Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks

Promotiecommissie

voorzitter:	Prof. dr. W.H.M. Zijm (Universiteit Twente)
promotor:	Prof. dr. ir. A. Nijholt (Universiteit Twente)
assistent-promotor:	Dr. ir. H.J.A. op den Akker (Universiteit Twente)
overige leden:	Prof. dr. H.C. Bunt (Universiteit van Tilburg)
	Prof. dr. ir. L.C. van der Gaag (Universiteit Utrecht)
	Prof. dr. J. Ginzburg (King's College, London)
	Prof. dr. C. Hoede (Universiteit Twente)
	Prof. dr. K. Jokinen (University of Helsinki)
	Prof. dr. F.M.G. de Jong (Universiteit Twente)

CTIT Ph.D.-thesis series No. 03-55
Centre for Telematics and Information Technology,
P.O. Box 217, 7500 AE Enschede

SIKS Dissertation Series No. 2003-11
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Publisher:

Twente University Press,
P.O. Box 217, 7500 AE Enschede, The Netherlands,
`www.tup.utwente.nl`

Twente University Press

Cover design: Simon Keizer
Cover illustration: Aurora Hilbrink
Print: Océ Facility Services, Enschede

ISBN 90-365-1951-9

ISSN 1381-3617 (CTIT Ph.D.-thesis series No. 03-55)

REASONING UNDER UNCERTAINTY IN NATURAL LANGUAGE DIALOGUE USING BAYESIAN NETWORKS

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. F.A. van Vught,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op woensdag 3 september 2003 om 15.00 uur

door

Simon Keizer
geboren op 12 september 1973
te Leeuwarden

Dit proefschrift is goedgekeurd door de promotor

prof. dr. ir. A. Nijholt

en de assistent-promotor

dr. ir. H.J.A. op den Akker

Hello, HAL do you read me, HAL?

Affirmative, Dave, I read you.

Open the pod bay doors, HAL.

I'm sorry Dave, I'm afraid I can't do that.

What's the problem?

I think you know what the problem is just as well as I do.

What are you talking about, HAL?

This mission is too important for me to allow you to jeopardize it.

I don't know what you're talking about, HAL.

I know you and Frank were planning to disconnect me, and I'm afraid that's something I cannot allow to happen.

Where the hell'd you get that idea, HAL?

Dave, although you took thorough precautions in the pod against my hearing you, I could see your lips move.

from: '2001: A Space Odyssey' (Kubrick, US, 1968)

Preface

Some five years ago, I started as a research member of the Parlevink Language Engineering Group. This developed into a Ph.D.-project under supervision of my promotor Anton Nijholt and my supervisor Rieks op den Akker, and the results have now been put down in the dissertation lying here before you.

As a student in applied mathematics, I became fascinated by natural language and wrote my master's thesis on *knowledge graphs* for natural language processing; the focus there was on representing words within a sentence. Now, I have also learned about utterances, within the context of a *dialogue* and the focus has moved to the *use* of language.

Moreover, despite the slight aversion I may have had against probability theory and statistics, I can now more appreciate them, after becoming acquainted with Bayesian networks and applying this technology to natural language processing.

I would like to express my sincere gratitude to Anton Nijholt for giving me the opportunity to do my Ph.D.-project in his research group. Without his positive approach at the times that I, ironically, felt *uncertain* about my progress, this thesis would never have come about.

Most of all, I thank Rieks op den Akker for being such a pleasant and helpful supervisor. The many discussions we have had about natural language, probability theory and other related issues were very inspiring to me. Rieks has given me many useful idea's to work out, idea's that have been essential in the development of the research and in preparing this dissertation. It has been a pleasure working with him.

I am also very grateful to Prof. dr. H.C. Bunt, Prof. dr. ir. L.C. van der Gaag, Prof. dr. J. Ginzburg, Prof. dr. C. Hoede, Prof. dr. K. Jokinen, and Prof. dr. F.M.G. de Jong for taking a place in my promotion committee. I feel very honoured.

Several of my colleagues have been of great help to me in the many fruitful discussions I have had with them. Special mention in this respect goes out to my roommate Roeland Ordelman, to Martijn van Otterlo, and to the members of my supervision committee, Dirk Heylen, Mannes Poel and Paul van der Vet.

I would also like to thank Jeroen van Barneveld for his work on annotating the SCHISMA corpus, Hendri Hondorp for helping me out many times when I had a problem with LaTeX, Java, or other computer stuff, and Charlotte Bijron and Alice Vissers of the secretariat, who have made life at the office a lot easier by taking care of so many things.

Last but not least, I would like to thank my friends and family for their love and support. Some have patiently listened to me trying to explain what my work is about, some provided an enjoyable distraction from everyday stress. I had great fun during the Monday evenings of playing indoor soccer with my colleagues from the UT-Kring, and watching and discussing movies, many many movies, with my dear friends has been, and still is, a very pleasurable waste of time (just to be sure: that last remark was a joke).

Simon Keizer

Enschede, August 2003

Contents

Chapter 1

Introduction

In which uncertainty in natural language dialogue is introduced as the central problem in the research described in this thesis. The idea of using of Bayesian networks is hypothesised as a possible solution to this problem. Dialogue acts are presented as the central notion in our approach to dialogue modelling and the task of recognising dialogue act types as the central topic in the experiments.

One of the most fundamental expressions of social and intelligent behaviour among human beings is natural language, be it either spoken or written language. Natural language dialogue can be seen as a very advanced and convenient – and indeed *natural* – way for people to interact with each other. This is one of the reasons why since the late 1960s there has been an ever growing interest in the field of computer science to develop computer programs that are provided with a natural language interface in order to make them more user-friendly. Today, this interest has developed into the specialised field of human-computer interaction involving not only natural language, but also other modalities of interaction, such as gestures and eye-movement that can nowadays be tracked by the state-of-the-art computer hard- and software.

Modelling a dialogue system, that is, a computer program that is able to process natural language utterances and give responses that are appropriate in the given state of the dialogue, is quite a challenging task, even in the case of relatively simple application areas such as that of providing public transport time-table information. The phenomena and mechanisms underlying natural language and the use of it in the communication between intelligent entities are very complex and contain many subtleties that are essential in the naturalness and (therefore) successfulness of the interaction.

1.1 Uncertainty in Human-human Dialogues

In this thesis we will discuss a phenomenon that plays an important role in understanding natural language dialogue, namely, that of *uncertainty*. By uncertainty we mean a qualification by a (human or artificial) agent to a state of affairs or the outcome of an event or[1]. In a dialogue, a hearer does not always have absolute certainty about what the speaker meant by his utterance and how he should react to it. This mental state of uncertainty arises because he might have missed or misheard parts of the utterance. Part of the information in a speaker producing an utterance is lost by the time the addressee actually hears the utterance. For the addressee, the information that is actually available to him is generally insufficient, and this keeps him from being absolutely certain about what was said. The addressee may for example have not enough information to know for sure if the speaker said "Austria" or "Australia".

Moreover, even what was actually said is often insufficient for interpreting utterances in a successful natural interaction. In many cases more is communicated than what was actually said. If a customer in a music shop approaches the counter with a CD in his hands, saying "this one please", the shop assistant will immediately understand that the speaker wants to buy the CD he is holding and will start the procedure for the transaction. Although the customer's intention is communicated, it is not directly expressed in the utterance. The given context of a music shop enforces certain tacit *conventions* our shop assistant uses to draw the conclusions given above. The customer uses these conventions when producing the utterance and expects the assistant to agree about these conventions. If I would go out on the street with a CD in my hands and say "this one please" to the first person I see passing by, he or she would not understand me at all, let alone if I had not even brought the CD.

The situation in which the utterance is made carries a potentially infinite number of factors that are relevant to the intention of the speaker. Because the perspectives the speaker and the hearer have of the situation might differ at some points – and therefore, the knowledge they have about the situation might not be identical – a hearer might misinterpret the speaker's utterance. For example, in stead of our shop assistant's interpretation that the customer wanted to buy the CD, the customer might have wanted to *listen* to the CD. Maybe the customer nodded in the direction of the headphones and the assistant did not notice that.

Besides loss of information in the communication channel, there is also a more fundamental ground for uncertainty, concerning the nature of meaning as such. A very common view in language philosophy is that language is used to express thoughts, in the sense of externalising them in time and space. A thought is something abstract that is emerges 'in a flash', while a spoken utterance is in fact an event, in which the speaker basically causes a series of changes in air-pressure. Even assuming that there is no loss of information in communicating the utterance itself, there is still no guarantee for *intersubjectivity*: that the thought expressed (or: 'encoded') by the speaker in the utterance is identical to the thought the hearer reproduces (or: 'decodes') when interpreting that utterance. Of course, intuitively we feel that in general, we are 'sharing thoughts'

[1] Translation of the English term "uncertainty" to Dutch with the word "onzeker" introduces a somewhat confusing connotation of a psychological disposition of a person being insecure or hesitant. In that respect, the word "onbepaaldheid" might be a better translation.

in the conversations we have, but a full account of the relation between 'the mental' and language is certainly not trivial and is an important issue in any serious philosophical theory of language and meaning[2].

1.2 Uncertainty in Human-Machine Dialogues

In the field of computational linguistics, natural language is considered from a technological perspective. This perspective is dominated by the classical descriptive theories of meaning, in which the meaning of an expression is taken to be a description of the way the expression refers to reality. In the technological perspective this boils down to the requirement of formal specifications. The problem of uncertainty arises basically because no accurate and complete formal specification of natural language has been found so far. Such a formal foundation may very well not exist at all. Any attempt to create a model for natural language dialogue introduces *abstractions* that may leave relevant phenomena and mechanisms not captured. In the case of human-computer interaction, this means that a computer system can never completely explain the natural communicative behaviour of human users and therefore is confronted with uncertainty with respect to its interpretations of user utterances, and has to deal with that in one way or another.

From a more practical computational point of view, it is not necessary to model all phenomena and mechanisms of natural interaction, as many aspects may be irrelevant given the specific application at hand. Also, a trade-off between computational efficiency and model complexity has to be taken into account. Especially in spoken dialogue systems, it is important not to disrupt the flow of the interaction by taking too much time in deliberating what utterance to produce. Besides that, it has been shown that users adapt their behaviour if they are aware that they are interacting with a machine, by avoiding complex behaviour in the interaction and use of language. Even *during* dialogues with the system users show adaptive behaviour: they *learn* how to behave in order to get the most out of the system.

Hence, we may conclude that there is a tension between on the one hand the user's expectations aroused by the fact that he can interact with a system by means of natural language, and on the other hand the limits that a formal specification of natural language poses on the possibilities of the interaction. This tension makes uncertainty an important issue to be dealt with.

Even disregarding everything not represented in the model, a dialogue system might still be confronted with uncertainty: situations of incomplete information may still occur. It might happen at times, that given the available information about the state of the dialogue, different alternative interpretations might still be considered possible by the system. This is a kind of uncertainty within the formal scope, stemming from *ambiguity*.

[2] Stokhof (2000) has published a very interesting overview.

1.3 Dealing with Uncertainty

There are two ways in which a dialogue participant can deal with uncertainty due to incomplete information. The first is by making an assumption by *directly* choosing between the alternatives and use that in the continuation of the dialogue. This choice may be based on principles varying from blind guessing to making more *educated guesses*, possibly using some measure of *plausibility* or *confidence*. Human dialogue participants do this very often, and in many cases they do that without even being aware of it.[3] The other way of dealing with uncertainty is by asking the speaker to rephrase his utterance, in stead of directly choosing one alternative interpretation. In that way, he can obtain additional information that might remove his uncertainty, or change his confidence in the various possible interpretations in such a way, that he is able to make more reliable assumptions. So, if the addressee considers one of the possible hypotheses to be much more plausible than the others (based on some threshold in the plausibility measure used), he may decide to use this hypothesis in the continuation of the dialogue; if no single hypothesis is plausible enough, he will try to obtain more information.

During the dialogue, a participant may conclude that assumptions made earlier on, now turn out to be wrong or less plausible. In the light of newly obtained information he may have reason to *revise* his assumptions. In the example dialogue below, participant B misheard the flight destination in A's utterance (A1), but he himself does not realise that at once. In processing the information obtained from A's utterance, he considers "Australia" to be most plausible and apparently decides it to be plausible enough not to verify it with A (B2). Later in the conversation however, B gets additional information that makes him revise his confidence in "Australia" (A4) and is has been able to recover from his mistake in the last utterance (B4).

B1 : *How can I help you?*

A1 : *I would like to go to Austria.*

B2 : *When would you like to leave?*

A2 : *On November 1.*

B3 : *There is a plane to Sydney leaving at 10am*

A4 : *No, I want to go to Austria, not Australia.*

B4 : *When would you like to leave for Austria?*

This thesis is concerned with how a conversational agent (more concretely, a dialogue system) can make educated guesses based on incomplete information when interpreting user utterances. The agent should be provided with a suitable formalism for processing new information in the light of his current

[3] This is not only the case in dialogue, but in all cases where human agents have perceptions of their environment; think for example of human vision.

knowledge. The formalism should enable the agent to decide whether or not verification is needed in cases of uncertainty, using some notion of plausibility. It should also support some way of detecting cases of new information causing previously made assumptions to be wrong and making consistent revisions in the agent's knowledge.

1.4 Bayesian Networks

We will argue for techniques based on probability theory to be the best way of dealing with uncertainty. We already indicated in Section 1.2 that in any formal model for natural language dialogue only a limited number of aspects are included and other possibly relevant aspects have been abstracted away from. Therefore the relations between the aspects in the model cannot be completely accurate. One way to account for this uncertainty is by adding a quantitative 'grey-scale' to the model, using some measure of plausibility assigned to propositions as already suggested in the previous section. Probabilities are very suitable for this purpose. Particularly, we will advocate the use of probabilistic models called *Bayesian networks*, sometimes referred to as *belief networks* or *probabilistic networks*. In such models, correlations between observable and unobservable aspects of some domain can be compactly represented by means of a graph structure. The nodes in the graph represent random variables and the arcs represent dependencies between those variables. To be more precise, the graph structure as a whole represents a set of conditional independency assumptions that give licence to a factored representation of the joint probability distribution of the model. With this factorisation, less probabilities need to be assessed for a complete specification of the probabilistic model and the computational complexity of inferences can be reduced.

Given some set of observations, the model can be queried for unobserved aspects an agent might be interested in. The results from these queries have the form of posterior probability distributions reflecting the plausibility of the possible scenarios, given the available information. The idea of educated guessing mentioned in Section 1.3 then amounts to choosing the scenario with the highest probability, provided that this probability is high enough to decide not to verify the hypothesis through information from further observations. A conversational agent can use a Bayesian network to take the partial information he has about an utterance of the user and about other aspects of the dialogue situation and determine how probable candidate interpretations of the utterance are, given the available information.

The construction of a Bayesian network consists of three parts:

1. choosing a set of random variables that describe the domain,

2. finding the graph structure that represents a set of conditional independencies, and finally,

3. assessing the numbers that are required for specification of the joint distribution.

One advantage of using Bayesian networks is that it is possible to combine expert knowledge and empirical data in constructing them. For example, given the set of variables, one could specify the graph structure for the network from expert knowledge and then assess the numbers from raw empirical data. One could also assess the graph structure from the data as well, or one could start with a complete Bayesian network obtained from expert knowledge and update it using the data.

In summary, Bayesian networks are probabilistic models that can be used to deal with uncertainty by making plausible assumptions based on incomplete information. Furthermore, in the absence of a complete specification of the domain at hand, the limited expert knowledge can be combined with empirical data to arrive at an approximation of the model that is statistically and logically sound.

1.5 Dialogue Acts

The central concept in our approach to dialogue modelling that will be discussed in great detail in the thesis, is the concept of *dialogue acts*. This notion has its origin in the findings laid down in various theories of language philosophy, emphasising that the meaning of expressions in language lies primarily in their *use*. Against the background of Wittgenstein's observations, later referred to under the slogan 'meaning-as-use', various other 'use-theories' of meaning evolved, including particularly the theory of *speech acts* due to Austin and Searle, which has become to be very influential in computational work on dialogue.

The notion of speech act stems from the insight that when a speaker produces a natural language utterance, therein he is also performing a conventional communicative act called *illocutionary act*. For example, producing the utterance "two tickets please" may be seen as a `request` by the speaker to the hearer to give him two tickets, while the utterance "I have two tickets" may be seen as an `assertion` or an `offer`. Although there are various other ways in which a speaker performs actions when producing an utterance (e.g. scaring, bugging or joking), the main focus of research has been on illocutionary acts. Nowadays, the term speech act usually refers to the illocutionary act.

The notion of *dialogue acts* has been introduced as an extension to speech acts. The traditional speech acts were only concerned with single utterances in isolation. However, the participants in a dialogue are involved in a process of communication, in which they have to coordinate their actions. Therefore, dialogue utterances are also used for maintaining a coherent interaction, involving issues such as grounding, turn-taking, social conventions and conversational structure.

Similar to the way in which speech acts are taken to be composed of an *illocutionary force* and a propositional content, dialogue acts are composed of a *dialogue act type* and a propositional, or more generally, *semantic content*. Different dialogue act types are organised in a *dialogue act taxonomy*. The choice for the types and the way a taxonomy is structured depends on a variety of factors, not only the (task-)domain or the physical circumstances of the dialogues involved, but also the research goals of constructing the taxonomy: is the aim purely linguistic or is there also a computational interest?

An important aspect in interpreting user utterances in terms of dialogue acts, is the task of *dialogue act recognition*. Given the various kinds of information the interpreter may have about an utterance, the state of the dialogue and the speaker's beliefs, desires and intentions, he has to choose which of the possible dialogue act types in the taxonomy is to be associated with the utterance. The experimental part of our research consisted of applying a machine-learning approach to dialogue act recognition. Using empirical data from a corpus, various *classifiers* for assigning dialogue act types to sets of *features* that can be extracted from an utterance in context, were trained and evaluated. In particular, Bayesian network classifiers were compared with other machine learning models and techniques.

1.6 Background

The Virtual Music Centre (VMC) is a virtual reality environment modelled after a real building called the 'Muziekcentrum', a musical theatre located in Enschede, The Netherlands. Users can explore this virtual environment from any computer connected to the Internet by moving around in the 3-dimensional space. This is done from a first person perspective: users do not see any avatar representing themselves when moving through the environment, but they see what they would see in reality when walking through an actual building. The environment is inhabited by a number of *embodied agents* the users can interact with. These are software agents that are visually represented in the environment; they can be receptive for communicative actions of users and possibly other agents, and are able to respond to these actions. In particular, interaction by means of multi-modal natural language dialogue might be supported: in the communication process, natural language information is integrated with non-verbal information like mouse-clicks on certain objects in the environment.

One of the agents is the navigation agent. This agent helps users move around through the building and provides information about various objects and locations in the environment (Luin et al., 2001). In Figure 1.1, a screen-shot of the interface is given. The screen contains the user's view from his current location in the environment, the course of the dialogue so far, and a more abstract, 2-dimensional view on the environment (from a 3rd person perspective). In (Hofs et al., 2003), a more recent version of the system is presented, introducing a speech interface and more advanced dialogue modelling, using dialogue acts.

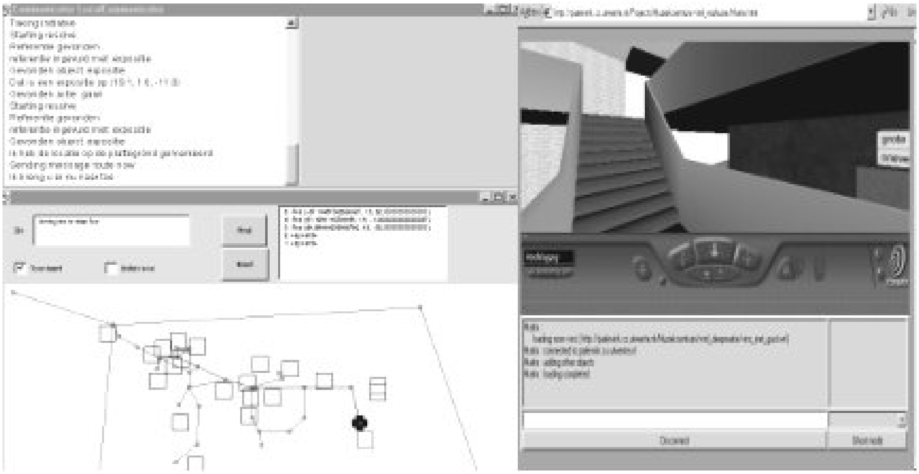


Figure 1.1: The navigation agent.

Another agent is the theatre information and booking agent 'Karin'. She provides information to users about theatre performances and can also make reservations if necessary. As in the case of the navigation agent, the dialogues with Karin are also multi-modal: besides the information given in the conversation itself, users can also get information about theatre performances by clicking on the graphical presentations of performance listings given by Karin during the dialogue (see Figure 1.2).

1.7 Objectives and Method

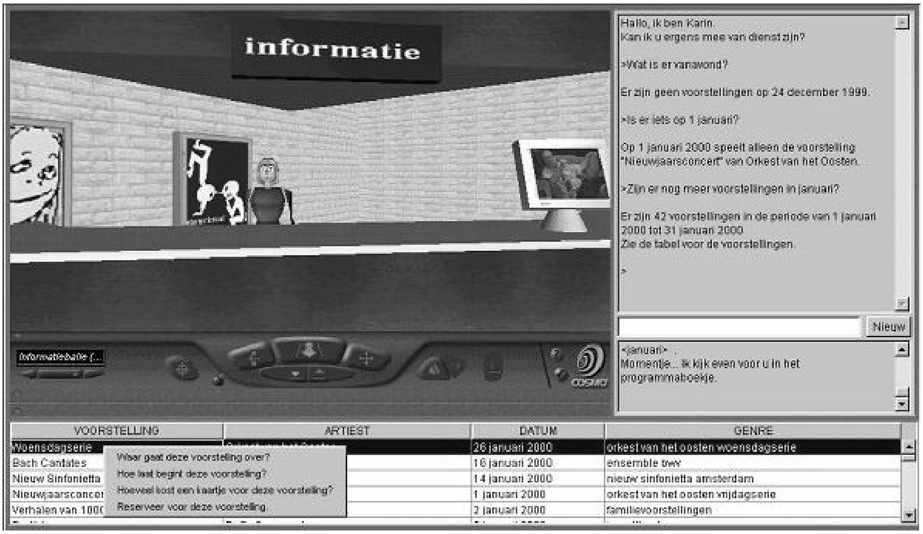
As a result of the general discussion in the previous sections, we can now formulate the following concrete questions we have aimed to answer in the research:

1. to what extent is uncertainty accounted for in current approaches to dialogue modelling, what techniques are used in current dialogue systems to deal with uncertainty, and how successful are those techniques?

2. what is currently known about the usefulness of probabilistic approaches and in particular, Bayesian networks, for dialogue modelling?

3. how can Bayesian networks be used in dialogue modelling and to what extent does this new approach improve a conversational agent? More specific questions in this respect are:

   (a) how can Bayesian networks be embedded into an architecture of a dialogue act based multi-modal dialogue system?



Hallo, ik ben Karin.
Kan ik u ergens mee van dienst zijn?

>Wat is er vanavond?

Er zijn geen voorstellingen op 24 december 1999.

>Is er iets op 1 januari?

Op 1 januari 2000 speelt alleen de voorstelling
"Nieuwjaarsconcert" van Orkest van het Oosten.

>Zijn er nog meer voorstellingen in januari?

Er zijn 42 voorstellingen in de periode van 1 januari
2000 tot 31 januari 2000
Zie de tabel voor de voorstellingen.

>

<januari> ..
Momentje... ik kijk even voor u in het
programmaboekje.

Waar gaat deze voorstelling over?
Hoe laat begint deze voorstelling?
Hoeveel kost een kaartje voor deze voorstelling?
Reserveer voor deze voorstelling.

VOORSTELLING	ARTIEST	DATUM	GENRE
Woensdagserie	Orkest van het Oosten	26 januari 2000	orkest van het oosten woensdagserie
Bach Cantates		16 januari 2000	ensemble bwv
Nieuw Sinfonietta		14 januari 2000	nieuw sinfonietta amsterdam
Nieuwjaarsconcer		1 januari 2000	orkest van het oosten vrijdagserie
Verhalen van 1000		2 januari 2000	familievoorstellingen

Figure 1.2: Karin.

(b) what dialogue act types should the system be able to distinguish between?

(c) what information is most relevant for the system to determine the dialogue act type in specific dialogue situations?

(d) how is the performance of Bayesian networks for this classification task in comparison with other techniques?

The work that has been done in order to answer the above questions, involved studies of 1) formalisms for dealing with uncertainty, including a thorough analysis of Bayesian networks, and 2) dialogue act theories, including the development of a dialogue act annotation scheme for the SCHISMA corpus. The annotated corpus has provided data for machine learning experiments concerning dialogue act classification. The experiments involved both finding features that are most informative for classification and comparing Bayesian network classifiers with other machine-learned classifiers. Software has been developed for 1) extracting the appropriate data-files from the annotated corpus for the machine-learning experiments, and 2) performing the machine-learning analyses with those data-files.

More generally, we have proposed a framework for using Bayesian networks for conversational agents, with the emphasis on combining different sources of evidence for interpreting user actions, in particular natural language utterances in terms of dialogue acts.

1.8 Overview of the Thesis

In Chapter 2, the phenomenon of natural language dialogue is discussed. The central viewpoint in this discussion is that of communicative action, leading to a dialogue modelling approach in terms of dialogue acts. The discussion includes some historical background, taxonomies of dialogue act types, dialogue act recognition, and the annotation of corpora with dialogue act types.

In Chapter 3, we discuss the phenomenon of uncertainty that was identified as an essential issue to deal with in modelling natural language dialogue, particularly in the process of dialogue act recognition. Uncertainty will be viewed as an epistemic phenomenon an agent is involved with. Various approaches concerning representation of and reasoning under uncertainty are described, of which the use of probability theory is the most important one. In addition, attention is given to machine learning techniques, used to derive models from empirical data.

Bayesian networks form a special kind of probabilistic models, to be discussed in Chapter 4. They will be defined as computational probabilistic models, in which probability theory and graph theory are combined. This way of defining is motivated by the model's foundation in probability theory and the visual and computational appeal of representing explicit assumptions of conditional independence. Further essential questions concerning Bayesian network models will be addressed, such as how they can be used and how they can be constructed. Additionally, Dynamic Bayesian networks, an extension of Bayesian networks, are discussed, some software tools for designing and developing Bayesian networks and an outline of common application areas.

