothesis selection and dialogue move confidence estimation

Confidence Scores in Speech Recognition give a figure of how confident the ASR system is of a hypothesis (see Section 2.1.4.1 for an introduction to confidence scoring). As discussed in Section 2.4.6, currently available confidence scoring is deficient and this has led researchers to attempt incorporating additional knowledge sources in order to achieve more accurate confidence scoring. This was the purpose of Chapter 8 where we achieved much more reliable confidence annotation by training a machine learner with higher order knowledge. Our machine-learnt confidence model annotated N-Best hypotheses on the utterance level with six confidence classes with very high precision. Section 8.4.5.3 discussed the possible improvement on grounding behaviour when applying this improved confidence model. In this section we will propose an approach for integrating this confidence annotation model into the GODIS dialogue system. Although the original confidence annotation model in Chapter 8 works on the utterance level we will discuss here how the results can be used at the dialogue move level to give confidence to each dialogue move in an utterance. In Chapter 7 we introduced the term *dialogue move confidence score*. In this chapter

we will propose how such scores can be integrated into our confidence annotation model from Chapter 8. Apart from the description of this new confidence annotation model in the GoDiS system we will also define a proposed implementation for selecting N-Best hypothesis (re-ranking) based on this confidence estimate.

9.2.1 Dialogue move confidence scores

It is very common in dialogue systems to make use of confidence scores on the utterance level for grounding utterances (see Section 3.3.5 for examples). However, certain parts of an utterance can be more reliable than others. In human miscommunication we would normally focus on the parts of the utterance that we failed to recognize, understand or are insecure about rather than trying to error handle the whole utterance. Speech recognizers can actually output a confidence score for each word of an utterance. What we will investigate in this section is how these word confidence scores can be used to derive scores on a semantic level. We want to estimate confidence of *dialogue moves* rather than utterances. In this way, by obtaining a confidence score for each dialogue move a dialogue system can apply different grounding strategies for each dialogue move. A dialogue system can for instance focus on verifying the parts that seem to be more likely to have been affected by recognition failure and accept the parts that seem to have been correctly recognized.

In related research, grounding has been handled in a more subtle way by focussing on word confidence scores rather than utterance scores. In this way grounding strategies have been applied by taking into account what words the system is confident about and what words could have been misrecognized (Hazen *et al.*, 2002). A normal strategy is to take into account the confidence the ASR has on content words or words triggering a specific slot in slot-filling semantics. In this way word confidence scores are used on a conceptual or semantic level. Skantze (2005b) uses such a strategy in early error detection to find out if the slot words in a sentence are reliable or not. In other studies confidence scores for semantic (or concept) slots are obtained by combining word confidence scores with additional features from the parser, from the ASR or other sources (Guillevic *et al.*, 2002; San-Segundo *et al.*, 2001b). Gabsdil and Bos (2003) show how word confidence scores could also be integrated into a more fine-grained semantics to obtain semantic content confidence scores. As discussed in Section 7.4, the GoDiS system has been extended to apply such an approach by giving each dialogue move associated with an utterance a confidence score: a *dialogue move confidence score*.

To exemplify the possible advantage of using confidence scores on a non-sentential level the example in 56 shows the output from the ASR and the semantic parser of a real user utterance from interactions with AgendaTalk. Consider the confidence score we get from the speech recognizer for the whole utterance in comparison to the list of scores for each word. As seen, in this example the confidence score for the whole utterance does not correspond well with the different parts as the content word "yoga" has much less confidence than the utterance as a whole. In fact, the ASR seems to rely more on the "what time" part.

(56) ASR HYP> Vilken tid är yogan?
Eng. What time is the yoga?
Utterance confidence score: 56
Word Confidence scores: [69, 60, 24, 43]
DM HYP: [ask(X^time(X)), answer(event(yoga))]

Ideally, the dialogue behaviour would be to act more optimistically when referring to the first part, that the user wanted to know something about the time of an event, but act more cautiously regarding the recognition (and understanding) of the yoga event. In the AGENDATALK system this utterance is interpreted into the two dialogue moves shown in 56. The aim is to distribute the word confidence scores over dialogue moves in order to be able to choose an appropriate grounding strategy for each move instead of using the utterance confidence score for all moves corresponding to the same user utterance. In this particular example we would like to give the "yoga" part corresponding to the `answer(event(yoga))` move a low confidence while the `ask(X^time(X))` move would be given a much higher confidence. By using confidence on the dialogue move level we would be able to go from the dialogue behaviour in 57 (where all dialogue moves are treated in the same way based on the reliability of the utterance) to the behaviour in 58 (where grounding strategies is applied based on the reliability of each dialogue move).

(57) USR> What time is the yoga?
SYS> You want to know about the time of an event
USR> yes
SYS> yoga?
USR> yes
SYS> it's at ten thirty

(58) USR> What time is the yoga?
SYS> yoga?
USR> yes
SYS> it's at ten thirty

By using the confidence of each dialogue move for grounding we can therefore get a more efficient behaviour with shortened dialogues but also a more precise grounding behaviour where only the critical parts are clarified.

To estimate confidence scores per dialogue move the ASR word confidence scores were integrated into the GODIS interpretation process. This was in order to distribute the word confidence scores over dialogue moves. This was achieved by changing the parsing algorithm to take account of word confidence scores. The word confidence scores for all words used in a rule to obtain a dialogue move were used to calculate the average score for that move[2]. This resulted in a new GODIS interpretation module which for each dialogue

[2]This implementation was carried out in the TALK project by David Hjelm.

move also outputted a dialogue move confidence score. The IS and several MIVs were also adapted to be able to pass more complex structures of dialogue moves. These new structures included word confidence scores, dialogue move scores but also the modality used to produce the dialogue move. As described in Section 3.4.1.5 and shown in Figure 13 (on page 80) dialogue moves were therefore represented as records in this new implementation.

The use of dialogue move confidence scores give much improved behaviour over using the confidence score based on the whole utterance for each dialogue move. A very common and critical misrecognition in the AgendaTalk system is for the ASR system to favour the recognition of the word "avsluta" (*Eng. quit*). This often leads to the system shutting down the whole application as was seen in the dialogue example in 24 (on page 134). Interestingly, although the ASR is often quite confident about the whole utterance the reliability of this particular word is often low. In 59 we can see how the ASR hypothesis does not match the real user utterance which leads to a partially wrong dialogue move sequence interpretation. The example shows how the estimation of dialogue move confidence scores would lead to a much lower score for the quit move than for the whole utterance. By using these dialogue move confidence scores with appropriate thresholds as a basis for grounding strategies we would be able to avoid a situation where the dialogue system accepts both moves as correctly recognized. Instead it would be able to reject the quit move while at the same time accepting the date answer move.

(59) USR> Vad gör jag på torsdag?
Eng. What am I doing on Thursday?
ASR HYP> Avsluta på torsdag?
Eng. Quit on Thursday?
DM HYP: [quit, answer(date(thursday))]
Utterance confidence score: 62
Word Confidence scores: [33, 80, 83]
Dialogue move confidence scores: [33.0, 81.5]

The implementation of dialogue move confidence scores in the TALK project was primarily motivated by the introduction of multimodality. The aim was to be able to ground graphical input more optimistically in multimodal systems as the interpretation of graphical input is more certain than speech. With this approach the dialogue moves in an utterance could be grounded separately depending on their modality source. However, the grounding strategies for spoken dialogue moves developed in the TALK project are still only dependent on ASR confidence scores (as described in Section 3.3.5) and do not take into account any other information. In the next sections we propose to change this.

9.2.2 Information state based dialogue move confidence scores

Encouraged by the results in Chapter 8 we want to eliminate the heavy dependence on the sometimes unreliable ASR confidence scores in GoDiS when selecting grounding strategies and therefore introduce confidence modelling that takes into account more knowledge

Table 9.1: Confidence class scores

Class	Score
OPT	95
POS	75
PESS	55
NEG	25
IGN	0
CROSS	0

sources. Our confidence classification from Chapter 8 works on the hypothesis level and does not classify each individual dialogue move. To be compatible with grounding on the dialogue move level, as explained in the previous section, we need to estimate confidence on the same level. We will do this by taking into account the hypothesis confidence class and the ASR-based dialogue move confidence scores (henceforth *ASR DM score*). However, the confidence class is not numeric. The question is how we could interpolate our classes with the numeric ASR DM scores. Firstly, we need to translate the Confidence Classes into scores. This could be set dynamically and left to the developer to optimize. Table 9.1 shows a proposal for translating confidence labels into values. It should be noted that the values presented are set arbitrarily and just to illustrate the approach.
With the use of these scores for each confidence class we can thereafter estimate a joint confidence score for each dialogue move by also taking into account the ASR DM score (estimated as explained in Section 9.2.1). Equation 9.1 shows how such a *information state based dialogue move score* (*IS DM Score*) can be estimated by linear combination.

$$\text{IS DM Score} = \boldsymbol{\lambda} * \text{Confidence Class Score} + \mathbf{(1 - \lambda)} * \text{ASR DM Score} \tag{9.1}$$

λ is here a weight that would be set by the developer after optimization. This weight makes it possible for the developer to decide how much weight she wants to put on either confidence method. If the developer has strong confidence in the recognizer she could for instance give more weight to the ASR DM Score. If the confidence class annotator is working well more weight could be given to the Confidence Class Score. Again, these weights will need to be optimized for best performance of the joint model.

Suppose we set this weight to 0.5 (giving equally importance to both confidence approaches) and that the confidence classes are set to the arbitrary scores in Table 9.1. For the same dialogue moves with the same ASR DM Scores we would then obtain quite different IS DM Scores depending on how our knowledge-based confidence annotator labels the whole hypothesis. This means that if the dialogue system has for example interpreted a user turn as consisting of a request and an answer move and then calculated ASR DM scores for each of these dialogue moves, as exemplified in the first two columns in Table 9.2, then the resulting IS DM Score (see column four) would vary significantly depending on the confidence class label given to the whole hypothesis (see column three). Table 9.2 illustrates this by showing the final IS DM scores for these both moves of the example

when marked with the labels OPT, PESS and IGN respectively and applying Equation 9.1 with the scoring from Table 9.1.

Table 9.2: Examples of estimation of information state based dialogue move score (IS DM score)

DM	ASR DM Score	Confidence Class	IS DM Score
Request	75	Opt	85
Answer	25	Opt	60
Request	75	Pess	65
Answer	25	Pess	40
Request	75	Ign	37.5
Answer	25	Ign	12.5

In a dialogue system we can use these IS DM scores against a threshold to decide whether to accept or reject a dialogue move. Suppose this threshold was set to 50. For the examples in Table 9.2 a dialogue system would then accept both the request and answer move if the confidence class was set to OPT for the ASR hypothesis as the IS DM Scores for both dialogue moves are above the proposed threshold (see the first row). With the confidence class set to PESS (as in the second row) the dialogue system would accept the request move but reject the answer move. If the confidence annotator classifies the hypothesis as IGN the result would be to reject both dialogue moves as the IS DM Scores would be below the threshold (see row three). In this way the utterance based confidence labelling from Chapter 8 in combination with dialogue move scores results in a new confidence measure that works on the dialogue move level.

In the GoDiS system we can go one step further and use IS DM Scores as a basis for selecting more fine-grained grounding strategies than the more simplistic Accept/Reject decision. Instead of the ASR DM scores, that are currently in use, we would use this new confidence measure. We could actually keep the current implementation of grounding in GoDiS in its current form and use this new estimation of the scores to condition the grounding behaviour. By using IS DM Scores we would take the confidence annotation from Chapter 8 into account and thereby base the grounding on much more knowledge (including dialogue context) than only the information from the ASR system. The results in Chapter 8 indicate that such an approach would lead to a much more accurate grounding behaviour. Still, the selection of grounding strategies would be dependent on thresholds that the developer would need to optimize. As previously explained in Section 3.3.5 GoDiS currently applies the grounding strategies and feedback moves shown in Table 3.1 which are based on three thresholds that are set manually by the developer. This includes the rejection threshold that the ASR system uses.

In Section 8.4.3 we showed how confidence thresholds can be optimized on training data on a conceptual level for better confidence classification. Although optimization was made on the hypothesis level the same approach can easily be applied for dialogue moves. In Chapter 8 we actually diverged from the approach in GoDiS and used four

thresholds (instead of three) in association to our five confidence classes. In this way we made a more fine-grained distinction of misrecognized utterances (into NEG and IGN) and in some experiments we even successfully recognized crosstalk. The usefulness of a more fine-grained classification of badly recognized user utterances needs to be evaluated. The value of being able to identify crosstalk was first discussed in Section 2.4.1.2 and then exemplified in Section 8.4.5.2. As discussed in Section 8.4.6 GoDiS actually has feedback moves and a grounding strategy for cases where the communication channel is not working properly. The same strategy could be used when crosstalk is identified. The identification of crosstalk goes against the idea that grounding should be carried out only on a dialogue move level. A well-founded grounding framework will probably need to work on both levels and include in its decision-making whether user input should be grounded on the dialogue move level or on the turn level. For severe miscommunication, channel problems or crosstalk grounding will probably need to be held on the turn level whereas for turns that the system is more confident about grounding would be applied on a dialogue move level. We will consider the latter case in this section.

The actual application of grounding strategies is out of the scope of this thesis and the aim of the confidence scoring proposed here is to be independent of the current grounding strategies in GoDiS. It can be used with any number of thresholds and to perform any type of grounding strategies. It could also be used for approaches which do not use thresholds but estimates costs as Skantze (2007) proposes. What was shown in Chapter 8 is that as many as six notions of confidence is actually possible for a classifier to distinguish using our approach. What we are showing here is how these levels in conjunction with ASR DM scores can improve the estimation of confidence for each dialogue move. For this purpose we will illustrate an approach with four confidence thresholds as in Table 9.3. It should be mentioned that we are here only discussing grounding from a perceptual point of view. However, grounding strategies in GoDiS are selected taking into account other levels such as semantic interpretation or pragmatic plausibility as discussed and exemplified in Section 3.3.5. For the ultimate decision of which grounding strategy and feedback moves to apply even more information would need to be taken into account. Skantze (2007) defined a grounding decision as based on three factors: the system's confidence in its understanding, the consequence or cost of falsely accepting or rejecting the hypothesis, the cost of performing the grounding move taking into account possible reactions to it. Here we are only considering the first of these factors: the impact of our estimation of confidence in system understanding on the grounding decision. The confidence annotation and the grounding strategies below should therefore be considered as input regarding the first factor for the grounding model. We will leave the discussion about other factors to consider for grounding decision for Chapter 10.

With the use of four thresholds we can distinguish five perceptual grounding strategies as shown in Table 9.3 (see also Section 3.3.5). *Optimistic acceptance* refers to a strategy where a dialogue system does not mention what has been perceived and understood but only reflects that something has been perceived, understood and accepted (for example by saying "OK."). *Implicit verification* and *explicit verification* are well-known strategies (see Section 3.3.5) where the perceived (and understood) dialogue moves are referred

Table 9.3: Confidence thresholds for perceptual grounding strategies

Confidence score	Grounding strategy
Score > T1	optimistic acceptance
T2 < Score > T1	implicit verification
T3 < Score < T2	explicit verification
T4 < Score < T3	explicit rejection
Score < T4	implicit rejection

to either explicitly in order for the user to confirm them or implicitly to give the user an opportunity to verify them. Following the same notions we have named the last two grounding strategies *explicit rejection* and *implicit rejection*. With explicit rejection the system will somehow inform the user about what has been perceived and signal its doubtfulness about the perception. On the other hand with implicit rejection nothing of what has been (mis)recognized and (mis)understood will be revealed.

It should be mentioned that as the developer can set the confidence class scores, the thresholds and the weight of the confidence models, this approach is independent of speech recognizer and can be adjusted to how well the ASR DM confidence scores are estimated or how well our confidence classification works. To illustrate the advantage of the use of the IS DM scores we will consider one of the hypotheses that was falsely accepted in our confidence annotation experiment in Chapter 8 (from Table 8.13) and show how a dialogue system would react by applying the different scoring approaches discussed. We will apply the optimized thresholds from Section 8.4.3 (shown in Table 9.4) with the grounding strategies presented in Table 9.3. Evidently, these are not the most optimized thresholds for the current task as we are now using them on the dialogue move level. However, in the absence of alternative thresholds they will be used here to illustrate the approach.

Table 9.4: Optimized thresholds for grounding

Threshold	Optimized Score
T1	70
T2	67
T3	53
T4	35

We will use the confidence class scores from Table 9.1 and the weight 0.5. The information related to the example hypothesis is shown in 60.

(60) USR> Har jag något bokat klockan sjutton noll noll på?
Eng. Do I have something booked at seventeen zero zero on?
USR DM: [ask(X^bookings(X)), answer(time(1700))]
ASR HYP> har jag något inbokat klockan sjutton och noll igår

Eng. Do I have something booked at seventeen and zero yesterday?
DM HYP: [ask(X^bookings(X)), answer(time(1700)), answer(date(yesterday))]
Utterance confidence: 55
Confidence class: POS

This partially misrecognized hypothesis was classified too optimistically with the confidence class POS by our confidence annotator. Although most of the user utterance has been correctly recognized the ASR has wrongly inserted the word "yesterday" in the end of the hypothesis which wrongly results in an additional dialogue move. Grounding strategies based on the confidence class or the utterance confidence score would ground all dialogue moves equally. For this particular example by using the utterance confidence score (in 60) against the thresholds in Table 9.4 GODIS would ground the correctly recognized parts too pessimistically. The system would perform three explicit confirmations. On the other hand, using the confidence class would result in all the moves being implicitly verified. This would be a more appropriate behaviour regarding the first two dialogue moves. However, this would imply that the misunderstood dialogue move answer(date(yesterday)) would be grounded too optimistically and actually be falsely accepted.

In previous sections we have presented two confidence scores related to the dialogue move level: ASR DM Score and IS DM Score. In (61) we see the Word Confidence Scores, the estimated ASR DM Score and the IS DM Score for each move of the example hypothesis in discussion. Based on the ASR DM Score in (61) and the thresholds in 9.4 the grounding behaviour would be much more accurate as the system would be able to explicitly verify the ask(X^bookings(X)) and answer(time(1700)) moves whereas it would correctly reject the answer(date(yesterday)) move implicitly.

(61) ASR HYP> har jag något inbokat klockan sjutton och noll igår
Word Confidence Scores: [66, 40, 77, 37, 51, 75, 47, 69, 25]
DM HYP: [ask(X^bookings(X)), answer(time(1700)), answer(date(yesterday))]
ASR DM Score: [61.0, 60.0, 25.0]
IS DM Score [68, 67.5, 50]

With IS DM scores we would obtain an even more appropriate behaviour by implicitly verifying the correctly understood dialogue moves while rejecting the incorrectly recognized date. This would result in a much smoother behaviour with less confirmations and more accurate confidence scoring. This example also shows that wrongly re-ranked and misclassified hypotheses from our experiments in Chapter 8 may with this approach not necessarily result in any negative impact on the dialogue in the end.

9.2.3 Confidence annotation and re-ranking model

There are several alternative ways of implementing and integrating a confidence annotation and re-ranking model based on the experiments in Chapter 8 into GODIS. In this section we will discuss what changes would be needed to the GODIS system in order to integrate

such a model. A proposal of how such a model could be implemented will also be presented. For a description of the GoDiS system see Chapter 3.

The confidence annotation would need access to information from the ASR system (the **input** module), the **interpret** module as well as from the IS. One approach would therefore be to intertwine the confidence annotation and re-ranking process with the interpretation and input processes. In this way we could control the interpretation process from the confidence annotation. This could lead to a more efficient implementation. A more modular approach is to have the confidence annotation and re-ranking as a separate module that is triggered after interpretation has been executed. As the objective here is an experimental system we will opt for the latter option. In this way it will be easier to switch confidence annotation and re-ranking on and off as well as interchange these processes for other ones. Regarding efficiency, as we are only working with shorter N-Best lists this should not cause any apparent troubles. As was discussed in Section 2.4.5 the most gain is to be found on the upper part of N-Best lists.

The confidence annotation module, **confidence**, would be called after the **interpret** module but before the **update** module as formalized in the modified control algorithm in Figure 9.11.

condition(alarm(asr)) => [apply_rule(moveInputQueueoInput,

[$input_buffer=Q],
[clear(input_buffer), input:=Q]),
print_state,
interpret,
confidence,
update,
print_state | SysTurn
]

Figure 9.11: The modified control algorithm including confidence annotation

This structure is somewhat unusual in dialogue systems. The normal approach is to have the ASR system decoupled from the rest of the dialogue system and confidence annotation and re-ranking being performed directly after recognition. However, to be able to make use of information from other parts of the dialogue system such as the output of the semantic parsing or the IS, we need to postpone the decision-making of confidence and N-Best selection until all necessary information can be obtained. Therefore we need this more tightly coupled approach where the final output of what hypothesis to use and what confidence to be given to it is taken at a much later stage.

The ASR OAA wrapper in use already supports handling N-Best lists. GoDiS on the other hand currently does not work with N-best lists. We would therefore need to adapt some of the modules, MIVs and data structures in order to handle N-Best lists. Even if we

would not want to apply the re-ranking approach and only confidence annotate topmost hypotheses we still need to handle N-Best lists as many of the most important features for confidence annotation turned out to be list-dependent. To be able to work on not only the top-ranked hypotheses the **input** module must be rewritten in order to pass on a more complex structure that holds a list of hypotheses with respective information from the ASR. The INPUT_BUFFER queue in GoDiS would therefore need to hold not only records of words and word confidence scores but we would need to group the words and scores of each hypothesis to create something as a queue of records of queue of records. The **interpretation** module algorithm would also need a slight change in order to parse and assign dialogue move confidence scores for each hypothesis in an N-Best list (extracted from the new INPUT_BUFFER structure). However, the interpretation process *per se* would not need any change.