Chapter 5 features the engineering side of dialogue, i.e., the development of dialogue systems, or, dialogue agents. The evolution of approaches to dialogue modelling will be outlined briefly, starting with the early primitive dialogue systems in the late 1960s. Additionally, we discuss two state of the art approaches to dialogue modelling that use some form of dialogue state, the BDI dialogue agent approach and the Information State approach. Then, an analysis is given of the way in which the problem of uncertainty is dealt with in dialogue systems, and finally, we will give a sketch of how Bayesian networks can be used within a dialogue system, especially in the light of the uncertainty problem.

The experiments that were performed concerning dialogue act classification with Bayesian networks are described in Chapter 6. This includes an outline of the annotation of the SCHISMA dialogue corpus, from which data have been extracted for the machine learning of the classifiers. The experiments were aimed at both finding relevant features for classification and comparing Bayesian network classifiers with other classifier types.

Finally, in Chapter 7 we draw some general conclusions.

Chapter 2

Dialogue Acts

In which the phenomenon of natural language dialogue is discussed. The central viewpoint in this discussion is that of communicative action, leading to a dialogue modelling approach in terms of dialogue acts. The discussion includes some historical background, taxonomies of dialogue act types, dialogue act recognition, and the annotation of dialogue corpora with dialogue act types.

2.1 Introduction

A natural language dialogue can be seen as a process of *communication* between two or more dialogue *participants*. Each of the participants tries to accomplish certain goals, essentially by sending messages to other participants, in the form of spoken or written natural language *utterances*. A dialogue is also a *joint activity* in which the participants try to *cooperate* in order to accomplish some *shared goal*. This shared goal can of course contribute to accomplishing the participants' individual goals. Consider for example Frank and Dave making plans to see a movie, discussing a place and time to meet and which movie to see. In the dialogue the participants make suggestions to each other, and in reaction, reject or accept suggestions. They also exchange information, particularly about the possible movies to see. So the shared goal is to make plans for seeing a movie, but this shared goal also serves the participant's individual goals, say, seeing an interesting movie, socialising with a friend, seeking some distraction from everyday work, etcetera.

As the accomplishment of the shared goal depends on the actions of both participants, the individual participants need to *coordinate* their actions. In achieving a shared goal, a participant is dependent of the actions of the other participant(s). In order to see a particular movie on a particular time and place, Frank needs Dave to agree with seeing that movie and meeting at that time and place. In other words, Frank and Dave need to have *mutual knowledge* about the movie and the time and place to meet. In fact, the participants need to obtain mutual knowledge about all sorts of things all through the dialogue, which actually already starts with getting each other's attention. In linguistics, this aspect of communication is called *grounding* and the resulting mutual knowledge is called the *common ground*.

The individual communicative actions of the participants can be either natural language utterances or non-verbal communicative actions such as gestures or facial expressions. If a speaker says "I want to see *that* movie", while pointing at an advertising poster of a particular movie, then the hearer will use information not only about the utterance itself but also about the pointing action and the object to which the speaker pointed, in order to get a complete interpretation. There are also non-verbal actions that are not really part of the communication process in the sense that they are not intentional, but may nevertheless influence the (interpretation of) communicative actions. For example, consider a teacher observing a student doing an exercise and giving instructions, where the student can also ask questions. Hence, there are all kinds of aspects of the given situation that may be relevant in the communication process.

If we shift our attention to an individual dialogue participant, we may see a dialogue as a special case of an agent in interaction with its environment. The agent in this case may be called a *dialogue agent* and its environment contains in particular the agent's dialogue partners (dialogue agents as well). Our dialogue agent *observes* his dialogue partner(s) performing (communicative) actions and plans his future (communicative) actions, based on the information obtained by these observations. If one of the participants says something to the agent, what happens is that the agent tries to figure out what the other participant *meant by* the utterance, based on the raw information the agent has about the utterance itself and other aspects of the circumstances in which the utterance was produced.

More specifically, a dialogue agent tries to find out what is the meaning of an utterance in a particular dialogue situation, in the broad sense of what was *intended* by the speaker. In general, an agent interprets observations from its environment using and updating his knowledge about that environment. In the case of natural language interaction (i.e. in a dialogue), this knowledge includes knowledge of the *conventions* that seem to underly the use of language and other, non-verbal, devices by which communication is made possible. Our approach to modelling dialogue is based on Austin and Searle's *speech act theory*, in which natural language utterances are analysed primarily in terms of usage and communicative action, in contrast to the ordinary syntax and truth-conditional semantics. The notion of *dialogue acts* is an extension of speech acts, where aspects beyond the scope of sentences in isolation are considered. This particularly involves the issue of how an utterance contributes to the dialogue as a whole, to a coherent interaction.

In the remainder of this chapter, we will first give some philosophical background concerning theories of language use in Section 2.2, involving particularly the theory of speech acts. In Section 2.2.4, this leads to the introduction of the notion of dialogue acts, which will be further elaborated in further sections on dialogue act taxonomies (Section 2.3), recognition of dialogue act types (Section 2.4) and the annotation of dialogue corpora with dialogue act types (Section 2.5).

2.2 Background: Theories of Language Use

2.2.1 Meaning As Use: Wittgenstein

In the first part of his 'Philosophical Investigations' (Wittgenstein, 1958)[1], Ludwig Wittgenstein (1889-1951) criticises the 'Augustinian view of meaning'. In this view, the (content) words of a language refer to things in reality. Sentences, being sequences of words with some underlying structure, refer to situations in reality built from the things that the words in the sentence refer to. In the sentence "John sees the moon", "John" refers to a person named "John", "sees" refers to an action and "moon" refers to a certain planet; the meaning of the sentence follows naturally: John is a person that performs the action of seeing and what he sees is the moon.

More generally, Wittgenstein criticises the 'mentalistic' viewpoint, in which understanding what an expression means consists of having a mental representation of its reference. The description of the meaning of the sentence "John sees the moon" above already suggested this: use is made of mental concepts such as PERSON, ACTION and PLANET.

Wittgenstein argues that the meaning of an expression is not given by what it refers to, but primarily by its use within context: 'meaning is use'.

> *For a large class of cases – though not for all – in which we employ the word "meaning" it can be defined thus: the meaning of a word is its use in the language.*
> *(Wittgenstein, 1958, I:43)*

Language is part of an activity, and Wittgenstein introduces the term *language game* (German: 'Sprachspiel') to refer to the entire complex of an expression used under particular circumstances with a particular purpose. Recall the example we gave in Section 1.1, where we sketched the scenario of a customer in a CD-shop intending to buy a CD he picked from one of the shelves in the store. This situation triggers a language game in which the dialogue participants have very specific roles, and utterances are used in a very specific way and the participants know – to some extent – how to play that game by the rules. In this way, the customer's utterance "this one please", in combination with presenting a particular CD, will make sense to the shop assistant (the customer indicates that he wishes to purchase that CD).

> *[. . .] I shall also call the whole, consisting of language and the actions into which it is woven, the "language-game".*
> *(Wittgenstein, 1958, I:7)*

[1] First published (posthumously) in 1953.

An interesting claim Wittgenstein makes, is that there are ′countless different kinds of use of what we call ″symbols″, ″words″, ″sentences″′. He illustrates this ′multiplicity of language-games′ with a list of examples, including ′giving orders, and obeying them′; ′reporting an event′; ′making a joke; telling it′; ′asking, thanking, cursing, greeting, praying′ (Wittgenstein, 1958, I:23). From this, we may conclude that Wittgenstein claims that any systematic classification of different uses of language can never be complete or accurate. This is an important point in our discussion on modelling dialogue in terms of language use, or more specifically, dialogue acts. We will return to this point in Section 2.4.3, where we discuss the notion of uncertainty in dialogue act recognition.

Against the background of Wittgenstein′s observations, several theories of meaning have emerged in which expressions are analysed primarily in terms of use, in stead of the traditional truth-conditions or reference. These theories are attempts to answer the questions of which kinds of language use can be distinguished and how these kinds of use can be described in terms of intended and actual effects.

2.2.2 Speaker′s Meaning: Grice

In his theory of meaning, H. Paul Grice (1913-1988) distinguishes *natural meaning* from *non-natural meaning*. Signs have natural meaning if there is a natural or causal relationship between two events or states-of-affairs. For example, dark clouds mean that it will probably start raining soon. Non-natural meaning is constituted by a conventional, intentional relationship. For example, hitchhikers use a particular form of gesturing (using their thumb) standing at the roadside, to make clear to car drivers passing by that they would like a ride. Linguistic expressions are typical examples of signs that have non-natural meaning. The meaning of a natural language utterance is given by this non-natural *speaker′s meaning*, and is different from the *sentence meaning*, which is given by the sentence itself, independent of the speaker′s beliefs and intentions.

Grice defines non-natural meaning of an expression x – verbal or non-verbal – produced by an agent S, addressed to an agent H, in terms of the effect e that S intends to achieve with H. For this intention to be the non-natural meaning (′meaning-nn′), Grice gives three conditions[2]:

I1: S has the intention to achieve e with H by means of producing x;

I2: S has the intention that H recognises I1 as such;

I3: S has the intention that I2 is instrumental in achieving e.

If John sneaks up on Mary and suddenly shouts ″relax!!″, John′s intention of giving Mary a fright is clearly not the meaning of the utterance. John intends to give Mary a fright by producing the utterance, but does not have the

[2] See Grice (1969); (Grice, 1991, Essay 5).

intention that Mary recognises this intention. Mary will be frightened because of the sudden noise John makes. Hence, I1 is satisfied in this case, but I2 is not.

Suppose Mary is talking to John, who is actually quite busy with other things and rather sees Mary leave. With the intention of getting rid of Mary, John asks Mary: "didn't you tell me you had some shopping to do?". Mary realises that she did and leaves. John may have the intention of Mary recognising his intention of getting her to leave, but she leaves John because she has some shopping to do. Therefore, condition I3 is violated, because Mary recognising John's intention of getting her to leave is not directly instrumental to the effect that Mary leaves John[3].

Now consider A's utterance in the example dialogue in Chapter 1, "I would like to go to Austria". A's intention of getting B to provide a ticket with destination 'Austria', cannot be seen as the direct meaning of the utterance, although I1-I3 have been satisfied.

Grice' definition of non-natural meaning adequately distinguishes between the effects of linguistic expressions that are related to the meaning and the effects that are not. However, it can not distinguish between the effects that are *directly* related to the meaning of a linguistic expression and those that are only *indirectly* related to that meaning.

2.2.3 Speech Acts: Austin and Searle

Austin

The main point that lies at the basis of speech act theory is the observation that not all utterances can be analysed in terms of truth conditions, in other words, not all expressions are *verifiable*. On the one hand, questions and commands are types of utterances of which we cannot say they are true or false (or imagine them to be true or false), but they can be easily filtered out from 'ordinary' statements on the grammatical level, via the general *sentence-types*: declarative sentences are ordinary statements, interrogative sentences are questions and imperative sentences are commands. On the other hand, many utterances with a declarative underlying sentence are not verifiable but are neither nonsense.

In the first part of 'How to Do Things with Words', (Austin, 1975)[4], John L. Austin (1911-1960) makes a distinction between *constative* and *performative* utterances. Constatives are statements that are verifiable, mostly descriptions of some event or state of affairs. Performatives are utterances that do not describe anything at all, are not true or false, and producing such utterances is part of the doing of the action. An example performative utterance is:

(1) *I name this ship the Queen Elizabeth.*

[3] A classic example is that of St. John the Baptist: Herod presents Salome with the head of St. John the Baptist to produce the belief that he is dead.

[4] Austin delivered this series of twelve lectures as the William James Lectures at Harvard University in 1955, but they were first published only in 1962.

for which producing the utterance *is* performing the action (of naming), while on the other hand, for the constative utterance:

(2) *He named this ship the Queen Elizabeth.*

producing the utterance *describes* the action (of naming).

As an extension to the notion of truth-conditions, Austin formulates a number of conditions for a performative to be 'happy'. These so-called *felicity conditions* concern the appropriateness of the circumstances in which the utterance is made and the participants involved. The conditions also serve as a classification of the ways in which something can go wrong in producing an utterance:

> *What these are we may hope to discover by looking at and classifying types of case in which something* goes wrong *and the act – marrying, betting, bequeathing, christening, or what not – is therefore at least to some extent a failure: the utterance is then, we may say, not indeed false but in general* unhappy.
>
> *(Austin, 1975, Lecture II)*

In the Queen Elizabeth example (1), the speaker has to be entitled to name the ship; otherwise the ship is not named at all and therefore the utterance is called *infelicitous*. In Austin's classification of infelicities, it is called a *misapplication*.

In the second part of (Austin, 1975), he elaborates his ideas about constatives and performatives and comes with a theory in which the distinction between the two is replaced by a more general distinction between the *locutionary* and *illocutionary* act. In fact, he distinguishes three different ways in which an utterance may be used, or 'senses' of using an utterance (Austin, 1975, Lecture VIII).

- the *locutionary act*: the act **of** uttering a sentence, i.e., the physical act of producing noises (the *phonetic act*), thereby producing words according to certain – grammatical – rules (the *phatic act*) and using that with a particular sense and reference (the *rhetic act*);
- the *illocutionary act*: the act **in** uttering a sentence; it expresses the communicative force of the utterance;
- the *perlocutionary act*: the act **by** uttering a sentence; it expresses the effect achieved in the particular situation.

Illocutionary acts are acts such as asserting something, asking a question, or giving orders. An illocutionary act is taken to be composed of an *illocutionary force*, specifying the type of the function of language use (that can be indicated by an explicit performative verb, e.g., assert, ask, command or promise), and the *propositional content*, specifying what is asserted, what is asked, what the hearer is commanded to do, or what the speaker is promising to do.

Perlocutionary acts are acts such as convincing, or frightening. Such acts describe effects that are achieved as a consequence of producing the utterance in the given circumstances, while illocutionary acts describe effects that are achieved by convention, in Austin's words, 'saying makes it so'. One method to distinguish illocutionary acts from perlocutionary acts is based on the assumption that the illocutionary force of any utterance can be made explicit by paraphrasing the original utterance U with a sentence in the first person indicative active in the simple present, using the adverb "hereby" (and possibly the personal pronoun "you" or the phrase "to you"), and an explicit performative verb PV that will correspond to the illocutionary force:

(3) *I hereby* $\langle PV \rangle$ *you that* $\langle U \rangle$.

For example, the utterance "it's warm in here" can be paraphrased by "I hereby assert to you that it is warm in here", but not with "I hereby convince you that it is warm in here". So, the act of assertion is an illocutionary act, but the act of convincing is a perlocutionary act.

Austin points out however, that besides locutionary, illocutionary and perlocutionary acts, there are yet other senses of language use, for example 'joking' or 'playing a part', and moreover, there are acts that cannot be clearly identified as either illocutionary or perlocutionary acts, e.g., 'insinuating'.

By using the 'hereby-method', Austin suggests that the space of illocutionary forces can be explored on the basis of performative verbs in the vocabulary of a language. Moreover, he suggests five preliminary classes of illocutionary forces, that should evolve 'naturally' from such an exploration:

1. *verdictives*: give a verdict about something, e.g., christening, sentencing (to 5 years of prison);

2. *exercitives*: exercise a right, e.g., appointing, voting;

3. *commissives*: commit the speaker to do something, e.g., promising, announcing an intention;

4. *behabitives*: concern attitudes and social behaviour, e.g., apologising, congratulating;

5. *expositives*: concern how an utterance fits into a conversation or argument, e.g., clarifying, illustrating.

The work of Austin on illocutionary acts is further developed by Searle. He improved Austin's classification of illocutionary forces, that was too much lead by the exploration of performative verbs, and therefore failed to serve as a classification for illocutionary acts in general. However, we will start our discussion of Searle's work with his idea's on language meaning in relation to Grice and Austin.

Searle

In reaction to Grice' and Austin's theories of meaning, John R. Searle gives a counter-example, in which the Gricean conditions for non-natural meaning are satisfied and also the intended effect of the utterance is illocutionary, but in which we cannot possibly identify this with the meaning of the utterance itself: suppose, in World War II, an American officer is imprisoned by the Italians and tries to tell the Italians that he is German. He knows one sentence in German and says to the Italians: ″Kennst du das Land wo die Zitronen blühen?″. The intended effect of making the Italians believe he is from Germany is recognised by the Italians as such and this is also instrumental to achieving the intended effect. The intended effect is also claimed to be illocutionary: it is an assertion. However, the method of paraphrasing mentioned above (the ′hereby-method′) seems to be problematic in this case.

Searle adds the notion of *convention* to his analysis of meaning, where Grice analyses meaning entirely in terms of intentions and effects. Using the notion of convention, Grice′ second condition for non-natural meaning can be modified to:

I2′: S has the intention that H recognises I1 as such through a conventional association between producing x and achieving e.

The notion of convention recovers a connection of the speaker meaning with the linguistic meaning that seems to be missing in Grice′ conditions for non-natural meaning. Clearly, there is no conventional association between producing the utterance "Kennst du das Land wo die Zitronen blühen?" and the effect of the speaker asserting he is from Germany. This assertion might be the speaker′s intention, but the speaker′s meaning is nothing more than a question.

Searle has concentrated his work on the systematisation of the illocutionary acts as introduced by Austin. He aims for a set of necessary and sufficient conditions for illocutionary acts to be performed successfully and non-defectively, a further development of Austin's felicity conditions. This has lead to rules for using linguistic clues to perform illocutionary acts, such as performative verbs, the sentence type and certain adverbs or particles (e.g., "certainly" or "please"). These clues are called *illocutionary force indicating devices*, referred to as IFIDs. In contrast to Austin's felicity conditions that indicate the circumstances in which illocutionary acts are unsuccessful and/or defective, Searle's rules are constitutive for the performance of illocutionary acts: using certain IFIDs in a certain way *counts as* performing a certain illocutionary act. So, Searle was looking for constitutive rules for using IFIDs to achieve illocutionary effects, like the rules of a game.

Four types in which the rules can be classified were defined by Searle: 1) rules concerning the propositional content, 2) preparatory rules, 3) sincerity rules, and 4) the essential rule. As an explanatory example, the rules for speaker S performing a REQUEST addressed to the hearer H along these dimensions, are given:

1. *propositional content*: future act a of H;

2. *preparatory*: 1) H is able to do a and S believes that H is able to do a and 2) it is not obvious to both S and H that H will do a in the normal course of events of his own accord;

3. *sincerity*: S wants H to do a;

4. *essential*: counts as an attempt to get H to do a.

As an alternative to Austin's classification, Searle defines five classes of (illocutionary) speech acts:

1. *representatives*: commit the speaker to the truth of the expressed proposition, e.g., asserting or concluding;

2. *directives*: attempts by the speaker to get the addressee to do something, e.g. requesting, questioning;

3. *commissives*: commit the speaker to some future course of action, e.g., promising, threatening or offering;

4. *expressives*: express a psychological state, e.g., thanking, apologising, congratulating;

5. *declaratives*: have immediate effect on the current state of affairs, e.g., declaring war, christening, marrying, firing from employment.

These speech act classes are established on criteria including the illocutionary force, the mental state of the speaker and the propositional content of the utterance. Especially concerning the speech acts of asserting and requesting were problematic in Austin's classification. An assertion could fall into any of Austin's categories, while a request is not covered by his categories at all.

So Searle's suggestion is that in each utterance, a speech act from one of five possible speech act classes is performed. However, he also observed that besides a so-called *direct speech act* also an *indirect speech act* may be performed that supersedes the direct speech act (Searle, 1975). In contrast to the direct speech act, the indirect speech act is not directly linked to the sentence type of the underlying sentence, i.e., it indicates a *non-literal use* of the expression. A few examples will clarify this phenomenon (the direct and indirect speech act are given between parentheses, in that order):

(4) *Can you pass the salt?* (yn-question/request)

(5) *I must ask you to leave* (statement/request)

(6) *I must ask him to leave* (statement/none)

In (4), the direct speech act 'yes/no-question' is related to from the interrogative sentence type. However, in most circumstances, the expression is used indirectly as a 'request', making the direct speech act secondary. In (5), the direct 'statement' is superseded by the indirect 'request'. However, not all expressions can evoke indirect use, as (6) shows. Searle points out, that expressions can be used indirectly if they address aspects specified in the felicity conditions for the indirect speech act. Expression (4) can be used indirectly as a request, because it addresses the preparatory condition for requests (the hearer is able to perform the act).

2.2.4 Dialogue Acts

In the previous section, we discussed speech act theory, in which natural language utterances were to be analysed in terms of actions, in stead of merely a sense and reference. However, the different possible speech acts that Searle specifies only refer to the use of single utterances in isolation of the context of a dialogue. That is to say, the realisation of the action expressed by the speech act may be determined by the context, but the speech acts themselves do not involve aspects concerning the coherence of the dialogue as a whole. As participants in a dialogue are actively involved in coordinating their actions, this should be accounted for in any dialogue model, and hence, any model based on some notion of speech acts.

Therefore, we have adopted the term *dialogue acts* to distinguish it from speech acts in the traditional – i.e., Searlean – sense, following for example, Traum (e.g., Poesio and Traum, 1998) and Bunt (1995, 1996). Others have used different terms, such as communicative acts (Allwood, 1976), dialogue moves (Kowtko et al., 1992), conversation acts (e.g., Traum and Hinkelman, 1992) and also the original term speech acts. All of them refer to basically the same concept and vary in their approaches with respect to application area, domain, formal treatment, etcetera.

As an illustration, consider utterances such as “yes”, or “on November 1”. Given some context, these utterances can be paraphrased by (grammatically) complete sentences like “yes, I would like that” respectively “I would like to leave on November 1”, revealing that they are – in this case – assertions. However, one can imagine that the utterances are also contributions to the coherence of the dialogue: “yes” might be an answer to a previously made question or request, and “on November 1” might be an answer to a previous question (“when would you like to go?”). On the other hand, “yes” might also be used by the speaker to indicate that he has understood what the other participant just told him:

> *then you turn left . . .*
>
> *yes*
>
> *ok, and then it's only three blocks to the main building.*

In the next sections, we will further develop the notion of dialogue acts, with a particular emphasis on dialogue act taxonomies, and then introduce the task of dialogue act recognition, where we will of course especially address the issue of uncertainty. One approach to deal uncertainty in the task of dialogue act recognition is using data from an annotated corpus to train dialogue act classifiers for this task. In a separate section, the annotation of dialogue corpora with dialogue act types is discussed.

2.3 Dialogue Act Taxonomies

Most speech act research has focussed on illocutionary acts. A substantial part of that work consists of making inventories of the types of illocutionary force that may occur in language. In the case of dialogue acts as introduced in Section 2.2.4, we will refer to such types as *dialogue act types* [5]. Such dialogue act types can be further structured into a number of categories like the two categorisations given by Austin and Searle (see Section 2.2.3). Each utterance will – hopefully – be of a particular speech act type and hence fall into one of the categories.

A more general and further refined system of dialogue act types and categories of dialogue act types can be specified by means of a *dialogue act hierarchy*, in the form of a lattice or tree structure. Dialogue act types that are further down the structure give a more specific account of a dialogue act performed in an utterance than the types on higher levels. For example, a dialogue act type `confirmation` gives a more specific account of the dialogue act performed in an utterance than a type `answer` does of that same utterance (see Figure 2.1). The nodes in a hierarchy may represent both individual types and classes of types. The categories are always represented by nodes higher up the lattice structure than the ones that represent the types themselves.

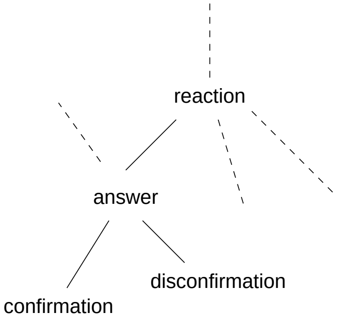


Figure 2.1: Detail of a dialogue act hierarchy.

Besides distinguishing between general types and more specific subtypes, various dialogue act taxonomies distinguish between different ways in which an utterance may contribute to the dialogue. Some dialogue act types may

[5] Again, different variations on this concept are around, e.g., speech act types, conversational move types, or communicative functions.

specifically refer to the task the dialogue participants are involved with, e.g., the task of making an appointment; other dialogue act types may be used to refer to the communication as such, e.g., to indicate that an utterance serves as an acknowledgement of what the other participant previously said, or that an utterance serves as giving feedback to the other participant, etcetera. Often, these different ways of contribution occur simultaneously in a single utterance. In that case, the taxonomy allows for assigning multiple dialogue act types to single utterances, each type accounting for one aspect of the contribution. This feature is commonly referred to as *multi-functionality*[6]. In several taxonomies that allow multi-functionality, the dialogue act types are organised into different *layers*, each layer containing a hierarchy of types that refer to one aspect of contribution. Other taxonomies that allow multi-functionality do not make use of such a layered system, but have to make explicit – in a more ad hoc way – which types may occur simultaneously and which may not.

Using the two general structural principles in a dialogue act taxonomy that were discussed above, viz. layeredness and hierarchy, a system of dialogue acts can be developed. In the process of specifying and defining the dialogue acts in the taxonomy, several considerations play a role (see Traum, 2000). For example, if one chooses dialogue acts purely based on intuitions concerning natural language use, one may find difficulties in developing a system of dialogue acts with desirable formal properties. Vice versa, for such desirable formal properties some properties of the intuitive concept may have to be sacrificed.

In the following sections, we will discuss two dialogue theories that make use of some notion of dialogue acts. We will especially focus on the taxonomy used in each theory.

2.3.1 Dynamic Interpretation Theory

In Dynamic Interpretation Theory (DIT) (see Bunt, 1995, 1996), dialogue acts are defined as *functional units* used by the speaker to change the context. Two main classes of dialogue acts are distinguished: *task-oriented acts*, which are motivated by the underlying task or goal of the speaker, and *dialogue control acts*, which are motivated by communicative goals, constructed in planning a way to achieve a non-communicative goal (as part of the underlying task). Dialogue control acts may also be motivated by another factor, *social obligation*, as natural communication is a social activity, in which certain norms and conventions will have to be observed.

Similar to the distinction between illocutionary force and propositional content made by Austin and Searle, dialogue acts in DIT consist of two components: the *semantic content* s, which contains the information relevant for the new context, and the *communicative function* F, which specifies the way in

[6] In a way, an indirect speech act (Section 2.2.3) also occurs simultaneously with the corresponding direct speech act, but the former supersedes the latter and therefore, they do not represent a form of multi-functionality in the way described above.

which the context is updated with that information. The complete dialogue act is then denoted $F(s)$.

The task domain assumed for this theory is that of information-exchange, which leads to a further elaboration of task-oriented acts into `info-seeking` and `info-providing` dialogue acts. For other task-domains, a different hierarchy of more specific task-oriented dialogue acts will have to be developed. The dialogue control acts on the other hand, are intended to cover any type of dialogue, in any domain.