In the experiments in Chapter 8 we made use of both the more robust key-phrase spotter written in Prolog and the more restrictive GF grammar. The GF grammar was used to obtain a grammaticality measure for a hypothesis which was shown to be a very informative feature. Following the strategy in the experiments we would need to control two different parsing processes which is not an optimal architecture. For this hypothetical implementation we assume we have a semantic parser that can do both, i.e. parse robustly and give us some kind of grammaticality measure. The N-Best interpretation module would then output all interpreted hypotheses with dialogue move scores for all dialogue moves (estimated as described in Section 9.2.1). It would also assign a grammaticality label to each of them. We would like to at least distinguish between fully parsed hypotheses (grammatical ones), robustly parsed hypotheses and unparsed hypotheses. The **interpretation** module would write its output to a new MIV: the LATEST_NBEST_MOVES. The processing steps to support confidence annotation and re-ranking would then be the following:

(62)

1 Input module passes on an N-Best list with all necessary ASR features for each hypothesis to INPUT_BUFFER on TIS.

2 Interpret module parses each hypothesis in the N-Best list into dialogue move sequences and calculates a dialogue move score for each dialogue move.[3]

3 Interpret module assigns a grammaticality label to each hypothesis. Output is written to LATEST_NBEST_MOVES.

4 Confidence module extracts and derives features to represent each hypothesis from LATEST_NBEST_MOVES as a feature vector.

5 Confidence module confidence classifies each of these feature vectors.

6 Confidence module selects the best hypothesis from the N-Best list based on the given confidence classes (re-ranking).

7 Confidence module updates the MIV CONF_CLASS in the IS to the confidence class of the selected hypothesis.

[3]A more efficient strategy would be to convert the N-Best list into a more compact representation, such as a word graph, and use a parsing algorithm able to parse graphs in one single pass instead of parsing individual sentences.

8 Confidence module estimates IS DM confidence scores for the dialogue moves of the selected hypothesis.

9 Confidence module updates LATEST_MOVES with the interpretation of the selected hypothesis and corresponding IS DM confidence scores.

10 Update module integrates selected dialogue moves taking into account the IS DM Confidence Scores and the Confidence Class.

An added MIV would be the CONF_CLASS which would work similarily to the current Utterance Confidence Score (SCORE) currently in use in GoDiS. The LATEST_MOVES variable would not need any modifications. The difference for this variable will not lie in the format or structure but in that the confidence scores held for each move will be IS-based instead of ASR-based.

The grounding model in GoDiS is part of the **update** and **select** modules in the DME. The DME would need to use the CONF_CLASS value to decide whether to ground on the turn level or the dialogue move level. If the value for example was CROSS a strategy on the turn level might be the best option as pointed out in Section 8.4.6. If the confidence class is IGN the system might need to decide whether it is worth grounding any of the dialogue moves (if only some of them have high scores) or if it would be more efficient to ask the user for a repetition or reformulation of the whole utterance. If the DME finally decides to ground on the dialogue move level then the IS DM Confidence scores will be considered for choosing feedback moves for each dialogue move. As mentioned earlier the ultimate grounding decision will need to take into account many other factors than the Confidence Class and the IS DM Confidence scores. However, the current grounding behaviour in GoDiS may well apply.

The approach presented below is for simplicity unimodal. For a multimodal setting we would need to take into account input from other sources such as from a GUI. In Chapter 10 we will discuss some possibilities regarding such an approach.

The following sections will further describe the implications of implementing the six processes in the **Confidence** module above. The module will take as input the information in the MIVs INPUT_BUFFER and LATEST_NBEST_MOVES. It will also have read access to the IS. It will write its output to the MIVs LATEST_MOVES and CONF_CLASS. In the second experiment in Chapter 8 we found that of our proposed 45 features only 39 were useful. These will need to be extracted or derived at run-time. One possibility is to distribute this feature extraction and derivation over different phases. In this way we would avoid having all the processing just after ASR. This might imply extending the current IS (or TIS) to keep track of more information during run-time. For the experiments in Chapter 8 we actually foresaw the dialogue history as a possible important feature for the experiments. We therefore extended the IS before the interactions with the AGENDATALK system were carried out (see 16 on page 84). Other features or additional derivation of features may be needed in the different stages beyond the confidence annotator.

9.2.3.1 Feature extraction

Our confidence classifier would need to have N-Best hypotheses represented as feature vectors. Many of the features needed for confidence classification are easily obtained from the ASR output or directly from the IS. Others need to be derived by the use of different sources. Some of the most informative features used in Chapter 8 were based on the interpretation of the hypotheses into dialogue moves. A necessary step is therefore to first interpret the hypotheses into dialogue move sequences. Conjointly we need to obtain dialogue move scores for each dialogue move following the approach in Section 9.2.1. As the interpretation and dialogue move score estimation will already have been performed when the confidence annotator goes into action due to the architecture we propose that the confidence annotator will only need to retrieve this information from LATEST_NBEST_MOVES.

To be able to extract the necessary features from the interaction logs for the experiments in Chapter 8 a feature extractor was implemented in Prolog. A great deal of this code is perfectly reusable for the implementation of a feature extracting phase for the confidence annotator. The code might need to be optimized to work efficiently in real-time at runtime. As discussed earlier, it might also be fruitful to divide the feature extraction over different phases and keep some of the needed features in the TIS. In this way we would have less processing in the actual recognition phase as some features would have already been estimated. Many features will also already have been estimated as they are used in the dialogue system for other processes. An example is the new dialogue move prediction presented earlier in this chapter. The feature extraction in the confidence annotator will be more straightforward than the feature extraction from the logs due to the difference in format but also when it comes to timing. It was cumbersome in the preparation of the experiments in Chapter 8 to determine which of the logged information states the values should be retrieved from as the IS had been set to be logged frequently. When retrieving directly from the IS this would not be a problem as we can decide at which moment to extract a feature.

Many of the features would have to be transformed into a more representative format or a format acceptable by the classifier. For example the GoDiS format for dialogue moves is not acceptable by TiMBL. In addition, the way we used dialogue moves in the experiments was either on the dialogue move type level or as dialogue moves without any values (see Section 3.3.2 regarding terminology). This means that a dialogue move such as answer(time(1000)) was represented as answer(time). This was done in order to reduce the features values so that the classifier could generalize more easily over them. This feature value reduction was implemented as part of the Prolog feature extractor referred to above and the code could easily be reused. For the experiments we used 45 features. For the implementation it is only interesting to extract the features that were shown to have discriminatory power. This leaves us with 39 features to extract or derive from different sources. We will use the same abbreviations for the features as in Section 8.4.4. It should be mentioned that future machine learning experiments based on much more training data might well reveal a slightly different feature selection than the one discussed here.

Eight of these features are actually accessible or can be derived even before ASR pro-

cessing. For example many of the features related to the dialogue history can be extracted before ASR. It was shown that the dialogue history or even parts of it were not informative for the classifier. However several features derived from the dialogue history were of interest. These must be extracted. The mean utterance confidence score (DiaHisConf) could either be extracted each time from the dialogue history, which keeps track of the confidence scores for all turns and dialogue moves, or we could add this feature as a variable to the TIS which is constantly updated. A problem is that if we change the type of scores we keep track of (IS-based instead of ASR-based) we would not have this information. This means we need to maintain the utterance confidence scores in the IS as well. The best thing would then probably be to use the current SCORE variable for this and maintain the mean ASR-based utterance confidence score in that holder and update this after each turn.

We would also need to change the dialogue history structure if we also want to keep track of the new confidence classes. The dialogue history stackset presented in (16) would be modified as in (63) with an additional field for the confidence class of the turn. The field score would stay intact but now hold the IS DM score instead of the ASR DM score for each move.

(63)
```
stackset( record( [ speaker : participant,
                    modality : set(modality),
                    conf_class : conf,
                    turn_cont : set( record( [ move : dmove,
                                               score : real ] ) ) ] ) )
```

The SysOnTrack, UsrOnTrack and DiaLen features are also extractable from the dialogue history. The code for the derivation of these features already exists and is perfectly reusable. The same occurs with the selected system dialogue move (SysDM), the last user dialogue move (LASTUSRDM) and the current shared action (ACT).

Twelve of the features are easily extracted and derived after recognition from the ASR information held in INPUT_BUFFER. The ASR would need to pass on more information than today such as the Probability scores in addition to the Confidence scores. Some of these features would then be possible to extract directly (such as HYPRank, HYPConf, HYPProb) whereas others need to be derived (HypMinWordConf, HypWordLen). A great deal of them are only possible to derive by taking into account the whole N-Best list (e.g. WordPurity, HypWordUniqueness, HypConfDropFromTop). The code already used for the experiments would be perfectly applicable with slight changes.

Eleven of the features can be extracted and derived after the semantic parsing based on the information in the new structure LATEST_NBEST_MOVES. HypGram and HypDM could be retrieved directly. HypDMLen, HypMinDMScore and HypDMScoreStDev would be easily derived reusing code from the experiments. To derive the features ListDMSeqVariety, ListDMVariety, FreqMatch, InclMajDM, DMPurity and HypDMUniqueness we need to process all the parsed hypotheses from LATEST_NBEST_MOVES. The derivation of these features has already been coded for the experiments and most of this code is reusable.

Finally, there are some features that can only be derived taking into account both the information in the IS as well as the parsed hypotheses in LATEST_NBEST_MOVES. To obtain the DMPredMatch feature we need the predicted dialogue move which is easily obtained from the IS as a result of the implementation presented earlier in this chapter. In the second experiment in Chapter 8 we had to perform dialogue move prediction afterwards using the implementation presented in this chapter. The DMPredMatch is easily derived if access to the IS field PREDDM is given. For the QAMatch feature we actually reused code from GoDiS in the experiments. The features RelQUD, RelISS, SolvePlan, PropRepeat, RepDM, PropActive are all derived by comparing the dialogue move interpretation of a hypothesis against QUD, Shared Issues, Plan, Dialogue History or Shared Commitments in the IS. Again, it would be possible to reuse part of the already implemented Prolog code from the experiments.

As seen, in order to represent each hypothesis in an N-Best list as a feature vector we would need to extract and derive a considerable number of features based on different information sources. It would be possible to distribute the feature extraction over different stages. Whether this is needed and fruitful would need to be investigated. As these features were already extracted and derived automatically from logs by implementing a program in Prolog for the experiments in Chapter 8 it would be pretty straightforward to re-implement this in GoDiS. The code may need some slight changes and perhaps needs to be optimized to work more efficiently. However, the work effort would be minimal.

9.2.3.2 Confidence classification model

The classifier could either be implemented by the use of rules learnt from the rule-based learner or by using the memory-based learnt classifier as an external server.

The rules in (54) gave a sample of how the rules of a rule-based confidence classifier could look. If we could use such rules the implementation in Prolog for GoDiS would be quite straightforward. In a similar manner to the dialogue move prediction implementation we could model these 102 if-then rules as GoDiS preconditions and effects. Also we would only need 25 features which would simplify the feature extraction phase.

Another option is to build the TiMBL classifier. In the experiments we used the TiMBL software from the command line. However, it is possible to use the TiMBL API to build a runtime classifier. Such a classifier would then lie as an external server waiting for input on a port. It would load the training patterns and run the the learning procedure at launch. The classifier would be called using the `classify` command sending a feature representation of a hypothesis. It would then classify the hypothesis and return its confidence class. To be able to communicate with such a TiMBL classifier we would need to build an interface that handles the communication. To easily integrate this into GoDiS and TrindiKit one approach would be to build a OAA wrapper in Java around the client as the TiMBL API uses C.

It should be mentioned when discussing machine learning that the implementation done for automatic labelling according to our confidence classes can easily be reused. It only needs logged interactions and transcriptions of the corresponding audio files. In this way

it is quite easy to create more training data. No manual annotation is needed.

9.2.3.3 N-Best hypothesis selection

The re-ranking or hypothesis selection could be intertwined with the confidence annotation. When we confidence classify we would start from the top of the N-Best list and send the highest ranked feature representation to the classifier. Whenever a hypothesis is classified as OPT we would stop the classification and select this hypothesis. It is only if no hypothesis is labelled as OPT that we would need to label the whole list and thereafter select the hypothesis with the highest confidence class. In this way we can save processing time and achieve a rapid classification and re-ranking when the speech recognizer is doing well and our confidence classifier is very confident on the ASR output. More processing with re-ranking and classification of entire 10-best lists would only be needed when the ASR is performing badly and our confidence classifier is doubtful about the ASR results. Only if none of the hypotheses has been classified as OPT would we have a whole N-Best list with confidence classes that we would need to re-rank. We would then proceed to select the hypothesis with the best confidence class using the re-ranking algorithm. The re-ranking algorithm is very simple. It would go through the list keeping track of the highest confidence class and the rank of that hypothesis. As we know that there is no hypothesis classified as OPT we could force the algorithm to stop as soon as it encounters a POS label. The algorithm would always find the most highly ranked hypothesis with the highest confidence label. The preference ordering of confidence labels would be: POS>PESS>NEG>IGN. A similar algorithm was already implemented in Prolog for the experiments to re-rank the experimental N-Best lists.

For the re-ranking experiments we wanted to bias the classifier towards perceptually correct hypotheses in order to minimize WER. However, in spoken dialogue systems it is irrelevant whether the system selects a hypothesis with less word errors as long as the semantic interpretation is correct. It is therefore not necessary to give the dialogue system the task of choosing between hypotheses with the same dialogue move interpretation. In Chapter 8 we therefore also estimated ranking accuracy without distinguishing the OPT and PESS class. Semantically equal hypotheses of the N-Best lists should therefore be collapsed. This will result in much shorter lists as the variation in dialogue move sequences is much lower than the variation in word sequences. Collapsed hypotheses will be represented with the highest confidence class of the affected hypotheses. It could also be possible to take into account all the confidence labels for all the hypotheses affected and estimate an average or boost the confidence class. It is clear that the confidence classification and re-ranking will definitely not put any important time constraints on processing. The most critical part is rather the feature extraction. The N-Best selection will set the variable CONF_CLASS to the confidence class of the selected hypothesis.

9.2.3.4 IS-based dialogue move confidence estimation

The IS DM confidence scores would be estimated as described in Section 9.2.2. This process would take as input the selected dialogue move sequence, its corresponding dialogue move scores and the confidence label set by the confidence annotator. It would make use of confidence class scores (similar to those in Table 9.1) in order to translate the confidence label into a numerical score. These values are to be set by the developer in the configuration file and are then loaded by the TIS represented as in Figure 9.12. The developer would also need to set the weight to be used in Equation 9.1 on page 235.

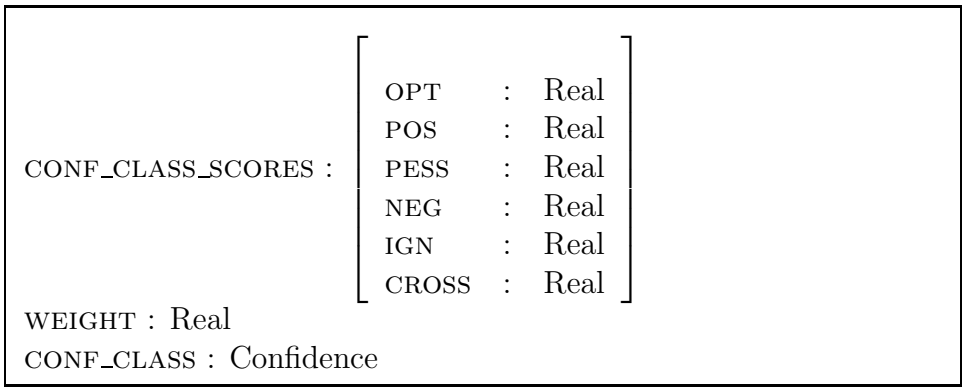

Figure 9.12: Module interface variables (MIVs) in GoDiS

An IS DM confidence score would be estimated for each dialogue move following Equation 9.1. This would be the final step of the **Confidence** module. The selected dialogue move sequence and the IS DM confidence scores would be written to LATEST_MOVES. This together with the value given to CONF_CLASS would be the output of the whole process. Also, we would need to update the mean utterance confidence score using the utterance score for the selected hypothesis as discussed earlier. The DME would then use the information in LATEST_MOVES to decide what grounding strategy to apply.

9.2.3.5 Putting it all together

To illustrate the confidence annotation approach we now go stepwise through the different stages with the N-Best list in Table 9.5. The N-Best list was produced by the ASR based on a Swedish utterance to AgendaTalk where the user said "lägga till" (*Eng. add*). The correct hypothesis can originally be found at rank 3. The **input** module would extract from the ASR system a ranked list of hypotheses with ASR confidence scores as represented in Table 9.5. This would be represented as a record and written to the MIV INPUT_BUFFER. The **interpret** module would then take this list together with the word confidence scores in order to return a parsed list with ASR dialogue move confidence scores. The result from the interpretation, presented in Table 9.6, would be written to the MIV LATEST_NBEST_MOVES.

The **confidence** module would make use of both the information in INPUT_BUFFER and LATEST_NBEST_MOVES as well as the current IS in order to derive a feature vector for

Table 9.5: N-Best list example with ASR confidence scores

Rank	Hypothesis	Word Confidence Scores	Confidence Score
1	mötet noll	42 17	31
2	mötet elva	42 22	31
3	**lägga till**	34 50	31
4	mötet tio	42 15	30
5	basket noll	31 34	30
6	mötet tolv	40 2	14
7	mötet imorrn	42 12	14
8	till mötet noll	5 42 35	14
9	mötet nio	42 11	14
10	match ett noll	32 34 19	14

Table 9.6: Parsed N-Best list with DM confidence scores

Rank	Dialogue Move Interpretation	ASR DM Scores
1	answer(event(meeting)) answer(number(0))	42 17
2	answer(event(meeting)) answer(time(1100))	42 22
3	request(add_event)	42
4	answer(event(meeting)) answer(time(1000))	42 15
5	answer(number(0))	34
6	answer(event(meeting)) answer(time(1200))	40 2
7	answer(event(meeting)) answer(date(tomorrow))	42 12
8	answer(event(meeting)) answer(number(0))	42 35
9	answer(event(meeting)) answer(time(900))	42 11
10	answer(event(match)) answer(time(100))	32 26

each hypothesis of the list. For illustration and comparison, in Table 9.7 we show the values from the 39-dimensional feature vectors derived from the experiments in Section 8.4.5.1 for the hypotheses on rank one and rank three for the current example. For a description of the features see Section 8.4.4.

Table 9.7: Feature vector values for hypotheses ranked as 1 and 3

Feature	Values for Rank 1	Values for Rank 3
HypRank	1	3
HypWordLen	2	3
HypConf	31	31
HypMinWordConf	17	34
HypProb	-80	-84
HypGram	gram	gram
HypDM	answer(event)-answer(number)	request(add_event)
HypDMLen	2	1
HypMinDMScore	17	42
HypDMScoreStDev	12	0
MeanWordLen	2	2
ListDMSeqVariety	0.9	0.9
ListDMVariety	0.6	0.6
HypConfDropFromTop	0	0
HypProbDropFromMean	-421	-25
HypConfUpFromLast	17	17
HypWLenComp	shorter	shorter
FreqMatch	freq	unfreq
InclMajDM	majdm	nomajdm
WordPurity	6	2
DMPurity	5	1
HypWordUniqueness	0	0.5
HypDMUniqueness	0	1
SysDM	askalt	askalt
DMPredMathc	unpredicted	predicted
QAMatch	noqamatch	noqamatch
LastUsrDM	[]	[]
RelQud	irrelevant	irrelevant
Act	top	top
RelISS	irrelevant	irrelevant
SolvePlan	unsolves	solves
DiaLen	3	3
SysOnTrack	0	0
UsrOnTrack	0	0
DiaHisConf	0	0
DiaHisConfStdDev	0	0
PropRepeat	newprop	noprop
RepDM	notrepeat	notrepeat
PropActive	newprop	noprop

The **Confidence** module would call the external confidence classifier and pass on one feature vector at a time for classification starting with the topmost hypothesis. It would stop the communication as soon as it would get back a feature vector classified as OPT or when all the feature vectors from the list had been classified. For our example list the confidence classifier from the experiments in Section 8.4.4 returned OPT for the third feature vector. This means it would not be necessary to classify the whole list. This would leave us with the three classified hypotheses shown in Table 9.8.

Table 9.8: Classified N-Best hypotheses

Rank	Dialogue Move Interpretation	Confidence Label
1	answer(event(meeting)) answer(number(0))	cross
2	answer(event(meeting)) answer(time(1100))	neg
3	request(add_event)	opt

As the third hypothesis has been classified with the maximum category OPT there would be no need for the re-ranking algorithm this time. The confidence annotator would directly select this hypothesis.

Our **Confidence** module has managed to select the correct hypothesis from the list. The next step would be to estimate the IS DM confidence score for the selected hypothesis based on the confidence label and the ASR DM scores according to Equation 9.1. The corresponding confidence class score would be taken from a table in the TIS as exemplified in Table 9.1 that would be configured by the developer. We will here use the proposed scores and the proposed weight from Section 9.2.2 for illustration. This would result in an IS DM Score of 68.5 which indicates quite high confidence. The **Confidence** module would finalize by updating the LATEST_MOVES variable in the IS as in (64).

(64) $$\text{LATEST\_MOVES} : \text{Oqueue}\left(\begin{bmatrix} \text{MOVE} & : & \text{request(add\_event)} \\ \text{MODALITIES} & : & \text{Set(speech)} \\ \text{SCORE} & : & 68.5 \end{bmatrix}\right.$$

Following the example thresholds in Table 9.4 and the approach presented in Section 9.3 the proposed grounding strategy would be to implicitly verify the utterance as the value lies between T1 and T2. A system response based on this example could then be "What type of event do you want to add?". Taking into account also the implementation of the first part of this chapter the system would, before realizing the selected system dialogue moves (including the chosen feedback move), predict the next user dialogue move. Based on this dialogue move prediction the system would prepare for loading an appropriate DMSLM.

9.3 Summary and conclusions

The first part of this chapter has sketched the straightforward implementation of the dialogue move prediction and DMSLM switching based on the experiments in Part II. A

dialogue move prediction module has been implemented for the GoDiS dialogue system which is applicable to any GoDiS application. In addition, we have also developed a strategy to switch DMSLMs on the go. The solution is domain and language independent. The dialogue move prediction can be used with or without DMSLM switching. The dialogue move prediction implementation has been applied to the AgendaTalk system together with DMSLM switching and tested for viability. The dialogue example in (55) shows the new AgendaTalk behaviour when running the new modules. By adding dialogue move prediction and DMSLM switching to GoDiS we have obtained a more context-aware speech recognition behaviour. However, to evaluate the real impact of this approach on speech recognition performance as well as on task performance an evaluation in a realistic setting with motivated users would be needed. Chapter 10 discusses the needs for such an evaluation and outlines a strategy.

The second part of this chapter first introduced ways of estimating confidence scores on the dialogue move level. It was shown how such approaches where confidence estimation works on a conceptual level rather than on a word or utterance level can result in a more appropriate grounding behaviour. We introduced a new confidence score called the *information state based dialogue move confidence score* (IS DM Score) which is based on dialogue move scores and the confidence labelling from Chapter 8. The final part of the chapter has described the proposed implementation of the new GoDiS module **Confidence**. This module would implement the feature extraction, confidence classification and re-ranking from Chapter 8 as well the IS DM Score estimation. This proposed integration of the results from the experiment in Section 8.4 would result in an approach involving a tighter coupling of the GoDiS system and an ASR system. The final decision of which hypothesis to select from an N-Best list and what confidence to assign to it would depend on many more knowledge sources such as the semantic parsing or the current IS. This means the decision-making is postponed to a later stage when all information is available. The integration will also require more processing at the moment of recognition and understanding. The implementation leaves it open for the developer to optimize the performance by modifying the weights and thresholds based on further experiments. Although this gives freedom to the developer it also puts responsibility on the developer to estimate optimized settings in order to achieve a proper performance.

As will be discussed further in Chapter 10 the actual grounding strategies discussed are only for the sake of illustration and more factors need to be taken into account. The proposed implementation is actually independent of particular grounding strategies and not bound to the current grounding implementation in GoDiS. Although much of the code needed for the proposed implementation was already developed for the experiments in Section 8.4.4 it would probably be necessary to optimize it for real-time performance. However, as we are restricting ourselves to shorter N-Best list (5-10) we do not expect any efficiency problems in an implementation. There are many ways of extending the proposed confidence annotation and alternative implementations which could be considered, as will be discussed in the following chapter.

Chapter 10

Future directions

The results from the experiments in **Part II** and **Part III** are encouraging and point to a number of future developments. In this chapter I will indicate some of these possible directions. In addition, recently published work related to the findings in this thesis will be discussed. The first part of this chapter considers future work building on the outcomes from the language modelling experiments and the dialogue move prediction in **Part II** as well as its implementation in Chapter 9. The second part of the chapter relates to the confidence annotation and re-ranking investigations from Chapter 8 in **Part III** and the proposed implementation of this in the previous chapter.

10.1 Grammar-based SLMs and dialogue move prediction

Part II started with the generation of SLMs from GF interpretation grammars exhibiting the benefit of this approach when creating initial models for spoken dialogue systems. The same approach was also used to develop semantically based context-specific models that would more appropriately fit a specific dialogue context. These were called dialogue move specific SLMs (DMSLMs). Dialogue move prediction using machine learning was proposed as an approach to automatically and dynamically choose the most appropriate of these DMSLMs given the current dialogue context. In Chapter 7 in **Part III** we applied the bootstrapping methodology from Chapter 4 to construct a simple dialogue move tagger. The experimental investigations in this thesis clearly display the strength and benefit of the proposed approaches. Notwithstanding fine-tunings, improvements and extensions of these approaches would inevitably obtain even better results. This will be discussed in the following sections.

10.1.1 Grammar-based SLMs

The development of grammar-based SLMs is an approach to bootstrapping preliminary SLMs for spoken dialogue systems when no or little training data exists. It has been

demonstrated in the experiments in this thesis that such a model performs much better than the initial grammar when used directly for ASR. Several recent studies have applied this methodology successfully for different spoken dialogue systems obtaining very positive results (Bangalore and Johnston, 2004; Raux *et al.*, 2003; Weilhammer *et al.*, 2006b). An exception is the study by Hockey *et al.* (2008) where the speech recognition grammar (SRG) strikingly outperforms the SLMs created from the grammar. Why this study making use of the Regulus platform for grammar development does not profit from SLM grammar generation as opposed to the other studies and the results presented in this thesis is hard to tell. A difference in their study, as opposed to the other studies and the work presented in this thesis, is their use of a seed corpus from which they derive the grammar. It is unclear how the seed corpus in Hockey *et al.* (2008) was created and if the grammar writer had knowledge of the test data that had been collected in previously versions of the system. Another distinction is the test setting they propose where subjects have a training phase with in-coverage sentences and learn to use the microphone etc. This could have led to a test set with less spontaneity, noise and disfluencies and more in-coverage and in-domain utterances than in the other studies which would favour grammar-based recognition.

Hockey *et al.* (2008) point out that with the methodology presented in this thesis (and also used in related work (Bangalore and Johnston, 2004; Weilhammer *et al.*, 2006b)) it is not clear what data was used to build the initial grammar or what utterances that the grammar writer had in mind. They therefore propose to start with a seed corpus on which the grammar is built (and trained) and then proceed to compare this grammar, the grammar-based SLM derived from it against an SLM built from the original seed corpus. In this way they argue it is possible to compare if the "roundabout" of grammar writing is fruitful. According to them a grammar built based on a seed corpus contains only a little more information than the corpus used to create it. We have a slightly different view of grammar writing where we see it as a compact way of generalizing over a corpus. The idea of generating a corpus from a grammar is in order to be able to obtain an extensive corpus rapidly. It is therefore not a roundabout in any sense but a straightforward way of producing an extensive coverage. With grammar generated corpora of around 10K - 1M utterances as used in Bangalore and Johnston (2004); Weilhammer *et al.* (2006b); Jonson (2006b) it would be unsustainable to write all the intuitions down in the form of sentences. The corpus generated is considered to be what was in the grammar writer's mind. Generation is actually a helpful and commonly used tool to check that a grammar is covering what the grammar writer had in mind. The grammar generated corpora in this thesis are exhaustive and therefore have exactly the same coverage as the original grammars. This make them perfectly comparable. In most other studies the corpora have been randomly generated to a certain extent.

The use of a grammatical framework such as GF or Regulus also makes it possible to make use of grammar resources to help grammar writers to generalize further and achieve more linguistic coverage. It should be mentioned that the grammars in this thesis were not written in order to render sentences as opposed to the grammars in Bangalore and Johnston (2004); Weilhammer *et al.* (2006b). They were built in the TALK project (by

others than the author[1]) to develop GF interpretation and speech recognition grammars for various experimental dialogue systems among others the two baseline systems presented in this thesis.

Whereas Hockey *et al.* (2008) and other studies (Knight *et al.*, 2001; Rayner and Hockey, 2003; Hockey and Rayner, 2005) have focussed on the comparison and discussion of advantages and disadvantages of SLMs and SRGs the ultimate purpose of this thesis is not to favour any of the directions but to illustrate how they can be combined and how statistical language modelling actually can profit from the use of grammars. The choice between grammars and SLMs is rather a matter of the purpose of the end applications than the performance of the different approaches. When users are expected to be mostly novices, applying spontaneous speech and when real data is supposed to be collected during the application's lifetime the choice should fall on SLMs. On the other hand, if the application is going to be exposed to users that will become experts and that will accept a more restricted coverage the alternative might be hand written SRGs. It is clear from the results in this thesis and from the results in (Knight *et al.*, 2001; Rayner and Hockey, 2003; Hockey and Rayner, 2005; Hockey *et al.*, 2008) that grammar-based approaches are more suitable for in-coverage data. However, it is questionable whether speech recognition models should be compared on their performance on in-coverage data as the grammar coverage is only an assumption of what expressions users will apply and mostly does not include expected phenomena such as disfluencies. The purpose of most spoken dialogue applications are to recognize utterances that are suitable in the context of the application, i.e. in-domain data. It would therefore be interesting to see the results when comparing performance on in-domain data vs out-of-domain data rather than in-coverage vs out-of-coverage. However, this distinction would require manual partitioning of the data. Such a distinction would also apply to comparison of performance of SLMs where no grammar exists.

Although the setting in this thesis has been two small experimental systems with a limited set of both naive and expert subjects the aim is towards methods applicable for commercial applications with unlimited numbers of naive users. In this case grammars are not a long-term possibility. However, what we have demonstrated is that grammars can be of use in the first stages of dialogue system development avoiding expensive WOz settings.

The use of artificially generated corpora from grammars is a way to generalize over a small amount of data as well as a way of introducing a dialogue system developer's intuitions about the domain. A difference in our experiments is the exhaustive non-probabilistic generation that we use. In this way the corpus covers everything that the grammar covers. This requires a grammar that does not overgenerate and is meticulously written. This approach is beneficial for word sequences reoccurring in different contexts but is clearly disadvantageous for short utterances that will only occur once in the corpus. A clear example was the Swedish words for "yes" and "no" that were badly trained in our models in Chapter 4. More accurate statistical estimations could have been achieved by using

[1]The GF grammars for AgendaTalk and DJ-GoDiS were written by Peter Ljunglöf, Ann-Charlotte Forslund, David Hjelm and Håkan Burden

a probabilistic grammar to generate artificial corpora randomly. However, real data is necessary to be able to estimate a PCFG and that was not available for the experiments.

Instead in Chapter 4 we made use of a Swedish spoken language corpus (GSLC) to incorporate more realistic estimates of spoken language to our grammar generated corpus. The results were very encouraging and surprisingly good considering the limited size of the GSLC corpus. It would be very interesting to see what impact a bigger spoken language corpus would have had and whether it would be possible to create in some sense a generic spoken language SLM to be used for interpolation with domain-specific models. Such a model would be able to contribute with the typical spoken language patterns that are so hard to capture. From the domain perspective a grammar-based SLM can also be used as a *seed SLM* in order to collect related corpora from the web as discussed in Section 2.4.4.4. Such a web selection approach was applied by Weilhammer *et al.* (2006b) giving a slight improvement to the model. The results of selecting in a simplistic manner the most related parts from the newspaper corpus as described in Section 4.2.1.3 gave a tremendous effect on performance. Bangalore and Johnston (2004) interpolated a grammar-based SLM with an SLM derived from a domain-specified generic corpus with positive results. The domain specification was made by tagging linguistic units, such as nouns, in the corpus and substituting them for linguistic units from the domain. These examples clearly makes evident the importance of exploring further more advanced selection techniques and methods of making use of generic corpora.

Another advantage of using a grammar-based SLM instead of the original grammar to collect real data with a running system is the ease of later incorporating and extending the model as training data becomes available. What has not been discussed is the use of grammars in on-line development. Grammar generated corpora could well be a strategy to more easily incorporate additions to an existing SLM in a running application. User language varies over time and new issues may come up that the original system does not handle. Developers are then often confronted with the problem of collecting new material. With the approach proposed in this thesis you can generate an additional corpus covering the new issues appearing and quickly incorporate them into the original SLM. In this way the system is able to cope, at least to some extent, with these new issues directly while data is collected.

What has not been investigated in this thesis is at what point a grammar-based SLM might become obsolete or whether a grammar-based SLM can be useful even when more training data exists.

10.1.2 Dialogue move specific SLMs

For DMSLMs the possible future approaches are quite similar to the above. The problem when building dialogue state specific SLMs by using user utterances collected in different dialogue states have been that some states are less visited than others and also that some options in a state are chosen less often. This will make some user possibilities less well represented in a context-specific SLM and some context-specific SLMs worse than others. With grammar-based SLMs for different contexts we assure a minimum coverage. In order

to extend DMSLMs user logs will be parsed and real user utterances will be added to the appropriate DMSLM. The data collected will also reveal what dialogue moves are expected in different contexts and give us training data for the dialogue move prediction. It might well turn out that the combination of dialogue moves in a DMSLMs will need to be differently distributed. Collected data will tell us how to combine them.

For the experiments in this thesis we were for technical reasons forced to prepare the DMSLMs beforehand. However, the idea is to be able to generate DMSLMs dynamically being able to combine different dialogue moves in order to create the most appropriate DMSLMs for different stages. We therefore discussed when it came to dialogue move prediction not only whether to predict the most likely dialogue move but also an N-Best list of the most likely moves in a certain context. Based on this list of possible dialogue moves we could then generate utterances representing these dialogue moves from the interpretation grammar. To this artificial corpus we would then add the collected user utterances having been interpreted as the predicted dialogue moves. Finally, we would interpolate this specific SLM with the generic SLM in order to obtain a new dynamic DMSLM. A less technically heavy approach would be to prepare DMSLMs representing each dialogue move off-line (combining grammar generated corpora and transcribed corpora) and then only make the interpolation of different DMSLMs and the generic SLM online.

In traditional state-specific SLMs the SLMs represent what words are used at different stages. We have proposed a conceptual approach where we group together possible dialogue moves that a user may perform in different contextual situations. However, the usage of one word or another is also dependent on the context, making some words belonging to the same category more likely than others. One example is the phenomenon of *lexical entrainment* as discussed in Section 2.4.7. It could well also be that the user has already chosen among alternatives and that these chosen alternatives are then the most likely to reoccur. An extension to our DMSLMs would therefore be to also allow them to prime words, e.g. by populating classes based on the context as Gruenstein *et al.* (2005) proposed (see Section 2.4.4.3). However, such an approach would require that we actually keep track not only of concepts in the information state but also of word choices. In order to be able to boost words in our models we need to predict not only dialogue moves but also which words are more likely to be used.

10.1.3 Dialogue move prediction

DMSLMs and dialogue move prediction are tightly coupled and we have therefore already discussed some possibilities for further work on dialogue move prediction in the previous section. For the experiments on dialogue move prediction we had very limited training data. More training data and training data from more realistic user settings would provide a better foundation for future work. As discussed in the conclusions of Chapter 6 one way to constantly improve the dialogue move prediction would be to use an adaptive learning method and learn from interactions with users. User dialogue moves performed in different prediction states would be used to update the model. ASR and interpretation errors might lead to the introduction of noise into the model. The use of our confidence annotator to

select only dialogue moves with high confidence to improve the models could be a way to avoid part of this noise.

As for all machine learning experiments there are broad possibilities for elaborating further both on the feature set, the feature representations and to explore other machine learning techniques than the two used in this thesis. What seems most necessary is to explore further how to represent dialogue moves and especially how to represent a dialogue history in a compact but informative manner. What is also of interest is to experiment only with features that are not domain specific or explore ways of customizing a generic dialogue move predictor to a new domain. In the final experiments we predicted dialogue moves types rather than dialogue moves. With more data it could well be possible to predict more specific representations, for example that the user is not only likely to perform an answer to a question but that that answer will include a song. This was explored in the first experiment but abandoned as such an approach makes the prediction very domain-dependent.

10.1.3.1 Planned evaluation

In order to estimate the real impact of the implementation of dialogue move prediction in combination with DMSLMs in GoDiS it would be necessary to carry out an evaluation with real users. In Chapter 5 and 6 we could clearly show that using DMSLMs would give us an interesting benefit if we could predict when to apply which DMSLM in the dialogue system. Our approach to dialogue move prediction was proved to be sufficiently accurate in order to maintain an overall recognition performance impact. The dialogue move prediction has been implemented as described in Chapter 9 and the DMSLMs described in Chapter 5 have been used to create a new version of the AgendaTalk system. This new version has a statistical and context-aware speech recognition approach as opposed to the grammar-based approach that was our starting point.

To evaluate the performed integration and the use of the methodologies in **Part II**, i.e. grammar-based SLMs, DMSLMs and dialogue move prediction, a collection of data with novice users with the new version of the AgendaTalk system would be necessary. In this way it would be possible to estimate the real impact of the application of these techniques together and evaluate both prediction accuracy and speech recognition performance. However, such an evaluation needs a realistic setting with motivated users. The impression from the data collected for the experiments in this thesis with the experimental system AgendaTalk is that it is very hard to make subjects behave as real users and that they therefore easily change goals, play around for fun or follow instructions too strictly. Such user behaviour would skew the results. We could either end up with subjects that are too compliant or too collaborative which would give us a prediction accuracy which is too high. On the other hand we could also end up with subjects that are not motivated and would accept misunderstandings or suddenly change topic when bored. This would result in too low a prediction accuracy. We have therefore chosen to postpone the evaluation until we have the opportunity to provide a more realistic setting where subjects would be able to perform tasks that are useful for them.

We will at any rate describe how such an evaluation could be carried out taking as an example the AgendaTalk system. We have already created two versions of AgendaTalk: the fullfledged dialogue move prediction and DMSLM switching version (version 2) and version 1 where dialogue move prediction is carried out implicitly but DMSLM switching is not applied. Having dialogue move prediction running in both versions (implicitly and explicitly) enables the analysis of dialogue move prediction accuracy on all collected data. Having DMSLM switching on or off makes it possible to compare both speech recognition performance and overall performance between versions.

The subjects should be given two very similar tasks. An example would be a first task involving deleting a booking, adding a booking and changing a time of a booking. In the second task the subject would then need to add the deleted booking from task one, delete the added booking from task one and change back the time of the booking in task one. In this way the calendar would look the same as in the beginning after carrying out the two tasks. Subjects are to be divided into two groups. The first group would use version 1 to start with (only dialogue move prediction) followed by version 2 (also DMSLM switching). The second group would test version 2 first and subsequently version 1. The objective of this is to eliminate the influence on the results of a possible learning curve.

We will evaluate both prediction accuracy and speech recognition performance. The prediction accuracy is easily obtained by comparing the predicted move (preddm) with the actual performed user move. The user dialogue moves can be manually tagged based on transcriptions of the recorded audio files. A more noisy but automatic comparison would be to compare against the system's interpreted dialogue moves. Prediction accuracy is not expected ever to be perfect as different users will behave differently in the same situations. What we want to achieve is to at least predict properly the most commonly performed dialogue moves in certain dialogue contexts. The analysis of this part of the evaluation would then be to see that the dialogue move prediction is not missing any clear patterns and that the patterns it predicts actually occur. Speech recognition performance can be evaluated in several ways. We can compare the overall performance in version 1 vs version 2. We can also compare the performance in version 2 against the performance on the same recordings when applying the generic model. Both comparisons would be influenced by the success of the dialogue move prediction. In order to compare with the starting point of **Part II** and also include the advantages gained in Chapter 4 we would need to also compare against the original SRG. This would give us the actual impact of the application of all the proposed approaches in **Part II**.

10.1.4 Dialogue move tagging

Chapter 7 proposed that GF grammars can be used to generate semantically annotated corpora. In this way semantic decoders can be bootstrapped without the need for manual annotation. The dialogue move taggers presented seemed to better cover unforeseen utterances that had not been anticipated in the grammar modelling. The purpose of the experiments in Chapter 7 was to quickly achieve more robust parsing than is possible with a GF grammar. The resulting taggers only perform a very shallow annotation and do not

capture deep semantic hierarchies. More sophisticated semantic coding can be achieved with the same methodology if the grammar and the corpus provide a more detailed hierarchically organised structure. This was shown in parallel with the work in this thesis by work by Weilhammer in the TALK project (Weilhammer *et al.*, 2006a). Two hierarchical parsers were developed from GF generated corpora in English for a Tourist domain. These two bootstrapped semantic decoders, a N-gram parser and HVS model were shown to perform much more robustly than the corresponding GF grammar in a similar manner to the dialogue move taggers. With this strategy it was also possible to perform more deep parsing and discover full semantic trees. In the experiments in Chapter 7 the taggers only made use of the actual words in order to determine dialogue moves. It is our belief that a dialogue move tagger would, just as was shown with the dialogue move predictor, profit from dialogue context and other knowledge in the semantic decoding process. However, in order to train such a tagger we would need training data from dialogue logs. Semantic decoding and dialogue act tagging is an extensive research area and the work in this thesis was only a sidestep in order to produce a tool for the other experiments in the thesis. The possibilities for future work is therefore immense. One possible use of a simplistic bootstrapped dialogue move tagger like the one presented here could be for raw automatic annotation to prepare a corpus for manual annotation.

10.2 Confidence annotation and re-ranking

The resulting confidence model from Chapter 8 looks very promising. An immediate future direction should of course be to implement the proposed model into the GoDiS platform following the specification in Chapter 9. With this in place it would be possible to carry out an evaluation of the real impact of this approach in a spoken dialogue system. Following the principle in the previous sections we would implement the confidence model and re-ranker such that these processes can be switched on or off. In this way it would be possible to make a comparison of this novel approach with the currently available strategies.

A more general future direction is to further explore the use of other knowledge sources and better ways to represent them computationally. Various recent publications have related to the work in this thesis and further investigated the use of higher level knowledge sources for confidence annotation and N-Best re-ranking. These studies seem to re-affirm the results in this thesis that the use of linguistic knowledge and knowledge regarding the dialogue is beneficial for the task.

Georgescul *et al.* (2008) actually carry out a similar experiment to the one in Chapter 8 in the same domain, a calendar system. They train a classifier using SVM learning to re-rank 5-best lists from a PCFG. Following the ideas of Gabsdil and Lemon (2004) and Jonson (2006a)[2] they make use of linguistic features to achieve a significant reduction in WER. Although they put forward many interesting features, a great deal of them were proved not to be informative in the experiment and had to be discarded. Their most informative feature is a semantic feature representing whether a query is underconstrained or not. They

[2]The first part of Chapter 8 is an extended version of this paper

also present interesting syntactic features such as if the hypothesis is an elliptical phrase, is in imperative form or includes an existential construction. Unfortunately, they do not relate their features to the dialogue context.

The work in this thesis is heavily influenced by Gabsdil and Lemon (2004). Lemon and Konstas (2009) build further on that same study and the work in this thesis (as published in Jonson (2006a)). In this bigger study they confidence classify and re-rank 60-best lists using memory-based learning and eventually make use of some of the features proposed in Jonson (2006a) such as introducing features representing the whole N-Best list. In addition, they introduce a novel feature from user simulation which estimates the likelihood of a hypothesis given the previous dialogue context. The dialogue context is here represented with 5-grams of consecutive dialogue moves. Dialogue moves are represented as pairs of the dialogue move type and the task with arguments (slots). The results obtained show a significant reduction in WER with the user simulation feature being the most contributive feature.