The hierarchies for both classes of dialogue acts consist of both communicative functions – denoted in CAPITALS – and classes of communicative functions – denoted in `typewriter` font. A communicative function F being a subtype of a communicative function G in the hierarchy means, that given some semantic content s, the dialogue act $F(s)$ conveys more information about the speaker than dialogue act $G(s)$.

Task-oriented Acts

We will now schematically describe the task-oriented communicative functions in the case of information-exchange (where S denotes the speaker and H the hearer).

1. `information-seeking` communicative functions:

   - YN-QUESTION: S wants H to tell him if something (specified in the semantic content of the dialogue act) is the case or not ("are there any planes leaving for Frankfurt?").
     - CONTRA-CHECK: S has a weak belief that the semantic content is false ("are there planes leaving for Frankfurt, then?").
     - CHECK: S has a weak belief that the semantic content is true.
       - POSI-CHECK: S verifies with H if something is true ("there are planes leaving for Frankfurt, right?").
       - NEGA-CHECK: S verifies with H if something is false, in surprise because he believed otherwise ("aren't there any planes leaving for Frankfurt, then?").
   - WH-QUESTION: S wants H to give him references of objects satisfying a given specification. ("which planes leave for Frankfurt?")
   - ALTS-QUESTION: S offers H alternative answers to choose from. ("economy or business class?")

2. `information-providing` communicative functions:

   (Some of the functions appear in an additional, less specific 'weak' variant, implying that the speaker *suspects* the information provided, in stead of *knowing* it.)

   - YN-ANSWER: answer to a YN-QUESTION
     - CONFIRM: S states that the semantic content of H's YN-QUESTION is true.

    - DISCONFIRM: S states that the semantic content of H's YN-QUESTION is not true.
- WH-ANSWER: S gives H the references H asked for in his WH-QUESTION.
- INFORM: S lets H know something
    - AGREEMENT: S informs H of S's agreement with something
    - DISAGREEMENT: S informs H of S's disagreement with something
        - ∗ CORRECTION: S informs H of his disagreement by means of a correction.

Dialogue Control Acts

The class of dialogue control acts consists of three subclasses: feedback, interaction management and social obligations management acts. Again, we will give a concise overview of all these dialogue control acts.

1. `feedback`:

    - `auto-fb`: S gives H information about S's processing of H's input:
        - `positive`: "uh-huh"
        - `negative`: "what?"
    - `allo-fb`: S gives H information about H's processing of S's input:
        - `positive`: "correct!"
        - `negative`: phrase substitution
        - `eliciting allo-fb`: "OK?"

    (The feedback communicative functions above may all be further specified by one of the four functions `perception`, `interpretation`, `evaluation` and `execution`.)

2. `interaction management` (in fact, all dialogue control functions not concerning feedback or social obligations):

    - `own communication management`: S tries to modify the utterances he himself made before.
        - RETRACTION: S takes back what he said before: "of nee, doe maar ..." *(Eng.: no, make it ...)*
        - SELF-CORRECTION: S makes a correction on what he said before: "ik bedoel, ... " *(Eng.: I mean, ...)*
    - `time management`: S tries to gain time for his own actions.
        - PROTRACTION: utterances for gaining a relatively small amount of time. ("eh, ... ")
        - PAUSE: utterances for gaining more time: "momentje, ik zal het voor u nakijken" *(Eng.: just a moment, I will check it for you)*
    - `turn management`:
        - TURN GIVING: S gives H the opportunity to produce the next utterance.

    - TURN TAKING: S takes the turn from H, by interrupting him or in response of H's act of TURN GIVING.
    - TURN KEEPING: S makes sure that he can continue his turn with further utterances; often in combination with PAUSE.
  - `contact management`: concerning aspects of presence and attention of the participants.
    - PRES/ATT -INDICATION: S notifies H that he is present and paying attention to H.
    - PRES/ATT -CHECK: S checks if H is present and paying attention.
    - PRES/ATT -REQUEST: S requests H to pay attention.
  - `discourse structuring`:
    - DIALOGUE ACT ANNOUNCEMENT: "I have a question: ..."
    - topic management:
      - TOPIC INDICATION
      - SHIFT ANNOUNCEMENT
      - CHANGE INDICATION

3. `social obligations management`: acts stemming from norms and conventions in communication:

   - `self-introduction`: "hi, my name is ..."
   - `greeting`
     - `opening`
     - `farewell`

     Both classes of greeting functions have an initiative variant (INIT) and a reactive variant (REACT).
   - `apology`
   - `gratitude expression`: this class has an initiative variant ("thanks") and a reactive ("you're welcome") variant as well.

Context change

We conclude this section with some short remarks on how the dialogue acts in the hierarchy change the context. In DIT, the context is characterised by five different aspects, i.e., 'categories of factors, relevant to the understanding of communicative behaviour' (Bunt, 1995):

1. **Linguistic Context**: the surrounding linguistic material;
2. **Semantic Context**: the current state of the underlying task;
3. **Cognitive Context**: the participants state of processing and models of each other's states;
4. **Physical and Perceptual Context**: the availability of communicative and perceptual channels; the partner's presence.

5. **Social Context**: the communicative rights, obligations and constraints of each participant.

We will not give an extensive description of the process of updating this context by a communicative function, given a semantic content, but we will conclude with the remark that every dialogue act changes the cognitive context, but task-oriented acts also change the semantic context; dialogue control acts change the social or physical context, but not the semantic context.

2.3.2 Conversation Act Theory

Another theory of dialogue in which a distinction is made between dialogue acts performed to further certain goals the participants have with a dialogue and dialogue acts performed to make the dialogue go smoothly, avoiding or recovering misunderstandings and following certain social conventions, is the multilevel theory of *Conversation Acts* (CAT) of Traum and Hinkelman (1992), further developed by Poesio and Traum (1997, 1998).

Traum and Hinkelman (1992) distinguished four classes or 'levels' of conversation act types. These levels have in fact later been adopted as 'annotation layers' in the DRI/DAMSL dialogue act coding scheme, which will be described extensively in Section 2.5.4. The original taxonomy contains the levels of *turn-taking* acts for coordinating who is speaking, *grounding* acts for coordinating the flow of mutual understanding, the *core speech acts* as the traditional illocutionary acts, and *argumentation* acts for managing the higher-level conversational structure. Here are some sample acts for the levels:

- **Turn-taking**: take-turn, keep-turn, release-turn, assign-turn.
- **Grounding**: initiate, continue, ack, repair, reqRepair, reqAck, cancel.
- **Core Speech Acts**: inform, ynq, check, eval, suggest, request, accept, reject.
- **Argumentation**: elaborate, summarize, clarify, q&a, convince, find-plan.

From this original four level taxonomy, a multi-level dialogue act taxonomy has been developed (Poesio and Traum, 1997, 1998). In this taxonomy, an additional level is defined for the locutionary acts. Starting with this level, we will discuss this taxonomy very briefly, as it is based on the taxonomy underlying the DAMSL annotation scheme, to be discussed in Section 2.5.4.

Locutionary Acts A locutionary act is characterised by the ternary predicate: $e : \mathtt{Utter}(\mathbf{A}, \mathbf{P})$, where A is an individual, P is a string, and e is an eventuality.

Core Speech Acts The core speech acts have evolved from a re-interpretation of the classic speech acts (i.e., illocutionary acts), in that the acts are viewed as joint actions: the acts only take on their full effect if they are *grounded*.

- **Forward-looking acts**: these are acts that introduce new social attitudes in the conversation that have to be addressed:
  - *Statement*: Assert, Reassert, Other-statement
  - *Influencing-addressee-future-action*: Open-option, Directive (Action-directive or Info-request)
  - *Committing-speaker-future-action*: Offer, Commit
  - *Conventional*: Opening, Closing
  - *Explicit-performative*
  - *Exclamation*
- **Backward-looking acts**: these are act that give a response to previous acts:
  - *Agreement*: Accept, Accept-part, Maybe, Reject, Reject-part, Hold
  - *Answer*

Grounding Acts These are acts that are performed for grounding the contributions made by either of the dialogue participants.

- **Understanding**:
  - *Signal-non-understanding*
  - *Signal-understanding*: Acknowledge, Repeat-rephrase, Completion
  - *Correct-misspeaking*

Turn-taking Acts These are acts for managing who has the turn at a given moment. These are currently not in the DRI/DAMSL scheme:

- *take-turn*
- *keep-turn*
- *release-turn*
- *assign-turn*

Argumentation Acts These acts involve the macro-structures of conversation, like games or rhetorical discourse structure.

During the dialogue, both of the dialogue participants keep track of the *Conversational Information State* (CIS), in which grounded conversational acts and also ungrounded contributions are recorded. A CIS is typically characterised by a feature structure containing embedded feature structures for both dialogue participants (see also Section 5.2.2). Again, we will not go further into the details of how updating of these structures on the basis of dialogue acts is done; for now, our main interest lies in the speech act/conversational act/dialogue act classifications.

2.3.3 Discussion

In Sections 2.3.1 and 2.3.2, we gave a sketch of two dialogue theories, which are based on some notion of speech acts. DIT is concerned with *Dialogue Acts*, which are seen as functions, that change the context. CAT, on the other hand, is concerned with *Conversational Acts*, which are seen as contributions which have to be grounded (acknowledged): they have yet to become part of the common ground. The information about grounded and ungrounded contributions is modelled by Information States, which can be characterised by feature structures.

Although both theories stress that dialogue acts for controlling the interaction and for following certain social conventions are just as important as task-oriented dialogue acts for furthering the task/goals at hand, only in DIT the distinction between these types of acts has been made explicit in the dialogue act hierarchy. On the other hand, the organisation of conversation acts in CAT leaves room for developers to extend the hierarchy at various positions with domain/task-specific acts. In particular, information-seeking acts may be defined as special cases of `info-requests` and information-providing acts will be `statements` on the one hand, and `answers` on the other.

However, in the DAMSL annotation scheme, an additional Information Level has been introduced, that accounts for explicitly labelling utterances as being `task`, `task-management` or `communication-management`. This may serve as a way to distinguish between task-oriented dialogue acts (`task` in DAMSL) and dialogue control acts (`communication-management`in DAMSL) as is done in DIT, and even adds an additional refinement in using the `task--management` label (see Section 2.5.4 for the exact explanations of these labels).

Furthermore, we should note that CAT uses a multi-level approach, giving a structural guideline for multi-functionality. DIT only uses the structural distinguishing between task-oriented dialogue acts and dialogue control acts to be performed simultaneously, hence only two levels. At a more specific level, CAT explicitly distinguishes between dialogue acts that refer to previous utterances and acts that 'look forward' on the level of core speech acts, DIT makes such a distinction in the hierarchy of task-oriented acts, in terms of the task-specific information-seeking and information-providing acts.

Feedback

With respect to acts for controlling the interaction, the DIT hierarchy contains a branch for feedback, distinguishing not only between positive and negative feedback, but also between feedback with respect to the hearer's processing of the speaker's utterance (allo-fb) and the speaker's processing of the hearer's utterance (auto-fb), while CAT contains `understanding` acts, which only cover positive (`signal-understanding`) and negative feedback (`signal-non-understanding`. Where DIT has further categories for indicating at which level the feedback is taking place (perception, interpretation, evaluation and execution), CAT doesn't seem to cover this distinction. Again, at the Information Level of the DRI scheme, the `communication-management` label is used for utterances that concern the understanding at the level of interpretation (in contrast to perception).

Turn-taking

The second topic we will discuss, is *turn-taking*: how do participants manage not to speech simultaneously all the time. In DIT, three communicative functions are distinguished, which are used for turn-taking: TURN-TAKING, TURN-GIVING and TURN-KEEPING. In CAT, the level of Turn-taking Acts is described: here, four different act are distinguished, where the extra act constitutes a further refinement of turn-giving: either by 'releasing' the own turn, giving the other the *opportunity* to take the turn, or by 'assigning' the turn to the other, to a certain extent *making* the other take the turn.

Topic-management

Another issue is management of the *topic* of the conversation: how do participants manage to maintain common knowledge of what the dialogue is about and using that common knowledge in their utterances. In the DIT hierarchy `discourse structuring` acts and more specifically, `topic management` acts are contained, which can be further specified by three possible communicative functions: TOPIC INDICATION, SHIFT ANNOUNCEMENT and CHANGE INDICATION. In CAT, there is no mention of topic-management, although there is the level of Argumentation acts, which may contain notions of discourse structuring, which could also include topic-management.

Social conventions

Finally, there is the issue of social conventions, that underly natural interaction and that the participants follow in order to make the dialogue go smoothly. DIT describes a number of `social-obligations-management` acts: `self--introduction`, `greeting` and `apology`. CAT only describes the two more general conventional acts: `conventional-opening` and `-closing`.

2.4 Dialogue Act Recognition

In this chapter we have been discussing the various ways in which expressions in language can be used, in particular, utterances in a dialogue. This has lead to the notion of dialogue acts as an extension of Searle's speech acts, where the different ways in which an utterance can contribute to a coherent interaction have been incorporated. We discussed two general theories of dialogue acts, thereby focussing on the dialogue act taxonomy.

Now, the question rises how a hearer identifies the dialogue act that a speaker performed in producing a natural language utterance in a particular dialogue situation. As a dialogue act is taken to be composed of a dialogue act type and a semantic content, this can be reformulated as the classification task of recognising the *dialogue act type* given the speaker's utterance.

It is not very surprising that Searle's work on the systematisation of illocutionary acts in a very formal way has been of great influence to technological approaches to dialogue, in which a formal account of dialogue phenomena is essential. His view that speakers follow conventional rules to use illocutionary force indicating devices (IFIDs) in order to perform a speech act (see Section 2.2.3) has a direct association with a computational account of dialogue act recognition: given the observable IFIDs in an utterance, i.e., the *input*, what is the illocutionary force, i.e., the *output*? This is illustrated in Figure 2.2, where the input to the recognition module (DAR) is an utterance u and the output a dialogue act (d, sc), consisting of a dialogue act type d and semantic content sc.

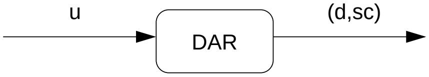


Figure 2.2: Dialogue Act Recognition (DAR).

The first IFID that comes to mind is the already in Section 2.2.3 mentioned *sentence-type*. At the grammatical level, this feature can be derived relatively easy and seems to directly tell us what is the illocutionary force:

SENTENCE-TYPE	ILLOCUTIONARY FORCE
declarative	statement
interrogative	yes/no- or wh-question
imperative	command

However, this view is complicated by the fact that these speech acts are often superseded by an indirect speech act (Searle, 1975). For example, interrogative sentences are often used indirectly as requests, in stead of yes-no-questions ("can you tell me where is room A?"). Additional IFIDs may be used to resolve this problem, e.g., the surface pattern "Can you X?" is an indication for an indirect request. On the other hand, Searle (1975) proposed an approach that is based on the view that the hearer *infers* the indirect speech act. This *inferential approach* has been further developed by AI-oriented researchers into a *plan* *inference* model based on beliefs, desires and intentions. Jurafsky and Martin (2000) distinguish this model from their own *cue-based model*, in which utterances are taken as a set of *cues* to the speaker's intentions. In the following two subsections, we will discuss both approaches in more detail.

2.4.1 The Plan Inference Model

The plan inference model[7] for dialogue act recognition is primarily aimed at the resolution of indirect speech acts. The model elaborates on Searle proposing that the hearer first recognises the direct speech act on linguistic grounds, and then *infers* the indirect speech act via a *chain of inference*. Using the general AI model of *Beliefs, Desires and Intentions* (BDI), Allen, Cohen and Perrault – among others – developed a model for speech acts (Cohen and Perrault, 1979; Perrault and Allen, 1980; Allen and Perrault, 1980). This model contains action specifications for speech acts and a set of *plan inference rules* that a hearer uses to infer an indirect act from a *surface-level act* (the direct speech act). Figure 2.3 gives a schematic view of the model: a DAR module is now used to determine a surface-level dialogue act (sd, sc), from which the actual dialogue act (d, sc) is inferred, using a BDI model.

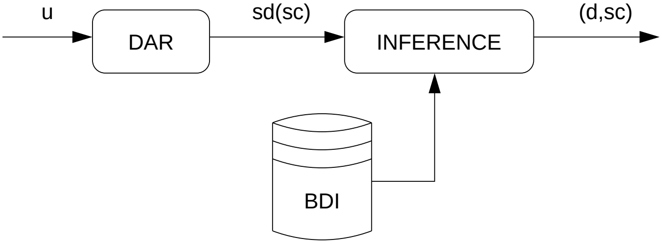


Figure 2.3: DAR: the plan inference model.

To make the plan based model a little bit more concrete, we give speech act specifications for a REQUEST and a surface-level request, S.REQUEST, below:

$\mathbf{REQUEST(S, H, A)}$:	
Constraints:	$Speaker(S) \wedge Hearer(H) \wedge Act(A) \wedge H$ is agent of A
Precondition:	$Want(S, Act(H))$
Effect:	$Want(H, Act(H))$
Body:	$Belief(H, Want(S, Act(H)))$

$\mathbf{S.REQUEST(S, H, A)}$:	
Effect:	$Belief(H, Want(S, Act(H)))$

An example of a plan inference rule is the 'Action-Effect Rule':

[7] We refer to Jurafsky and Martin (2000) for a more detailed overview of the plan inference model.

> **(PI.AE) Action-Effect Rule**: For all agents S and H, if Y is an effect of action X and if H believes that S wants X to be done, then it is plausible that H believes that S wants Y to obtain.

Now, the hearer may first interpret the utterance "can you tell me where is room A?" as a direct question, i.e., a request of the speaker to the hearer to inform him whether he (the hearer) can show him (the speaker) the location of Room A:

> S.REQUEST(S,H,InformIf(H,S,CanDo(H,Show(H,S,Location(RoomA)))))

From this direct speech act, the hearer may then use the plan inference rules to infer the indirect request of the speaker to the hearer to show him (the speaker) the location of Room A:

> REQUEST(S,H,Give(H,S,Location(RoomA)))))

The plan inference model is a powerful model that uses rich knowledge structures and inference mechanisms to model human reasoning in interpreting natural language utterances. It also serves as a firm basis for designing conversational agents (see Chapter 5). A major drawback of this model however, is the computational complexity of the inference processes. Moreover, the development of accurate plan inference rules is a complex and time-consuming enterprise. A more fundamental objection against the model is its requirement that any utterance to be interpreted has to have a direct speech act to draw further inferences from in terms of an indirect speech act. This view of a two-stage interpretation of utterances has been criticised by several researchers in (psycho-)linguistics.

2.4.2 The Cue-based Model

As an alternative to the plan inference model, the so-called cue-based model is much more attractive from a computational point of view. The hearer takes an utterance as a set of *cues* for detecting the dialogue act type. These cues may be based on various sources of information, including lexical, syntactic, prosodic or conversational-structure information. In cue-based modelling, the assessment of cues that are informative for detecting the various dialogue act types can be done via corpus studies and psycholinguistic research. Additionally, machine learning techniques can be used to model the relation between cues and dialogue act types, more specifically, to construct a dialogue act classifier. In this way, the informativeness of the cues can be evaluated in terms of performance of such a classifier, but the classifier can also be used as a dialogue system component. Another application of this research is using dialogue act information for improving speech recognition (Stolcke et al., 2000).

Many of the cues used in a dialogue act interpretation model stem from analyses of corpora. In (Jurafsky et al., 1997) various informative cues have been found along three different dimensions of information.

- **Prosodic Information**: concerns features such as speaking rate and pitch; rising pitch at the end of an utterance is a good cue for a yes-no-question.
- **Words and Word Grammar**: e.g., "please" or "would you" is a good cue for a request.
- **Discourse Grammar**: a "yeah" which follows a proposal is probably an agreement; a "yeah" which follows an informing utterance is probably an acknowledgement.

Usually, cue-based dialogue act recognition is based on a probabilistic model of the relation between cues and dialogue act types. Empirical data can be used to assess such models; mostly, the models are defined as classifiers, that can determine the most likely dialogue act type (a *class*), given a set of cues (the *features*). We may now present the cue-based model schematically as in Figure 2.4. An utterance in context is represented by a set of feature-value pairs $(f_1 = v_1, \ldots, f_n = v_n)$ and the classifier (DAR) that is machine-learned (ML) from data in a dialogue corpus determines which dialogue act type d is the most likely one.

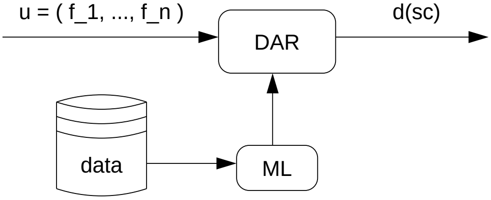


Figure 2.4: DAR: the cue-based model.

Experimental results from research that may be categorised as cue-based, have been presented by for example, Nagata and Morimoto (1994); Samuel et al. (1998); Kipp (1998); Wright (1998); Wright et al. (1999); Stolcke et al. (2000); Black et al. (2003). In our research on dealing with uncertainty in dialogue, we have done some experiments on dialogue act classification that can be qualified as cue-based. After a general proposal for using Bayesian networks for this task and some preliminary experiments (Keizer, 2001a; Keizer et al., 2002), this work has lead to the current findings described in Chapter 6.

2.4.3 Uncertainty

In the process of dialogue act recognition, the interpreter has to deal with uncertainty. Firstly, uncertainty arises because of the loss of information when perceiving an utterance. This is even more the case in the formal setting of cue-based recognition, where utterances and context are represented by means of cues, or more technically, a set of feature-value pairs. Any utterance within context has a potentially infinite range of characteristics and can therefore never be completely represented in terms of feature-value pairs.

Secondly, in using a finite set of dialogue act types, many subtleties in the use of an utterance may be abstracted away. In Section 2.2.1, we cited Ludwig Wittgenstein's remark that there are infinitely many ways of using language. So in that light, any classification would be incomplete: many ways of language use would not be covered by the set of dialogue act types and furthermore, in some dialogue act types several ways of language usage may be clustered and hence their differences wiped out. However, the classification of ways of language use within a framework of dialogue acts has in fact been restricted to the illocutionary acts originally introduced by Austin. He intended to find a classification of illocutionary acts by inventarisation of performative verbs, which seems to be a more feasible enterprise than finding classifications for the general Wittgensteinean 'language-games'.

Searle also accounts for the implications of his attempt to find a system in the ways in which language is used, thereby responding to Wittgenstein's observation that the similarities this 'system' should reveal are 'family resemblances' at the most:

> *The concepts of game, or chair, or promise do not have absolutely knockdown necessary and sufficient conditions, [...] But this insight into the looseness of our concepts, and its attendant jargon of "family resemblance" should not lead us into a rejection of the very enterprise of philosophical analysis; rather the conclusion to be drawn is that certain forms of analysis, especially analysis into necessary and sufficient conditions, are likely to involve (in varying degrees) idealisation of the concept analyzed.*
>
> *(Searle, 1969, Chapter 3)*

But there are still problems with the approach of studying illocutionary acts only. Austin already admitted that there are some ways of language use that cannot be clearly categorised as illocutionary or perlocutionary. He gives the act of 'insinuating' as an example. Of course, one cannot say "I hereby insinuate that you have lied", but in non-performative utterances, it might be hard to distinguish between insinuating and accusing, which is more clearly illocutionary: "I hereby accuse you of lying".

All of these complications in finding classifications of dialogue acts and specifying utterance features render a complete model for dialogue act recognition impossible. In other words, one can never find an exact and complete mapping from the space of features to the set of dialogue act types. Therefore, any dialogue act recogniser can only output a dialogue act type that is considered *plausible*. For this reason, most research on dialogue act recognition has focussed on the cue-based model, using machine learning techniques to learn dialogue act classifiers from data in an annotated dialogue corpus. In the next section, the annotation of dialogue corpora with dialogue act types will be discussed.

2.5 Dialogue Act Annotation

2.5.1 Introduction

In Section 2.3 we discussed the organisation of dialogue acts into a dialogue act taxonomy. Two theories of dialogue acts were discussed that used such a taxonomy: Dynamic Interpretation Theory and Conversation Act Theory. This section will be about the more experimental or, empirical, approach to dialogue act research, in the form of annotating dialogue corpora. This is an essential aspect in applying the cue-based approach to dialogue act recognition, discussed in Section 2.4.

Dialogue act annotation is concerned with labelling the utterances in a corpus of dialogues with dialogue act types from a dialogue act taxonomy. The purposes of dialogue act annotation are twofold: it may serve as 1) a framework for recording corpus studies of dialogue acts and to disclose dialogue act information in empirical data, and 2) as a database for machine learning of relations among dialogue acts and relations of dialogue acts with other aspects of a dialogue, such as surface features of utterances. For our research, the second general purpose will be most important; more specifically, we are interested in using data to learn dialogue act classifiers.

A *dialogue corpus* is a set of recorded dialogues, in spoken or written form, which can serve as examples of how a dialogue should and could develop. Depending on what phenomena are considered to of interest in the dialogues, the corpus needs to be *annotated*, by specifying how it is structured w.r.t. these phenomena. An obvious first aspect of annotating a corpus of dialogues is specifying where a dialogues starts and ends. Further structuring may be done in terms of *turns*, *utterances*, etcetera. For dialogue act annotation, the units to which a dialogue act type is to be assigned, are required to be specified.

Various schemes for annotating dialogue corpora have been developed. The phenomena covered in these schemes may vary in a number of ways, depending on the character of the dialogues to be analysed. This character may be influenced by a number of factors:

1. *Task domain*: what the dialogues are about, e.g. theatre performances, appointments, travelling (flight and/or train schedules), how to use some complex device (e.g. a printer), etcetera.

2. *Modalities of interaction*: are the dialogues spoken or (tele)typed? Is there any additional non-verbal communication – in the form of mouse and/or keyboard actions, gestures, or facial expressions?

3. *Types of activity*:

    - Cooperative negotiation: the participants have to reach a form of agreement, such as making an appointment or a transaction.
    - Information extraction: one of the participants tries to obtain information he thinks the other can give him. The other might be cooperative and give the information if he has it.

   - Problem solving: the participants cooperate to solve a problem (possibly a problem of one of the participants, but it may also be a shared problem).
   - Tutoring/instruction: one of the participants helps the other in performing a certain task, for example, by giving clues.
   - Counselling: one of the participants gives advice to the other concerning some complex (legal) issue.

4. *Source of the dialogue corpus*: how have the data been obtained?

   - Human-human dialogues: conversations between human participants, either spontaneous or elicited to some extent.
     - Machine-mediated dialogues: telephone conversations or conversations in which utterances are typed on a computer which possibly does some processing (e.g., translation) and sends a message to another computer, which also possibly does some processing and displays an utterance for the other dialogue participant.
     - Non-machine-mediated dialogues: direct spoken dialogues between two participants, not mediated by a machine.
   - Human-machine dialogues: obtained from interaction of a human user with a prototype dialogue system.
   - Simulated human-machine dialogues: dialogues between human participants, in which one participant uses instructions to simulate a dialogue system (the human user assumes he is interacting with a machine); such a setting is called a Wizard of Oz experiment.

   o In addition, multi-party conversations, in which more than two people are involved, may be of interest.