Wu and Seneff (2007) refer to the author's work, as presented in Jonson (2006a), and state that it is "well-known that, by taking into account dialogue context information, speech understanding performance can be improved". They are able to show that a high level description of the semantics of the preceding dialogue assists in the process of selecting N-Best candidates. They use a learning classifier system (LCS) to re-rank and give confidence to ASR hypotheses following the ideas in Gabsdil and Lemon (2004) and Jonson (2006a). In order to obtain more training material they use simulated user dialogues. Dialogue context is represented as a sequence of consecutive high level speech acts (similar to dialogue move types). They achieve an important reduction of SER when re-ranking 5-best lists by selecting the hypothesis with the highest combined contextual, acoustic and parsing score.

The studies by Wu and Seneff (2007) and Lemon and Konstas (2009) give support for our results in Section 8.3 that dialogue context matters and indicate that an extended dialogue context is indeed of value. That our explicit dialogue history features in Section 8.4.4 did not give the expected result might be due either to a representation which is too complex or lack of training data. It would be worth considering a new representation such as a score of the likeliness of a move given the previous context. In fact we did use a similar feature with our predicted dialogue move which was indeed shown to be fruitful. Also, our information state based features that represent dialogue context but in a more implicit way did prove to be discriminative. As pointed out in Section 10.1.3 new ways of representing dialogue context will be necessary. In the same way we need to explore how to best represent dialogue moves. In an abstract domain-independent way as dialogue move types as in Wu and Seneff (2007) or in a more specific domain-dependent way with arguments as in Lemon and Konstas (2009) or Kim *et al.* (2007)?

Other studies reaffirm the importance of semantics. Balakrishna *et al.* (2006) introduce generic semantic features derived from WordNet and PropBank to estimate the semantic coherence of a hypothesis. These semantic features were shown to contribute most when re-ranking N-Best hypotheses in a large vocabulary task. In Kim *et al.* (2007) semantic features are also shown to be of high importance. They estimate probability scores for

speech acts (similar to dialogue move types), main actions and speech act attributes (slots) that are used by the classifier. Although they do not take the context into account, but only give a semantic probability score to each hypothesis based on the wording, they obtain an interesting word and concept error reduction.

In Chapter 9 we suggested the use of a semantic parser that could both parse robustly and give a syntactic score (grammaticality score). In a similar manner to Gruenstein (2008) it would be helpful to at least be able to distinguish between fully parsed hypotheses (grammatical ones), robustly parsed hypotheses and unparsed hypotheses. An even better option would be to obtain a semantic score as in Kim *et al.* (2007) or a syntactic one as in Balakrishna *et al.* (2006) or Lemon and Konstas (2009). In Purver *et al.* (2006) the scores from two different semantic parsers (deep and shallow) are actually combined. Purver *et al.* (2006) also take into consideration some of the semantic and contextual features from Gabsdil and Lemon (2004). As the weighting of scores from multiple knowledge sources is defined manually, as well as the confidence thresholds used for grounding, it would be interesting to apply some of their ideas to a machine learnt classifier.

Georgescul *et al.* (2008) proposed the system response type that each hypothesis would trigger as an additional feature. They were, however, not able to show any informative value of this feature. In a framework like ours where dialogue context is represented in a principled manner and system responses can be represented on a semantic level as dialogue moves this might, however, be an interesting feature. Gruenstein (2008) actually recasts the problem of re-ranking ASR hypotheses into re-ranking system responses. He suggests estimating confidence of the system responses that the ASR N-Best candidates will evoke. It is shown that the number of unique system responses are not only less than the number of hypotheses but also less than the number of unique parses. This makes the selection task much simpler. He uses SVMs to train a classifier that discriminates between acceptable or unacceptable system responses. Many of the features used are taken from Hazen *et al.* (2002). Additional features such as the distribution of responses, type of response and the parse status are also applied. Although this approach is very interesting, it somehow simplifies the role of the dialogue manager and does not take into account the fact that user utterances do not only evoke verbal system responses but many different system actions. However, the novel features presented seem suitable also for an approach that focuses on re-ranking and estimating confidence for N-Best hypotheses.

Each of the studies discussed above use different features and different ways of representing them. The experiments are carried out in distinct domains on different data. Therefore, it is very hard to compare and draw any conclusions about what features are fruitful in general and not only to a specific domain and dialogue system. A thorough investigation of what features are useful against a common baseline would be a desired future direction. We think the ISU framework would be an exceptional experimental setting in this connection. In addition, it would be desirable to carry out more experiments with human subjects, similar to the one in Chapter 8, in order to investigate further what features humans seem to make use of in dialogue and how we best can represent these computationally.

In order to find general features we need to keep to domain-independent features. In

our study only four of the thirty-nine selected features were domain-dependent in the sense that their values included domain-related information. As was shown in the experiment presented in Section 8.4.5.2 we achieved quite good results even without these domain features. Either we opt for a domain-independent model or a model that is easily adapted to new domains. Domain specification may actually be necessary in order to model certain patterns. Bohus (2007) carried out a cross-domain evaluation where confidence models trained on data from one domain were tested in other domains. It was shown that the models more or less generalized but did not perform as well as domain models. Bohus (2007) also tested adapting a model to a new domain with a small amount of in-domain data. Such an investigation would be a natural follow-up of our domain-independent confidence model as the aim is to have one single confidence model in the GoDiS ISU framework.

In a similar manner we want to opt for language independent features and try to avoid features, for example, that represent words explicitly. Although, on the surface all our features were language independent there might be features whose values are not reusable across languages. Hypothesis word length is one example. Another example, is all features based on acoustic confidence scores as the acoustic models in an ASR system might be distinct from language to language. Also, the values of these same features are highly dependent on the ASR system in use. It would therefore be necessary to evaluate our model on data other than Swedish. As was shown in the experiments in Chapter 8 it is possible to obtain reliable confidence models even without ASR features. Such a model would be independent of the ASR system. One problem with traditional confidence scoring is that, as it depends only on acoustic information, the scores are biased by how well the acoustic models model the actual speaker. Some speakers will continually get low scores. This problem might be avoided when more knowledge sources are taken into account. It would therefore be interesting to investigate the impact of this model for low-performing speakers – introduced as "goats" in Section 2.4.3.1.

There are some evident features that we have not been able to explore in our work. For example, in both Gabsdil and Lemon (2004) and Lemon and Konstas (2009) the use of amplitude features in order to capture prosody are shown to be fruitful. Prosodic features are also taken into account in the confidence model presented in Bohus (2007). In a similar manner, Ananthakrishnan and Narayanan (2007) have shown how prosody can be used as an external knowledge source to refine N-Best rescoring in large vocabulary tasks. They represent prosody as a statistical prosody model built on the syllable level. This type of acoustic feature is a knowledge source that we have not taken into account in our work. It would be very interesting to see if prosody would improve, for example, our model's ability to distinguish noise or crosstalk from other utterances.

The approach presented in Section 9.2.3 does not consider multimodality. For a multimodal setting we would need to take into account input from other sources such as from a GUI. As the structures passed around in GoDiS include the modality value it would not be a problem to add separate processing for different modalities. It would be interesting to take multimodality into account for the confidence annotation and re-ranking. In the experiments in Chapter 8 we actually had some features that could take on values from

other modalities. An example was that the feature Action would hold the latest action performed which could well have been an action performed in the GUI or on the device such as the music being turned on or off. Due to the small amount of multimodal training data this information did not affect the classifier. The human subjects on the other hand seemed to take such information into account when re-ranking. With more training data it might turn out that this information is of relevance. However, it would be interesting to also take into account moves performed in other modalities in the same turn. Parallel (or immediately preceding or succeeding) graphical input might well make one hypothesis more plausible than another. In order to take into account input from other sources than the ASR the **Confidence** module would need to consider such input when confidence classifying and re-ranking. Kim *et al.* (2007) introduced a multimodal reference resolution score as a feature in their N-Best re-ranking model. Unfortunately, they could only show a very small improvement due to the distribution of training data where the evidence for multimodal references was scarce. With more training data including more multimodality it might be possible to prove the importance of multimodal information as an additional knowledge source.

More training data is also important in order to better capture dialogue context patterns and other features. An optimal way to achieve more training data is to collect data with real systems. However, such an approach requires a deployed system and additionally requires an important time effort for annotation. Although our approach does not require any manual annotation of the labels it does require manual transcription of recordings which is a time-consuming task. One possible shortcut is to produce data artificially with user simulation as in Wu and Seneff (2007) and Lemon and Konstas (2009). Another tempting idea to avoid manual annotation is to confidence annotate hypotheses based on users' reactions to system understanding. Bohus and Rudnicky (2007) propose such an implicit supervised learning strategy where user corrections in dialogue are used to automatically confidence annotate utterances. Specifically, they take into account users' responses to explicit confirmations in order to select and label new data to refine the confidence model. If the user responds in an affirmative way to the previous user utterance it will be labelled as correct whereas if the user rejects the system's understanding of the preceding user contribution it will be labelled as incorrect.

If we consider such as strategy in GoDiS with our new confidence model we could, for example, gradually learn to better distinguish utterances classified as PESS by labelling them as either POS or NEG dependent on the user reaction to the explicit confirmation. In this way we would be able to minimize explicit confirmations in the long run. We could also consider taking into account user reactions to implicit confirmations or even to explicit rejections. A non-reaction to an implicit confirmation would be considered as confirming the current labelling as correct. A correction of it would mean re-labelling it to, for example, NEG. The same would hold for an explicit rejection but in the opposite way. In this way the classifier could be reinforced and learn from future experience. As Bohus and Rudnicky (2007) point out one might need to take into account not only the direct reaction in a confirmation subdialogue but also whether the propositions and actions introduced are accepted throughout the dialogue. In the ISU framework this would be quite straight-

forward as long as we also keep hold of which utterance gave rise to what propositions. Nevertheless, we would need to decide the minimum confidence needed for the user reactions in order to trust this implicit source. Furthermore, if we consider the confidence and grounding approach proposed in Section 9.2.3 where grounding is not applied on the turn level for each utterance but for each dialogue move we would rather need to automatically label dialogue moves than utterances. How implicit learning can be applied to the dialogue move level would need further exploration.

In this work we have examined two different machine learning techniques. We did not obtain any big difference in performance between them. In a similar manner Skantze and Edlund (2004) compared memory-based learning and transformation-based learning for the task of detecting word errors. Again, there were no apparent differences in performance. In Bohus (2007) several machine learning techniques were tested for the task of confidence annotation. No major differences could be shown although linear regression seemed to be more apt for the task. It therefore seems that further comparison of different techniques is not as interesting as further exploration of different features.

Although many studies have chosen to estimate confidence scores or make a binary decision (accept or reject, correct or incorrect) our approach was to discriminate between several different confidence levels. Our 6-way classification task is much more fine-grained than many other studies. It would of course be possible to merge some of our classes. In fact, our automatic labelling appraised correct perception over correct understanding (by distinguishing OPT and POS) as the ultimate task was to improve also speech recognition accuracy. However, hypotheses labelled as OPT or POS were in fact semantically identical according to our labelling conditions. From a semantic perspective there is therefore no need to make such a distinction and we could well have labelled them identically. In the same way it could be argued that there is no need to distinguish between IGN and NEG. However, in our labelling schema IGN was applied to complete non understandings whereas NEG was given to misunderstandings where at least some concept or part of the utterance was more or less correct. Lemon and Konstas (2009) adopt four of the labels from Jonson (2006a): OPT, POS, NEG and IGN. In their classification schema CROSSTALK falls under IGN whereas the hypotheses that we label as either PESS or NEG are all classified as NEG. An optimal labelling schema would need to be investigated further as well as more optimal conditions for labelling.

It is a matter of choice whether to distinguish CROSSTALK separately or not. In order to perform grounding more precisely we need not only to detect whether a hypothesis is incorrect but also to know what is incorrect about it. Unlike the other labels CROSSTALK should be given to a whole N-Best list and not to individual hypotheses. A possibility would therefore be to use a binary classifier that detects crosstalk and only passes on non-crosstalk N-Best lists to the confidence classifier. A preliminary binary classifier presented in Section 8.4.5.2 had an almost perfect crosstalk detection (98.9% accuracy).

As discussed in the previous chapter confidence scoring on the utterance level might not be optimal. A future direction might therefore be to investigate how we could train a confidence classifier on the dialogue move level directly. Such a classifier would not give confidence to utterances or to dialogue move sequences but to the possible dialogue moves

in an N-Best list. This is based on the importance we have seen of the homogeneity of an N-Best list. Rather than re-ranking we would attempt to produce the best possible dialogue move sequence from the dialogue moves in the whole list. The idea resembles work on re-scoring of lattices but carried out on a semantic level. This would imply interpreting all the hypotheses in a list and then represent the N-Best list as all unique dialogue moves (for example as a graph) which would then be given confidence scores. The confidence for repetitive dialogue moves would be boosted. The next step would be to select the most plausible or highest scored dialogue moves from the list. In this way we would build up a new dialogue move sequence. This dialogue move sequence could well be one already existing in the list or a complete new combination. The hope is that we in this way would pass on only the most reliable moves and discard already in this phase incorrect dialogue moves. As Bohus and Rudnicky (2005a) point out the real aim in ASR performance for dialogue systems is to maximize the number of correctly transferred concepts and minimize the number of incorrectly transferred concepts and not necessarily to optimize word accuracy. In spite of this view they like many others estimate confidence on the utterance level. With confidence given on the dialogue move level it should be much easier to optimize this task and pass on the correct moves from an utterance while rejecting incorrect ones.

In this thesis we have focussed on ways of preventing errors from happening as well as detecting them at an early stage. However, we have not paid much attention to how to react when an error is detected. A new more accurate way of estimating confidence will most certainly facilitate more reliable grounding and error-handling strategies. With more error awareness dialogue systems will be better at knowing when something is wrong and what it is that is wrong. Notwithstanding, it is not straightforward to know how to proceed with erroneous ASR output. Research on error-handling has shown that the current grounding strategies in spoken dialogue systems are rather poor and do not seem to align well with human strategies. It has been shown that humans rather than explicitly signalling misunderstanding or non-understanding often opt for moving on with the dialogue by for example applying an alternative dialogue plan (Skantze, 2005a). Bohus and Rudnicky (2005b) showed how this same strategy, which they called *Move On*, was the most successful strategy considering error-recovery rate. Studies by Bohus and Rudnicky (2005b); Shin *et al.* (2002) also show that users are, for example, more likely to rephrase than repeat in order to correct an error and that different correction methods are dependent on the system's error-recovery strategy. Other promising error-recovery strategies shown in Bohus and Rudnicky (2005b) were different types of help messages to guide the user of how to direct the system. This makes studies such as Gorrell *et al.* (2002) with targeted help interesting. However, in order to use appropriate help strategies the system needs more knowledge of what is going wrong: whether it is noisy, the user is out of scope or out of grammar or speaking at too low or high a volume. If the system knows it is misrecognizing an in-domain utterance it could actually try to re-recognize the input with another SLM before communicating the problem to the user. If the system knows the user is out-of-domain there is no use in re-recognizing or asking the user to repeat. It might be more successful to adopt a totally different strategy, for example, trying to guide the user

with more information about the system functionality. Furthermore, the repeat strategy might well be appropriate if some disturbing noise interfered with the user utterance. To conclude, better ways of detecting not only if errors occur but also type of errors will open up for better application of error-recovery strategies.

According to Bohus and Rudnicky (2005b) to recover from a misunderstanding is twice as costly as recovering from a non-understanding. To falsely accept (misunderstand) is much more costly than to falsely reject. Whether the success of error-recovery depends on the type of error or the existence of suitable techniques for recovering from certain errors is unclear. The introduction of cost as a basis for choosing error-recovery or grounding strategies is appealing. Bohus and Rudnicky (2005a) apply a data-driven approach to determine optimal rejection threshold by recasting the problem into considering the cost of an understanding error. Skantze (2007) proposes not to use thresholds at all but calculate costs for different feedback moves. He bases this approach on the use of confidence scores from a commercial speech recognizer. There is nothing that hinders his approach to apply the improved confidence scoring in this thesis combining both approaches. Bohus and Rudnicky (2005a) show that the cost of understanding error varies across dialogue states and they propose that rejection thresholds should therefore be state-specific. It would be interesting to push this a step further and estimate costs for different dialogue moves in line with Skantze (2007). Some types of dialogue moves will apparently be more costly or riskful to falsely accept than others. For example, if the system is going to execute a bank transaction this may require a more pessimistic grounding strategy as the cost of a false acceptance is extremely high. A clear example from the AgendaTalk experiments was the falsely acceptance of the quit move which would shut down the calendar application (see dialogue example 59 in Section 9.2.1). This move would typically have a higher cost. We therefore propose to integrate different types of moves more easily or with more difficulty in ISU-based systems. More cautious grounding might be needed for costly actions but not for harmless actions.

Skantze (2007) also takes into account the possible cost of the reaction to a grounding move as different grounding strategies have shown to evoke different user strategies. We would like to add to this that grounding strategies cannot be based only on one dialogue participant's view of the dialogue. The decision whether the system should confirm or not and in what way (for example explicitly or implicitly) should not be made only in accordance with the system's confidence of its perception/understanding or the cost of the action. We also have to take into account the user's confidence that the system is recognizing and understanding correctly. Therefore, a less optimistic strategy might be needed even if the system is very confident. Hopefully, with less errors the user will get more confident and will then require less confirmation. For such a strategy we will need a way to model the user's confidence of the system's understanding. In summary, there is a need for more research in order to formalise a much more advanced grounding model. However, such a grounding model is only possible with better error awareness.

We should not neglect that some problems in error-recovery can most probably be pin-pointed to the ASR system and not to the dialogue system. As discussed in Section 2.4.3.2 users tend to change their way of speaking when faced with a non-understanding

system. This error-behaviour is counterproductive as acoustic models have normally not been trained on frustrated, loud or hyper-articulated speech. Another possible reason for spirals in error-handling might be that the SLMs are not properly trained on error-handling user responses. Even for explicit confirmation dialogues most SLMs are too simplistically trained without considering many possible ways of responding to a yes/no question. In Shin *et al.* (2002) it is shown that error-recovery is especially hard when it is the user that identifies the error, for example after an implicit confirmation. A common strategy in these situations was shown to be to contradict the system or cancel the current action. It is unclear from this study what role the language model in these failed attempts plays. It might actually be that the problem is that the language models cannot handle contradictions very well. From our studies it was clear that our models needed to reinforce their language coverage when it came to user corrections. Grounding and error-recovery strategies therefore need to be accompanied with proper SLMs. We could for example easily imagine using DMSLMs where error-corrective expressions have been boosted. However, more studies on how users express corrections need to be collected and investigated. As pointed out in Section 2.4.3.2 it might well also be necessary to use appropriate acoustic models.

When applying SLMs in error-handling dialogues we should also take into account the user's reaction to the previous misrecognition. A simple example is that the re-ranking model should never select a previously rejected dialogue move and therefore must take into account the beliefs in the information state. In Section 2.4.6 we discussed the approach proposed by Higashinaka *et al.* (2006) where Grice's maxim of quantity was taken into account by lowering the probability of user utterances involving the same concepts as the system was trying to confirm. Our feature PropRepeat was an attempt to model something similar. We could well also take this into account when regenerating DMSLMs by excluding the rejected dialogue moves. Orlandi *et al.* (2003) proposed such an approach by removing hypotheses from the recognition grammar that were semantically equivalent to the rejected hypothesis when re-recognizing. In a DMSLMs framework where DMSLM can be generated on the fly this could be achieved straightforwardly by excluding all rules for the rejected dialogue move.

Reliable confidence scores are not only useful at run-time in order to select appropriate system reactions but can also be used to automatically partition recordings for transcription. Manual transcription is a labour-intensive task. Nakano and Hazen (2003) propose a methodology where hypotheses with high word confidences can be used to feed an SLM directly without the need for manual transcription. They use the confidence scoring presented in Hazen *et al.* (2002) and an optimized threshold. With the highly accurate confidence estimation presented in this thesis it would be straightforward to select all hypotheses labelled as OPT and POS to be used as transcriptions. As we are also selecting the best choices from N-Best lists the statistical estimates will be reinforced and we would expect improved models. A more cautious approach would be to apply the implicit supervised learning strategy from Bohus and Rudnicky (2007) to this task by taking into account the grounding actions of the users. We could for example automatically select hypotheses that have been explicitly or implicitly confirmed by users in dialogue with the system for

automatic language model training.

However, without adding manual transcriptions the SLM will be biased towards what it is already good at (or doing badly on) and will not extend its vocabulary or ways of expression. In commercial applications with high call volume it is sometimes impossible to transcribe all calls and in particular it is hard to know what transcriptions will favour the model most. Nakano and Hazen (2003) propose that the order of transcription can also be based on confidence scores by selecting utterances that will boost the SLM most to be transcribed first. They selected utterances with a high number of low word confidence scores. Using our confidence model we can assume that utterances marked as IGN or CROSS are less interesting to transcribe than the ones marked as PESS or NEG. Being able to automatically discard crosstalk or utterances with a lot of noise or out-of-domain utterances would save transcribers' time dramatically. The possible contribution of our confidence model to such a cost saving would be very interesting to evaluate.

10.3 Summary and conclusions

In this chapter we have discussed possible future research based on the work in this thesis. We have also highlighted some recent research related to the work in this thesis.

We can envision many possible ways to improve the art of language modelling. There must be shortcuts that we have not yet discovered which would enable us to avoid having to wait until large amounts of data have been collected.

We believe that the use of linguistic knowledge in ASR will become more common. We have shown that there are many possibilities for finding new features and better computational representations of them. A common framework to work in would benefit the research of exploring additional knowledge sources.

We have also discussed the possibilities that appear when it comes to grounding and error-recovery with a much more reliable confidence model available like the one presented in this thesis. In addition, we have pointed out how more reliable confidence scores could also be useful for the task of automatically selecting data for SLM training data or for transcription.