5. *Purpose of annotation*: where are the annotated data used for?

   - Development of a conversational agent (human-machine interaction)
   - Machine-mediated human-human dialogues, e.g. for translation in dialogues between persons, each speaking their own, different language.
   - More indirectly for obtaining data for understanding dialogue or testing a dialogue theory, or for building language models for speakers in the context of a dialogue.

In the remainder of this section, we will discuss a number of projects in which dialogue corpus annotation has been an important topic. First, we discuss the VERBMOBIL project, dealing with human-human dialogues concerning appointment scheduling (Section 2.5.2). Then, we describe the annotation of the

MAPTASK corpus concerning route planning. Finally we present the annotation scheme, developed by the Multi-party Discourse Group of the Discourse Resource Initiative (DRI): a standard for annotating task-oriented dialogues (Section 2.5.4). We also describe some concrete schemes that made use of this standard.

2.5.2 VERBMOBIL

This German project has been concerned with speech-to-speech translation of dialogue. In the first phase of the project, the translation into English of spoken utterances in dialogues between German-speaking and Japanese-speaking business partners was dealt with. These were appointment scheduling dialogues. The second phase concerned robust bi-directional translation of spontaneous speech in dialogues, where the language pairs German/Japanese and German/English were covered. Now, the dialogues additionally concerned travel planning and making hotel reservations.

In VERBMOBIL, dialogue acts are defined as 'primary communicative intentions' (Alexandersson et al., 1998). In this light, dialogue act information plays an important role in translation of dialogue utterances. The information can be used for selecting templates for generating target language utterances. For example, consider the German utterance:

(7) Können sie vielleicht einen Vorschlag machen?

being translated into the English utterance

(8) Could you make a suggestion?

The word "vielleicht" is not translated here, because the utterance is identified as a `request` (further descriptions of this and the other dialogue acts in VERBMOBIL will be presented in a minute), in which "vielleicht" just serves as a politeness marker, which in English is covered by the verb form.

Especially in parts of the system where statistical modelling is applied, dialogue acts are used for annotation. Utterance segments are seen as the basic processing units and are annotated with dialogue acts. In addition of tags to indicate dialogue acts, a `:PHASE` tag is used to indicate which of the five possible phases the dialogue can be in: **Hello**, **Opening**, **Negotiate**, **Closing**, and **Good-Bye**. In the definitions of the dialogue acts (see Alexandersson et al., 1998), some reference to the notion of dialogue phase may be given: some acts may typically co-occur with certain phases. The dialogue act `suggest` (see below) will typically occur in the negotiate-phase of a dialogue. Furthermore, in the approach of Verbmobil, the meaning of an utterance consists of the dialogue act *and its propositional content*. Given a dialogue act, one can pose some restrictions on the accompanying propositional content and this can be used in the annotation process. For example, the propositional content of a `suggest` contains the suggested proposition, e.g., a date or duration, a location, a selection of transportation of accommodation, or an action. Other dialogue acts may not have any propositional content (e.g., a `request-comment`, in which the speaker explicitly asks the hearer to comment on a proposal). Clearly, the information regarding propositional content is strongly related to the domain.

The Dialogue Act Hierarchy underlying the annotation scheme (taken from Alexandersson et al., 1998) consists of three branches. The first is the **control-dialogue** branch containing tags for segments that are solely concerned with social interaction, relating to the dialogue itself, or for smoothing the communication. The second branch contains the **manage-task** acts, for annotating segments which concern managing or controlling the task. Finally, the **promote-task** branch deals with all other cases (more specifically, for segments concerning performing the task.

We give a short overview in which we describe the meaning of some of the tags and give example utterances.

1. control-dialogue
   - `greet`: ("Good morning").
   - `bye`: ("See you!").
   - `introduce`: ("My name is John").
   - `politeness-formula`: used to stabilise the relationship with the other participant or to fulfill certain conventions ("how are you?").
   - `thank`: ("Thank you very much")
   - `deliberate`: S intends to gain dialogue time, e.g. by thinking aloud ("let me see ... on Monday I have to go to Amsterdam, so ...") or using certain formulas ("hold on, ...").
   - `backchannel`: used for signal understanding: S acknowledges successful communication, without really taking the turn.
2. manage-task
   - `init`: S initialises the task by motivating, explaining or mentioning it for the first time ("I would like to make an appointment for next week").
   - `defer`: S proposes or states to postpone the task ("Maybe we could discuss further arrangements at a later time").
   - `close`: S considers the task as done ("so, that's settled then").
3. promote-task
   - `request`
     - `request-suggest`: S requests H for information regarding an action to take ("what does your schedule look like?").
     - `request-clarify`: S requests H to comment on a statement or suggestion made in the previous discourse ("I'm free in the afternoon from three to five;" - labelled `suggest` - "how is then for you?").
     - `request-comment`: S requests H for clarification of information ("did you say Wednesday the third?"). This act is used for verification purposes, in stead of `backchannel`, which concerns understanding utterances in direct communication.

  - `request-commit`: S requests H to commit to performing some action (“will you do that?”).
- `suggest`: S proposes an explicit instance or aspect of the negotiated topic.
- `inform`:
  - `digress`: deviation from the expected course of the dialogue.
    - `deviate-scenario`: for segments in which the topic does not belong to the scenario.
    - `refer-to-setting`: (“this keyboard is very annoying”).
  - `exclude`: S informs H of the impossibility or unsuitability of a certain action.
  - `clarify`: S presents more information about something that has already been, either explicitly or implicitly, introduced in the course of the dialogue.
  - `give-reason`: S gives a reason for, or justifies, or motivates, a previous segment.
- `feedback`: S gives a reaction to the previous discourse.
  - `feedback-negative`: negative reaction of S to a contribution of H.
    - `reject`: S rejects a previously introduced proposal.
  - `feedback-positive`: positive reaction of S to a contribution of H.
    - `accept`: S explicitly accepts a proposal.
    - `confirm`: S accepts a proposal by wrapping up the result of a negotiation.

  There is also a common sub-category of `give-reason` and `reject`, namely `explained-reject`; it is used in cases where the speaker obviously rejects a proposal formerly introduced in the dialogue, by stating a reason (John: “could we meet at 9:30h?”; Peter: “I already have another appointment at that time”).
- `commit`: S commits him-/herself to performing an action, which will actually be performed, unless explicitly rejected by H.
- `offer`: S offers to perform an action. This act requires a reaction of H: if H approves, S will perform the action.

The task-domain the annotated dialogues of VERBMOBIL and especially the types of activity connected with it, are clearly reflected in the dialogue act hierarchy used. Acts such as `suggest` and `request-suggest` are essential in cooperative negotiation dialogues. Furthermore, with extending the task domain with travel planning in the second phase of the project, new kinds of communicative actions appeared in the dialogues, which lead to extension and/or modification of the hierarchy by adding dialogue acts like `offer` and `commit`. So, not only very specific, domain-dependent additions to the hierarchy are obtained, but also a wider range of domain-independent types of dialogue utterances.

The dialogues annotated in VERBMOBIL were spoken dialogues, so part of the hierarchy contains dialogue acts for controlling the dialogue and especially the direct understanding of utterances (`backchannel` for example).

Furthermore, it is noted that human-human dialogues are under consideration, so to some extent, the behaviour of the participants may be different from human-computer dialogues. Furthermore, the dialogue acts are defined from the perspective of translation: the project deals with computer-mediated interaction between humans, in stead of human-computer interaction, where we are interested in understanding (observation and interpretation) and action. The dialogue acts are not defined in terms of belief states of the participants, context operators or plan operators, but as serving language-independent representations of natural language interaction.

2.5.3 MapTask

In the Dialogue Modelling project of HCRC Edinburgh, a corpus of task-oriented, spoken dialogues, has been created: the MapTask corpus. Where the VerbMobil project had a computational purpose of developing a dialogue translation system, here the purpose is theoretical, i.e., conversations have been elicited to obtain data for studying linguistic behaviour.

In the elicited dialogues in the corpus, the two dialogue participants each had a slightly different version of a map (the maps had a restricted number of characteristics in common). On one of the maps a route was written, which the owner of the other map had to reconstruct on his own map, using the directions/instructions he got from the other participant during the dialogue.

The annotation of the corpus (Carletta et al., 1996) was based on the notion of dialogue structure, which can be analysed at three levels:

1. *transactions*: sub-dialogues that accomplish one major step in the participants' plan for achieving the task.

2. *conversational games* (also called 'dialogue games', 'interactions', or 'exchanges'): a set of utterances starting with an initiation and encompassing all utterances up until the purpose of the game has been either fulfilled or abandoned; these can also be nested.

3. *conversational moves*: different kinds of initiations and responses classified according to their purpose.

So, the content and structure of transactions depend more on the task domain at hand, while the conversational games of which they consist can be identified by more domain-independent phenomena. These conversational games are in turn sequences of conversational moves, which form the most substantial part of the annotation scheme. Most of the corpus was annotated only on the levels of games and moves. With respect to the coding of games, we will confine ourselves to the remark, that the purpose of the game is indicated and also some notion of how games are interrelated.

The Move Coding Scheme

The scheme on the level of conversational moves is based on a hierarchy of move categories. In the clarification of the moves below, example utterances are given. Some of these examples consist of sequences of utterances, involving an ′instruction Giver′, which is denoted by G, and an ′instruction Follower′, denoted by F.

- Initiation Moves (moves which set up the expectation of a response):
  - `instruct`: H is commanded to carry out any action, except for implicit actions in queries (see below) (“I want you to go to the left-hand side of the page”).
  - `explain`: S states information, which has not been elicited by the partner; in the latter case, it would be a response (“I'm in between the remote village and the pyramid”).
  - `check`: S requests H to confirm information that S has some reason to believe, but is not sure about.

    “I told you about the land mine, didn't I?”

    G: ″Ehm, curve round slightly to your right″
    F: ″To my right?″
    G: ″Yes″
    F: ″As I look at it?″

    - in the second example, of course the moves in the second and fourth utterance are the `check`'s.
  - `align`: S checks the attention or agreement of H, or his readiness for the next move.

    G: “This is the left-hand edge of the page, yeah?”
    F: “Yeah, okay”

    - the move in the first utterance is the `align`.
  - `query-yn`: questions taking a “yes” or “no” answer, unless the question is already a `check` or `align` (“Do you have a stone circle at the bottom?”).
  - `query-w`: any query not covered by the other categories; mostly wh-questions, but also questions asking to choose from a set of alternatives.
- Response Moves: moves after an initiation, in order to fulfill the expectations set up in the dialogue (game).
  - `acknowledge`: verbal response which minimally shows that S has heard the move to which it responds, and often also demonstrates that the move was understood and accepted.

    G: “Ehm, if you... you're heading southwards”
    F: “Mmhmm”

    The utterance by F is the `acknowledge`.
  - `reply-y`: any reply to any query with a yes/no surface form which means “yes”, however that is expressed.

G: "See the third seagull along?"
F: "Yeah"

Again, the utterance by F is the acknowledge.

- reply-n: similar to reply-n, where the reply means "no".
- reply-w: any reply to any query, which doesn't simply mean "yes" or "no".

G: "And then below that, what've you got?"
F: "A forest stream"

The second utterance is a reply-w.

- clarify: a reply to some kind of question in which S tells H something over and above what was strictly asked, but which is not substantial enough to code it as two moves: a reply and an explain.

It has been considered to add a move like object to the hierarchy, to cover utterances, in which S refuses to take on a proposed goal. Because these cases appeared to be rare in the corpus, this move, seen as negative counterpart of acknowledge, has not been added.

- Preparation Move
  - ready: occurs after the close of a dialogue game and prepares the conversation for a new game to be initiated (often by means of utterances like "OK" and "right").

The moves defined in the move coding scheme clearly reflect the specific task-domain of the dialogues and the types of activity performed. The moves are not domain-dependent, but an initiation move such as instruct typically refers to the type of activity performed in this task. Moreover, this move is generally attributed to one specific participant in the dialogue, namely the one playing the role of instructor (having the map with the route on it).

There is no extensive coverage of dialogue control acts, although the corpus does contain spontaneous spoken dialogue: only the move acknowledge refers to some notion of (perceptual) understanding, while no attention is given to cases in which the hearer didn't understand what the speaker was saying. One could say that these situations are implicitly identified by repair- or clarification sub-dialogues, initiated by a yn-question and would be treated at the game coding level.

Furthermore, no moves are given for other important issues in spoken dialogues, like turn-taking, topic-management and social conventions. The dialogues in the corpus are characterised by the restrictedness of the task: the task is quite clear, the participants seem to have a clear idea of what to expect from the dialogue partner and they also have shared knowledge in the form of maps of the same area, even though they are not identical. The task at hand is relatively well defined.

2.5.4 DAMSL

Where the dialogue act schemes in VERBMOBIL and MAPTASK were aimed at particular tasks, either for the development of a specific dialogue system, or for merely studying the dynamics of dialogue, DAMSL (Dialogue Act Markup in Several Layers) is an annotation scheme standard, to be extended for development of various dialogue systems, each with their own specific properties, like a particular (task-)domain. The scheme was developed in the framework of the Discourse Research Initiative (DRI) by the Multiparty Discourse Group, during the 2nd and 3rd meeting in Dagstuhl ('96) resp. Chiba ('98) (Allen and Core, 1997). Many researchers, involved in related projects around the world themselves, participated in these meetings.

After describing DAMSL, we will briefly touch upon some projects dealing with dialogue corpus annotation, that have built their scheme on top of this standard, namely:

1. COCONUT : Pittsburg University and SRI International;

2. TRAINS: University of Rochester.

The DAMSL Dialogue Act Hierarchy

In the DAMSL scheme, four dimensions (also referred to as 'levels', or 'layers', see also Section 2.3.2) are distinguished, along which the annotation should be performed. Each of the dimensions refers to a (more or less) independent aspect of the dialogue process. The four dimensions are:

1. Communicative Status: is the utterance intelligible, and was it successfully completed?

2. Information Level: what type of information contains the utterance?

3. Forward Looking Function: what are the effects of the utterance on the future belief and actions of the participants and on the future discourse?

4. Backward Looking Function: how does the utterance relate to the previous discourse?

Communicative Status At this level, the annotator can indicate whether an utterance is intelligible and and whether it was successfully completed. The following tags are available:

- `uninterpretable`: the utterance is not comprehensible;

- `abandoned`: the speaker doesn't finish the utterance, but breaks it off;

- `self-talk`: the speaker utters something which is not intended for the communication.

If none of the situations described above occur, annotation at this level is simply omitted. The aspects of utterances dealt with at this level, are especially relevant in spoken dialogues and are of less interest in typed dialogues, like those in the SCHISMA corpus. However, a user may be typing an utterance so badly, that it becomes uninterpretable.

Information Level The tags at the Information level are used to indicate at an abstract level what the utterance is about. The following tags are used for the classification at this level:

- `task`: the utterance is part of **doing the task**;
- `task-management`: the utterance is part of **talking about the task**;
- `communication-management`: the utterance is part of maintaining the communication;
- `other-level`: other aspects not covered by the other categories, like jokes, non-sequiturs and small-talk.

Forward Looking Function The third layer is the **Forward Looking Function** (FLF), describing a hierarchy of categories of dialogue acts characterising the effect an utterance has on the subsequent dialogue and interaction. The annotation is guided by a decision tree: at every junction in the dialogue act hierarchy, the annotator has to choose which branch down the hierarchy to choose, based on some question about the utterance he has to answer. We will describe very briefly which kinds of utterances are labelled with which tags.

- `statement`: S makes a claim about the world.
  - `assert`: S tries to change the belief of H.
  - `reassert`: S thinks the claim has already been made.
  - `other`.
- `influencing-addressee-future-action`: S is suggesting potential actions to H, beyond answering a request for information.
  - `action-directive`: S is creating an obligation that H do the action unless H explicitly indicates otherwise.
  - `open-option`: S is not creating this obligation.
- `info-request`: S is requests (by just asking or in another, indirect way) H for information.
- `committing-speaker-future-action`: S is potentially committing to intend to perform a future action.
  - `offer`: S's commitment is contingent on H's agreement.
  - `commit`: otherwise.

- `conventional`:
    - `opening`: S summons H and/or starts the interaction.
    - `closing`: S closes the dialogue or is dismissing H.
- `explicit-performative`: S is performing an action by virtue of making the utterance.
- `exclamation`.
- `other-forward-function`.

Backward Looking Function This fourth layer is used for characterising the relation an utterance has on the previous dialogue. The annotation is again guided by a decision tree; again, we give a short characterisation of the kind of utterances labelled with the various tags. When an utterance is tagged at this level, also a reference is added to the segment to which this segment is relating (the *antecedent*).

- `agreement`: S is addressing a previous proposal, request, or claim.
    - `hold`: S is not stating his attitude towards the proposal, request, or claim.
    - `accept`: S is agreeing to all of the proposal, request, or claim.
    - `accept-part`: S is agreeing only partly to the proposal, request, or claim.
    - `reject`: S is disagreeing to all of the proposal, request, or claim.
    - `reject-part`: S is disagreeing only partly to the proposal, request, or claim.
    - `maybe`: S is stating his attitude, but is non-committal to the proposal, request, or claim.
- `understanding`: utterances concerning the understanding between S and H, ranging from merely hearing the words to fully identifying intentions.
    - `signal-non-understanding`: S asks H what H said or meant (not to be confused with cases of hold, in which also a request for clarification is made, but more in the sense of additional information; it does not signal non-understanding.
    - `signal-understanding`: S explicitly signals understanding (not complementary to signal-non-understanding!).
        - `acknowledge`: S signals understanding, but not necessarily acceptance.
        - `repeat-rephrase`: in order to signal understanding.
        - `completion`: S signals understanding by finishing (part of) the utterance H is making.
    - `correct-misspeaking`: S adds a correction, indicating that he believes that H has not said what he actually intended.
- `answer`: standard reaction of S to an info-request action by H.

- `information-relation`: this is a tag which should capture how the content of this utterance relates to the content of its antecedent; it is still subject of future study.

TRAINS

This project aimed at the development of an intelligent planning assistant. The primary application has been a planning and scheduling domain involving a railroad freight system, where the human manager and the system must cooperate to develop and execute plans. The current prototype is called TRIPS (The Rochester Interactive Planning System), see also Section 5.2.1.

The dialogue corpus consists of spoken dialogues between one person playing the role of a user with a certain task to accomplish and another person, playing the role of a planning assistant. The annotation of the dialogues was established using the original DAMSL annotation scheme (Allen and Core, 1997). The project members were closely involved in the development of the annotation standard of the DRI.

COCONUT

The COCONUT (*Co*operative, *Co*ordinated *N*atural language *ut*terances) project was a cooperation of the University of Pittsburgh Intelligent Systems Program and the Natural Language Group at SRI International, concerned in general with analysing interactive discourse. In the project, a corpus of human-human dialogues was obtained for analysis of both interpretation and generation in problem-solving and collaboration processes.

The dialogues were collected using a setting in which two human participants communicated with each other by means of typed sentences. In the setting, both subjects were assigned an inventory of furniture and a budget, and they had to collaborate by sharing their inventories and budgets and negotiate about buying furniture from each other, in order to furnish their respective living and dining rooms. Certain priority rules had to be followed and the participants' goals were further directed by means of a point system in which they could *jointly* earn points for achieving certain situations. So, the participants did not have private goals, but made joint decisions for purchases.

Both participants had a computer interface, on with the course of the typed dialogue was monitored. Additionally, both of the participants were able to maintain their own information concerning the budgets and inventories (including those of the other participant); also a floor plan of their rooms was displayed. Besides the typed dialogue utterances, also the other information maintained by the participants was recorded for the corpus.

The obtained corpus has been annotated using a DAMSL-based coding scheme (Eugenio et al., 1998). In Chapter 6, we will describe a scheme for annotating a corpus of Dutch dialogues that was based on the DAMSL scheme in a similar way. At the top level of the dialogue act hierarchy of the COCONUT scheme, the level Communicative Status was left out, while two additional levels were added: *Topic* and *Surface Features*. The Topic level deals with what the utterance is about, and we will seen that it contains rather domain-specific tags. Surface Features concern syntactic features like tense, mood, subject and modals (indicated by words like "lets", "should"). In the overview below, new acts are given in boldface.

Information Level

- `task`
  - **evaluate-plan**: used when plans are explicitly talked about as objects; these may be both complete and partial solutions, but not single steps in a plan ("that sounds a very reasonable plan to me.").
  - **game-procedure**: utterances concerning what the players are allowed to do ("we can't exceed our budget").
- `task-management`
  - **strategize-action**: concerning strategies to follow in the current problem solving scenario ("Let's start from the living room").
- `communication-management`
- `other-level`

Forward-Communicative Function

- `statement`
  - `assert`
  - `reassert`
  - `other-statement`
- influence-on-listener (`influencing-addressee-future-action`)
  - `open-option`
  - **directive**
    - `info-request`: a top-level Forward-looking function in DAMSL.
    - `action-directive`
- influence-on-speaker (`committing-speaker-future-action`)
  - `offer`
  - `commit`
- `other-forward-function`: the other tags from DAMSL have been given a less important status by subcategorising them under this tag.
  - `conventional-opening`
  - `conventional-closing`
  - `explicit-performative`
  - `exclamation`

Topic (dealing with what the utterance is about)

- TOPIC PROPER
  - Tags related to Furniture Items
    - `need-item` ("we need a cheaper sofa").
    - `have-item` ("I don't have a cheaper sofa").
    - `get-item` ("buy the chairs").
    - `elaborate-item`: elaborations on a description of a specific item already introduced ("How much is the red sofa", "so are the chairs 25 each?", "No, the chairs are 100 each").
    - `other-item`
  - Tags related to Budget
    - `budget-amount`: utterances where the available initial budget is mentioned ("I have 550").
    - `budget-remains`: utterances where the amount of money left is discussed

      "let's do the yellow/blue mix

      "leaving us with 250"
    - `cost-accum`: utterances where the costs of various choices are listed ("spent: 300 (sofa), 250 (lamp), 200 (table), and 300 on chairs")
  - Tags related to Points
    - `point-amount`: utterances where the points associated with a certain choice are mentioned ("gained 200 for the sofa")
    - `point-accum`: utterances where the points associated with the solution are summed up ("how about ... and give us 630 pts.").
- ATTITUDE: of the speaker towards: either furniture items, budget amounts and costs, and point amounts; or a (possibly partial) solution or plan.
  - `eval`, with possible values *pos* and *neg*.
  - `relate`, with possible values *better*, *worse*, *same* and *dfft* (different).

In the scheme, an additional 'menu' is contained, called *ItemFeature*, which contains tags, specifying which properties of items are discussed, compared or evaluated:

- `price`, `color`, `type`, `points`.
- `typeColor`, `typePrice`, `colorPrice`, `all`: combinations of furniture item properties.
- `genl`: general assessment about a solution or a plan ("that sounds good" - in the appropriate context).

Surface Features (concerning syntactic features of the utterance)

- `word-surface-features`
  - `matrix`: possible values *iSay*, *iGuess*, *iThink*, *iKnow*, *letMeKnow*, *otherMatrix*, *noMatrix*.
  - `modal`: possible values *howAbout*, *lets*, *can*, *could*, *must*, *should*, *would*, *otherModal*, *noModal*.
  - `subject`: possible values *i*, *you*, *we*, *otherSubject*.
- `syn-surface-features`
  - `tense`: possible values: *pres*, *past*, *future*, *presProg*, *pastProg*, *presPerfect*, *pastPerfect*, *otherTense*, *noTense*.
  - `mood`: possible values: *questionY/N*, *questionWH*, *indir-question* ("I forgot how much the sofa was"), *imperative*, *declarative*, *otherMood*.
  - `neg-polarity`: when an explicit negation is present.

Backward-Communicative Function

- **initiate**: indicating that the utterance is unsolicited, meaning in fact that there is no backward function.
- `agreement`
  - `accept`
  - `accept-part`
  - `maybe`
  - `reject-part`
  - `reject`
  - `hold`: capturing a clarification request in the sense of being a request for additional information in order to help the participant make a decision.
  - **clarification-request**: in the sense of the participants reviewing their decisions ("did we decide on the red sofa?"), or co-occurring with a `signal--non-understanding`, or while trying to repair a `signal-non-understanding`.
- `understanding`
  - `signal-non-understanding`
  - `signal-understanding`
    - `acknowledge`
    - `repeat-rephrase`
      (`completion` has been omitted)
  - `correct-misspeaking`: correction at the speaking level.
  - **correct-assumption**: correction of wrong assumptions; in DAMSL also covered by `correct-misspeaking`.

- `answer`
- `information-relations`: mostly concerning information about budget and specific furniture items.
  - `Act:Condition`: relation between an existing act and a necessary condition (precondition) of that act.
  - `Act:Consequence`: the utterance contains a consequence of an existing act; the act can be expressed as an action or a state, resulting from an action.
  - `other-info`: the utterances are related at the informational level, but neither of the above tags apply.

  In the coding of an utterance, the values *LabelOrder* and *ReverseOrder* are used in addition to `Act:Condition` or `Act:Consequence` to indicate whether the coded utterance contains a precondition or consequence of a previously introduced act, or describes an act, referring to its previously mentioned precondition or consequence, respectively.
- **coreference/set-relations**
  - `same-item`: the utterance discusses the same item as its antecedent (it does not mean that it provides more information about the item, as indicated by the `elaborate-item` tag at the Topic level).
  - `subset`: usually, but not always, co-occurring with `reject-part` or `accept-part`.

    S1: "I have a red table for 200 and a yellow one for 400"

    S2: "Let's go with your red table."
  - `mut-exclusive` (a set relation): relation linking two propositions p_1 and p_2, where p_1 mentions a set of items S_1, and p_2 provides an alternative S_2 to that same set of items and S_1 and S_2 are mutually exclusive:

    S1: "let's get a blue sofa"

    S2: "no, let's get a yellow table"

    or:

    S1: "do you have any high tables?"

    S2: "yes I do [have high tables]"

Apart from some minor changes, like defining `info-request` and `action--directive`as special cases of `directives`, the COCONUT scheme has taken the DAMSL scheme and just added a number of additional features which were relevant for the task-domain at hand. First, an extra level called Topic has been added, which contains a number of domain-specific tags. Furthermore, other additional tags have been added, which are domain-independent, e.g., the tags `clarification-request` and `correct-assumption`. Also, some tags describing the action-consequence and action-condition relations have been defined as tags under `information-relation`. Furthermore, an extra branch has been added at the Backward-Communicative Function level, with tags concerning coreference and set-relations.

There are still some additional tags in the scheme, but those are of less interest to us and have therefore been omitted from our description.

2.5.5 Discussion

Now that we have described these annotation schemes, we can compare them, using a number of general criteria. Most of these criteria can be directly derived from the factors, mentioned in Section 2.5.1. In Table 2.1, a schematic overview is given of the schemes in relation to the criteria, which is partly based on Klein et al. (1998). In the overview, three additional schemes are given: the annotation scheme used by Traum (supporting his theory of conversational acts and information state updates, described in Section 2.3.2), the SWBD-DAMSL scheme for the SWitchBoarD corpus of non-task-driven telephone conversations (used for training stochastic discourse grammars in order to build better language models for automated speech recognition), and the annotation scheme for the SCHISMA corpus that will be discussed in Chapter 6.

	COCONUT	DAMSL	MapTask	SWBD-DAMSL	Traum	VerbMobil	Schisma
# dialogues	16	18	128	1155	36	1172	64
Languages	English	English	English	English	English	Eng., Jap., Ger.	Dutch
Participants	2	2	2	2	2	2	2
Modalities	typed	spoken	spoken	spoken	spoken	spoken	typed
Task orientation	Task-driven	Task-driven	Task-driven	Non-Task-driven	Non-Task-driven	Task-driven	Task driven
Domain restriction	furnishing rooms	no	giving directions	no	no	business appointments	theatre performances
Activity type	cooperative negotiation	several	problem solving	several	cooperative negotiation	cooperative negotiation	info-exchange transaction
Human/Machine Machine-mediated	HH MM	HH –	HH non-MM	HH MM	HH non-MM	HH MM	HH non-MM

Table 2.1: Annotation schemes compared along a set of criteria.

Chapter 3

Uncertainty

In which we discuss the phenomenon of uncertainty, that was identified as an essential issue to deal with in modelling natural language dialogue. Uncertainty will be viewed as an epistemic phenomenon an agent is involved with. Various approaches concerning representation of and reasoning under uncertainty are described, of which the use of probability theory is the most important one. Bayesian networks form a special kind of probabilistic models, to be discussed in Chapter 4. In addition, some attention is given to machine learning techniques, used to derive models from empirical data.

3.1 Introduction

In the previous chapters we discussed the problem of uncertainty in modelling natural language dialogue. It was pointed out that in a dialogue between a human user U and an artificial agent S, S can never be absolutely certain about what was meant by an utterance of U addressed to him, given his knowledge about the context in which the utterance was made and the information S has about U's utterance. Information is lost, not only in the *noisy* channel between U producing the utterance and S perceiving it, but also in the fact that S makes abstractions in order to interpret the utterance. S has a model of the situation, in which both observed and unobserved information about the utterance (such as the occurrence of certain keywords, syntactic features, intonation, etc.) may be represented, together with more (unobserved) abstract aspects considered relevant, such as dialogue act types as discussed in Chapter 2. Because our agent S has incomplete information and has no complete formal specification of a dialogue theory, in which all aspects considered relevant are interrelated, he is *uncertain* about what U meant by his utterance.

As we are dealing with natural interaction between two or more participants and have particularly computational motives, we will take the *agency-approach* stemming from the field of Artificial Intelligence (AI). In the approach, the participants of the interaction are referred to as *conversational agents* or *dialogue agents*. The *knowledge* such agents have about themselves, about the state of the interaction, about the environment, and particularly about other agents in that environment they are interacting with, is captured in a *mental model*. This means, that part of a conversational agent's mental model consists of mental models of other, possibly human, agents that it is in interaction with. The field of *user modelling* is particularly involved in designing and managing models of users of applications such as information systems or tutoring systems.

In this agency-perspective, uncertainty is taken to be an *epistemic* phenomenon, as opposed to an *ontological* one: uncertainty is always regarded with respect to an agent that is uncertain about some event or state of affairs, *given* his background knowledge, and not as a (physical) property of reality, independent of any agent.

In this chapter we will discuss some general approaches and techniques that have been taken concerning reasoning under uncertainty. Our idea to investigate the use of Bayesian networks in dialogue modelling was motivated in a general sense by Pearl's plea for probability theory, as it provides an intelligent agent with a coherent account of common sense reasoning under uncertainty (Pearl, 1988). First, we will take a closer look at the uncertainty phenomenon itself (Section 3.2) and then discuss some requirements for solutions to dealing with uncertainty, that have been captured under the term *plausible common-sense reasoning* (Section 3.3).

3.2 Rational Agency and Uncertainty

An artificial agent that is to behave intelligently, will have to be able to:

1. make observations of its environment,
2. process the information obtained from those observations, and
3. act upon its environment.

In Figure 3.1, it is illustrated that in general, an agent can observe only part of its environment, while the agent's actions may effect any part of the entire environment.

Especially in the case of a dialogue agent, part of the environment may consist of one or more (human) agents. The observable part of the environment may then concern for example, a speech signal or a keyboard typed text. More advanced (multi-modal) systems may also be able to observe non-verbal aspects of the behaviour of agents in the environment, such as hand gestures.

Agents that are *embodied* also have a physical aspect to them that may especially be of relevance to the communicational functionality of the agent. For example, it may have a face that can take on expressions of joy, anger, etcetera, and change the physical viewpoint with respect to the environment, i.e., in drawing the attention of or giving attention to particular other agents in the environment. The arms can also be of importance for communication with

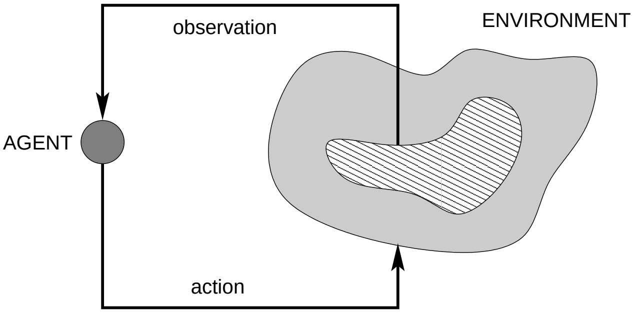


Figure 3.1: Agent in interaction with its environment.

other agents, e.g., in pointing to certain objects or emphasising utterances. So, the agent himself can also be part of reality and hence be part of its own environment (see Figure 3.2). Materialists would focus on this viewpoint and reject the situation in Figure 3.1: everything, including concepts such as reasoning and knowledge, are eventually material and thus part of the physical world.

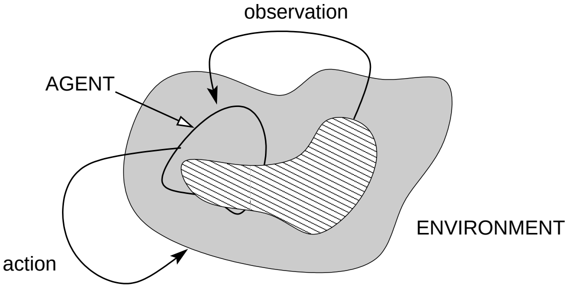


Figure 3.2: Agent as part of its environment.

If we descend from our third person perspective – in which we consider an agent and its environment – to the agent's perspective, the (physical) environment is just something (metaphysically) presupposed. The agent receives information through observations, supposedly originating from that environment, and sends information through actions, supposedly effecting that environment. In fact, the only thing the agent is really concerned with are his actual observations, based on which he builds a *mental model* of the environment; his actions are based on this model and the agent's preferences and goals. This situation has been illustrated in Figure 3.3: information about the environment is obtained through observation, using sensors such as eyes or a microphone; the agent changes the environment through action, using effectors such as arms or speech. During this process, the agent maintains a model of its environment, i.e., he has *knowledge* about the environment, and uses this model in deciding on how to act.

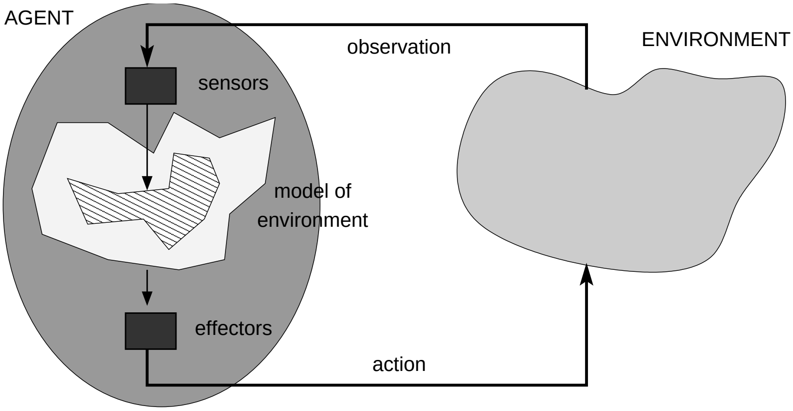


Figure 3.3: Agent maintaining a model of its environment.

In his model of the environment, the agent has performed some form of *abstraction* in order to represent the information he receives and to *understand* the presupposed environment as a whole: he has a *theory* about the domain in order to be able to reason about it. The extent to which abstractions, or rather, simplifications have been made in this theory is also determined by the task requirements or goals of the agent: many aspects of the environment may not be relevant for him in deciding on his actions. Because the agent 1) has only partial information about the environment and 2) his theory of the domain may not be complete, he cannot be absolutely certain about the unobserved aspects of the environment. As we have seen in the previous chapters, both factors are very significant in the domain of natural language dialogue.

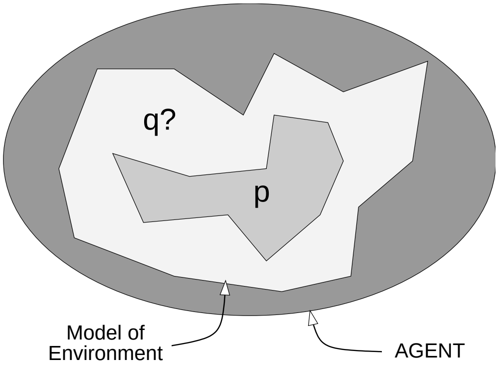


Figure 3.4: Uncertainty due to incomplete information.

In Figure 3.4 the situation of uncertainty due to incomplete information is illustrated: assuming that the model is represented by a set of propositions, the agent's observation produces information in the form of, for example, a proposition p (for example, p means "it has rained"). However, the agent may be uncertain about an unobserved aspect q (for example, q means "the football pitch is slippery"). Based on some theory in which p and q are related, the agent may determine whether or not q is the case, or, how *plausible* q is.

3.3 Plausible Common-sense Reasoning

For the problem of dealing with uncertainty in dialogue modelling, we found probability theory (more particularly, Bayesian networks) to be a formalism that supports a number of desired patterns of plausible common-sense reasoning (see Pearl (1988) for a detailed discussion):

- *consistency*: the formalism may not allow conclusions to be drawn that are inconsistent with the current knowledge base, i.e., new information may not lead to a mental model with conflicting beliefs;
- *non-monotonic reasoning*: enables reasoning about context-sensitive beliefs (new information may cause beliefs to be less plausible);
- *deductive reasoning*: application of the logical inference rule 'modus ponens' (see Table 3.1: above the horizontal line the premises are given and below it, the conclusion is given).
- *abductive reasoning*: works in the opposite direction of the implication, but needs a weaker formulation in terms of plausibility (see Table 3.2; 'm.p.' means 'more plausible').
- *explaining-away*: combines reasoning in both directions of logical implications (see Table 3.3).

$$\begin{array}{|l|l|}\hline \begin{array}{l} \mathrm{A} \rightarrow \mathrm{B} \\ \mathrm{A} \\ \hline \mathrm{B} \end{array} & \begin{array}{l} \mathrm{A} \rightarrow \mathrm{B} \\ \neg\, \mathrm{B} \\ \hline \neg\, \mathrm{A} \end{array} \\ \hline \end{array}$$

Table 3.1: Deduction.

$$\begin{array}{|l|l|}\hline \begin{array}{l} \mathrm{A} \rightarrow \mathrm{B} \\ \mathrm{B} \\ \hline \mathrm{A\ m.p.} \end{array} & \begin{array}{l} \mathrm{A} \rightarrow \mathrm{B} \\ \neg\, \mathrm{A} \\ \hline \neg\, \mathrm{B\ m.p.} \end{array} \\ \hline \end{array}$$

Table 3.2: Abduction.

$$\begin{array}{|l|}\hline \begin{array}{l} \mathrm{A} \rightarrow \mathrm{B} \\ \mathrm{C} \rightarrow \mathrm{B} \\ \mathrm{B} \\ \hline \mathrm{C} \rightarrow \neg\, \mathrm{A\ m.p.} \end{array} \\ \hline \end{array}$$

Table 3.3: Explaining away.

Conventional rule-based models have the following properties that can be safely assumed if there is no uncertainty:

1. *modularity*:

   (a) *detachment*: the impact of a new fact to the knowledge base can be calculated in (independent) stages, i.e., in sequences of applying reasoning mechanisms such as Modus Ponens, regardless of how the premises were obtained.

   (b) *locality*: the impact of a new fact to the knowledge base by application of a logical rule (e.g., Modus Ponens) can be calculated regardless of any other knowledge.

2. *monotonicity*: the knowledge base grows monotonically along with obtaining new facts.

If there is uncertainty however, these properties can no longer be assumed without any verification of irrelevancy. If our knowledge base contains rule $A \rightarrow B$ and we find A, we are allowed to draw conclusions about B, regardless of how A was obtained or what other information is in the knowledge base. So the irrelevancy of the other information has already been implicitly assumed (and hopefully, verified as well). Now, consider what we may call the 'probabilistic counterpart' of this rule, the conditional probability of B given A, $P(B|A)$. This statement does not allow us at all to draw conclusions about B on finding A, *unless* we can verify that the other information in the database is irrelevant. In that sense, a probabilistic approach is more secure.

In the following section we will discuss two logical (rule-based) formalisms that have been designed for dealing with uncertainty.

3.4 Default Reasoning

In default reasoning, one deals with incomplete information by 'jumping to conclusions'. These conclusions are plausible, but not provable from the information at hand: more information may be obtained later, making the agent decide withdraw some of his earlier conclusions. For example, utterances in a dialogue that start with a wh-word are information requests, e.g., "what time do you want to go?". However, there are exceptions in which the utterance should not be interpreted as such, e.g., the utterance "who knows".

The *non-monotonicity* of this type of reasoning lies in the fact that it may occur at some point that we obtain information and conclude to new beliefs, thereby causing inconsistencies in our knowledge. In order to remove these inconsistencies, we will have to retract one or more of the beliefs we had adopted until that point.[1] This kind of reasoning violates the monotonicity property in classical logic: if $A \vdash q$ and $A \subset B$ then $B \vdash q$, for any proposition q and sets of propositions A and B. This property tells us that after adding new facts to A, resulting in B, all conclusions from A can still be drawn from B.

Two specific forms of non-monotonic reasoning will now be discussed very briefly: Default Logic and Non-monotonic Logic. Other forms to be mentioned, but not discussed, are Circumscription (McCarthy, 1980) and Truth Maintenance Systems (Doyle, 1979).

3.4.1 Default Logic

In Default Logic (Reiter, 1980), one 'jumps to conclusions' by making default assumptions using 'typicality' information, such as "typically, birds fly". When observing an individual bird, the agent assumes 'by default' that it flies. Exceptions to this rule (e.g., "Tweety does not fly") are possible, but not likely, given

[1] In the field of *Belief Revision*, this process is given a formal treatment (see Gärdenfors, 1992).

the currently available information. As an example from the natural language domain, if an utterance contains a wh-word and it is consistent to assume that the utterance is not one of a fixed set of special expressions (including for example "who knows"), conclude that it is an information request. This can be expressed in the following *default rule*:

$$\frac{\mathit{containsWHWord(utt)} : \neg \mathit{specialExpr(utt)}}{\mathit{infoRequest(utt)}} \tag{3.1}$$

3.4.2 Non-monotonic Logic

Another form of non-monotonic reasoning is an extension of standard logic, called Non-monotonic Logic (McDermott and Doyle, 1980). In this logic, a modal operator M is introduced, where a formula $\mathsf{M}\,p$ should be read as "p is consistent with all current beliefs". The example default rule given in the previous section, may be expressed in non-monotonic logic by the following formulas:

$$\begin{aligned} &\mathit{containsWHWord(utt)} \;\wedge\; \mathsf{M}\ \mathit{infoRequest(utt)} \;\rightarrow\; \mathit{infoRequest(utt)} \\ &\mathit{specialExpr(utt)} \;\rightarrow\; \neg\ \mathit{infoRequest(utt)} \end{aligned} \tag{3.2}$$

In contrast to the principle of jumping to conclusions in default reasoning, quantitative approaches provide a more refined way of representing uncertainty. One of these approaches is probability theory, a well-founded formalism for plausible common-sense reasoning.

3.5 Probability Theory

3.5.1 Bayesian Belief as Probability

Suppose we would like to construct and maintain a model of a particular part of the real world. Concerning our knowledge of that domain, we are restricted to our senses: based on perceptual information we continually update and revise our idea of what the reality is like. Because the information we get does not completely describe all things in the domain that we are interested in, we are uncertain about many elements in our domain. In order to deal with these uncertainties, we may use probability theory in order to reason about this world efficiently and coherently. We can describe states of affairs in our world by means of *propositions*, i.e., sentences that can be either true or false. Because we often don't have absolute certainty about a proposition, we may assign measures of *plausibility* to them.

In the Bayesian view, these plausibilities are called *beliefs*, which behave according to classical probability theory, i.e., a belief assignment $P(.)$ adheres to the *Axioms of Probability Calculus*:

Axiom 3.1 $0 \leq P(A) \leq 1$ *, for any proposition* A*;*

Axiom 3.2 $P(A) = 1$ *, for any tautology* A*;*

Axiom 3.3 $P(A \vee B) = P(A) + P(B)$ *, for any pair* A, B *of mutually exclusive propositions.*

The axioms say that our beliefs are represented using numbers between 0 and 1 (Axiom 3.1), tautologies (i.e., sure propositions such as $A \vee \neg A$) are assigned the belief 1 (Axiom 3.2), and if we have two propositions A and B that can never be true at the same time, then we may take the sum of our beliefs w.r.t. the individual propositions to assess our belief w.r.t. $A \vee B$ (Axiom 3.3).

We think that probabilities should be taken as *objective* (as opposed to subjective) and *mental* (as opposed to physical) belief assignments. In a subjective interpretation of probability, probability assignments are adopted by individual agents. If two different agents adopt different probability assignments to one particular proposition, one of them might be wrong. Subjective probability assignments are arbitrary, while objective probability assignments are assignments that all agents will agree upon in the end. In a physical interpretation of probability, probability assignments are seen as properties of the independent physical world.

In Axiom 3.3, we have a partitioning of the states of the world in which $A \vee B$ is true into states where A is true and states where B is true. More generally, we can partition the states where A is true by means of a set of mutually exclusive propositions B_i and calculate $P(A)$ by *marginalising over* B_i :

Theorem 3.1 (Law of Total Probability) *If* $\{B_1, \ldots, B_n\}$ *is a set of exhaustive and mutually exclusive propositions, then:*

$$P(A) = \sum_i P(A, B_i), \text{ where } P(A, B_i) \text{ is short for } P(A \wedge B_i). \tag{3.3}$$

In the Bayesian view however, probabilities of joint events such as $A \wedge B$ are not the most basic expressions. In stead, *conditional probabilities* are taken to be much easier to estimate. These are probabilities of individual propositions, *given* information concerning the truth of some other propositions. In stead of directly assessing our belief in $A \wedge B$, we can decompose this assessment in first estimating our belief w.r.t. B, and then estimate our belief in A, assuming that we know B :

$$P(A, B) = P(B) \cdot P(A|B) \;\; (= P(A) \cdot P(B|A) \;) \tag{3.4}$$

As indicated between brackets, we can also do the reverse assessment: first estimate our belief in A and then our belief in B, given that we know A. It depends on what A and B represent in practice, which beliefs are actually the easiest to estimate.

This decomposition can be generalised to any set of propositions $A_1, \ldots,$ A_n, where $P(A_1, \ldots, A_n)$ can be assessed by estimating $P(A_1)$, $P(A_2|A_1)$, ..., $P(A_n|A_1, \ldots, A_{n-1})$ subsequently, yielding the *Chain Rule*:

Theorem 3.2 (Chain Rule) *The joint probability of* $(A_1, \ldots, A_n)$ *can be written as:*

$$P(A_1, \ldots, A_n) = \prod_i P(A_i|A_1, \ldots, A_{i-1}) \tag{3.5}$$

The central formula in Bayesian Inference in Statistics is the Bayesian Inversion Formula, or *Bayes' Theorem* 3.3. For mathematicians this is merely a formula that can be simply derived by applying twice the classical definition of conditional probability, $P(A|B) = P(A, B)/P(B)$. However, for people in AI and statistics, it is essential to probabilistic inference, in which experimental data can be organised in such a way, that accurate inferences are possible. In many cases it is hard to assess our belief w.r.t. B, given the truth value of A ($P(B|A)$) directly from experiential knowledge. However, it can be made easier by using Bayes' Theorem to derive this belief from the 'reverse' belief $P(A|B)$, and our *prior* belief $P(B)$, beliefs that may be much easier to assess ($P(A)$ is merely a normalisation constant an therefore needs not to be estimated). In medicine for example, experts find it much easier to estimate the belief that a patient shows certain symptoms, given a particular disease, than the belief that a patient has a particular disease, given that he has certain symptoms, although this is the belief of interest if we want to make a diagnosis of a patient based on the symptoms.

Theorem 3.3 (Bayes' Theorem)

$$P(B|A) = \frac{P(A|B) \cdot P(B)}{P(A)} \tag{3.6}$$

In the following section, we will further generalise our setting of probabilistic modelling from propositions describing the domain to sets of variables.

3.5.2 Random Variables and Conditional Independence

A probabilistic model of our domain consists of a set of *Random Variables* (RVs) $\{X_1, \ldots, X_n\}$. Random variables are events (inquiries, observations, measurements) that can have a number of possible outcomes. [2] An RV taking a certain value, $X_i = x_i$, corresponds to a proposition A. Therefore, every X_i induces a partition of our world into states in which $X_i = x_i$ is true, for each possible value x_i from the domain D_{X_i} of X_i. Every instantiation of all RVs X_i in the model, corresponds to an individual, elementary state of our world, so our model will be completely described by a probability function, assigning beliefs

[2] Jaynes (2003) rejects using the term 'randomness' in probabilistic terminology.

to all possible configurations of the RVs: the *Joint Probability Function* (JPD). This JPD satisfies the Chain Rule (see Theorem 3.2):

$$P(X_1, \ldots, X_n) = \prod_i P(X_i | X_1, \ldots, X_{i-1}) \tag{3.7}$$

The problem of course, is the enormous amount of parameters needed to specify this JPD: if all n variables have at most m values, we need at most $(m-1) \cdot \sum_{i=1}^{n} m^{i-1}$ parameters, a number which increases exponentially in the number of variables ($\mathcal{O}(m^n)$). In order to overcome this problem, *conditional independence* relationships among the variables should be identified, that can strongly simplify the product in the Chain Rule formula and therefore decrease the number of parameters specifying the JPD to be assessed.

In the following, we use some shorthand notation: $P(x)$ is short for $P(X = x)$, a variable X may refer to either an individual variable or a set of variables. The value (or *assignment*) z of a set $Z = \{X, Y\}$ refers to a set of values $\{x, y\}$ for the individual variables in Z: $X = x$ and $Y = y$. So, we have for example $P(z) \equiv P(Z = z) \equiv P(X = x, Y = y)$.

Definition 3.1 (Conditional Independence) *Let U, V and W be any three subsets of RVs from a finite set of variables $\{X_1, \ldots, X_n\}$. U and V are said to be* conditionally independent *given W, notation: $U \perp\!\!\!\perp V \mid W$, if*

$$P(u|v, w) = P(u|w) \text{ , whenever } P(v, w) > 0 \tag{3.8}$$

In other words, new information $V = v$ does not give us additional information on the value of U, given that we know the value of W.

The factors in the product of Equation 3.7, $P(X_k | X_1, \ldots, X_{k-1})$, can be simplified to $P(\ X_k \mid \Pi_k\ )$, where Π_k is a subset of $\{X_1, \ldots, X_{k-1}\}$, if the following conditional independencies can be identified:

$$X_k \perp\!\!\!\perp \{X_1, \ldots, X_{k-1}\} \setminus \Pi_k \ \mid \ \Pi_k \tag{3.9}$$

and hence:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | \Pi_i) \tag{3.10}$$

Now, again assuming that each of the n variables have m possible values, and assuming that each variable has at most k parents ($k \leq n - 1$), then, in stead of at most $(m-1) \cdot \sum_{i=1}^{n} m^{i-1}$ ($= \mathcal{O}(m^n)$), we need at most $(m-1) \cdot n \cdot m^k$ ($= \mathcal{O}(n \cdot m^k)$) parameters.

A set Π_k is called *Markovian Parents* of X_k, if it is a minimal set of predecessors of X_k (i.e., a subset of $\{X_1, \ldots, X_{k-1}\}$) that renders X_k independent of all its other predecessors (i.e., for which Equation 3.9 holds). Given an arbitrary ordering on the variables and a fully specified joint probability distribution, the Markovian Parents of each of the variables can be found, following that variable ordering. If the joint distribution is strictly positive, there is a *unique* set of Markovian Parents for each variable (Pearl, 1988).

Definition 3.1 gives a quantitative account of the notion of conditional independency. However, people seem to be able to identify conditional independencies relatively easy, without the assessment of any exact probabilities. This qualitative account of conditional independence can be reflected by probabilistic models using graphs, in which the nodes represent RVs and the connections represent dependencies between these RVs. In the case of undirected graphs, these models are called *Markov Networks* (also called Markov Random Fields); in the case of directed graphs, *Bayesian Networks*. In Chapter 4, we will discuss Bayesian Networks in detail.

3.6 Other Quantitative Approaches

We will close our discussion of the various approaches to dealing with uncertainty by briefly outlining two other techniques that employ numeric assignments to propositions. Again, we refer to Pearl (1988) for a detailed discussion.

In the *Certainty Factors* framework, that was developed for the MYCIN medical diagnosis system Shortliffe (1976), combination functions are used to compute the numeric certainty factors of conclusions from the certainty factors of the premises and numeric 'credibilities' associated with the rules that were used in drawing the conclusions. This formalism is attractive from a computational point of view, but may lead to inferences that are semantically inappropriate, because in many domains the assumptions of locality and detachment (see Section 3.3) necessary for this formalism are too strong.