In this thesis we have only been able to conduct tests using experimental systems with limited user data. Applying the ideas to large amounts of real data in real systems will be the real challenge. The most appealing future direction is therefore to put the results of this thesis to work in such systems.

Chapter 11

Conclusions

The main purpose of this thesis has been to investigate whether we can benefit from the use of higher level knowledge sources in automatic speech recognition (ASR) for spoken dialogue systems (SDSs). I have presented several experimental studies exploring different approaches to the enhancement of ASR by incorporating more linguistic knowledge. The previous chapter also discussed some possible future research directions related to the work presented in this thesis. In this concluding chapter we will recapitulate the major findings from the investigations presented in the previous chapters. A concluding discussion of the major contributions of this thesis will then follow.

11.1 Thesis summary

At the beginning of this thesis we described the complexity of spoken natural language together with the difficulties for ASR. We emphasized the lack of incorporation of linguistic knowledge in the speech recognition process specifically when recognizing speech in dialogue systems where dialogue context and syntactic, semantic and pragmatic knowledge is available. We also pointed out the chicken and egg problem involved in building initial spoken language models for SDSs in which user behaviour is unknown. On the one hand user data is necessary in order to model user language. On the other hand a first language model is needed to be able to collect user data.

In Chapter 2 a brief introduction to ASR, SDSs and the metrics for evaluating ASR in the context of SDSs was given. This was followed by a survey of previous attempts to improve ASR on different levels with the main focus on studies that have exploited the introduction of linguistic knowledge and knowledge of language in the ASR process. The parts of the survey relating to the main issues of this thesis (language modelling, N-Best re-ranking and confidence estimation) were more extensively described.

Chapter 3 introduced the information state update (ISU) approach to dialogue management with focus on the TrindiKit toolkit for ISU-based dialogue system development. A description of the most essential parts of GoDiS, an ISU-based dialogue system built with TrindiKit, was given. The two baseline systems, the GoDiS applications AgendaTalk

and DJ-GoDiS, were also presented. Additionally, a short description of the two machine learning toolkits (TiMBL and JRip) that have been used for some of the experiments in this thesis was given.

In the context of the two experimental GoDiS applications, AgendaTalk and DJ-GoDiS, we have explored ways of improving the speech recognition performance of these two systems. As these two applications make use of the ISU approach to dialogue management we have had the opportunity of exploiting the actual *information state* as an additional knowledge source for ASR. The starting point of this thesis was two applications with a loosely coupled ASR module using hand written speech recognition grammars (SRGs).

Part II of this thesis addressed the issue of rapidly obtaining better initial language models for SDSs by taking advantage of grammatical frameworks. It also explored how to take into account dialogue context in language modelling and to automatically predict when to use such models based on the information state. The purpose of this part of the thesis was to investigate approaches to enhance a SDS's initial ASR performance.

Part III explored the potential of exploiting different knowledge sources in SDSs when re-ranking N-Best lists and estimating confidence for the ASR output. The purpose in this part was to enhance a SDS's use of the ASR output by improving a dialogue system's decision-making of how to proceed and make use of ASR hypotheses. In **Part IV** we outlined the integration of the results of the experimental work from **Part II** and **Part III** into the GoDiS framework. In Chapter 10 we discussed possible future directions based on the results presented in this thesis.

This thesis has presented several experimental studies addressing the issues described above. The experimental setting was such that first a pilot experiment was carried out in the MP3 player domain (with DJ-GoDiS) followed by a more extensive and thorough experiment in the Calendar domain (with AgendaTalk). Although both of these domains are quite restricted, in this way, we aimed for minimizing domain-dependence of the results. We will here provide a brief summary of the results and contributions.

11.1.1 Enhancing a dialogue system's initial ASR performance

The three chapters in **Part II** focussed on enhancing the initial ASR performance of a SDSs.

11.1.1.1 Grammar-based SLMs

The experimental work of this thesis started in Chapter 4 by facing the problem of no available training data in order to build appropriate spoken language models. We revived an old idea of generating artificial corpora from grammars (Zue *et al.*, 1991; Jurafsky *et al.*, 1995). The novelty of our approach was to use GF interpretation grammars and generate all meaningful sentences from such grammars exhaustively. The resulting corpora were used to build trigram statistical language models (SLMs). Although the statistical estimates in such models are artificial it became clear that they are sufficiently close to be able to

improve the speech recognition performance. In the first preliminary experiment we built an SLM based on 300 000 Swedish utterances generated from the DJ-GoDiS GF grammar. In comparison with the GF grammar compiled directly into an SRG the grammar-based SLM gave a relative improvement in WER of 37%. In our second experiment the grammar-based SLM was built on a grammar-generated corpus of 1.7 million Swedish utterances covering expressions anticipated for the AgendaTalk domain. With the use of this class-based model we were able to lower the WER from 39% to 29%, a relative improvement of 26%. By measuring dialogue move error rate (DMER) (an evaluation metrics introduced in Chapter 2) we could show that this improvement also propagated well to the understanding performance (a 24% relative improvement).

We also investigated the performance of the grammar-based SLMs vs the SRGs on different types of users and data. All models performed substantially better with expert users than naive users. Expert users were revealed to keep more to the coverage of the hand written grammars and the vocabulary simplifying the recognition task. The SRG showed a slightly better performance on in-coverage data than the grammar-based SLMs which also was to be expected. However, it is our belief that ASR performance in SDSs should be evaluated on in-domain data, i.e. user expressions related to the domain, rather than on in-coverage data, i.e. user expressions anticipated by developers. On this perspective the grammar-based SLMs outperform their corresponding SRGs substantially.

In order to add more accurate statistical estimates and show how grammar-based models can be further extended we explored the use of external corpora in Swedish. At our disposal we had a Swedish news corpus and the Gothenburg spoken language corpus (GSLC). It was demonstrated that the more appropriate data the better. The news corpus was only valuable as a resource when we applied a simple domain selection technique to extract the most related parts from it. The GSLC corpus proved to be very useful by adding real estimates of spoken language patterns that helped to improve our models significantly. The SLM used in the second experiment, which was built by incorporating the GSLC corpus to the grammar-based AgendaTalk SLM, was shown to be much more robust to naive users, good on in-coverage data and able to improve the performance overall with a 17% WER reduction. In conclusion, the contribution of Chapter 4 was to demonstrate the compromise between the ease of grammar writing and the robustness of SLMs in the first stage of dialogue system development.

11.1.1.2 Dialogue move specific SLMs

With SLMs it is possible to capture the lexical context statistics of specific language uses. However, the statistical distribution in a dialogue is not static but language use is determined by the dialogue context. To take into account that the probability of a user's dialogue moves is not fixed during a dialogue we showed in Chapter 5 that the same methodology as in Chapter 4 can be used to generate contextually optimized SLMs where certain dialogue moves are more probable than others. These so called *Dialogue Move specific SLMs (DM-SLMs)* can be used at different points of a dialogue depending on contextual constraints without constraining the user. They ensure a minimal coverage of expressions for a certain

context. The use of DMSLMs on data from the AgendaTalk domain gave an important reduction in both WER (24% relative) and DMER (18% relative) in comparison to the general grammar-based model.

11.1.1.3 Dialogue move prediction

The use of context-specific models such as DMSLMs poses a new problem. How can we accurately select the most appropriate model for a certain situation? In Chapter 6 we experimented with dialogue move prediction by using machine learning to explore the usage of the information state for the prediction of user dialogue moves. The ultimate goal of dialogue move prediction in this work was to use it to switch between the DMSLMs from Chapter 5. However, the predicted dialogue move proved to be useful also as a feature in the experiments in Chapter 8.

The dialogue move predictor presented in the first prediction experiment classified 19 dialogue moves with 67% accuracy. With a majority baseline of 32% the improvement was substantial. In accordance with the experiments in Chapter 5 where we had experimented with four different DMSLMs based on the request, answer, ask and yn moves the second experiment and third experiment of dialogue move prediction explored the possibilities of a classifier that could predict only these moves. We achieved very similar results for both the AgendaTalk and DJ-GoDiS domain with an accuracy of over 70%.

11.1.1.4 Overall results for Part II

If we consider the results from Chapter 4 and 5 as a whole we obtained an overall reduction in WER of 46% (40% in DMER). We then assume an optimal method for selecting the best suited DMSLM at every moment instead of using the baseline; the compiled GF grammar. To get a more realistic figure we need of course to take into account the dialogue move prediction accuracy. Considering a dialogue move prediction accuracy of around 70% would still leave us with a substantial WER reduction of 32%. It became evident that wrongly applied DMSLMs performed well even on dialogue moves not in focus due to their unrestrictedness. This means that even if the prediction fails this does not necessarily mean that the ASR will break down. The real impact of using DMSLMs is heavily dependent on the success of the dialogue move prediction but also on the user behaviour, i.e. how much the users follow the expected patterns. Perfect dialogue move prediction is impossible as user behaviour is not totally predictable. However, the ultimate goal is overall improvement of ASR accuracy. **Part II** clearly demonstrated that it is possible to make a system more prepared for the most expected user contributions and being able to recognize these with less WER without constraining the users.

The Integration chapter (Chapter 9) presented the implementation of the dialogue move prediction strategy and DMSLM switching in the GoDiS framework. Chapter 10 pointed out that a real evaluation is needed to estimate the real impact of the use of this implementation of dialogue move prediction and DMSLM switching.

11.1.2 Enhancing a dialogue system's use of ASR output

The two chapters in **Part III** had as focus to explore how a dialogue system can make better use of the ASR output.

11.1.2.1 Dialogue move tagging

The dialogue move tagging experiment in Chapter 7 was originally a small diversion in order to get a tool for tagging N-Best lists more robustly in the subsequent experiment in Chapter 8. However, it turned out that this simple way of developing a machine-learnt tagger led to a much more robust semantic decoding than with the GF parser.

Chapter 7 demonstrated that robust dialogue move taggers can be bootstrapped by following the same methodology as in Chapter 4. Once again a GF interpretation grammar was used to generate an artificial corpus. This time the corpus was also automatically annotated with dialogue moves. Two different dialogue move taggers were developed for the DJ-GoDiS domain from a GF generated corpus of 55702 utterances marked with 3873 possible dialogue move combinations. The first tagger was utterance-based and built with the memory-based learner TiMBL. The second tagger was built with MBT and worked on the word level in a similar way to a POS tagger. We evaluated the taggers against a test set of 263 transcribed and manually annotated Swedish user utterances collected with the DJ-GoDiS system. These utterances included both words and constructions not covered by the grammar. By applying the bootstrapped utterance based tagger we achieved a substantial 34% increase in tagging accuracy in comparison to the GF grammar (from 59% to 79%). The MBT tagger yielded a 79% tagging accuracy which again boosted the perfomance by 34%. Although these taggers were not capable of capturing deep semantic relationships their interpretation capability was sufficient for interpretation within the domain and for the experiment in Chapter 8.

11.1.2.2 Confidence annotation and re-ranking

In **Part II** we pointed out that there seemed to still be room for improvement in the ASR performance in our systems considering the N-Best Error rates. We referred to techniques discussed in Section 2.4.5 where additional knowledge sources had been used to re-rank N-Best lists. Although we had managed to improve the ASR with better initial models and more context-awareness the ASR was still too unreliable to know whether its hypotheses were more or less accurate or wildly wrong. This would seriously affect our dialogue system and impair the grounding behaviour. In Chapters 5 and 6 we had seen the benefit of contextual constraints on ASR performance and the encouraging results when using information state based features for dialogue move prediction. In Chapter 8 we therefore opted to explore a better use of the output from the ASR by taking into account dialogue context and other linguistic knowledge sources. The investigation had two main directions. The preliminary focus was to explore a more reliable estimate of the ASR performance, i.e. a better confidence model. The spin-off effect was to explore the possibilities of improving

the ASR outcome by means of selecting more accurate hypotheses from N-Best lists with this new confidence estimation as guidance.

The first experiment in Chapter 8 continued the exploration of the actual benefit of dialogue context when applied to N-Best list re-ranking. This study first investigated to what extent human subjects could take advantage of dialogue context when charged with the task of selecting the most plausible hypothesis in an N-Best list. The subjects had to complete four tasks on a smaller test set: N-Best lists taken from DJ-GoDiS logs. These lists were provided with more or less dialogue context. The outcome from the experiment was that dialogue context clearly contributed to an improvement in re-ranking performance. The more dialogue context available the better the performance. The subjects managed to increase sentence accuracy (SA) and dialogue move sequence accuracy (DMSeqA) by 41 and 52 percentage points respectively. It was revealed that it was not only the immediate context that mattered but by making the previous dialogue flow available the subjects got a better idea of the dialogue situation and could more easily select the most appropriate hypothesis. From this study it was clear that humans benefit from the use of more dialogue context in ASR. However, it was unclear what information they actually made use of and how they structured it.

In the following experiment in Chapter 8 we tried to represent dialogue context computationally in order to automate this task for application in SDSs. We used TiMBL to first confidence classify and thereafter re-rank ASR hypotheses. We converted the N-Best lists with adjacent dialogue logs into 21-dimensional feature vectors. The features were chosen in an attempt to approximate the possible features the human subjects might have used. They were mainly taken from the information state. We divided the features into four feature groups in accordance with the four context levels in the experiment with human subjects. An additional fifth group was created with features representing the patterns of the N-Best list a hypothesis belonged to. Many of the human subjects had mentioned that they often considered the list as a whole to find a suitable choice. The fifth group was therefore an attempt to represent this potential knowledge source.

We labelled the training hypotheses with the following five confidence labels: Optimistic (OPT) (certainly correctly recognized), Positive (POS) (probably correctly recognized), Pessimistic (PESS) (possibly correctly recognized), Negative (NEG) (probably a misrecognition) and Ignore (IGN) (certainly a misrecognition). This was done automatically based on the similarity of a hypothesis and its transcription on both the word and conceptual level. The machine-learnt confidence classifier and ranker was then trained on this smaller amount of data and tested both in a *leave-one-out* setup and on the smaller test set representing the N-Best lists used for the human experiments. It was shown that this computational representation of dialogue context actually contributed to the task of automatic confidence classification of ASR hypotheses and subsequent re-ranking. The results indicated that the performance improved incrementally for every contextual feature group added. The 5-way classifier yielded a confidence classification accuracy of 87% in comparison to the baseline's 44%. By analysing the confusions made by the classifier it was determined that the possible false acceptance (FA) and false rejection (FR) rate when using this confidence labelling for grounding was extremely low. This indicated that using this more reliable

confidence model would evoke a much smoother dialogue flow.

An evaluation of the ranking task showed that the automatic classifier, like the human subjects, outperformed the baseline (i.e. always choosing the topmost of an N-Best list) and that the performance improved considerably when exploiting more contextual cues. In fact, the automatic classifier performed slightly better than the human subjects and reduced SER considerably in comparison to the baseline, showing an improvement of 48 percentage points. The improvement was even more if we consider the conceptual level.

As this experiment had been based on a smaller amount of data from DJ-GoDiS and evaluated on a test set with a disproportionally high amount of "re-rankable" lists, in order to exploit the possibilities of context, the aim of the next experiment was to carry out a similar experiment in the AgendaTalk domain with more data and a more proportional test set. The second experiment focussed on exploiting different levels of linguistic knowledge such as acoustic, lexical, semantic and pragmatic knowledge. The main objective was to obtain a much more solid confidence annotator much better suited to SDSs by using these knowledge sources than by only taking into account the ASR confidence score and ranking. As discussed in Section 2.4.6 a speech recognizer's confidence accuracy is crucial in order to use speech recognition successfully. We reused many of the features, including the information state based features, from the previous experiment but this time we grouped them according to their linguistic use, as acoustic, lexical, semantic and pragmatic features. Many additional features were also added resulting in a 39-dimensional feature set. This time we had a more extensive training set taken from AgendaTalk logs. We trained two classifiers using two different machine learning approaches: memory-based and rule-based. With the ambition of also exploring the possibilities of identifying crosstalk we introduced this as a sixth confidence class. In fact, the classification of crosstalk turned out to be highly accurate. In general the performance of this 6-way classifier was over expectations with a classification accuracy of 81% with FA and FR rates below 1%.

We also compared our 5-way knowledge-based classifier (excluding crosstalk) to a more simple one that used the ASR confidence scores with confidence thresholds optimized for the 5-way task as only knowledge source. Our knowledge-based model was clearly shown to be a more reliable confidence model increasing classification accuracy from 39% to 79%. When exploring the impact of the linguistic knowledge features the lexical, semantic and pragmatic features were revealed to contribute more to the task than the acoustic features. In fact, we were able to improve classification accuracy by 61% relative when adding the lexical, semantic and pragmatic features to the acoustic features. The classifier's more accurate way of giving confidence also propagated to recognition accuracy by being able to select more accurate hypotheses based on a straightforward re-ranking strategy. A relative improvement of 7.7% in Sentence Accuracy (SA) and 13% in Dialogue Move Sequence Accuracy (DMSeqA) (see Chapter 2 regarding the metrics) indicated that when there was a hypothesis in the list that could be interpreted as the same dialogue move sequence as the user's utterance our approach was relatively good at finding it. The relative improvement in Dialogue Move Accuracy (DMA) was even greater (24%) which indicated that by exploting contextual cues our classifer was able to select hypotheses that capture more of the message that the user tried to convey.

The results left no doubt that it is possible to profit from increased linguistic knowledge when assigning confidence to ASR hypotheses. The confidence model we achieved by taking into account much more than just the ASR confidence score was a significantly more accurate model. With a confidence model that is better on knowing when the ASR has recognized something correctly or has performed a misrecognition the grounding behaviour will improve significantly. We analysed the impact of this approach on grounding behaviour and could envision a more optimistic grounding behaviour with less explicit confirmations, especially less explicit confirmations of misrecognitions, and less implicit confirmations of correct recognitions. In other words, we had achieved a confidence annotation model that was much better at knowing when the recognizer was doing right or wrong. This would most certainly lead to a smoother dialogue behaviour.

Chapter 8 showed that we could obtain not only more accurate speech recognition but also a more reliable model for determining the success of the ASR process. In Chapter 9 we outlined a possible implementation in the GoDiS framework of this new confidence model and re-ranking approach. We discussed the concept of *dialogue move confidence scores* and showed how our confidence levels can be integrated with such scores in order to achieve confidence for each dialogue move instead of for each utterance (or dialogue move sequence). Moreover, we discussed how these *information state based dialogue move confidence scores* would evoke a more appropriate grounding behaviour in GoDiS and discussed the implementation of this approach. Finally, we outlined the processing steps of the confidence model and re-ranker and described how each of these processes should be implemented.

11.1.3 Knowledge sources of interest for ASR

A different perspective on the results in this thesis is given by considering what knowledge sources were found to contribute to the task of recognizing speech. The impact of the immediate dialogue context, for example the latest move, was apparent in all experiments. Although it was clear from the experiments on human subjects that they benefitted from a more extended dialogue context it was harder to prove the actual benefit of this in the computational tasks. There are probably two reasons for this: the need for a more compact and better way of representing dialogue history and the lack of training data in order to capture dialogue history patterns. In both the dialogue move prediction experiments and the confidence classification experiments, several information state features were shown to contribute significantly to the tasks. Information state features such as QUD, Shared Commitments or the Shared Plan (see Chapter 3 for an introduction to the GoDiS information state) actually represent dialogue context but in a less salient way than an explicit dialogue history. These results indicate that information state features might be a more compact and informative way of representing dialogue context. Another dilemma is how to represent dialogue moves. In the experiments we represented dialogue moves on different levels of specificity for example as dialogue move types or dialogue moves with arguments (see Section 3.3.2 for definition). By normalizing dialogue moves to dialogue move types it is possible to generalize over the data and obtain a reduction in feature

values. This facilitates the learning task. Another advantage is that the features become domain-independent so that these predictors can easily be reused in other domains. The downside is that such a generalization can lead to a loss of vital information. We might end up with features that are not commensurate with the task. Although the representation of dialogue moves should be further explored the representation in the experiments in this thesis was successful enough to show the predictive value of semantic features in the form of dialogue moves. Overall, the most important features were revealed to be on the semantic level. An example is the FreqMatch feature that represented whether a hypothesis was semantically equal to the most frequent semantic representation in its N-Best list. In fact, some relations would not have been possible to represent without taking it to a conceptual level. Also, some features would not had been such good predictors on a lexical level. An example is the immediate dialogue context that would hardly had have the same impact if we had represented utterances only by the words involved. In the confidence classification experiments it was interesting to see how little importance the actual ASR confidence score had. The linguistic features were revealed to be much more informative: for example, grammaticality.

N-Best lists from crosstalk do not show the homogeneity (in terms of repeated patterns) which is common for N-Best lists. Therefore, features representing the N-Best list as a whole, such as the variety of dialogue moves in the lists, were demonstrated to be significant for the task of identifying crosstalk. In general, the fifth group of features (see group **List** in Section 8.3.3.2) in the confidence classification experiments, which represents the patterns and characteristics of a whole N-Best list, made an important contribution to the overall performance. Our novel approach of representing such features on the semantic level was shown to be very promising.

During the thesis work we also came to adapt the GoDiS information state to include more features that were considered necessary for the experiments. We expanded the information state to keep track of confidence scores, dialogue history, referents used for anaphora resolution and the predicted dialogue move based on the implementation of the classifier in Chapter 6. Other features were extracted based on the information in the logs. This indicates that we might need a different information state and a different representation of the information that is more apt for ASR for future experiments.

11.2 Concluding remarks

This thesis proposes moving towards an information state based approach to automatic speech recognition (ASR). Such an approach implies the use of the information state and other linguistic knowledge sources to alleviate the search problem and reliability estimation in ASR. The ten experimental studies presented have demonstrated different methodologies to approach the deficiencies of ASR in order to obtain enhanced speech recognition in spoken dialogue systems (SDSs). In order to fulfill the requirement of immediately applicable techniques the focus has been on methodologies that can be applied without interfering with the internal ASR process. In this way the thesis is an attempt to determine how you

can make the most of the ASR system which is available in a SDS.