Another quantitative approach is *Dempster-Shafer Theory* (Dempster, 1968; Shafer, 1976), a theory that was designed to deal with the distinction between uncertainty and *ignorance*. In the approach of probability theory, a complete specification of the probabilistic model is required. Dempster-Shafer theory however, allows for incomplete specifications. Furthermore, in Bayesian belief assignments the beliefs of A and $\neg A$ add up to 1 for any proposition A, whereas this is not necessarily the case in Dempster-Shafer Theory.

3.7 Machine Learning

Each formalism for dealing with uncertainty has its own way of representing the entities of interest and its own way of reasoning about these entities. Besides these two aspects, there is the question of how to construct a model based on such a formalism. A model can be constructed either from domain expertise or by learning from experience, i.e., by machine learning[3] from empirical data.

[3] (see Mitchell, 1997) for a comprehensive overview.

In this section, we present some of the most common learning techniques, especially those that are relevant with respect to Bayesian networks and alternative methods that have been used in the experiments on dialogue act recognition described in Chapter 6.

3.7.1 Bayesian Learning

In the Bayesian approach to learning, a probabilistic point of view is taken, in which the entities of interest are represented by probability distributions and the learning process is governed by manipulation of these distributions, following the rules of probability theory. Central to Bayesian learning is Bayes' Theorem (see also Formula 3.3):

$$P(h|\mathcal{D}) = \frac{P(\mathcal{D}|h) \cdot P(h)}{P(\mathcal{D})} \tag{3.11}$$

Bayes' Theorem expresses that the *posterior* probability distribution over a hypothesis h, given available data $\mathcal{D}$, is to be computed from the *likelihood* $P(\mathcal{D}|h)$, i.e., the probability distribution of the data given a hypothesis, and the *prior* $P(h)$, i.e., the prior probability distribution over possible hypotheses. The probability distribution $P(\mathcal{D})$ serves as a normalisation constant. Hence, in Bayesian learning, prior knowledge about possible hypotheses is combined with information obtained from experience to derive new, posterior knowledge about the hypotheses.

The most straightforward form of Bayesian learning is *Maximum A-Posteriori estimation* (MAP), in which the solution of the learning problem is given by the hypothesis that maximises the posterior distribution:

$$\begin{aligned} h_{MAP} &= \underset{h}{argmax}\, P(h|\mathcal{D}) \\ &= \underset{h}{argmax}\, \frac{P(\mathcal{D}|h) \cdot P(h)}{P(\mathcal{D})} \\ &= \underset{h}{argmax}\, P(\mathcal{D}|h) \cdot P(h) \end{aligned} \tag{3.12}$$

If we assume no prior knowledge about the hypothesis space and use the Principle of Indifference, i.e., we assume that $P(h)$ is uniformly distributed, the MAP estimation is equivalent to *Maximum Likelihood estimation* (ML), i.e., choosing the hypothesis that maximises the likelihood.

$$h_{ML} = \underset{h}{argmax}\, P(\mathcal{D}|h) \tag{3.13}$$

In the *Minimum Description Length* (MDL) principle, introduced by Rissanen (1978), an information theory interpretation of MAP estimation is given. First, the MAP estimation is reformulated in terms of numbers of bits (using the $\log_2$ function):

$$
\begin{aligned}
h_{MAP} &= \underset{h}{argmax}\, P(h|\mathcal{D}) = \underset{h}{argmax}\, P(\mathcal{D}|h) \cdot P(h) \\
&= \underset{h}{argmax} \log_2 P(\mathcal{D}|h) + \log_2 P(h) \\
&= \underset{h}{argmin}[-\log_2 P(\mathcal{D}|h) - \log_2 P(h)]
\end{aligned}
\tag{3.14}
$$

This reformulation shows that in MAP estimation the hypothesis is chosen that minimises the sum of two terms that measure amounts of information: $log_2 P(\mathcal{D}|h)$ measures the information that is revealed when the data $\mathcal{D}$ is communicated by means of an optimally compact binary encoding (or: *description*) C, if we *already* know hypothesis h; similarly, $log_2 P(h)$ measures the information that is revealed when hypothesis h itself is communicated. Now, the *description length of a message m with respect to an encoding C* refers to the number of bits required to encode m using C, denoted with $L_C(m)$. The MDL principle is now given by:

$$
h_{MDL} = \underset{h}{argmin}\, L_{C_1}(h) + L_{C_2}(\mathcal{D}|h) \tag{3.15}
$$

If the encodings we choose are optimal, i.e., if $L_{C_1}(h) = -\log_2 P(h)$ and $L_{C_2}(\mathcal{D}|h) = -\log_2 P(\mathcal{D}|h)$, then $h_{MDL} = h_{MAP}$.

EM-Algorithm

In many cases of learning from data, some of the variables of interest may not have been observed in the data at all. In such cases, the data is called *incomplete*. The EM (*Expectation Maximisation*) algorithm is designed especially for cases of incomplete data. The algorithm calculates the expected values of the unobserved variables given the current hypothesis and uses them to find an improved hypothesis in the next iteration of the algorithm. The EM-algorithm converges to a local maximum likelihood hypothesis.

Each entry d_j in the data $\mathcal{D}$ is a tuple of individual variable instances. For the case of incomplete data, let the unobserved variables be denoted $Z = (Z_1, \ldots, Z_s)$, and the observed variables $X = (X_1, \ldots, X_r)$. A data entry d_j then corresponds to an instance of the observed variables, say $(X_1 = x_{1j}, \ldots$ $\ldots, X_r = x_{rj})$, while the unobserved variables $Z_1, \ldots, Z_s$ are unknown. The *full data* can be seen as generated by a probability distribution over a variable $Y = X \times Z$. The learning problem consists of finding the parameters θ that describe this distribution.

In each iteration of the EM-algorithm, an improved hypothesis h' for the values of the parameters θ given the current hypothesis h is found by seeking the hypothesis h' that maximises $E[\ln P(Y|h')]$, which can be seen as a function of the likelihood $P(Y|h')$ of the full data given hypothesis h'. As the distribution over Y is determined by the parameters θ that are to be estimated and therefore unknown, the current hypothesis h will be used instead, i.e. we assume $\theta = h$. Now, we define a function $Q(h'|h)$ that gives the expected value $E[\,\ln P(Y|h')\,]$ as a function of h', under the assumption $\theta = h$ and given the observed part X of the full data Y.

$$Q(h'|h) = E[\,\ln P(Y|h')\mid h, X\,] \tag{3.16}$$

The algorithm repeats the steps of *Expectation* (i.e., computing $Q(h'|h)$) and *Maximisation* (i.e., finding an improved hypothesis h') until a local optimum is found (see Algorithm 3.1).

Algorithm 3.1 EM

```
repeat
    Expectation: calculate $Q(h'|h)$ using the current hypothesis $h$ and the observed data $X$ to estimate the probability distribution over $Y$.
        $Q(h'|h) := E(\ln P(Y|h')|h, X)$    (3.17)
    Maximisation: replace hypothesis $h$ by the hypothesis $h'$ that maximises this $Q$ function.
        $h := \underset{h'}{argmax}\, Q(h'|h)$    (3.18)
until $h$ local optimum for $Q(h'|h)$
```

Classification

When using machine learning techniques for classification, the hypotheses h may be conceived as functions $h : X \to C$ that map instances $x \in X$ to class values $c \in C$. Such functions may however also be probabilistic and map to probability distributions over the class values. The instances are specified using a vector of attributes $(a_1, \ldots, a_n)$ or feature-values pairs.

In *Bayes Optimal Classification*, the probability distribution over all possible hypotheses, given the data, is integrated in the classification process:

$$v^* = \underset{v_j \in V}{argmax} \sum_{h_i \in H} P(v_j|h_i)P(h_i|\mathcal{D}) \tag{3.19}$$

The Bayes optimal classification v^* is generally not equal to the classification by the MAP hypothesis, i.e., there may be a new instance x for which $v^*(x) \neq h_{MAP}(x)$.

BAYESIAN CLASSIFICATION:

$$\begin{aligned} v_{MAP} &= \underset{v_j \in V}{argmax}\, P(v_j|a_1,\ldots,a_n) \\ &= \underset{v_j \in V}{argmax}\, P(a_1,\ldots,a_n|v_j)\cdot P(v_j) \end{aligned} \tag{3.20}$$

NAIVE BAYES CLASSIFIER: in this classifier the attributes a_i are assumed to be conditionally independent, given the target value v_j, hence:

$$v_{NB} = \underset{v_j \in V}{argmax}\, P(v_j)\cdot \prod_i P(a_i|v_j) \tag{3.21}$$

3.7.2 Decision Trees

A decision tree is a classifier that is represented by means of a tree structure. Except for the leaf nodes, each node in the tree represents an attribute test on a given instance to be classified, in other words, at each node a particular feature is evaluated leading to a different new node down the tree for each of the possible resulting values. The process of classifying a given instance corresponds to a path down the tree, starting at the root node and ending at one of the leaf nodes, each of which corresponds to a class value.

One of the state-of-the-art algorithms for learning decision trees from data is the C4.5 algorithm (Quinlan, 1993). This decision tree learner has been used in our experiments on dialogue act classification (Chapter 6). In Figure 3.5, part of a decision tree that was learned from data in our experiments on dialogue act classification is shown. Each new instance, i.e., an utterance represented by a set of feature-value pairs, is classified by this decision tree by first testing on the feature **flf-1** (this refers to a dialogue act type of the previous utterance; for further details on the features and the class variable **blf** used in this decision tree, see Section 6.3). For example, if **flf-1** has value *query_ref* for the instance to be classified, the feature **sp-1** is to be tested next. If this feature has value *null*, the instance is classified as `positive_answer`; if the feature has value *S*, a further feature has to be evaluated: **startsWithWHExpr**.

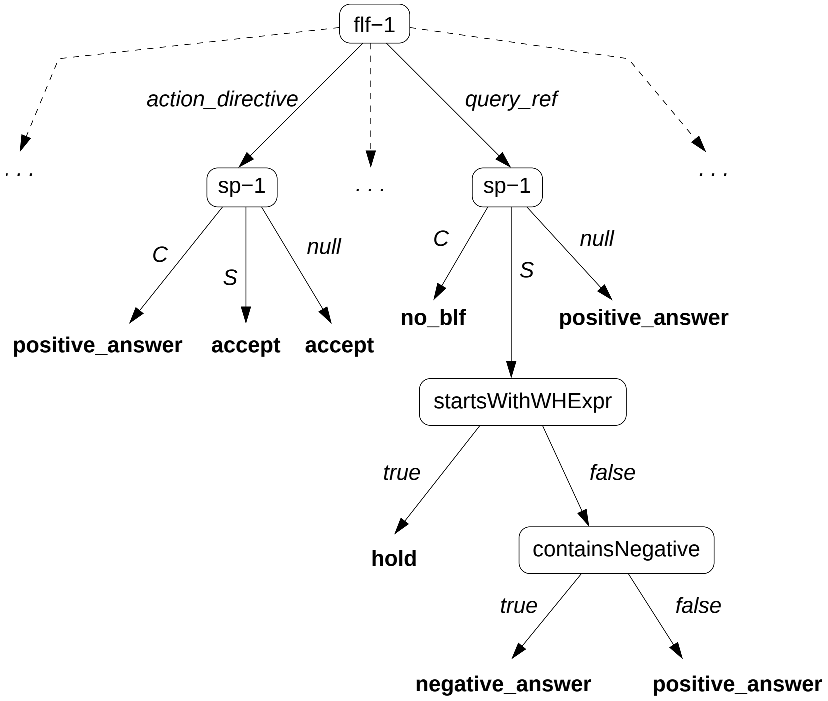


Figure 3.5: Example of a decision tree for dialogue act classification.

Chapter 4

Bayesian Networks

In which Bayesian networks will be defined as computational probabilistic models, thereby combining probability theory and graph theory. The definition is motivated by the model's foundation in probability theory and the visual and computational appeal of representing explicit assumptions of conditional independence. Further essential questions concerning Bayesian network models will be addressed, such as how they can be used and how they can be constructed.

4.1 Introduction

In the theory of Bayesian networks, probability theory and graph theory are combined into a framework for probabilistic inference. By means of a directed graph, i.e., a mathematical structure consisting of nodes and arcs (i.e., ordered pairs of nodes), conditional independencies between the different (stochastic) variables in a probabilistic model can be specified. These independencies allow the joint probability distribution to be factorised into a set of conditional probability distributions. This makes the model more easy to be constructed and allows for more efficient algorithms for reasoning with the model.

Reasoning (or *inference*) in a Bayesian network consists of computing posterior probabilities, given newly obtained information. The new information is entered into the network by setting some of the variables to particular values. This information on the observed variables is then propagated through the network (the information ′flows′ along the arcs), causing the probability distributions of the individual, unobserved variables to be updated. The overall joint probability distribution can be updated as well and possibly a new configuration (also referred to as ′instantiation′ or ′assignment′) of the variables will maximise this distribution. This configuration is also called the *most probable explanation* for the information.

An essential question to ask is of course how a Bayesian network is designed for the particular domain of interest. One needs to specify the variables of interest, including *hypothesis variables* – variables that we cannot observe directly but are of interest to us because we might for example let our future actions depend on them – and *information variables* – variables that we can observe and are in some way related to the hypothesis variables. Next, we have to find out how exactly these variables are interrelated and how this is expressed by means of the arcs between the nodes representing the variables. Finally, the required probabilities need to be assessed. This assessment is an important point of criticism against Bayesian networks and Bayesian probability in general: where do the numbers come from? Well, roughly speaking, there are two ways of assessing the numbers. Either we use the judgements made by domain experts, or we use empirical data and construct the Bayesian network using statistical techniques. However, techniques that combine these two approaches are also possible. This will result in Bayesian network models that are based on expert knowledge and fine-tuned by using raw empirical data.

In the following sections, we will define the Bayesian network model and try to shed some light into the problem of constructing Bayesian networks. After that, some extensions of the general Bayesian network model are discussed, software tools for developing and employing Bayesian networks that have been used during our research, and an outline of applications of Bayesian networks in various domains are discussed.

4.2 Definition

In the literature, different definitions of Bayesian Networks have been given. Some of them are equivalent, other are clearly not. After we have outlined the definition used in this thesis, including many general notions that also play a role in the alternative definitions, we will discuss the main differences in Section 4.2.1.

In this thesis, Bayesian networks are defined as probabilistic models over sets of stochastic variables. More specifically, a Bayesian network consists of two parts: a graph representation of conditional independence assumptions among the set of variables, and a set of conditional probability distributions associated with that graph. In Definition 4.1, a more formal account is given.

Definition 4.1 (Bayesian Network) *A Bayesian Network is a probabilistic model over a set of stochastic variables* $\{X_1, \ldots, X_n\}$*, consisting of two parts:*

1. *a Directed Acyclic Graph (DAG) G, i.e., a directed graph without any directed cycles.*

   The graph G is defined by a pair (V, A)*, where:*

   (a) *V is a set of nodes ('vertices')* $\{V_1, \ldots, V_n\}$*, where each node* V_i *represents exactly one variable* X_i *and vice versa, each variable* X_i *is represented by exactly one node* V_i *(i.e., there exists a 1-1-mapping between V and X).* [1]

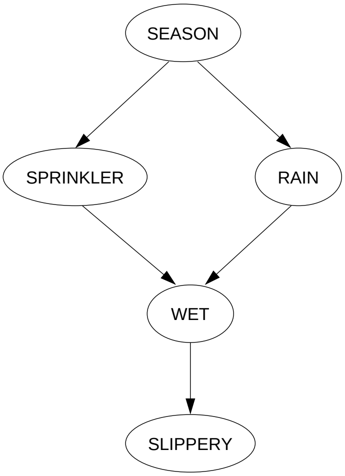
[1] After this definition, we will use the same label for both node and represented variable, unless explicitly stated otherwise

(b) *A is a set of arcs between the nodes: an arc from a node V_i to a node V_j represents an informational dependency between the variables represented by these nodes: observing the value of the one gives us information about the value of the other.*

2. *a set of Conditional Probability Distributions (CPDs).*

   With each variable X, a conditional probability distribution $P(X|pa(X))$ is associated, where $pa(X)$ is the set of variables represented by the parents *in G of the node V representing X.*

In Figure 4.1 an example is given. All variables are Boolean-valued, except $Season$ that takes values in $\{spring, summer, autumn, winter\}$. So for $P(Sprinkler|Season)$ we need to assess $(2-1)\cdot 4 = 4$ numbers and for $P(Wet|Sprinkler, Rain)$ we need $(2-1)\cdot(2\cdot 2) = 4$ numbers.



The required CPDs are:

- $P(Season)$
- $P(Sprinkler|Season)$
- $P(Rain|Season)$
- $P(Wet|Sprinkler, Rain)$
- $P(Slippery|Wet)$

Figure 4.1: Example Bayesian Network

In Definition 4.1, an arc from node V_i to node V_j was said to indicate an informational dependency between the corresponding variables. A more basic relation expressed by the graph structure as a whole, is that of conditional independence. It is this relationship that makes it sufficient to specify only the conditional probability distributions of all variables, given their parents in the network.

The connection between conditional independency of variables and properties of the network is founded in the graph theoretical concept of *d-separation* (*d* for *directional*). Informally, the variables in X are conditionally independent of those in Y, given the variables in E, if the nodes in E 'block' the information flow between nodes in X and Y. The information flow between X and Y takes place along *undirected paths*: paths in the underlying undirected graph from nodes in X to nodes in Y. Undirected paths in the network of Figure 4.1

include $Sprinkler, Wet, Slippery$ but also $Sprinkler, Season, Rain$, where the arcs involved do not all point in the same direction.

Definition 4.2 (d-separation) *Let G be a DAG and let E, X and Y be disjunct sets of nodes in G. Then E* d-separates *X and Y, if every undirected path from a node in X to a node in Y is* blocked, *given E. Notation: $\langle X \mid Z \mid Y \rangle$*

Every node Z on an undirected path (the starting and ending point excluded), is connected with two other nodes P_1 and P_2 in one of three possible ways. On the path, Z may be 1) *linear*: (P_1, Z) and (Z, P_2) are arcs in the graph, 2) *diverging*: (Z, P_1) and (Z, P_2) are arcs in the graph, or 3) *converging*: (P_1, Z) and (P_2, Z) are arcs in the graph. An undirected path being blocked by a set of nodes is defined along these three situations in Definition 4.3.

Definition 4.3 *An undirected path P is* blocked *given a set of nodes E, if there is a node Z on the path for which one of three conditions holds:*

1. *$Z \in E$ and Z is linear on P.*
2. *$Z \in E$ and Z is diverging on P.*
3. *$Z \notin E$ and $Q \cap E = \emptyset$, where Q is the set of descendants of Z (the nodes to which there is a directed path starting from Z) and Z is converging on P.*

The conditions are illustrated in Figure 4.2.

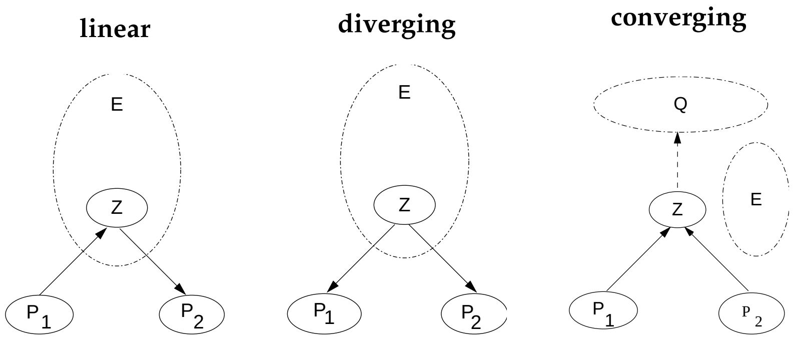


Figure 4.2: Schematic view of the conditions in Definition 4.3

In the example of Figure 4.1, the set of nodes $\{Sprinkler, Rain\}$ d-separates $\{Season\}$ and $\{Wet\}$, because $Sprinkler$ blocks the path $(Season, Sprinkler, Wet)$ and $Rain$ blocks the path $(Season, Rain, Wet)$. In both cases, the blocking node is linear on the path. Similarly, $\{Wet\}$ d-separates $\{Sprinkler, Rain\}$ and $\{Slippery\}$. Furthermore, $\{Season\}$ d-separates $\{Sprinkler\}$ and $\{Rain\}$, where $Season$ is diverging on the only path between them, $(Sprinkler, Season, Rain)$. The empty set d-separates $\{Sprinkler\}$ and $\{Rain\}$, because Wet is converging on $(Sprinkler, Wet, Rain)$ and Wet nor its descendant $Slippery$ is in the empty set. Because Wet and $Slippery$ neither are elements of the set $\{Season\}$, $\{Season\}$ also d-separates $\{Sprinkler\}$ and $\{Rain\}$. These relations of d-separation are summarised in Equation 4.1.

$$\begin{aligned}
&\langle\ \{Season\}\ \mid\ \{Sprinkler, Rain\}\ \mid\ \{Wet\}\ \rangle\\
&\langle\ \{Sprinkler, Rain\}\ \mid\ \{Wet\}\ \mid\ \{Slippery\}\ \rangle\\
&\langle\ \{Sprinkler\}\ \mid\ \{Season\}\ \mid\ \{Rain\}\ \rangle\\
&\langle\ \{Sprinkler\}\ \mid\ \{Season\}\ \mid\ \{Rain\}\ \rangle\\
&\langle\ \{Sprinkler\}\ \mid\ \emptyset\ \mid\ \{Rain\}\ \rangle
\end{aligned} \tag{4.1}$$

The DAG G of a Bayesian Network defines a factorisation of the JPD P for the probabilistic model, analogous to Equation 3.10. Thus, G defines a class of probability distributions that allow this factorisation, i.e., 'are Markov-compatible with G'.

Definition 4.4 (Markov-compatibility) *If a joint probability distribution P allows the factorisation*

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | pa(X_i)) \tag{4.2}$$

relative to a DAG G, P is said to be Markov-compatible with G.

In all distributions P that are Markov-compatible with G, (sets of) variables X and Y are independent given Z, if X and Y are d-separated by Z:

$$\langle X \mid Z \mid Y \rangle_G \implies X \perp\!\!\!\perp_P Y \mid Z, \text{whenever } P \text{ compatible with } G \tag{4.3}$$

Generally, the reverse is *not* the case: there might be JPDs compatible with G, containing conditional independencies that are not reflected in terms of d-separation in G. However, the independencies that hold in *all* distributions compatible with G *are* reflected:

$$X \perp\!\!\!\perp_P Y \mid Z \text{ for all } P \text{ compatible with } G \implies \langle X \mid Z \mid Y \rangle_G \tag{4.4}$$

With respect to any joint distribution compatible with the DAG in Figure 4.1, various conditional independencies hold, for example, that $Sprinkler$ and $Slippery$ are conditionally independent, given Wet. If we already know that the ground is wet, then any new information on whether the sprinkler was on or not is irrelevant to the question whether or not the ground is slippery. In the same way, if we know what season it is, any information on whether or not it has rained is irrelevant to the question of whether or not the sprinkler was on. But: any new information concerning whether it rained is irrelevant to the question whether or not the sprinkler was on, *unless* we already know that the ground is wet or slippery. In that case, a dependency emerges that can be understood intuitively using a causal interpretation of the arcs in the graph: it can be seen as a dependency between two common *causes* of the same known *effect* and learning about one cause makes the other cause less probable, because the first cause already explains our information about the effect. This form of reasoning is called *explaining away*.

We can distinguish various patterns of plausible common-sense reasoning that are supported by Bayesian networks (see also Section 3.3). These patterns can be explained using the example network in Figure 4.1:

- *predictive*: suppose we believe that if it has rained, it becomes more plausible for us that the ground is wet. If we find out that it has rained, the ground being wet has become more plausible for us. This pattern can be seen as a weakened form of deduction.

- *diagnostic*: suppose we believe that if the ground is wet, it becomes more plausible for us that the ground is slippery, and we find out that the ground is slippery; then the ground being wet has become more plausible for us. This pattern can be seen as a weakened form of abduction.

- *explaining away*: suppose we believe that both an activated sprinkler and the fact that it has rained makes the ground being wet more plausible for us, and we know that the ground is actually wet; if we find out that the sprinkler was activated, the fact that it has rained becomes less plausible for us.

In Section 4.3, we will give a more technical and quantitative account of using a Bayesian network.

Summarising, a Bayesian Network is a probabilistic model on a set of stochastic variables, consisting of a DAG G that specifies a set of conditional independencies among the variables, thereby justifying a compact representation of the joint probability distribution, and a set of conditional probability distributions that required to make the full specification of the joint distribution complete. Given such a Bayesian network, we can do plausible reasoning: draw conclusions in terms of probabilities given newly obtained information.

4.2.1 Discussion

We will now discuss some of the differences in defining Bayesian networks that are around and place our definition into that context.

Pearl

Pearl (1988) makes a clear distinction between a probabilistic model (PM) and a Bayesian network (BN), which he sees merely as a directed acyclic graph (DAG) with certain properties that can be used to express certain properties of a PM. A PM is a pair $M = (U, P)$ where U is a set of variables and P is a joint probability distribution over U. A DAG is a pair $G = (V, A)$ where V is a set of nodes and A is a set of arcs between the nodes. In a PM, *conditional independency relationships* between the variables in U can be identified (see Definition 3.1), while in a DAG, *d-separation relationships* can be identified (see Definition 4.2).

Pearl connects DAGs with PMs by means of the notion *I-map*: a DAG $G = (V, A)$ is an I-map of a PM $M = (U, P)$ if:

a) there is a 1-1-mapping between U and V, and

b) for all disjoint subsets R, S and T of V, if $\langle R \mid S \mid T \rangle_G$ then $X \perp\!\!\!\perp_M Y \mid Z$, where X, Y and Z are associated with R, S and T respectively, according to the 1-1-mapping in a).

Finally, Pearl defines a DAG G to be a Bayesian network for a PM M, if G is a *minimal* I-map of M, i.e., removing any arcs in G will cause it not to be an I-map of M anymore.

So, Pearl sees a Bayesian network as a DAG that reflects conditional independence relationships of a given PM, and *not* as a probabilistic model in itself, which is in contrast with our definition. Because we are discussing Bayesian networks from a computational point of view, we prefer to see them as computational models, requiring both the independency relationships reflected by the DAG and the associated conditional probability distributions.

Another important difference with Pearl's definition is the minimality-constraint. In our definition, a fully-connected DAG corresponding to any ordering on the variables, i.e., a DAG in which the parents of all nodes R are exactly the nodes that represent the predecessors of variable X represented by R, is allowed for a Bayesian network by virtue of the Chain Rule 3.7. However, Pearl's definition may not allow this DAG to be a Bayesian network for the probabilistic model, because there may be predecessors of X that are independent of X, given the other predecessors of X.