As ASR vendors do not provide packaged solutions speech developers will need to provide SLMs and grammars themselves. Therefore, ways to overcome ASR deficiencies will be necessary and ways to bootstrap SLMs indispensable. In this thesis we have demonstrated how we can combine the art of grammar writing with the power of statistics by bootstrapping SLMs for SDSs from grammars written using the Grammatical Framework (GF). It should be clear that in no way are we proposing grammar-based SLMs as the final models to use in a SDS. What we propose is a methodology to bootstrap preliminary SLMs based on interpretation grammars to get a better starting point when no or little training data exists. We suggest that this approach gives a better start than with grammars alone and is much more straightforward and cheaper than collecting WOz data. What we then encourage is to incorporate data from other sources to these models as well as use these models to collect real data. When enough data has been collected grammar-based SLMs should be converted into more realistically estimated ones.

With DMSLMs we have demonstrated how we can use the same approach also to build context-aware models that improves the performance further but without restricting the user. This reaffirms that taking into account statistical language variation during a dialogue will give us more accurate recognition. The method we use here has the advantage that we can build statistical context-specific models even when no data is available, ensuring a minimal coverage. Using interpolation with a general model we do not need to constrain the users' language use with an overly restrictive context-specific model. The dialogue move prediction approach indicates that it is possible to automatically predict which context-aware models to use instead of manually predicting this with rules. This is especially important in ISU-based systems where the number of states are not enumerated or easily predicted. The practical outcome of this first series of experiments is an implementation of the proposed approach as a new GoDiS module.

Another important outcome of this thesis is a novel ASR confidence model which achieves very high reliability by taking advantage of higher level knowledge. We maintain that a more reliable confidence model is critical for a SDS and will improve both its grounding behaviour and its ASR performance.

One of the major contributions of this thesis has been to bring to light the benefit of taking into account additional knowledge sources in ASR processing. The intention was to provide more insight into what type of knowledge sources in SDSs would be potential contributors to the task of ASR. In particular we have exploited the use of the information state and revealed that the knowledge encoded in the information state is indeed a valuable resource. We have proved that both humans and machines benefit from contextual cues in dialogue and that other linguistic knowledge, in particular semantic knowledge should not be neglected. The experiments in this thesis have exemplified how such knowledge can be represented computationally. The results suggest that higher level linguistic knowledge matters and that its computational representation should be further explored.

As the approach presented has been developed in the ISU-based framework most of the methods are easily applicable to any ISU-based system. Furthermore, many of the ideas and knowledge sources used are not bound to the ISU-based approach, nor to the domains

of the experimental baseline systems, and are therefore perfectly applicable to any SDS. We conclude that by not treating ASR as a separate module but instead coupling it more tightly to the dialogue system, that is, to the information state, will lead to more successful spoken language understanding.

We hope that the results of this thesis will encourage more research on how to make better use of additional knowledge sources in ASR. The survey in Chapter 2 showed that the use of additional knowledge sources for ASR is still in its infancy. In this thesis we have tried to overcome some of the problems that arise when using ASR for SDSs and have revealed the benefit of taking into account the information available in a SDS. However, as discussed already in the survey and later based on the findings in this thesis it will be necessary also to overcome some of the causes of breakdown in ASR such as brittle end-pointing, poor robustness to noise, extralinguistic sounds, too simple turn-taking when applying barge-in or even bad adjustments or use of microphones. Without a proper treatment of these problems breakdowns will occur at such an early stage in ASR processing that it will be hard to recover from them whatever knowledge sources are used.

In spite of this we envision a future architecture of SDSs that goes away from the picture in Figure 2.3 (on page 24) and presents a much more tightly coupled approach. In such an architecture ASR, SLU and dialogue management would not be isolated processes but concurrent intertwined processes that share information. Dialogue designers would not any longer see ASR as a decoupled black box but a provider of possibilities. In a similar manner, speech recognition engineers would not avoid taking advantage of all the available information in dialogue systems. It is only with such an approach that we will be able to explore the real potential of these technologies in combination and start to develop real spoken dialogue systems and not just adding speech interfaces to text-based dialogue systems.

Bibliography

Adda-Decker, Martine and Lori Lamel (2005) Do Speech Recognizers Prefer Female Speakers?, in *Proceedings of Interspeech 2005 – Eurospeech*, pp. 2205–2208, Lisbon, Portugal.

Akbacak, Murat, Yuqing Gao, Liang Gu and Hong-Kwang Jeff Kuo (2005) Rapid Transition to New Spoken Dialogue Domains: Language Model Training Using Knowledge from Previous Domain Applications and Web Text Resources, in *Proceedings of Interspeech 2005 – Eurospeech*, pp. 1873–1876, Lisbon, Portugal.

Allen, James F. (1991) Discourse Structure in the TRAINS Project, in *Proceedings of the Speech and Natural Language Workshop*, pp. 325–330, Pacific Grove, CA.

Allwood, Jens (1981) On the Distinction between Semantics and Pragmatics, in W. Klein and W. Levelt (eds.), *Crossing the Boundaries in Linguistics*, pp. 177–189, D. Reidel Publishing Company.

Allwood, Jens (1998) Some Frequency Based Differences between Spoken and Written Swedish, in *Papers from the 16th Scandinavian Conference of Linguistics*, p. 18, Turku, Finland.

Allwood, Jens (1999) The Swedish Spoken Language Corpus at Göteborg University, in *Fonetik 99, Gothenburg Papers in Theoretical Linguistics 81*, Dept. of Linguistics, University of Gothenburg.

Allwood, Jens (2000) An Activity Based Approach to Pragmatics, in H. Bunt and W. Black (eds.), *Abduction, Belief and Context in Dialogue: Studies in Computational Pragmatics*, pp. 47–80, John Benjamins.

Ananthakrishnan, Sankaranarayanan and Shrikanth Narayanan (2007) Improved Speech recognition using Acoustic and Lexical correlates of Pitch Accent in a N-Best rescoring framework, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 4, pp. 873–876, Honolulu, Hawaii.

Ang, Jeremy, Rajdip Dhillon, Ashley Krupski, Elizabeth Shriberg and Andreas Stolcke (2002) Prosody-Based Automatic Detection of Annoyance and Frustration in Human-Computer Dialog, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pp. 2037–2040, Denver, CO.

Ang, Jeremy, Yan Liu and Elizabeth Shriberg (2005) Automatic Dialog Act Segmentation and Classification in Multiparty meetings, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1, pp. 1061–1064, Philadelphia, PA.

Aust, Harald, Martin Oerder, Frank Seide and Volker Steinbiss (1995) The Philips automatic train timetable information system, *Speech Communication*, Vol. 17, No. 3-4, pp. 249–262.

Baggia, Paolo, Elisabetta Gerbino, Egidio Giachin, Claudio Rullent and Cselt Centro (1991) Efficient Representation of Linguistic Knowledge for Continuous Speech Understanding, in *Proceedings of IJCAI-91*, pp. 653–677, World Scientific Publishing Company, Sydney, Australia.

Baggia, Paolo, Danieli Morena, Elisabetta Gerbino, Loreta Moisa and Cosmin Popovici (1997) Contextual Information and Specific Language Models for Spoken Language Understanding, in *Proceedings of SPECOM, 2nd International Conference "Speech and Computer"*, pp. 51–56, Cluj-Napoca, Romania.

Balakrishna, Mithun, Dan Moldovan and Ellis Cave (2006) N-Best list reranking using higher level phonetic, lexical, syntactic and semantic knowledge sources, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1, pp. 413–416, Tolouse, France.

Bangalore, Srinivas and Michael Johnston (2004) Balancing Data-Driven and Rule-Based Approaches in the Context of a Multimodal Conversational System, in *Proceedings of HLT-NAACL*, pp. 33–40, Boston, MA.

Becker, Tilman, Peter Poller, Staffan Larsson, Oliver Lemon, Guillermo Pérez, Jan Schehl and Karl Weilhammer (2006a) Software Infrastructure, Deliverable D5.1, TALK Project. http://www.talk-project.org/fileadmin/talk/publications_public/deliverables_public/D5_1_2.pdf.

Becker, Tilman, Peter Poller, Jan Schehl, Nate Blaylock, Ciprian Gerstenberger and Ivana Kruijff-Korbayová (2006b) The SAMMIE system: multimodal in-car dialogue, in *Proceedings of the COLING/ACL on Interactive presentation sessions*, pp. 57–60, Association for Computational Linguistics, Morristown, NJ.

Bellegarda, Jerome R (1998) Multi-Span statistical language modeling for large vocabulary speech recognition, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pp. 2395–2399, Sydney, Australia.

Berton, André, Pablo Fetter and Peter Regel-Brietzmann (1996) Compound words in large-vocabulary German speech recognition systems, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pp. 1165–1168, Philadelphia, PA.

Bilmes, Jeff (2003) Buried Markov Models: A Graphical-Modeling approach to Automatic Speech Recognition, *Computer, Speech and Language*, Vol. 17, No. 2–3, pp. 213–231.

Bilmes, Jeff (2004) What HMMs Can't do, Invited paper and lecture at ATR Workshop "Beyond HMMs", Kyoto, Japan. Available at: http://ssli.ee.washington.edu/people/bilmes/mypapers/whathmmscannotdo04.pdf.

Bohus, Dan (2007) *Error Awareness and Recovery in Task-Oriented Spoken Dialog Systems*, PhD dissertation, Carnegie Mellon University, Pittsburgh.

Bohus, Dan and Alexander Rudnicky (2005a) A principled approach for rejection threshold optimization in spoken dialog systems, in *Proceedings of Interspeech 2005 – Eurospeech*, pp. 2781–2784, Lisbon, Portugal.

Bohus, Dan and Alexander Rudnicky (2005b) Sorry, I Didnt Catch That! - An Investigation of Non-Understanding Errors and Recovery Strategies, in *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue*, pp. 128–143, Lisbon, Portugal.

Bohus, Dan and Alexander Rudnicky (2007) Implicitly-supervised learning in spoken language interfaces: an application to the confidence annotation problem, in *Proceedings of the 8th SIGDial Workshop on Discourse and Dialogue*, pp. 256–264, Antwerp, Belgium.

Boros, Manuela, Wieland Eckert, Florian Gallwitz, Günther Görz, Gerhard Hanrieder and Heinrich Niemann (1996) Towards Understanding Spontaneous Speech: Word Accuracy vs. Concept Accuracy, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pp. 1009–1012, Philadelphia, PA.

Bos, Johan, Ewan Klein, Oliver Lemon and Tetsushi Oka (2003) DIPPER: Description and Formalisation of an Information-State Update Dialogue System Architecture, in *Proceedings of the 4th SIGdial Workshop on Discourse and Dialogue*, pp. 119–122, Sapporo, Japan.

Bousquet-Vernhettes, Caroline and Nadine Vigouroux (2003) Recognition Error Handling by the Speech Understanding System to Improve Spoken Dialogue Systems, in *Proceedings of ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems*, pp. 113–118, Chateau-d'Oex-Vaud, Switzerland.

Boye, Johan, Joakim Gustafson and Mats Wirén (2006) Robust spoken language understanding in a computer game, *Speech Communication*, Vol. 48, No. 3–4, pp. 335–353.

Bradlow, Ann R and Tessa Bent (2002) The clear speech effect for non-native listeners, *Journal of Acoustic Soc Am.*, Vol. 112, pp. 272–284.

Brennan, Susan E. (1996) Lexical Entrainment in Spontaneous Dialog, in *International Symposium on Spoken Dialog*, pp. 41–44, Philadelphia, PA.

Brill, Eric, Radu Florian, John C. Henderson and Lidia Mangu (1998) Beyond N-Grams: Can Linguistic Sophistication Improve Language Modeling?, in C. Boitet and P. Whitelock (eds.), *Proceedings of ACL and COLING*, pp. 186–190, Morgan Kaufmann Publishers, San Francisco, CA.

Bringert, Björn (2008) *Programming Language Techniques for Natural Language Applications*, PhD dissertation, Chalmers University of Technology and University of Gothenburg, Sweden.

Bringert, Björn, Robin Cooper, Peter Ljunglöf and Aarne Ranta (2005) Development of multimodal and multilingual grammars: viability and motivation, Deliverable D1.2a, TALK Project. http://www.talk-project.org/fileadmin/talk/publications_public/deliverables_public/TK_D1-2-1.pdf.

Brown, Peter F., Vincent J. Della Pietra, Peter V. Desouza, Jennifer C. Lai and Robert L. Mercer (1992) Class-Based N-gram Models of Natural Language, *Computational Linguistics*, Vol. 18, No. 4, pp. 467–479.

Bulyko, Ivan and Mari Ostendorf (2003) Getting More Mileage from Web Text Sources for Conversational Speech Language Modeling using Class-dependent Mixtures, in *Proceedings of HLT-NAACL 2003*, Vol. 2, pp. 7–9, Edmonton, Canada.

Burke, Carl, Christy Doran, Abigail Gertner, Andy Gregorowicz, Lisa Harper, Joel Korb and Dan Loehr (2003) Dialogue complexity with portability? Research directions for the Information State approach, in *Proceedings of the HLT-NAACL 2003 Workshop on Research Directions in Dialogue Processing*, Vol. 7, pp. 13–15, Edmonton, Canada.

Carletta, Jean (1996) Assessing agreement on classification tasks: the Kappa Statistic, *Computational Linguistics*, Vol. 22, pp. 249–254.

Carpenter, Paul, Chun Jin, Daniel Wilson, Rong Zhang, Dan Bohus and Alexander Rudnicky (2001) Is this conversation on track?, in *Proceedings of Eurospeech*, pp. 2121–2124, Aalborg, Denmark.

Carter, David, Jaan Kaja, Leonardo Neumeyer, Manny Rayner, Fuliang Weng, Mats Wirén and Millers Yard (1996) Handling Compound Nouns in a Swedish Speech-Understanding System, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pp. 26–29, Philadelphia, PA.

Chelba, Ciprian and Frederick Jelinek (1999) Recognition Performance of a Structured Language Model, in *Proceedings of Eurospeech*, pp. 1567–1570, Budapest, Hungary.

Chen, Stanly F. and Joshua T. Goodman (1999) An Empirical Study of Smoothing Techniques for Language Modeling, *Computer Speech and Language*, Vol. 13, pp. 359–397.

Chotimongkol, Ananlada and Alexander Rudnicky (2001) N-Best Speech Hypotheses Reordering Using Linear Regression, in *Proceedings of Eurospeech 2001*, pp. 1829–1832, Aalborg, Denmark.

Chung, Grace and Stephanie Seneff (1998) Improvements in Speech Understanding Accuracy Through the Integration of Hierarchical Linguistic, Prosodic and Phonological Constraints in the Jupiter Domain, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pp. 935–939, Sydney, Australia.

Clark, Hebert H. and Susan E. Brennan (1991) Grounding in communication, in L. B. Resnick, J. M. Levine and S. D. Teasley (eds.), *Perspectives on socially shared cognition*, pp. 127–149, APA Books, Washington, DC.

Clark, Herbert H. and Edward F. Schaefer (1989) Contributing to discourse, *Cognitive Science*, Vol. 13, No. 2, pp. 259–294.

Clark, Herbert H. and Deanna Wilkes-Gibbs (1986) Referring as a collaborative process, *Cognition*, Vol. 22, No. 1, pp. 1–39.

Clarkson, P.R. and Ronald Rosenfeld (1997) Statistical Language Modeling Using the CMU-Cambridge Toolkit, in *Proceedings of Eurospeech*, pp. 2707–2710, Rhodes, Greece.

Cohen, Michael Harris, James P. Giangola and Jennifer Balogh (2004) *Voice User Interface Design*, Addison-Wesley Professional.

Cooper, Robin and Staffan Larsson (2009) Compositional and ontological semantics in learning from corrective feedback and explicit definition, in E. et. al. (ed.), *Proceedings of DiaHolmia, 2009 Workshop on the Semantics and Pragmatics of Dialogue*, Stockholm, Sweden.

Creutz, Mathias, Teemu Hirsimaki, Mikko Kurimo, Antti Puurula, Janne Pylkkonen, Vesa Siivola, Matti Varjokallio, Ebru Arisoy, Murat Saraclar and Andreas Stolcke (2007) Morph-Based Speech Recognition and Modeling of Out-of-Vocabulary Words Across Languages, *ACM Transactions on Speech and Language Processing*, Vol. 5, No. 1, p. 3.

Daelemans, Walter, Jakub Zavrel, Antal van den Bosch and Ko van der Sloot (2003) MBT: Memory Based Tagger, version 2.0, Reference Guide. http://ilk.uvt.nl/mbt/.

Daelemans, Walter, Jakub Zavrel, Ko van der Sloot and Antal van den Bosch (2001) TiMBL: Tilburg Memory-Based Learner - version 4.0 Reference Guide. http://ilk.uvt.nl/timbl/.

Doddington, George, Walter Liggett and Mark Przybocki (1998) Sheep, goats, lambs and wolves: A statistical analysis of speaker performance in the NIST 1998 speaker recognition evaluation, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, Sydney, Australia. Paper 0608.

Dzikovska, Myroslava O., Charles B. Callaway, Elaine Farrow, Manuel Marques-Pita, Colin Matheson and Johanna D. Moore (2007) Adaptive Tutorial Dialogue Systems Using Deep NLP Techniques, in *Proceedings of the 2007 Meeting of NAACL and HLT, Demo Session*, pp. 5–6, Rochester, NY.

Eckert, Wieland, Florian Gallwitz and Heinrich Niemann (1996) Combining stochastic and linguistic language models for recognition of spontaneous speech, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 423–426, Atlanta, Georgia.

Eklund, Robert and Elisabeth Shriberg (1998) Crosslinguistic Disfluency Modeling: A comparative Analysis of Swedish and American English Human–Human and Human–Machine Dialogs, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pp. 2631–2634, Sydney, Australia.

Ericsson, Stina, Gabriel Amores, Björn Bringert, Håkan Burden, Ann-Charlotte Forslund, David Hjelm, Rebecca Jonsson, Staffan Larsson, Peter Ljunglöf, Pilar Manchón, David Milward, Guillermo Pérez and Mikael Sandin (2006a) Software illustrating a unified approach to multimodality and multilinguality in the in-home domain, Deliverable D1.6, TALK Project. http://www.talk-project.org/fileadmin/talk/publications_public/deliverables_public/D1_6.pdf.

Ericsson, Stina, Ciprian Gerstenberger, Pilar Manchón and Jan Schehl (editor) (2006b) Plan library for multimodal turn planning, Deliverable D3.2, TALK Project. http://www.talk-project.org/fileadmin/talk/publications_public/deliverables_public/D3_2.pdf.

Fügen, Christian, Hartwig Holzapfel and Alex Waibel (2004) Tight coupling of speech recognition and dialog management - dialog-context dependent grammar weighting for speech recognition, in *Proceedings of Interspeech 2004 and 8th International Conference on Spoken Language Processing (ICSLP)*, pp. 169–172, Jeju Island, South Korea.

Frankel, Joe, Mathew Magimai-Doss, Simon King, Karen Livescu and Özgür Çetin (2007) Articulatory Feature Classifiers Trained on 2000 hours of Telephone Speech, in *Proceedings of Interspeech 2007*, pp. 2485–2488, Antwerp, Belgium.

Gabsdil, Malte and Johan Bos (2003) Combining Acoustic Confidence Scores with Deep Semantic Analysis for Clarification Dialogues, in *Proceedings of International Workshop on Computational Semantics (IWCS)*, pp. 137–150, Tilburg, The Netherlands.

Gabsdil, Malte and Oliver Lemon (2004) Combining Acoustic and Pragmatic Features to Predict Recognition Performance in Spoken Dialogue Systems, in *Proceedings of ACL*, pp. 343–350, Barcelona, Spain.

Galescu, Lucian, Eric Ringger and James Allen (1998) Rapid language model development for new task domains, in *Proceedings of the ELRA First International Conference on Language Resources and Evaluation (LREC)*, pp. 807–812, Granada, Spain.

Gallwitz, Florian, Elmar Nöth and Heinrich Niemann (1996) A category based approach for recognition of out-of-vocabulary words, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pp. 228–231, Philadelphia, PA.

Gao, Jianfeng, Mingjing Li and Kai fu Lee (2000) N-gram distribution based language model adaptation, in *Proceedings of International Conference on Spoken Language Processing (ICSLP 2000)*, pp. 497–500, Beijing China.

Georgescul, Maria, Manny Rayner, Pierrette Bouillon and Nikos Tsourakis (2008) Discriminative Learning using Linguistic Features to Rescore N-Best Speech hypotheses, in *Proceedings of Spoken Language Technology Workshop*, pp. 97 – 100, Goa, India.

Georgila, Kallirroi, James Henderson and Oliver Lemon (2005) Learning User Simulations for Information State Update Dialogue Systems, in *Proceedings of Interspeech – Eurospeech*, pp. 893–896, Lisbon, Portugal.

Ginzburg, Jonathan (1996) Interrogatives: Questions, Facts and Dialogue, in S. Lappin (ed.), *The Handbook of Contemporary Semantic Theory*, pp. 385–422, Blackwell, Oxford.

Ginzburg, Jonathan (Forth) *The Interactive Stance: Meaning for Conversation*, Forthcoming book, available at: http://www.dcs.kcl.ac.uk/staff/ginzburg/papers-new.html.

Glass, James (1999) Challenges for Spoken Dialogue Systems, in *Proceedings of IEEE ASRU Workshop*, pp. 307–310, Keystone, CO.

Godfrey, John J. and Edward Holliman (1997) Switchboard-1 Release 2, Linguistic Data Consortium, Philadelphia.

Godfrey, John J., Edward C. Holliman and J. McDaniel (1992) SWITCHBOARD: telephone speech corpus for research and development, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1, pp. 517–520, San Francisco, CA.

Gold, Ben and Nelson Morgan (2000) *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*, John Wiley and Sons, Inc.

Goldwater, Sharon, Dan Jurafsky and Christopher D. Manning (2008) Which Words Are Hard to Recognize? Prosodic, Lexical, and Disfluency Factors that Increase ASR Error Rates, in *Proceedings of ACL-08: HLT*, pp. 380–388, Association for Computational Linguistics, Columbus, Ohio.