Neapolitan

Neapolitan (1990) defines Bayesian networks in a similar way as Pearl does, but his definition is not entirely equivalent and furthermore, refers to them as *causal networks* in stead of Bayesian networks or belief networks. Pearl has recently discussed causal networks as well, but defines those as a specific type of Bayesian network, related to the notion of *interventions*, i.e.,*forcing* a variable to some value (in stead of *observing* the value). For further details, see (Pearl, 2000).

Neapolitan also considers a joint probability distribution P and then defines a causal network as a pair (P, G), where G is a DAG with nodes representing the variables over which P is specified (via a 1-1-mapping, as in Definition 4.1) and in which each variable X is conditionally independent of the variables represented by the non-descendants of the node V that represents X, given the parents of V, relative to distribution P.

Just like in our definition, no minimality-constraint is used: a fully-connected DAG constructed relative to an arbitrary ordering on the variables is allowed for a causal network by virtue of the Chain Rule 3.7, although after removal of some arcs, the resulting tighter independency relationships reflected might still hold for P.

Furthermore, Neapolitan sees a causal network not just as a graph, but as a joint probability distribution combined with that graph. However, he does not use the notion of conditional probability distributions, as we do.

Russell & Norvig

In our definition, we follow the definition of *belief networks*, given by Russel and Norvig (1995), who also define it as a DAG plus associated conditional probability distributions. A Bayesian network is seen as a piece of *syntax* with a local and a global semantics that are equivalent: the *local semantics* states that the DAG represents a set of conditional independence assertions, while the *global semantics* states that the joint probability distribution is given by the product of the conditional probability distributions.

4.3 Use

Suppose that we have a Bayesian network that is completely specified with the conditional probability distributions (CPDs) required by the DAG of the model. Then the joint probability distribution is given by the product of these CPDs:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | pa(X_i)) \tag{4.5}$$

Now we can use this completely specified probabilistic model for inference: we can draw conclusions about the variables when new information is obtained. This new information is called *evidence* and has the form of an instantiation (also called 'assignment' or 'configuration') of a number of variables in the network. Given some new evidence, we can perform two types of inference: *belief updating* and *belief revision*. Belief updating concerns the computation of (marginal) probabilities over variables, while belief revision concerns the computation of the most probable assignment (or configuration, or instantiation) of variables. These terms should not be confused with updating and revision in the general theories of belief change. In those theories, belief revision is about changing our beliefs concerning a static world, because of new information we get about that world, and belief updating is about changing our beliefs concerning a dynamic world, because of changes occurring in that world. For more information on these notions of belief change, see for example (Gärdenfors and Rott, 1995; Katsuno and Mendelzon, 1992). In Section 4.5.3 we will discuss Dynamic Bayesian networks, which may account for worlds in which changes may occur.

Suppose we obtain the evidence e ($E = e$), denoting the instantiation of a set of variables E in the network. Then our posterior belief in $X = x$ (the instantiation of either an individual RV or a set of variables), can be obtained from the JPD as follows:

$$P(x|e) = \frac{P(x,e)}{P(e)} = \frac{\sum\limits_{s} P(x,e,s)}{\sum\limits_{x,s} P(x,e,s)} \ , \tag{4.6}$$

where s denotes the values of all variables, excluding X and E. In the process of belief revision, we determine the configuration y^* of the variables in such a subset Y, that maximises the joint probability over these variables, given the evidence e:

$$y^* = \underset{y}{argmax}\, P(y|e) \tag{4.7}$$

In Equation (4.6) we have shown how posterior beliefs can be derived from the joint probability distribution (JPD), by using the definition of conditional probability and then take summations of the JPD in both numerator and denominator. However, as inferences in Bayesian networks have been shown to be NP-hard (Cooper, 1990), algorithms are needed that take care of the inferences more efficiently. This can be done by utilising the compact representation of the joint distribution (Equation 4.5), following from the independence assumptions.

Pearl (1982) developed a *Message-passing* algorithm for networks that are *tree-structured*: the networks are DAGs in which every node has at most one parent. In the algorithm, every node in the network is assigned a processor that receives information from its children, processes it and passes updated information to its parent and other children and vice versa: processes information from its parent to send updated information to its children. This flow of information is started by the evidence observed at some of the variables in the network and continues until an equilibrium has been established. In this equilibrium, the processors at the nodes will have stored the marginal probability distributions. This message-passing technique has been extended by Kim and Pearl (1983) for *singly-connected* networks (also called *polytrees*): DAGs in which there is at most one directed path between any two nodes.

Two main approaches have been taken towards inference in *multiply-connected* networks, i.e., DAGs where multiple directed paths between two nodes are allowed. In the *Join-tree propagation* approach (Lauritzen and Spiegelhalter, 1988), the DAG is transformed into a polytree. This is done through graph-theoretical operations like *moralisation* and *triangulation* and then clustering the nodes according to the cliques in the graph. The resulting polytree is called *junction tree* or *join-tree*. Two variants on this approach we mention here, are *Bucket elimination* (Dechter, 1996) and *Variable elimination* (Zhang and Poole, 1996).

The second approach is called *Cutset conditioning* (Pearl, 1986) and (Jensen, 1996; Jensen et al., 1990), in which the multiply-connected DAG is transformed into several simpler singly-connected networks, in each of which a certain variable has been instantiated to a different definite value.

If the complexity of the algorithms still exceeds reasonable bounds, approximation methods like *Stochastic simulation* can be applied.

4.3.1 Classification

In Section 3.7, we already discussed the Decision Tree and the Naive Bayes classifiers. Bayesian networks can also be seen as a classifier and in this sense, the Naive Bayes classifier is a special case of a Bayesian network classifier (hence the qualification 'naive').

Consider the Bayesian network in Figure 4.3 that may be used for dialogue act classification. In Chapter 6, more refined versions of this model will be discussed and experimented with. The network contains one node that represents the dialogue act type of a given utterance (this is the class variable *DA*), three nodes that represent surface features of that utterance (*NumWords*, *CanYou* and *IWant*) and one node that represents a feature of the context in which the utterance was produced, the dialogue act type of the previous utterance (*PrevDA*). Classification using this Bayesian network operates in a straightforward way using the maximisation specified in (4.7), with the simplification that evidence is available for all feature variables, i.e., all variables except the class variable, and the most probable class value is found by maximising over the values of the single class variable.

In a Naive Bayes classifier, the features are considered conditionally independent, given the class variable. Such a model however can also be expressed in a Bayesian network: see Figure 4.4. Therefore, we may conclude that in theory, Bayesian network classifiers can be found that perform at least as good as Naive Bayes classifiers.

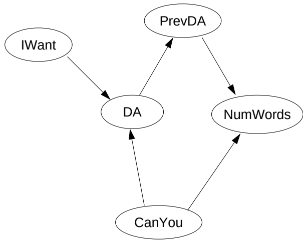


Figure 4.3: A Bayesian Network for Dialogue Act Recognition.

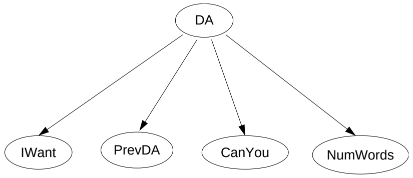


Figure 4.4: Naive Bayes classifier as Bayesian Network.

4.4 Construction

When designing and constructing a Bayesian network for a particular domain, we first need to specify what aspects of the domain are considered relevant. We are restricted however to a characterisation of the domain in terms of a set of stochastic variables (continuous or discrete). The set of variables should include both variables that are observable – these will be called *information variables* – and variables that are not observable – these are referred to as *hypothesis variables*.

After having defined the variables, any conditional independencies among them can be specified by means of building a directed acyclic graph in which each node represents one variable (more precisely, there is a mapping between the set of nodes in the graph and the set of variables in the model in which each variable is represented by exactly one node and, vice versa, each node represents exactly one variable). This can be done either by using domain experts to decide on the conditional independencies, or by inducing the network structure from raw, empirical data using machine learning techniques. The latter method will be discussed in Section 4.5.

Finally, the conditional probability distributions that are required by the network structure chosen need to be assessed. Here, the solutions may range from well-founded theories of the domain to subjective estimates made by (human) domain experts. As was the case for structure learning, we can also use suitable raw data to induce the probability distributions, see Section 4.5.

In general, the method given in Algorithm 4.1 can be followed in constructing a Bayesian network (we use the same names for both variables and corresponding nodes in the network).

The choice of the ordering of the variables in step 2 is a crucial one. In principle, any ordering is allowed, but the next step of specifying the Markovian parents can turn out to be rather difficult and error-prone in many orderings. Pearl (2000) claims that using causal information in constructing Bayesian networks leads to more reliable models, because conditional independence relationships are more accessible to the mind when anchored onto causal relationships. Therefore, a variable ordering in which causes precede effects should be preferred.

However, in principle, alternative orderings on the variables – and even when using the constraint of causality, several alternatives remain – lead to different Bayesian networks. Theoretically speaking, even from a *given* joint probability distribution P, where conditional independencies can be identified exactly by computation (see Definition 3.1), different orderings generally lead to different network structures, and therefore to different representations of the same joint distribution. If P is strictly positive, then the set of Markovian Parents for each variable is *unique* (see also our remark on the Markovian Parents in Section 3.5 in the previous chapter), and therefore the DAG G found by construction is unique. (Pearl, 1988, Section 3.3, Corrolary 3)

Besides using causal intuition, it may sometimes be convenient to introduce additional *mediating* variables, in order to make the assessment of conditional

Algorithm 4.1 General method for the construction of a Bayesian network.

```
Choose the set $X$ of relevant variables to describe the domain;
choose an ordering for the variables, say $X = (X_1, \ldots, X_n)$;
add nodes for $X_1$ and $X_2$ to the network and decide whether they are dependent; if so, draw an arc from $X_1$ to $X_2$;
for $i = 3$ to $n$ do
    add a node for $X_i$ to the network;
    select any minimal set of $X_i$'s predecessors $\Pi_i \subseteq \{X_1, \ldots, X_{i-1}\}$, that 'screens off' $X_i$ from its other predecessors:

        $P(x_i|x_1, \ldots, x_{i-1}) = P(x_i|\Pi_i)$

    and draw arcs from each $X_j \in \Pi_i$ to $X_i$, i.e., set $pa(X_i) := \Pi_i$, where $pa(X_i)$ are the parents of $X_i$ in the underlying DAG of the Bayesian network; $\Pi_i$ is the set of Markovian Parents of $X_i$, see Section 3.5.2;
end for
for $i = 1$ to $n$ do
    determine the conditional probability distributions $P(X_i|\Pi_i)$ (= $P(X_i)$, if $\Pi_i = \emptyset$);
end for
```

independencies more easy and therefore the network more accurate (we refer to (Jensen, 1996, Section 3.1) for more details).

In Chapter 6, we will describe how this modelling process was applied to the domain of human-machine dialogue, for the specific task of recognising dialogue acts.

4.5 Machine Learning

In this section, we will describe how Bayesian networks can be constructed from empirical data using machine learning techniques. This learning process can be subdivided into two tasks: 1) learning the network structure (the DAG), and 2) learning the conditional probability distributions of the variables, given their parents as indicated by that network structure.

For more detailed overviews on learning Bayesian networks and more in general, graphical models, see Heckerman (1995) or Buntine (1994).

4.5.1 Learning the Network Structure

In learning the network structure for a Bayesian network, a Directed Acyclic Graph, representing a set of conditional independency assumptions on the variables, two general approaches can be distinguished:

1. *constraint-based* : starting with a fully connected graph, arcs are removed if certain conditional independencies are measured in the data.
2. *search-and-score* : a search is performed through the space of possible DAGs, and either the best one found, or a sample of the models found is returned; because the number of possible DAGs is exponential in the number of nodes, a set of assumptions and a heuristic search method is required, e.g., a local search algorithm like greedy hill climbing or a global search algorithm like Markov Chain Monte Carlo.

The second type of structure learning is the most popular approach; in the following section we will discuss the well-known K2 algorithm, which is also used in the experiments as described in Chapter 6.

The K2 algorithm

In Section 4.4 a general method was given for constructing a Bayesian network (Algorithm 4.1). After defining an ordering on the variables, the DAG for the model was to be built, given that ordering. The K2 algorithm for learning the DAG for a Bayesian network from data, developed by Cooper and Herskovits (1992), also takes as starting point an ordered set of nodes $X_1, \ldots, X_n$ and for each node X_i, selects a subset of its predecessors X_1,. . . ,X_{i-1} to be its parents in a DAG. In this algorithm however, selecting the parents is not based on expert judgements concerning conditional independencies, but on raw empirical data and some statistical measure of the quality of a DAG.

In the original form, the algorithm uses as quality measure for DAGs their probability, given the data. So basically, the idea is finding the DAG G that is most probable, given the data D, i.e., to find the DAG G that maximises $P(G|D)$. However, the method of K2 in fact searches for DAGs that are more probable than the current DAG found so far. Therefore it is sufficient to consider only the probability for a DAG G_1, *relative* to another DAG G_2. Furthermore, it is mathematically more convenient to change the focus from comparing conditional probabilities of DAGs, $P(G|D)$, to joint probabilities, $P(G, D)$. This can be done because of the following equivalence:

$$\frac{P(G_1|D)}{P(G_2|D)} = \frac{P(G_1, D)}{P(G_2, D)} \tag{4.8}$$

Now, let G be a DAG with nodes corresponding to the discrete variables $X_1, \ldots, X_n$ and D a database consisting of m cases, where each case is an instantiation $x_1, \ldots, x_n$ of the variables. Under the assumption that the cases in the database are independent, given a Bayesian network model, that D does not contain cases with missing values, and that all possible DAGs are equally probable, prior to the data, it can be shown that:

$$P(G, D) = P(G) \cdot \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \tag{4.9}$$

where q_i is the number of unique instantiations of the parents of X_i, relative to G, N_{ijk} the number of cases in D in which X_i takes its k-th value (of the r_i possible ones) and the parents of X_i take their j-th possible instantiation (according to some ordering on all possible instantiations of those parent variables); $N_{ij} = \sum_k N_{ijk}$. This probability is called the *Bayesian measure*.

As the K2-method only needs the probabilities for DAGs, relative to another DAG and all possible DAGs are considered equally possible, the factor $P(G)$ is irrelevant in the algorithm. The rest of the formula is a product of n factors, each corresponding to one variable. The DAGs can now be found by selecting parent nodes for each variable X_i individually from its predecessors, relative to a given ordering on the nodes. In each of these selection procedures, it suffices to use only the factor corresponding to that variable:

$$g(i, \pi_i) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \tag{4.10}$$

where π_i is the set of parents of X_i, selected from its predecessors $Pred(X_i)$. In Algorithm 4.2, the specification of K2 is given.

Algorithm 4.2 K2

Input: n nodes, an ordering on the nodes, an upper bound u on the number of parents a node may have, and a database D containing m cases.

Output: for each node X_i a set of parent nodes π_i.

```
for $i = 1$ to $n$ do
  $\pi_i := \emptyset$; $P_{old} := g(i, \pi_i)$; $OK := true$;
  while $OK \wedge |\pi_i| < u$ do
    $z^* := \underset{z \in Pred(X_i)}{argmax}\ g(i, \pi_i \cup \{z\})$;
    $P_{new} := g(i, \pi_i \cup \{z^*\})$;
    if $P_{new} > P_{old}$ then
      $P_{old} := P_{new}$;
      $\pi_i := \pi_i \cup \{z^*\}$;
    else
      $OK := false$;
    end if
  end while
end for
```

Variants on the K2 procedure are mostly based on variants of the quality measure for DAGs. In stead of the probability of DAGs, a measure based on entropy or the MDL principle (see also Section 3.7) may be used (Bouckaert, 1993; Lam and Bacchus, 1994). There have also been various attempts to deal with the problem of the K2 algorithm requiring an ordering on the variables, most notably the use of genetic algorithms to find optimal orderings for structure learning (Hsu et al., 2002, see e.g.,).

4.5.2 Learning the Conditional Probability Distributions

After finding a graph structure for a Bayesian network, whether by using machine learning or not, the required conditional probability distributions for the model can be assessed. Let $\boldsymbol{X} = \{X_1, \ldots, X_n\}$ be a set of variables and S_m a fixed structure of a Bayesian network with nodes representing these variables. Further, let $\boldsymbol{\Theta_m}$ be a random variable with parameter sets $\boldsymbol{\theta_m}$ as possible values, each of which contains a set of real numbers specifying the 'true' conditional probability distributions required for a Bayesian network with structure S_m. Suppose we have raw data in the form of a number of *cases*, in each of which the values of all variables are known. This is called a *random sample* $\mathcal{D} = \{\boldsymbol{x_1}, \ldots, \boldsymbol{x_N}\}$, where each $\boldsymbol{x_k}$ is an instantiation of the variables in $\boldsymbol{X}$. We now would like to derive values in $\boldsymbol{\theta_m}$ that accurately explain these data and thus provide the conditional probability distributions required for the Bayesian network. The classical method of assessing these values is Maximum Likelihood Estimation.

Maximum Likelihood Estimation (MLE)

Maximum Likelihood Estimation consists of choosing the parameter set $\boldsymbol{\theta_m}$ that maximises the *Sample Likelihood* $P(\mathcal{D}|S_m, \boldsymbol{\theta_m})$, see also (3.13):

$$\boldsymbol{\theta^*_m} = \underset{\boldsymbol{\theta_m}}{argmax}\, P(\,\mathcal{D} \mid S_m,\, \boldsymbol{\theta_m}\,) = \underset{\boldsymbol{\theta_m}}{argmax} \prod_k P(\,\boldsymbol{x^k} \mid S_m,\, \boldsymbol{\theta_m}\,) \tag{4.11}$$

where it is assumed that given the 'true' model (S_m and $\boldsymbol{\theta_m}$), the cases are independently and identically distributed. Another important assumption is that the data in $\mathcal{D}$ are complete, i.e., in each data-entry a value is given for each variable.

The case probabilities can be found from the conditional probability distributions required by the network structure S_m, that are specified by the $\boldsymbol{\theta_m}$:

$$P(\,\boldsymbol{x_k} \mid S_m,\, \boldsymbol{\theta_m}\,) = \prod_i P(\,x_i^k \mid \boldsymbol{pa_i^k},\, \boldsymbol{\theta_{m,i}}\,) \tag{4.12}$$

The maximum likelihood optimisation can now be decomposed into individually optimising the parameter sets $\boldsymbol{\theta_{m,i}}$, representing the conditional probability distributions, associated with individual variables X_i :

$$\boldsymbol{\theta^*_{m,i}} = \underset{\boldsymbol{\theta_{m,i}}}{argmax} \prod_k P(\,x_i^k \mid \boldsymbol{pa_i^k},\, \boldsymbol{\theta_{m,i}}\,) \tag{4.13}$$

If X_i is a binary RV (e.g., either $X_i = true$ or $X_i = false$), then we are dealing with a *binomially distributed* sample likelihood. If $\mathcal{D}_i$ denotes that part of the data concerning only the values of X_i, while p_i and n_i denote the number of occurrences of X_i being $true$ and $false$ respectively, we have:

$$P(\mathcal{D}_i|\theta_{m,i}) = \theta_{m,i}^{p_i} \cdot (1 - \theta_{m,i})^{n_i} \tag{4.14}$$

In this case, the maximum likelihood is given by the *observed frequency*:

$$\theta_{m,i}^{*} = \frac{p_i}{p_i + n_i} \tag{4.15}$$

For each instantiation of the parent variables of X_i, this optimisation can be performed, thus obtaining the conditional probability distribution of X_i, given it parents $pa(X_i)$. For each configuration of the parents $pa(x_i)$, p_i (respectively, n_i) denotes the number of occurrences of $X_i = true$ (respectively, $false$), *in the samples consistent with* $pa(x_i)$.

The Maximum Likelihood approach however is prone to sparse data. It may well be the case that many of the possible configurations of the variables have not been observed in the data, and this leads to 'divide-by-zero' problems in the estimation. In case of sparse data, one could use Dirichlet-priors that can be used in Maximum A-Posteriori (MAP) estimation.

Maximum A-Posteriori Estimation (MAP)

Learning the conditional probability distributions for a given Bayesian network structure by Maximum A-Posteriori Estimation (see also (3.12)) will be outlined in two steps. First the case in which there is only one random variable in the model is discussed, resulting in an estimation of the distribution of that variable specified in (4.24). Then the case of more variables is discussed, resulting in an estimation of the conditional distribution of a variable given its parent variables specified in (4.30).

One variable First, we assume that our network consists of only one RV and that it is two-valued: say $X = t$ or $X = f$. Suppose furthermore that we have empirical data in the form of a number of instantiations of X (called *cases*), $\mathcal{D} = \{x_1, \ldots, x_N\}$ ($\mathcal{D}$ is called *random sample*). We would now like to derive a probability distribution over X that accurately describes the data in our random sample. This can be used as an estimation of the unknown physical probability distribution over X which we will represent by what is called the *uncertain variable* Θ. The estimation of $P(X = t|\mathcal{D}, \xi)$ is based on our uncertainty on this physical distribution, expressed by $P(\theta|\xi)$. This uncertainty can be updated with our data using Bayes' Theorem:

$$P(\theta|\mathcal{D}, \xi) = \frac{P(\mathcal{D}|\theta, \xi) \cdot P(\theta|\xi)}{P(\mathcal{D}|\xi)} \tag{4.16}$$

$P(\mathcal{D}|\theta,\xi)$ is the *likelihood function* of *binomial sampling*:

$$P(\mathcal{D}|\theta,\xi) = \theta^{N_t} \cdot (1-\theta)^{N_f} \ , \tag{4.17}$$

where N_t resp. N_f is the number of occurrences of $X = t$ resp. $X = f$ in the data $\mathcal{D}$.

For the prior $P(\theta|\xi)$, we can take the *Beta-distribution* $Beta(\theta|\alpha,\beta)$:

$$P(\theta|\xi) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\cdot\Gamma(\beta)} \cdot \theta^{\alpha-1} \cdot (1-\theta)^{\beta-1} \ , \tag{4.18}$$

where $\Gamma(.)$ is the *Gamma-function* which satisfies $\Gamma(x+1) = x\cdot\Gamma(x)$ and $\Gamma(1) = 1$ and $\alpha,\beta > 0$.

The probability of $X = t$, given data $\mathcal{D}$ and any background knowledge ξ, can now be found by averaging over all possible values θ:

$$\begin{aligned} P(X{=}\,t|\mathcal{D},\xi) &= \int P(X{=}\,t|\theta,\xi)\cdot P(\theta|\mathcal{D},\xi)\ d\theta \\ &= \int \theta \cdot P(\theta|\mathcal{D},\xi)\ d\theta \\ &\equiv E_{P(\theta|\mathcal{D},\xi)}(\theta) \end{aligned} \tag{4.19}$$

Applying (4.16), (4.17) and (4.18) and using the properties of the Beta-distribution, we finally get:

$$P(X{=}\,t|\mathcal{D},\xi) = \frac{\alpha + N_t}{\alpha+\beta+N_t+N_f} \tag{4.20}$$

Now suppose that X is multi-valued and takes values from $\{x^1,\ldots,x^r\}$. Then we try to estimate the physical distribution $\boldsymbol{\theta} = (\theta_1,\ldots,\theta_r)$.

$$P(X{=}\,x^k|\mathcal{D},\xi) = \int P(X{=}\,x^k|\boldsymbol{\theta},\xi)\cdot P(\boldsymbol{\theta}|\mathcal{D},\xi)\ d\boldsymbol{\theta} = \int \theta_k \cdot P(\boldsymbol{\theta}|\mathcal{D},\xi)\ d\boldsymbol{\theta} \tag{4.21}$$

Generalising from the two-valued case to multiple values we get *multinomial sampling*:

$$P(\mathcal{D}|\boldsymbol{\theta},\xi) = \prod_k \theta_k^{N_k} \ , \tag{4.22}$$

where N_k is the number of occurrences of $X = x^k$.

For the prior $P(\boldsymbol{\theta}|\xi)$ we may now choose in stead of the Beta-distribution, the *Dirichlet-distribution* $Dir(\boldsymbol{\theta}|\alpha_1,\ldots,\alpha_r)$:

$$P(\boldsymbol{\theta}|\xi) = \frac{\Gamma(\alpha)}{\prod_k \Gamma(\alpha_k)} \prod_k \theta_k^{\alpha_k - 1} \,, \tag{4.23}$$

where $\alpha = \sum_k \alpha_k$.

Analogously to the two-valued, single parameter case, the probability distribution for the next case is given by:

$$P(X = x^k|\mathcal{D}, \xi) = \frac{\alpha_k + N_k}{\alpha + N} \tag{4.24}$$

where $\alpha = \sum_k \alpha_k$ and $N = \sum_k N_k$.

Two or More Variables Let $\boldsymbol{X} = \{X_1, \ldots, X_n\}$ be a set of variables and S^h a hypothetical fixed structure of a Bayesian network with nodes representing these variables. Further, let $\boldsymbol{\Theta^h}$ be a the uncertain variable representing parameters $\boldsymbol{\theta^h} = (\boldsymbol{\theta_1}, \ldots, \boldsymbol{\theta_n})$ containing the parameter sets specifying the 'true' conditional probability distributions specifying the model, given the network structure S^h.

Furthermore, we have a *random sample* $\mathcal{D} = \{\boldsymbol{x^1}, \ldots, \boldsymbol{x^N}\}$, where each $\boldsymbol{x^k}$ is an instantiation of the variables in $\boldsymbol{X}$. Given these data, we may derive the values in $\boldsymbol{\theta^h}$ that accurately explain these data and thus provide the completion of a possible Bayesian network model.

Using the factorisation indicated by S^h, we can write for the physical joint probability distribution of $\boldsymbol{X}$:

$$P(\boldsymbol{x}|\boldsymbol{\theta^h}, S^h) = \prod_i P(x_i|\boldsymbol{pa_i}, \boldsymbol{\theta_i}, S^h) \tag{4.25}$$

Each factor in (4.25) can be seen as a collection of multinomial distributions: one for each configuration $\boldsymbol{pa_i^j}$ of the parents of X_i. So, we assume:

$$P(x_i^k|\boldsymbol{pa_i^j}, \boldsymbol{\theta_i}, S^h) = \theta_{ijk} > 0 \tag{4.26}$$

If we make the assumptions of the random sample $\mathcal{D}$ having no missing data and the parameter vectors $\boldsymbol{\theta_{ij}}$ being mutually independent (which remain independent, given a random sample), we get:

$$P(\boldsymbol{\theta^h}|\mathcal{D}, S^h) = \prod_i \prod_j P(\boldsymbol{\theta_{ij}}|\mathcal{D}, S^h) \,, \tag{4.27}$$

Thus, we can update each $\boldsymbol{\theta_{ij}}$ independently like in the single variable case:

$$P(\boldsymbol{\theta_{ij}}|\mathcal{D}, S^h) = \frac{P(\mathcal{D}|\boldsymbol{\theta_{ij}}, S^h) \cdot P(\boldsymbol{\theta_{ij}}|S^h)}{P(\mathcal{D}|S^h)} \tag{4.28}$$

Again, we take Dirichlet-distributions for the prior distributions of the $\boldsymbol{\theta_{ij}}$:

$$P(\boldsymbol{\theta_{ij}}|S^h) = Dir(\boldsymbol{\theta_{ij}}|\alpha_{ij1}, \ldots, \alpha_{ijr_i}) \, , \tag{4.29}$$

where r_i is the number of values that X_i may take.