Gorrell, Genevieve (2003) Using statistical language modelling to identify new vocabulary in a grammar-based speech recognition system, in *Proceedings of Eurospeech*, pp. 2729–2732, Geneva, Switzerland.

Gorrell, Genevieve (2006) Generalized Hebbian Algorithm for Incremental Singular Value Decomposition in Natural Language Processing, in *Proceedings of EACL*, pp. 97–104, Trento, Italy.

Gorrell, Genevieve (2007) *Generalized Hebbian Algorithm for Dimensionality Reduction in Natural Language Processing*, PhD dissertation, Linköping University, Sweden.

Gorrell, Genevieve, Ian Lewin and Manny Rayner (2002) Adding Intelligent Help to Mixed Initiative Spoken Dialogue Systems, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pp. 2065–2068, Denver, CO.

Gorrell, Genevieve and Brandyn Webb (2005) Generalized Hebbian Algorithm for Latent Semantic Analysis, in *Proceedings of Interspeech 2005 – Eurospeech*, pp. 1325–1328, Lisbon, Portugal.

Grice, Herbert Paul (1975) Logic and Conversation, in P. Cole and J. L. Morgan (eds.), *Syntax and Semantics*, Vol. 3, pp. 41–58, Academic Press.

Grönqvist, Leif (2006) *Exploring Latent Semantic Vector Models Enriched With N-grams*, PhD dissertation, Växjö University, Sweden.

Grosz, Barbara and Candace Sidner (1986) Attention, Intentions, and the Structure of Discourse, *Computational Linguistics*, Vol. 12, No. 3, pp. 175–204.

Gruenstein, Alexander (2008) Response-based Confidence Annotation for Spoken Dialogue Systems, in *Proceedings of the 9th SIGDial Workshop on Discourse and Dialogue*, pp. 11–20, Columbus, Ohio.

Gruenstein, Alexander, Chao Wang and Stephanie Seneff. (2005) Context-sensitive statistical language modeling, in *Proceedings of Interspeech 2005 – Eurospeech*, pp. 17–20, Lisbon, Portugal.

Guillevic, Didier, Simona Gandrabur and Yves Normandin (2002) Robust Semantic Confidence Scoring, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pp. 853–856, Denver, CO.

Gurevych, Iryna and Robert Porzel (2003) Using knowledge-based scores for identifying best speech recognition hypothesis, in *Proceedings of ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems*, pp. 77–81, Chateau-d'Oex-Vaud, Switzerland.

Gustafsson, Joakim, Linda Beskow, Johan Boye, Rolf Carlson, Jens Edlund, Björn Granström, David House and Mats Wirén (2000) AdApt - a multimodal onversational dialogue system in an apartment domain, in *Proceedings of International Conference on Spoken Language Processing (ICSLP 2000)*, pp. 134–137, Beijing, China.

Hacioglu, Kadri and Wayne Ward (2001) Dialog-context dependent language modeling combining n-grams and stochastic context-free grammars, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1, pp. 537–540, Salt Lake City, Utah.

Hazen, Timothy J. and Issam Bazzi (2001) A Comparison and Combination of Methods for OOV Word Detection and Word Confidence Scoring, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 397–400, Salt Lake City, Utah.

Hazen, Timothy J., Joseph Polifroni and Stephanie Seneff (2002) Recognition confidence scoring for use in speech understanding systems, *Computer Speech and Language*, Vol. 16(1), pp. 49–67.

Henderson, James, Oliver Lemon and Kallirroi Georgila (2005) Hybrid Reinforcement/Supervised Learning for Dialogue Policies from COMMUNICATOR Data, in *Proceedings of IJCAI workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pp. 68–75, Edinburgh.

Hermansky, Hynek (1998) Should recognizers have ears?, *Speech Communication*, Vol. 25, No. 1–3, pp. 3–27.

Hernandez-Abrego, Gustavo and Jose Marino (2000) Contextual Confidence Measures for Continuous Speech Recognition, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. III, pp. 1803–1806, Istanbul, Turkey.

Higashinaka, Ryuichiro, Katsuhito Sudoh and Mikio Nagano (2005) Incorporating discourse features into confidence scoring of intention recognition results in spoken dialogue systems, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 25–28, Philadelphia, PA.

Higashinaka, Ryuichiro, Katsuhito Sudoh and Mikio Nagano (2006) Incorporating discourse features into confidence scoring of intention recognition results in spoken dialogue systems, *Speech Communication*, Vol. 48, No. 3–4, pp. 417–436.

Hirschberg, Julia, Diane Litman and Marc Swerts (2000) Generalizing prosodic prediction of speech recognition errors, in *Proceedings of International Conference of Spoken Language Processing (ICSLP 2000)*, pp. 254–257, Beijing China.

Hjelm, David, Ann-Charlotte Forslund, Staffan Larsson and Andreas Wallentin (2005) DJ GoDiS: Multimodal Menu-based Dialogue in an Asynchronous Information State Update System, in *Proceedings of DIALOR'05: 9th workshop on the semantics and pragmatics of dialogue*, pp. 159–163, Nancy, France.

Hockey, Beth Ann and Manny Rayner (2005) Comparison of Grammar Based and Statistical Language Models Trained on the Same Data, in *Proceedings of the the AAAI Workshop on Spoken Language Understanding*, Pittsburgh, PA. Available at: http://ti.arc.nasa.gov/m/pub/1001h/1001

Hockey, Beth Ann, Manny Rayner and Gwen Christian (2008) Training Statistical Language Models from Grammar-Generated Data: A Comparative Case-Study, in B. Nordström and A. Ranta (eds.), *GoTAL* (*Lecture Notes in Computer Science* 5221), pp. 193–204, Springer.

Holzapfel, Hartwig and Alex Waibel (2006) A Multilingual Expectations model for Contextual Utterances in Mixed-Initiative Spoken Dialogue, in *Proceedings of Interspeech 2006 - ICSLP Ninth International Conference on Spoken Language Processing*, Pittsburgh, PA. Paper 1614.

Huang, X., Alex Acero and H-W Hon (2001) *Spoken Language Processing: A guide to theory, algorithm and system development*, Prentice Hall.

Janiszek, David, Renato De Mori and Frederic Bechet (2001) Data Augmentation And Language Model Adaptation, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1, pp. 549–552, Salt Lake City, Utah.

Jelinek, Frederick (1991) UP FROM TRIGRAMS! The struggle for improved language models, in *Proceedings of Eurospeech 91*, pp. 1037–1040, Genova, Italy.

Jelinek, Frederick (1997) *Statistical Methods of Speech Recognition*, 3rd edition, MIT Press.

Jelinek, Frederick and Ciprian Chelba (1999) Putting language into language modeling, in *Proceedings of Eurospeech*, p. Keynote Paper 1, Budapest, Hungary.

Ji, Gang and Jeff Bilmes (2005) Dialog Act Tagging using Graphical Models, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1, pp. 33–36, Philadelphia, PA.

Jiang, Hui (2005) Confidence measures for speech recognition: A survey, *Speech Communication*, Vol. 45, No. 4, pp. 455–470.

Jokinen, Kristina (2009) *New Trends in Speech Based Interactive Systems*, Springer Science+Business Media, Inc.

Jonson, Rebecca (2000) Agenda Talk: A Talking Filofax Developed with the TrindiKit toolkit. Master's thesis, University of Gothenburg.

Jonson, Rebecca (2006a) Dialogue Context-Based Reranking of ASR Hypotheses, in *Proceedings of Spoken Language Technology Workshop*, pp. 174 – 177, Palm Beach, Aruba.

Jonson, Rebecca (2006b) Generating statistical language models from interpretation grammars in dialogue systems, in *Proceedings of EACL*, pp. 57–65, Trento, Italy.

Jonson, Rebecca (2007) Grammar-based context-specific statistical language modelling, in *Proceedings of the ACL Workshop on Grammar-Based Approaches to Spoken Language Processing (SpeechGram)*, pp. 25–32, Prague, Czech Republic.

Jurafsky, Daniel and James H. Martin (2008) *Speech and Language Processing*, Prentice Hall.

Jurafsky, Daniel, Chuck Wooters, Jonathan Segal, Andreas Stolcke, Eric Fosler, G. Tajchaman and Nelson Morgan (1995) Using a stochastic context-free grammar as a language model for speech recognition, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1, pp. 189–192, IEEE Computer Society, Detroit, Michigan.

Kellner, A. (1998) Initial language models for spoken dialogue systems, in *Proceedings of ICASSP'98*, pp. 185–188, Seattle,WA.

Kemp, Thomas and A. Jusek (1996) Modelling unknown words in spontaneous speech, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 530–533, Atlanta, Georgia.

Kim, Kyungduk, Minwoo Jeong and Gary Geunbae Lee (2007) Improving speech recognition using semantic and reference features in a multi-modal dialog system, in *Proceedings of the 16th IEEE International Symposium on Robot and Human Interactive Communication (IEEE RO-MAN 2007)*, pp. 416–420, Jeju Island, South Korea.

Knight, Sylvia, Genevieve Gorrell, Manny Rayner, David Milward, Rob Koeling and Ian Lewin (2001) Comparing Grammar-Based and Robust Approaches to Speech Understanding: A Case Study, in *Proceedings of Eurospeech*, pp. 1779–1782, Aalborg, Denmark.

Kruijff-Korbayová, Ivana, Gabriel Amores, Johan Bockgård, Stina Ericsson, Ciprian Gerstenberger, Rebecca Jonson, Oliver Lemon, Pilar Manchón, David Milward, Peter Poller, Aarne Ranta and Jan Schehl (2006) Modality–Specific Resources for Presentation, Deliverable D3.3, TALK Project. http://www.talk-project.org/fileadmin/talk/publications_public/deliverables_public/D3_3.pdf.

Kuhn, Roland and Renato De Mori (1990) A Cache-Based Natural Language Model for Speech Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 6, pp. 570–583.

Lager, Torbjörn and Natalia Zinovjeva (1999) Training a Dialogue Act Tagger with the u-TBL system, in *Proceedings of the Third Swedish Symposium on Multimodal Communication*, Linköping, Sweden. http://www.ida.liu.se/~ssomc/papers/Lager.pdf.

Larsson, Staffan (2002) *Issue-based Dialogue Management*, PhD dissertation, University of Gothenburg.

Larsson, Staffan, Alexander Berman, Jacob Hallenborg and David Hjelm (2004) TrindiKit 3.1 Manual. Department of Linguistics, University of Gothenburg. Version June 3, 2004.

Larsson, Staffan and Robin Cooper (2009) Towards a formal view of corrective feedback, in *Proceedings of the EACL Workshop on Cognitive Aspects of Computational Language Acquisition*, pp. 1–9, Athens, Greece.

Larsson, Staffan, Peter Ljunglöf, Robin Cooper, Elisabet Engdahl and Stina Ericsson (2000) GoDiS - An Accommodating Dialogue System, in *Proceedings of ANLP/NAACL-2000 Workshop on Conversational Systems*, pp. 7–10, Seattle, Washington.

Larsson, Staffan and David Traum (2000) Information State and Dialogue Management in the TRINDI Dialogue Move Engine Toolkit, *Natural Language Engineering*, Vol. 6, pp. 323–340.

Lemon, Oliver (2004) Context-sensitive speech recognition in ISU dialogue systems: results for the grammar switching approach, in *Proceedings of CATALOG, 8th Workshop on the Semantics and Pragmatics of Dialogue*, pp. 49–55, Barcelona, Spain.

Lemon, Oliver, Anne Bracy, Alexander Gruenstein and Stanley Peters (2001a) Information states in a multi-modal dialogue system for human-robot conversation, in *Proceedings of the 5th Workshop on Formal Semantics and Pragmatics of Dialogue (Bi-Dialog)*, pp. 57–67, Bielefeld, Germany.

Lemon, Oliver, Anne Bracy, Alexander Gruenstein and Stanley Peters (2001b) The WITAS Multi-Modal Dialogue System I, in *Proceedings of Eurospeech*, pp. 1559–1562, Aalborg, Denmark.

Lemon, Oliver and Alexander Gruenstein (2004) Multithreaded context for robust conversational interfaces: Context-sensitive speech recognition and interpretation of corrective fragments, *ACM Transactions on Computer-Human Interaction (TOCHI)*, Vol. 11, No. 3, pp. 241–267.

Lemon, Oliver and Ioannis Konstas (2009) User Simulations for Context-Sensitive Speech Recognition in Spoken Dialogue Systems, in *Proceedings of EACL*, pp. 505–513, Association for Computational Linguistics, Athens, Greece.

Lendvai, Piroska, Antal Van den Bosch, Emiel Krahmer and Sander Canisius (2004) Memory-based Robust Interpretation of Recognised Speech, in *Proceedings of SPECOM*

'04, 9th International Conference "Speech and Computer", pp. 415–422, St. Petersburg, Russia.

Lesser, Victor R., Richard D Fennell, Lee D Erman and D Raj Reddy (1975) Organization of the Hearsay-II Speech Understanding System, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, No. 1, pp. 11–24.

Levinson, Stephen (1983) *Pragmatics*, Cambridge University Press, Cambridge.

Levow, Gina-Anne (1998) Characterizing and recognizing spoken corrections in human-computer dialogue, in *Proceedings of ACL and COLING*, pp. 736–742, San Francisco, CA.

Lewis, David (1979) Scorekeeping in a Language Game, *Journal of Philosophical Logic*, Vol. 8, pp. 339–359.

Lieberman, Henry, Alexander Faaborg, Waseem Daher and José Espinosa (2005) How to wreck a nice beach you sing calm incense, in *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces*, pp. 278–280, ACM Press, New York, NY.

Lindblom, Björn (1990) Explaining phonetic variation: a sketch of H&H theory, in W. Hardcastle and M. A. (eds.), *Speech production and Speech modeling*, pp. 403–439, Kluwer.

Lippmann, Richard (1997) Speech recognition by machines and humans, *Speech Communication*, Vol. 22, No. 1, pp. 1–16.

Litman, Diane, Marylin Walker and M. Kearns (1999) Automatic detection of poor speech recognition at the dialogue level, in *Proceedings of ACL*, pp. 309–316, Maryland.

Litman, Diane J., Julia B. Hirschberg and Marc Swerts (2000) Predicting automatic speech recognition performance using prosodic cues, in *Proceedings of the 1st Annual Meeting of the North American Association for Computational Linguistic*, pp. 218–225, Morgan Kaufmann Publishers Inc., San Francisco, CA.

Liu, Yang (2003) Automatic disfluency identification in conversational speech using multiple knowledge sources, in *Proceedings of Eurospeech*, pp. 957–960, Geneva, Switzerland.

Livescu, Karen, James Glass and Jeff Bilmes (2003) Hidden feature models for speech recognition using dynamic Bayesian networks, in *Proceedings of Eurospeech*, pp. 2529–2532, Geneva, Switzerland.

Mangu, Lidia, Eric Brill and Andreas Stolcke (1999) Finding Consensus Among Words: Lattice-Based Word Error Minimization, in *Proceedings of Eurospeech*, pp. 495–498, Budapest, Hungary.

Manning, Christopher and Hinrich Schütze (1999) *Foundations of Statistical Natural Language Processing*, 6th edition, MIT Press.

Martin, David L., Adam J. Cheyer and Douglas B. Moran (1999) The Open Agent Architecture: A Framework for Building Distributed Software Systems, *Applied Artificial Intelligence*, Vol. 13, No. 1-2, pp. 91–128.

McTear, Michael F. (1999) Software to Support Research and Development of Spoken Dialogue Systems, in *Proceedings of Eurospeech*, pp. 339–342, Budapest, Hungary.

McTear, Michael F. (2002) Spoken dialogue technology: enabling the conversational interface, *ACM Computing Surveys*, Vol. 34, No. 1, pp. 90–169.

McTear, Michael F. (2004) *Spoken dialogue technology: toward the conversational user interface*, Springer Verlag, London.

Meza-Ruiz, Ivan and Oliver Lemon (2005) Using dialogue context to improve parsing performance in dialogue systems, in *Proceedings of International Workshop on Computational Semantics (IWCS)*, Tilburg. Available at: http://homepages.inf.ed.ac.uk/olemon/iwcs05.pdf.

Misu, Teruhisa and Tatsuya Kawahara (2006) A bootstrapping approach for developing language model of new spoken dialogue systems by selecting web texts, in *Proceedings of Interspeech 2006 - ICSLP Ninth International Conference on Spoken Language Processing*, Pittsburgh, PA. Paper 1167.

Mitchell, Tom M. (1997) *Machine Learning*, McGraw-Hill, New York.

Moore, Robert C. (1999) Using Natural-Language Knowledge Sources in Speech Recognition, in *Computational Models of Speech Pattern Processing*, pp. 304–327, Springer-Verlag.

Moore, Roger K (2003) A comparison of the data requirements of automatic speech recognition systems and human listeners, in *Proceedings of Eurospeech*, pp. 2582–2584, Geneva, Switzerland.

Moore, Roger K. (2005) Results from a survey of attendees at ASRU 1997 and 2003, in *Proceedings of Interspeech 2005 – Eurospeech*, pp. 117–120, Lisbon, Portugal.

Moore, Roger K (2007) Spoken Language Processing: Piecing together the puzzle, *Speech Communication*, Vol. 49, pp. 418–435.

Moore, Roger K and Anne Cutler (2001) Constraints on theories of human vs. machine recognition of speech, in *Proceedings of SPRAAC (Workshop on Speech Recognition as Pattern Classification)*, pp. 145–150, Nijmegen, Netherlands.

Nagata, Masaaki and Tsuyoshi Morimoto (1993) An experimental statistical dialogue model to predict speech act type of the next utterance, in *Proceedings of the International Symposium on Spoken Dialogue*, pp. 83–86, Tokyo, Japan.

Nakano, Mikio and Timothy J. Hazen (2003) Using untranscribed user utterances for improving language models based on confidence scoring, in *Proceedings of Eurospeech*, pp. 417–420, Geneva, Switzerland.

Ng, Tim, Mari Ostendorf, Mei-Yuh Hwang, Ivan Bulyko, Manhung Siu and Xin Lei (2005) Web-data Augmented Language Model for Mandarin Speech Recognition, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1, pp. 589–592, Philadelphia, PA.

Noord, Gertjan Van, Gosse Bouma, Rob Koeling and Mark jan Nederhof (1999) Robust grammatical analysis for spoken dialogue systems, *Journal of Natural Language Engineering*, Vol. 5, pp. 45–93.

Nuance (2006) Nuance Communications. http://www.nuance.com.

Olsson, Anna and Jessica Villing (2005) Dico - a dialogue system for a cell phone. Master's thesis, Department of Linguistics, University of Gothenburg.

Orlandi, Marco, Christopher Culy and Horacio Franco (2003) Using dialog corrections to improve speech recognition, in *Proceedings of ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems*, pp. 47–51, Chateau-d'Oex-Vaud, Switzerland.

Ostendorf, Mari and Ivan Bulyko (2002) The impact of speech recognition on speech synthesis, in *Proceedings of 2002 IEEE Workshop on Speech Synthesis*, pp. 99–106, Santa Monica, CA.

Ostendorf, Mari, Izhak Shafranz and Rebecca Bates (2003) Prosody Models for Conversational Speech Recognition, in *Proceedings of the 2nd Plenary Meeting and Symposium on Prosody and Speech Processing*, pp. 147–154, Tokyo, Japan.

Oviatt, Sharon, Margaret Maceachern and Gina-Anne Levow (1998) Predicting hyperarticulate speech during human-computer error resolution, *Speech Communication*, Vol. 24, pp. 87–110.

Pakhomov, Sergey, Michael Schonwetter and Joan Bachenko (2001) Generating Training Data for Medical Dictations, in *Proceedings of the NAACL*, pp. 1–8, Pittsburgh, PA.

Pao, Christine, Philipp Schmid and James Glass (1998) Confidence Scoring for Speech Understanding Systems, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, Sydney, Australia. Paper 0392.

Peckham, Jeremy (1991) Speech Understanding and Dialogue over the Telephone: An Overview of the ESPRIT SUNDIAL Project, in *Proceedings of the Speech and Natural Language Workshop*, pp. 14–27, Pacific Grove, CA.

Perrault, Raymond C., James F. Allen and Philip R. Cohen (1978) Speech acts as a basis for understanding dialogue coherence, in *Proceedings of the 1978 workshop on Theoretical issues in natural language processing*, pp. 125–132, Association for Computational Linguistics, Morristown, NJ.

Poesio, Massimo and Andrei Mikheev (1998) The predictive power of game structure in dialogue act recognition: Experimental results using maximum entropy estimation, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, Sydney, Australia. Paper 0606.

Popovici, Cosmin and Paolo Baggia (1997a) Language Modelling For Task-Oriented Domains, in *Proceedings of EUROSPEECH'97*, Vol. 3, pp. 1459–1462, Rhodes, Greece.

Popovici, Cosmin and Paolo Baggia (1997b) Specialized Language Models using Dialogue Predictions, in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP'97)*, Vol. 2, p. 815, Munich, Germany.

Pradhan, Sameer and Wayne Ward (2002) Estimating Semantic Confidence for Spoken Dialogue Systems, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1, pp. 233–236, Orlando, Florida.

Purver, Matthew (2002) Processing Unknown Words in a Dialogue System, in *Proceedings of the 3rd SIGDial Workshop on Discourse and Dialogues*, pp. 174–183, Philadelphia, PA.

Purver, Matthew, Florin Ratiu and Lawrence Cavedon (2006) Robust Interpretation in Dialogue by Combining Confidence Scores with Contextual Features, in *Proceedings of Interspeech 2006 - ICSLP Ninth International Conference on Spoken Language Processing*, Pittsburgh, PA. Paper 1314.

Quesada, José, Gabriel Amores, Pilar Manchón, Guillermo Peréz, Silvia Knight, David Milward and James Thomas (2002) Possibilities for Enhancing Speech Recognition by Consulting Information States, Deliverable D2.3, SIRIDUS. Available at: http://www.ling.gu.se/projekt/siridus/Publications/deliv2-3.pdf.

Rangarajan, Vivek, Srinivas Bangalore and Shrikanth Narayanan (2007) Exploiting prosodic features for dialog act tagging in a discriminative modeling framework, in *Proceedings of Interspeech 2007*, pp. 150–153, Antwerp, Belgium.

Ranta, Aarne (2004) Grammatical Framework. A Type-Theoretical Grammar Formalism, *Journal of Functional Programming*, Vol. 14, No. 2, pp. 145–189.

Raux, Antoine, Dan Bohus, Brian Langner, Alan Black and Maxine Eskenazi (2006) Doing Research on a Deployed Spoken Dialogue System: One Year of Let's Go! Experience, in *Proceedings of Interspeech 2006 - ICSLP Ninth International Conference on Spoken Language Processing*, Pittsburgh, PA. Paper 1794.

Raux, Antoine, Brian Langner, Alan Black and Maxine Eskenazi (2003) LET'S GO: Improving Spoken Dialog Systems for the Elderly and Non-natives, in *Proceedings of Eurospeech*, pp. 753–756, Geneva, Switzerland.

Raux, Antoine, Brian Langner, Dan Bohus, Alan Black and Maxine Eskenazi (2005) Let's go public! taking a spoken dialog system to the real world, in *Proceedings of Interspeech 2005 – Eurospeech*, pp. 885–888, Lisbon, Portugal.

Rayner, Manny, David Carter, Vasilis Digalakis and Patti Price (1994) Combining Knowledge Sources to Reorder N-Best Speech Hypothesis Lists, in *Proceedings of the 1994 ARPA Workshop on Human Language Technology*, pp. 219–221, Plainsboro, NJ.

Rayner, Manny, Genevieve Gorrell, Beth Ann Hockey, John Dowding and Johan Boye (2001) Do CFG-based language models need agreement constraints?, in *Proceedings of North American Chapter of the Association for Computational Linguistics on Language technologies (NAACL '01)*, pp. 1–9, Association for Computational Linguistics, Morristown, NJ.

Rayner, Manny and Beth Ann Hockey (2003) Transparent combination of rule-based and data-driven approaches in a speech understanding architecture, in *Proceedings of EACL'03*, pp. 299 – 306, Budapest, Hungary.

Rayner, Manny, Beth Ann Hockey and Pierrette Bouillon (2006) *Putting Linguistics into Speech Recognition: The Regulus Grammar Compiler*, CSLI Publications.

Rayner, Manny, Beth Ann Hockey, Nikos Chatzichrisafis, Kim Farrell and Jean-Michel Renders (2005) A voice enabled procedure browser for the International Space Station, in *ACL '05: Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*, pp. 29–32, Association for Computational Linguistics, Morristown, NJ.

Rayner, Manny, Beth Ann Hockey, Frankie James, Elizabeth Owen Bratt, Sharon Goldwater and Jean Mark Gawron (2000) Compiling Language Models from a Linguistically Motivated Unification Grammar, in *Proceedings of International Conference on Computational Linguistics (COLING)*, Vol. 2, pp. 670–676, Saarbruecken, Germany.

Reithinger, Norbert, Ralf Engel, Michael Kipp and Martin Klesen (1996) Predicting Dialogue Acts for a Speech-To-Speech Translation System, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, Vol. 2, pp. 654–657, Philadelphia, PA.

Reithinger, Norbert and Martin Klesen (1997) Dialogue Act Classification using language models, in *Proceedings of Eurospeech*, pp. 2234–2238, Rhodes, Greece.

Renders, Jean-Michel, Manny Rayner and Beth Ann Hockey (2005) Simple Kernel Methods for Identification of Cross-Talk and Misrecognition. Available at: http://ti.arc.nasa.gov/m/pub/1000h/1000

Riccardi, Guiseppe, Alexandros Potamianos and Shrikanth Narayanan (1998) Language model adaptation for spoken language systems, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, Sydney, Australia. Paper 1052.

Roque, Antonio, Anton Leuski, Vivek Rangarajan, Susan Robinson, Ashish Vaswani, Shrikanth Narayanan and David Traum (2006) Radiobot-CFF: A Spoken Dialogue System for Military Training, in *Proceedings of Interspeech 2006 - ICSLP Ninth International Conference on Spoken Language Processing*, Pittsburgh, PA. Paper 1828.

Rosenfeld, Ronald (2000a) Incorporating Linguistic Structure into Statistical Language Models, in *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, Vol. 358, pp. 1311–1324, The Royal Society.

Rosenfeld, Ronald (2000b) Two decades of statistical language modeling: Where do we go from here?, *Proceedings of the IEEE*, Vol. 88, No. 8, pp. 1270–1278.

Ross, Robert and John Bateman (2009) Daisie: Information State Dialogues for Situated Systems, in *Proceedings of Text, Speech and Dialogue (TSD)*, Vol. 5729/2009, pp. 379–386, Springer Berlin / Heidelberg, Pilsen, Czech Republic.

Rosset, Sopie and Delphine Tribout (2005) Multi-level information and automatic dialog acts detection in human-human spoken dialogs, in *Proceedings of Interspeech 2005 – Eurospeech*, pp. 2789–2792, Lisbon, Portugal.

Rotaru, Mihai and Diane J. Litman (2005) Using Word-Level Pitch Features to Better Predict Student Emotions During Spoken Tutoring Dialogues, in *Proceedings of Interspeech 2005 – Eurospeech*, pp. 881–884, Lisbon, Portugal.

Rudnicky, Alexander and Wei Xu (1999) An Agenda-Based Dialog Management Architecture For Spoken Language Systems, in *Proceedings of IEEE ASRU Workshop*, pp. 337–340, Keystone, CO.

Sacks, Harvey, Emanuel A. Schegloff and Gail Jefferson (1974) A Simplest Systematics for the Organization of Turn-Taking for Conversation, *Language*, Vol. 50, No. 4, pp. 696–735.

Samuel, Ken, Sandra Carberry and K. Vijay-Shanker (1998) Dialogue Act Tagging with Transformation-Based Learning, in *Proceedings of ACL and COLING*, pp. 1150–1156, San Francisco, CA.

San-Segundo, Ruben, Juan Manuel Montero, J Ferreiros, Ricardo Córdoba and Juan M Pardo (2001a) Designing confirmation mechanisms and error recover techniques in a Railway Information system for Spanish, in *Proceedings of the 2nd SIGdial Workshop on Discourse and Dialogue*, pp. 1–4, Association for Computational Linguistics, Morristown, NJ.

San-Segundo, Ruben, Bryan Pellom, Kadri Hacioglu, Wayne Ward and Juan Pardo (2001b) Confidence Measures for Spoken Dialogue Systems, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1, pp. 393–396, Salt Lake City, Utah.

Sarikaya, Ruhi, Agustin Gravano and Yuqing Gao (2005) Rapid Language Model Development Using External Resources for New Spoken Dialog Domains, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1, pp. 573–576, Philadelphia, PA.

Scharenborg, Odette (2007) Reaching over the gap: A review of efforts to link human and automatic speech recognition research, *Speech Communication*, Vol. 49, No. 5, pp. 336–347.

Searle, John R. (1969) *Speech Acts: An essay in the philosophy of language*, Cambridge University Press, Cambridge, England.

Searle, John R. (1975) *A taxonomy of illocutionary acts*, University of Minnesota Press, Minneapolis.

Seneff, Stephanie (1992) TINA: A Natural Language System for Spoken Language Applications, *Computational Linguistics*, Vol. 18, No. 1, pp. 61–86.

Seneff, Stephanie, Ed Hurley, Raymond Lau, Christine Pao, Philipp Schmid and Victor Zue (1998) Galaxy-II: A Reference Architecture For Conversational System Development, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pp. 931–934, Sydney, Australia.

Sethy, Abhinav, Panayiotis Georgiou and Shrikanth Narayanan (2005) Building topic specific language models from webdata using competetive models, in *Proceedings of Interspeech 2005 – Eurospeech*, pp. 1293–1296, Lisbon, Portugal.

Shin, Jongho, Shrikanth Narayanan, Laurie Gerber, Abe Kazemzadeh and Dani Byrd (2002) Analysis of User Behaviour under Error Conditions in Spoken Dialogs, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pp. 2069–2072, Denver, CO.

Shriberg, Elizabeth (2005) Spontaneous Speech: How people really talk and why engineers should care, in *Proceedings of Interspeech 2005 – Eurospeech*, pp. 1781–1784, Lisbon, Portugal.

Shriberg, Elisabeth, Rebecca Bates, Andreas Stolcke, Paul Taylor, Daniel Jurafsky, Klaus Ries, Noah Coccaro, Rachel Martin, Marie Meteer and Carol van Ess-Dykema (1998) Can prosody aid the automatic classification of dialog acts in conversational speech?, *Language and speech*, Vol. 41 (Pt 3-4), pp. 443–492.

Shriberg, Elisabeth and Andreas Stolcke (2004) Prosody Modeling for Automatic Speech Recognition and Understanding, *Mathematical Foundations of Speech and Language Processing, IMA Volumes in Mathematics and Its Applications*, Vol. 138, pp. 105–144.

Shriberg, Elizabeth, Andreas Stolcke, Dilek Hakkani-Tür and Gükhan Tür (2000) Prosody-based automatic segmentation of speech into sentences and topics, *Speech Communication*, Vol. 32, No. 1-2, pp. 127–154.

Skantze, Gabriel (2005a) Exploring Human Error Handling Strategies: Implications for Spoken Dialogue Systems, *Speech Communication*, Vol. 45, No. 3, pp. 325–341.

Skantze, Gabriel (2005b) Galatea: a discourse modeller supporting concept-level error handling in spoken dialogue systems, in *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue*, pp. 178–189, Lisbon, Portugal.

Skantze, Gabriel (2007) *Error Handling in Spoken Dialogue Systems - Managing Uncertainty, Grounding and Miscommunication*, PhD dissertation, Department of Speech, Music and Hearing, KTH, Sweden.

Skantze, Gabriel and Jens Edlund (2004) Early Error Detection on Word Level, in *ISCA Tutorial and Research Workshop (ITRW) on Robustness Issues in Conversational Interaction*, Norwich, UK. Paper 17.

Smith, Ronnie W. and Richard D. Hipp (1995) *Spoken natural language dialog systems: a practical approach*, Oxford University Press, Inc., New York, NY.

Solsona, Roger Argiles, Eric Fosler-Lussier, Hong-Kwang Jeff Kuo, Alexandros Potamianos and Imed Zitouni (2002) Adaptive Language Models for Spoken Dialogue Systems, in *Proceedings of the International Conference on Acoustic Speech and Signal Processing (ICASSP)*, Vol. 1, pp. 37–40, Orlando, Florida.

Soltau, Hagen, Florian Metze and Alex Waibel (2002) Compensating for Hyperarticulation by Modeling Articulatory Properties, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pp. 841–844, Denver, CO.

Soltau, Hagen and Alex Waibel (2000) Specialized acoustic models for hyperarticulated speech, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1779–1782, Istanbul, Turkey.

Stolcke, Andreas (2002) SRILM - An Extensible Language Modeling Toolkit, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pp. 901–904, Denver, CO.

Stolcke, Andreas, Noah Coccaro, Rebecca Bates, Paul Taylor, Carol Van Ess-Dykema, Klaus Ries, Elizabeth Shriberg, Daniel Jurafsky, Rachel Martin and Marie Meteer (2000) Dialogue act modeling for automatic tagging and recognition of conversational speech, *Computational Linguistics*, Vol. 26, No. 3, pp. 339–373.

Stolcke, Andreas and Elisabeth Shriberg (1996) Statistical language modeling for speech disfluencies, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 405–409, Atlanta, Georgia.

Surendran, Dinoj and Gina-Anne Levow (2006) Dialog Act Tagging with Support Vector Machines and Hidden Markov Models, in *Proceedings of Interspeech 2006 - ICSLP Ninth International Conference on Spoken Language Processing*, Pittsburgh, PA. Paper 1831.

Taylor, Paul, Simon King, Stephen Isard and Helen Wright (1998) Intonation and dialogue context as constraints for speech recognition, *Language and Speech*, Vol. 41, pp. 489–508.

Traum, David R. (1995) *A computational theory of grounding in natural language conversation*, PhD dissertation, University of Rochester, Rochester, NY.

Traum, David R. (2000) 20 Questions for Dialogue Act Taxonomies, *Journal of Semantics*, Vol. 17, pp. 7–30.

Traum, David R. and Staffan Larsson (2003) The information state approach to dialogue management, in R. W. Smith and J. Kuppevelt (eds.), *Current and New Directions in Discourse & Dialogue*, pp. 325–353, Kluwer.

Trimmis, Nikolaos, N Markatos, K Malaperdas and Evangelos Papadeas (2007) Word Recognition Scores by Native and Non-Native Speakers of Modern Greek Language, in *Proceedings of 8th EFAS Congress (European Federation of Audiology Societies)*, Heidelberg, Germany. Available at: http://www.uzh.ch/orl/dga2007/program/scientificprogram/ Trimmis_N._Markatos_N._Malaperdas_K._Papadeas_E..pdf.

Turunen, Markku and Jaakko Hakulinen (2003) Jaspis - An Architecture for Supporting Distributed Spoken Dialogues, in *Proceedings of Eurospeech*, pp. 1913–1916, Geneva, Switzerland.

Venkataraman, Anand, Luciana Ferrer, Andreas Stolcke and Elisabeth Shriberg (2003) Training a Prosody-based Dialog Act Tagger from unlabeled data, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1, pp. 272–275, Hong Kong.

Villing, Jessica and Staffan Larsson (2006) Dico - A Multimodal Menu-based In-vehicle Dialogue System, in *Proceedings of Brandial (10th workshop on the semantics and pragmatics of dialogue)*, pp. 187–188, Potsdam, Germany.

Wai, Carmen, Roberto Pieraccini and Helen Meng (2001) A Dynamic Semantic model for Re-scoring Recognition Hypotheses, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1, pp. 593–596, Salt Lake City, Utah.

Walker, Marilyn, Irene Langkilde, Jerry Wright, Alien Gorin and Diane Litman (2000a) Learning to predict problematic situations in a spoken dialogue system: Experiments with how may I help you, in *Proceedings of North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 210–217, Seattle, Washington.

Walker, Marilyn, Jerry Wright and Irene Langkilde (2000b) Using Natural Language Processing and Discourse Features to Identify Understanding Errors in a Spoken Dialogue System, in *Proceedings of 17th International Conference on Machine Learning*, pp. 1111–1118, Morgan Kaufmann, San Francisco, CA.

Wang, Wen, Yang Liu and Mary Harper (2002a) Rescoring effectiveness of language models using different levels of knowledge and their integration, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1, pp. 785–788, Orlando, Florida.

Wang, Wen, Andreas Stolcke and Mary Harper (2004) The use of a linguistically motivated language model in conversational speech recognition, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1, pp. 261–264, Montreal, QC, Canada.

Wang, Ye-Yi, Alex Acero and Ciprian Chelba (2003) Is word error rate a good indicator for spoken language understanding accuracy?, in *Proceedings of Eurospeech*, pp. 609–612, Geneva, Switzerland.

Wang, Ye-Yi, Alex Acero, Ciprian Chelba, Brendan Frey and Leo Wong (2002b) Combination of Statistical and Rule-based approaches for Spoken Language Understanding, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, Denver, CO.

Ward, Nigel G., Anais G. Rivera, Karen Ward and David G. Novick (2005) Root Causes of Lost Time and User Stress in a Simple Dialog System, in *Proceedings of Interspeech 2005 – Eurospeech*, Lisbon, Portugal.

Ward, Wayne (1991) Understanding spontaneous speech: the Phoenix system, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 365–367, IEEE Computer Society, Washington, DC.

Ward, Wayne and Sunil Issar (1994) Recent improvements in the CMU spoken language understanding system, in *HLT '94: Proceedings of the workshop on Human Language Technology*, pp. 213–216, Association for Computational Linguistics, Morristown, NJ.

Ward, Wayne and Sunil Issar (1996) A class based language model for speech recognition, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 416–418, IEEE Computer Society, Atlanta, Georgia.

Webb, Nick, Mark Hepple and Yorick Wilks (2005) Dialog Act Classification Based on Intra-Utterance Features, in *Proceedings of the AAAI Workshop on Spoken Language Understanding*, Pittsburgh, PA.

Wee, Chze Ling (2004) Web data for language modelling of conversational telephone speech. Master's thesis, Cambridge University Engineering Dept, Machine Intelligence Laboratory, Trumpington Street, Cambridge, CB2 1PZ, United Kingdom.

Weilhammer, Karl, Rebecca Jonson, Håkan Burden, Jost Schatzmann, Matt Stuttle and Steve Young (2006a) Integrating multiple modalities into SLMs and parsing the output of SLMs, Deliverable D1.4, TALK Project. http://www.talk-project.org/fileadmin/talk/publications_public/deliverables_public/D1_4.pdf.

Weilhammer, Karl, Matthew Stuttle and Steve Young (2006b) Bootstrapping Language Models for Dialogue Systems, in *Proceedings of International Conference on Spoken Language Processing (ICSLP 2006)*, Pittsburgh, PA.

Weintraub, Mitch, Kelsey Taussig, Kate Hunicke-Smith and Amy Snodgrass (1996) Effect of speaking style on LVCSR performance, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pp. 16–19, Philadelphia, PA.

Wilson, Stephen, Ayse Pinar Saygin, Martin Sereno and Marco Iacoboni (2004) Listening to speech activates motor areas involved in speech production, *Nature NeuroScience*, Vol. 4, pp. 701–702.

Witten, Ian, Eibe Frank, Len Trigg, Mark Hall, Geoffrey Holmes and Sally Cunningham (1999) Weka: Practical machine learning tools and techniques with java implementations, in *Proceedings of ICONIP/ANZIIS/ANNES'99 Int. Workshop: Emerging Knowledge Engineering and Connectionist-Based Info. Systems.*, pp. 192–196, Perth, Australia.

Wright, Helen (1998) Automatic utterance type detection using suprasegmental features, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pp. 1403–1406, Sydney, Australia.

Wright, Helen, Massimo Poesio and Stephen Isard (1999) Using high level dialogue information for dialogue act recognition using prosodic features, in *ESCA Tutorial and Research Workshop on Dialogue and Prosody*, pp. 139–143, Eindhoven, The Netherlands.

Wu, Hsu-Chih and Stephanie Seneff (2007) Reducing Recognition Error Rate based on Context Relationships among Dialogue Turns, in *Proceedings of Interspeech 2007*, pp. 2701–2704, Antwerp, Belgium.

Xu, Wei and Alexander Rudnicky (2000a) Can artificial neural networks learn language models?, in *Proceedings of International Conference on Spoken Language Processing (ICSLP 2000)*, Vol. 1, pp. 202–205, Beijing, China.

Xu, Wei and Alexander Rudnicky (2000b) Language modeling for dialog system, in *Proceedings of International Conference on Spoken Language Processing (ICSLP 2000)*, Vol. 1, pp. 118–121, Beijing, China.

Young, Steve (1996) Large Vocabulary Continuous Speech Recognition, *IEEE Signal Processing Magazine*, Vol. 13, pp. 45–75.

Young, Sheryl and Wayne Ward (1993a) Learning New Words From Spontaneous Speech, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 2, pp. 590–591, Minneapolis.

Young, Sheryl and Wayne Ward (1995) The role of higher-level semantic, pragmatic and discourse knowledge in recognizing and understanding new spoken words and phrases, in *Proceedings of ESCA Workshop on Spoken Dialogue Systems*, pp. 29–32, Vigso, Denmark.

Young, Sheryl R. (1989) The MINDS system: Using Context and Dialog to Enhance Speech Recognition, in *HLT '89: Proceedings of the workshop on Speech and Natural Language*, pp. 131–136, Association for Computational Linguistics, Philadelphia, PA.

Young, Sheryl R., Alexander G. Hauptmann, Wayne Ward, Edward Smith and Philip Werner (1989) High Level Knowledge Sources in Usable Speech Recognition Systems, *Communications of ACM*, Vol. 32, No. 2, pp. 183–194.

Young, Sheryl R. and Wayne Ward (1993b) Recognition Confidence Measures for Spontaneous Spoken Dialog, in *Proceedings of Eurospeech*, pp. 1177–1179, Berlin, Germany.

Yu, Don, Yun Cheng Ju, Ye-Yi Wang and Alex Acero (2006) N-gram based filler model for Robust Grammar Authoring, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1, pp. 565–568, Toulouse, France.

Zhang, Rong and Alexander Rudnicky (2001) Word Level Confidence Annotation using Combinations of Features, in *Proceedings of Eurospeech*, pp. 2105–2108, Aalborg, Denmark.

Zhang, Rong and Alexander Rudnicky (2002) Improve Latent Semantic Analysis based Language Model by Integrating Multiple Level Knowledge, in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pp. 893–896, Denver, CO.

Zhu, Xiaojin and Ronald Rosenfeld (2001) Improving Trigram Language Modeling with the World Wide Web, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1, pp. 533–536, Salt Lake City, Utah.

Zue, Victor, James Glass, David Goodine, Hong Leung, Michael Phillips, Joseph Polifroni and Stephanie Seneff (1991) Integration of speech recognition and natural language processing in the MIT VOYAGER system, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 713–716, IEEE Computer Society, Washington, DC.

Appendix A

Discarded Features

In the confidence classification experiment in Section 8.4.3 we had at first many more features as preliminary candidates. These were later discarded since they did not contribute to successful results. However, it might be of interest to know which features were not informative for the task. The discarded features are therefore listed below.

(65)
HypWordConf: List of word confidence scores
ListLen: Number of hypotheses in list
ListMeanConf: Mean confidence score of list
ListConfStdDev: Standard deviation of confidence scores in list
ListMinConfScore: Minimum confidence score in list
ListTopConfScore: Top confidence score in list
ListMeanProb: Mean probability score of list
ListProbStdDev: Standard deviation of probability in list
HypDMScore: Dialogue move confidence scores
DMPred: Predicted dialogue move
PrevDM: Previous dialogue move
SHCOM: Shared commitments
FSTCOM: First item on shared commitments
FSTOFPLAN: First construct in plan
DREF: List of discourse referents
DIAHIS: Dialogue history in reduced form
ACTION: Last performed action