Using (4.24), all this results in the following conditional probability distributions for new cases:

$$P(X_i = x_i^k|\mathcal{D}, S^h) = \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}} \, , \tag{4.30}$$

where $\alpha_{ij} = \sum_k \alpha_{ijk}$, $N_{ij} = \sum_k N_{ijk}$ and N_{ijk} is the number of cases where $X_i = x_i^k$ and $\boldsymbol{Pa_i} = \boldsymbol{pa_i^j}$.

The assumption that MAP relies on, is that of full observability: all data-entries yield an instantiation of all variables in the model. If some of the variables are hidden, i.e., we do not know the value in each data-entry, then the method of Expectation Maximisation may be used.

Expectation Maximisation (EM)

In case the available data is *incomplete*, i.e., there are variables for which we do not have a value for all cases in the data, the expected values of these variables can be used instead. These can be computed using an inference algorithm. The details will not be discussed here, partly because the datasets used in our experiments (Chapter 6) were not incomplete.

4.5.3 Dynamic Bayesian Networks

A normal Bayesian network represents the state of belief of an agent with respect to its environment. This belief state may change when new information about the environment becomes available. However, this does not necessarily mean that the environment *itself* has changed.[2] In order to explicitly take into account the agent's environment being subject to *change*, an extension of of the regular Bayesian network model has been introduced, known as *Dynamic Bayesian networks* (DBNs).

In a DBN, the state of (the model of) the environment at different time-points has been made explicit by means of variables carrying a time-index:

[2] If one wrongly draws such a conclusion, one has become the victim of what Jaynes called the 'Mind Projection Fallacy' (Jaynes, 1990) in which epistemic judgements are confused with ontological judgements and projected *as such* onto reality.

in stead of variables X characterising a static environment, variables X_t for different time-indices t are introduced, referring to the state of the environment at different time-points. Given some event or action that may cause the environment to change, the time-periods before and after the event or action (referred to as *time-slices*) are indicated by means of subsequent (discrete) time-indices. Now the network consists of two parts: the *Observation* or 'intra-slice' model and the *State Evolution* or 'inter-slice' model. In the Observation model $\mathbf{P}(\mathbf{O}_t|\mathbf{S}_t)$, the correlations between hypothesis and observation variables referring to the environment during a single time-slice are specified; the State Evolution model $\mathbf{P}(\mathbf{S}_t|\mathbf{S}_{t-1})$, specifies correlations between aspects of the environment at different time-points, reflecting the way in which the state of the environment evolves in time.

Of course, distinguishing variables at different time-points in a Bayesian network does not make them essentially different. Some of the variables just have a temporal aspect in their definition, which is not formally disallowed in the definition of regular Bayesian networks (Definition 4.1). However, what makes DBNs essentially different, is in the processing of observations at different time-points (the 'monitoring' process). This process consists of 3 steps: *prediction*, *roll-up*, and *estimation*.

1. *Prediction*: Suppose our agent has some belief with respect to the current state of the environment, say $Bel(\mathbf{S}_{t-1})$. Here, $\mathbf{S}_{t-1}$ denotes a vector of random variables, each of which additionally carry one shared time parameter value. So each instantiation of these variables represents a state at time $t-1$, but the agent's belief w.r.t. that state is a probability distribution over the possible states at that time. Then, using the State Evolution model, the agent predicts the state of the environment in the next time-slice, by computing the posterior distribution over $\mathbf{S}_t$ as follows:

$$\widehat{Bel}(\mathbf{S}_t) = \sum_{\mathbf{s}_{t-1}} \mathbf{P}(\mathbf{S}_t|\mathbf{S}_{t-1} = \mathbf{s}_{t-1}) \cdot Bel(\mathbf{S}_{t-1} = \mathbf{s}_{t-1}) \qquad (4.31)$$

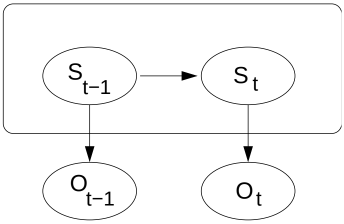


Figure 4.5: Prediction.

2. *Roll-up*: Now the time-slice $t-1$ is *removed*, and for for the state variables at time t, $\widehat{Bel}(\mathbf{S}_t)$ is inserted as the new prior.

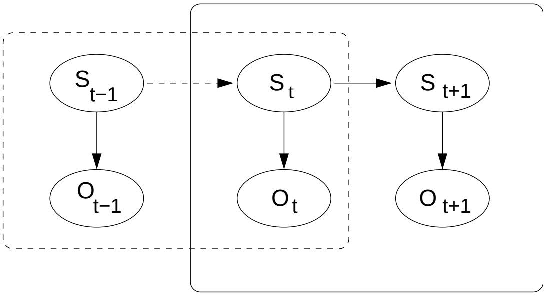


Figure 4.6: Roll-up.

3. *Estimation*: In the final step of the processing cycle, the agent processes new information about the environment obtained by observation. Based on the new evidence, the belief concerning $\mathbf{S}_t$ is updated:

$$Bel(\mathbf{S}_t) = \alpha \cdot \mathbf{P}(\mathbf{O}_t|\mathbf{S}_t) \cdot \widehat{Bel}(\mathbf{S}_t) \tag{4.32}$$

where α is a normalisation constant.

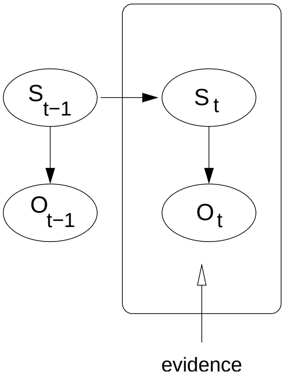


Figure 4.7: Estimation.

4.6 Software Tools

When developing Bayesian networks, tools are needed to easily construct, edit and test networks. There are many tools around, varying in aspects such as functionality (support for learning and inference algorithms, graphical interface for presenting and editing networks, export possibilities for re-using networks in other software) and availability (free/share-ware, open-source, programming language). In the following subsections we briefly describe some of the tools that are available and that have played a role during the course of our research.

4.6.1 Hugin

With Hugin (Andersen et al., 1989), Bayesian networks and their extension Influence Diagrams (networks to create decision support directly through *action nodes*) can be created and edited in ′edit-mode′. Both discrete nodes and to some extent continuous nodes (with a Gaussian distribution) can be defined in the models. In ′run-mode′, both belief updating (′Prop Sum Normal′) and revision (′Prop Max Normal′) can be performed. In Hugin, a junction tree algorithm is used for inference (see Section 4.3).

The Hugin System can be used through Hugin Runtime, an easy-to-use graphical environment. Also the Hugin API (Application Program Interface) can be used, which comes as a library for C (or C++). With the API, models can be constructed as components in an application (mostly) in the area of decision support and expert systems.

4.6.2 JavaBayes

The JavaBayes system (Cozman, 1998) is the first full implementation of Bayesian networks in the Java programming language. It is is composed of a graphical editor to create and modify the networks, a core inference engine, and a parser.

The parser allows you to import Bayesian networks in a variety of formats. The engine is responsible for manipulating the data structures that represent Bayesian networks. The engine performs both Bayesian updating and belief revision and can produce the expectations for univariate functions (for example, the expected value of a variable).

The tool supports two different inference algorithms: variable elimination (Zhang and Poole, 1996) and bucket tree elimination (Dechter, 1996): see Section 4.3.

4.6.3 BayesNet Toolbox

The BayesNet Toolbox is an open-source Matlab package for directed graphical models, created by Kevin Murphy (Murphy, 2001). It supports many kinds of nodes, exact and approximate inference, parameter and structure learning, and static and dynamic models. Some early experiments in our research on using Bayesian networks for dialogue act classification were performed using this toolbox.

4.6.4 Weka

WEKA (Waikato Environment for Knowledge Analysis) is a toolbox with implementations of various machine-learning algorithms, including the Bayesian Network classifier. For the classification experiments described in Chapter 6, we made use of the WEKA API to set up all experiments.

4.7 Applications

The first applications of Bayesian networks were diagnostic systems in the medical field. More generally, Bayesian networks are primarily used in expert systems.

4.7.1 Medical Diagnosis

In examining a patient, symptoms are observed and can be used as evidence in a Bayesian network in order to find explanations for these symptoms, i.e., find probable diagnoses. In the Pathfinder system, diseases of the lymph node can be diagnosed. Bayesian theory is expanded with decision theory (resulting in *influence diagrams*) to decide on additional medical tests to perform, in order to make a more reliable diagnosis.

4.7.2 Map Learning

This is used in robot technology: imagine a robot moving through some space, continuously receiving sensory inputs and using these data as evidence in a Bayesian network in order to build and maintain a map of the space the robot is navigating through.

4.7.3 Language Understanding

Words, for example English words, can be used as evidence to derive the meaning of the text these words constitute. Bayesian theory is applied in this way in story understanding: from the words the author has used to tell a story (e.g., an experience he once had), this story is retrieved to some extent, using probabilistic relationships in a network.

In the Lumière project (Horvitz et al., 1998), Bayesian theory is applied to the help-feature of Microsoft Excel, the 'Office Assistant'. A Bayesian network is used to derive goals and needs of software users from the user's text queries (Heckerman and Horvitz, 1998) and other events as well, during the use of the software.

Chapter 5

Dialogue Systems and Bayesian Networks

In which we discuss the engineering side of dialogue, i.e., the development of dialogue systems, or, dialogue agents. The evolution of approaches to dialogue modelling will be outlined briefly. Additionally, we discuss two state of the art approaches to dialogue modelling that use some form of dialogue state. Then an analysis is given of the way in which the problem of uncertainty is dealt with in dialogue systems, and finally, we will give a sketch of how Bayesian networks can be used within a dialogue system.

5.1 Introduction

One of the earliest programs that was capable of some interaction with human users through natural language is ELIZA, created by Weizenbaum (1966). The system is supposed to simulate a Rogerian psychotherapist that produces responses to user utterances by means of recognising patterns in the utterances and using these patterns to transform them into responses through a set of production rules (user: "my wife doesn't understand me" / system: "how do you feel about your wife not understanding you?").

The SHRDLU system, realised by Winograd (1972), involved more elaborate (syntactic) analysis of user utterances. The system supports simple dialogues about the blocks world, i.e., the user can give commands to the system, such as "pick up a big red block", and the system may respond with "OK" and carry out this action. The state of and actions in the blocks world during the conversation are visualised on the computer screen. One of the main issues of language understanding in SHRDLU is reference resolution. If the user asks the system to "grasp the pyramid", the system tries to resolve which object the user is referring to when speaking of *the* pyramid. If it fails to find a unique object satisfying the user's specification, it will ask for further information ("I don't understand which pyramid you mean").

Much research related to and used for the development of dialogue systems has concentrated on improving the analysis of single natural language utterances in isolation, with the primary emphasis on constructing internal representations based on syntactic and semantic analysis. Another aspect in designing dialogue systems that is receiving more and more attention however, is the idea of taking into account the global structure of a dialogue when analysing an utterance at a certain point in that dialogue. As discussed extensively in the chapter on dialogue acts (particularly, Section 2.2), each utterance is produced under particular circumstances and should be interpreted as such. The same expression may have different interpretations under different circumstances. In a formal setting, one may refer to these circumstances as the *dialogue state*. In the evolution of dialogue system design, more and more use has been made of some explicit notion of dialogue state in interpreting utterances and planning responses. Both of these aspects also involve maintenance, i.e., *updating* the dialogue state according to the course of events.

5.2 State-based Dialogue Modelling

The simplest type of dialogue systems using some notion of dialogue state are finite state-based systems, in which a *finite state automaton* is used to specify all possible courses of events in a dialogue. In each state, the system has a finite number of possible interpretations of the user's utterance, each of them causing a state transition in the automaton and a system reply. In a time-travel information system for example, the user may be asked to state the desired departure city, destination city, and the dates of departure and arrival, in that order (see Figure 5.1). In these finite state models, the system has the initiative during the entire dialogue and the user is very restricted in his actions.

Of course, finite state based systems can be extended with thousands of states. In this way, many variations in the behaviour of users can be supported, thereby giving the user more freedom in his action. However, in case more complex behaviour is to be supported, alternative representations are more efficient. Alternative representations involving stacks, frames or feature structures can support infinitely many states, in contrast to finite state automatons.

Users can be given more freedom of movement by using *templates* containing slots in which pieces of information to be provided by the user – such as departure date and time of a train connection – can be stored by the system during a dialogue. Now, the user may give more information about the desired train connection than prompted for, or may influence the order in which the required pieces of information are given. The actions of the system now depend on the state of the template. Dialogue systems using such templates are also known as *frame-based* systems.

The dialogue states as represented in finite state automatons are 'atomic', i.e., they do not have an internal structure. In more advanced approaches to

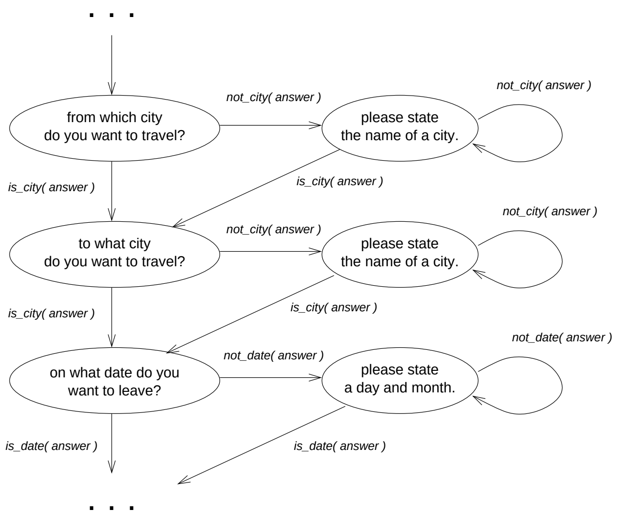


Figure 5.1: A finite-state automaton for time-travel information dialogues.

dialogue modelling, dialogue states are represented by several parameters concerning aspects such as a notion of dialogue phase (opening, closing, negotiation), the mental or cognitive states of the participants (their beliefs, intentions, etc.), the state of the task being performed (concerning issues such as name and address information of the user or preferred departure times in a flight booking system), etcetera. The frame-based systems can be seen as a very simple form of such advanced approaches.

In the following sections, we will discuss two examples of advanced dialogue modelling: the *BDI-agent* approach and the *Information State* approach. For an extensive overview on dialogue modelling techniques and existing dialogue systems, see for example (McTear, 2002).

5.2.1 BDI-agent Dialogue Modelling

In this approach, the dialogue state is a mental state of a *conversational agent*. The agent has a mental model consisting of *Beliefs, Desires and Intentions (BDI)* and is typically represented by means of some logical formalism, such as epistemic predicate logic. In Figure 5.2, a simple scheme for a BDI-agent is given.

- **Beliefs**: the agent's representation of the present state of the world, including beliefs about the beliefs of other agents, e.g., a human user; by *perceiving* its environment, the agent is able to update his beliefs on the basis of new information.

- **Desires**: (positive or negative) attitudes with respect to possible states of the world; for example, a user may desire a state of the world in which he/she possesses tickets for a theatre performance that he/she likes.

- **Intentions**: make up the course of action decided on; the agent plans the *actions* that will lead to a desired state of the world and *commits* to these actions.

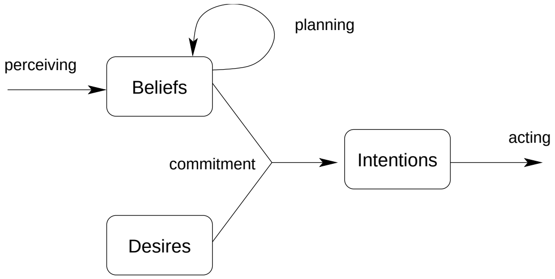


Figure 5.2: BDI-agent (taken from Allen, 1987).

In the BDI-approach, communication is seen as a social, multi-agent activity. The essence of communication lies in intention and recognition of intention. The task of dialogue act recognition in a BDI-based dialogue system is primarily based on the plan-inference model as described in Section 2.4.1.

In the TRAINS project (Allen et al., 1995)[1], an agent-based dialogue system is developed, that assists users in planning the transport of cargo within a complex logistic system. In Figure 5.3, an strongly extended version of the general agent model in Figure 5.2 is given, in which aspects involving natural language communication have been incorporated, such as *shared beliefs*, *discourse obligations*, and *speech acts*.

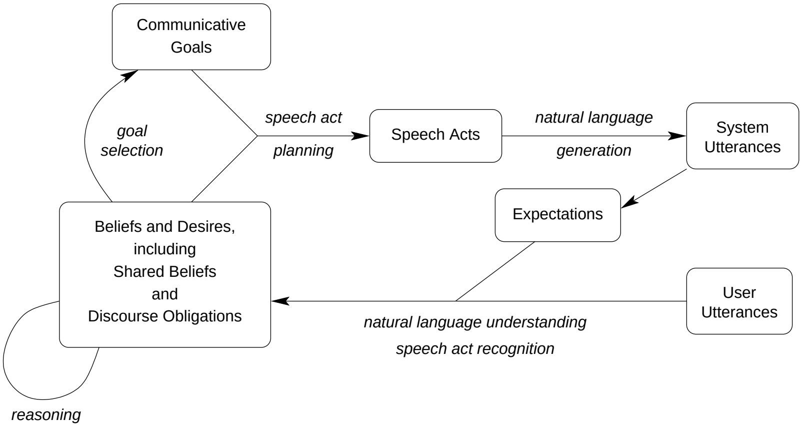


Figure 5.3: TRAINS as a conversational agent (taken from Allen et al., 1995).

[1] Its successor project is TRIPS: The Rochester Interactive Planning System (Ferguson and Allen, 1998).

5.2.2 Information State Dialogue Modelling

In this approach, the dialogue state is not primarily a mental state, but rather a more neutral state of information, as seen from the perspective of a third party not involved in the dialogue. This *information state* can be represented by feature structures, lists, sets, etcetera, and it may cover various relevant aspects such as participants, beliefs, common ground, or intentions. In GoDiS (the Gothenburg Dialogue System) for example, feature structures are used to represent the information state of a participant – so in this specific case the information state may be seen as a mental model of a dialogue agent. These feature structures consist of two parts: one part for what is *private* to the agent and one part for what is assumed to be *shared* between the dialogue participants. In Figure 5.4, an example feature structure is given; for further details we refer to (Larsson and Traum, 2000; Trindi, 2001).

$$
\text{is} : \left[\begin{array}{ll}
\text{private} : & \left[\begin{array}{l}
\text{plan} : \text{StackSet(Action)} \\
\text{agenda} : \text{Stack(Action)} \\
\text{bel} : \text{Set(Prop)} \\
\text{tmp} : \left[\begin{array}{l}
\text{bel} : \text{Set(Prop)} \\
\text{qud} : \text{Stack(Question)} \\
\text{lu} : \left[\begin{array}{l}
\text{speaker} : \text{Participant} \\
\text{moves} : \text{assocSet(Move,Bool)}
\end{array}\right]
\end{array}\right]
\end{array}\right] \\
\text{shared} : & \left[\begin{array}{l}
\text{bel} : \text{Set(Prop)} \\
\text{qud} : \text{StackSet(Question)} \\
\text{lu} : \left[\begin{array}{l}
\text{speaker} : \text{Participant} \\
\text{moves} : \text{assocSet(Move,Bool)}
\end{array}\right]
\end{array}\right]
\end{array}\right]
$$

Figure 5.4: An information state in GoDiS.

Another important component in any system taking the information state approach is the notion of *dialogue moves*, representing the actions underlying the utterances of the participants. Using *update rules* it can be specified how the information state is updated when a dialogue move is performed, in terms of applicability conditions and effects. The TrindiKit is a toolkit for building and experimenting with dialogue systems based on a *dialogue move engine* and *information states*. The GoDiS system was developed using this toolkit. In this system, dialogue moves that are performed by a user are identified basically from lexical and semantic information extracted from utterances using a grammar. So, what we referred to as dialogue act recognition seems to be done here in a rule-based manner, without using any contextual aspects.

5.3 Uncertainty

In most approaches to dialogue modelling used in the design of existing dialogue systems, the problem of uncertainty is dealt with in two senses: 1) using probabilistic models on the low level of speech recognition, and 2) using dialogue strategies of verification on the highest level of interpretation.

For the process of speech recognition, generally a probabilistic approach is chosen, hence accounting for the noisy character of the speech signal. The idea is to find the most likely word sequence w^*, given the speech signal in the form of acoustic features a, through an *acoustic model* $P(a|w)$ and a *language model* $P(w)$, which have been trained separately from empirical data, see (5.1). Alternatively, a ranked list of the most likely potential word sequences is generated that can be disambiguated at higher levels of analysis.

$$w^* = \underset{w}{argmax}\, P(w|a) = \underset{w}{argmax}\, P(a|w) \cdot P(w) \tag{5.1}$$

At the higher levels of interpretation however, the approaches are mostly deterministic. Anaphora resolution for example, is mostly modelled in a rule-based fashion, implementing heuristics about the salience of potential referents. Dialogue act types (or any variant of this concept) are usually derived from syntactic analysis only. The dialogue manager of a system is also modelled deterministically, and uncertainty at the top-level is dealt with through the dialogue strategies used. The system can deal with the possibility that its initial interpretations are wrong using verification subdialogues that provide feedback, and employing some notion of dialogue state that can be updated and/or revised during the dialogue.

5.4 Bayesian Networks

In Figure 5.5, a simple blackboard architecture of a dialogue agent is given. The dialogue manager sends syntactic and semantic information extracted from a user utterance to a dialogue act recognition component, that returns a single dialogue act type as the most likely one, or several likely candidate dialogue act types with associated probabilities. The dialogue act recogniser could be any type of classifier, for example a Bayesian network.

However, Bayesian networks may also be used in dialogue systems for other tasks than dialogue act classification. Another important aspect in the interpretation of user utterances for example, is anaphora resolution. Lemon et al. (2002) present some initial work on using Bayesian networks for that purpose, where the construction of the models is completely based on expert knowledge on anaphora resolution used in an existing dialogue system.

There are several factors involved in determining the most likely referent of an anaphor in a user utterance, such as the recency of potential referents and the intra-sentential location of the relevant noun phrases in the preceding dialogue. In multi-modal dialogues, there may be additional information, concerning the non-verbal behaviour of the user, in the form of gesturing for example, that may influence the process of anaphora resolution. The more

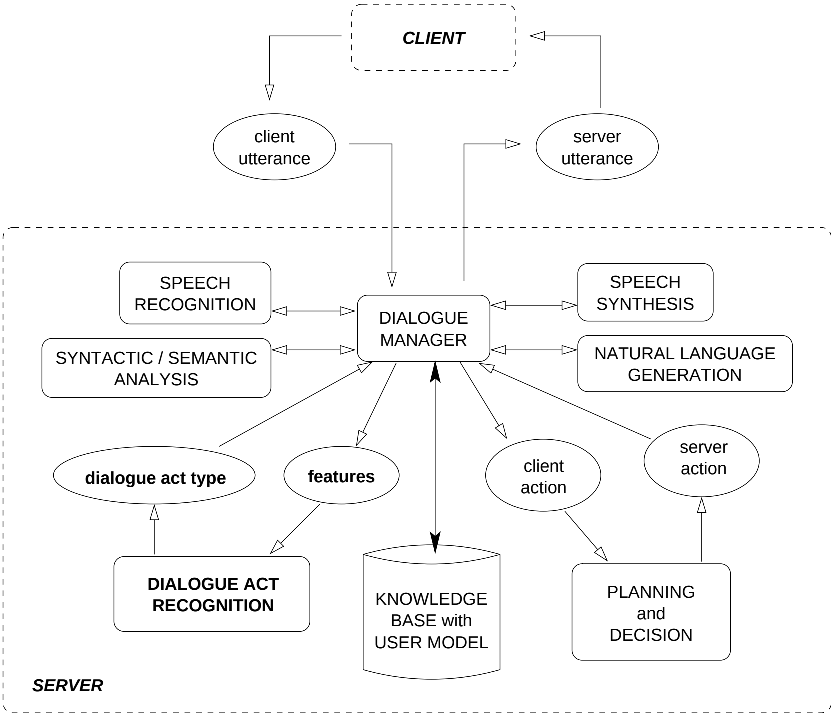


Figure 5.5: Dialogue System Architecture with Dialogue Act Recognition component.

complex the settings get, the more need there will be for non-deterministic approaches like probabilistic models, that can account for exceptions to the rules implemented that are bound to occur.

Bayesian networks may be very suitable to be used in a more general framework for interpretation of utterances in a dialogue under incomplete information. Different tasks, such as speech recognition, dialogue act recognition, anaphora resolution, and topic identification can be integrated into one probabilistic model. Results from speech recognition are used in the process of dialogue act recognition, but it has been shown that vice versa, dialogue act information can also be of use for improving speech recognition results (Stolcke et al., 2000). The model may represent a diversity of aspects, involving the beliefs and desires of the conversational agent, the state of the dialogue, the state of the task at hand, a user model, and so on. One of the advantages of Bayesian networks is that it is not a functional model with fixed input and output attributes. A Bayesian network incorporates all attributes (in the form of random variables) regardless of them being input (observed) or output (unobserved) attributes; evidence on any subset of the variables can be entered and the most likely configuration of the unobserved variables can be computed, whether they concern the recognised words in an utterance, a dialogue act type or the referent of an anaphor. This approach can be seen as a form of agent-based dialogue modelling: the Bayesian network is a (mental) belief model of the conversational agent.

In Figure 5.6, a prototypical, schematic, dynamic Bayesian network is given, in which a broad range of aspects in the process of dialogue act modelling can be incorporated. Note that each of the nodes represents a sub-network of the Bayesian network: the node *Dialogue_Act* $_t$ may involve both dialogue act type(s) and elements of the semantic content (including referents of anaphora); the nodes *Dialogue_State* $_t$ and *Dialogue_State* $_{t+1}$ may involve, for the states before and after the user's utterance, elements of the dialogue context, such as preceding dialogue act types; in the same way, the nodes *User_Model* $_t$ and *User_Model* $_{t+1}$ involve beliefs and preferences of the user; the node *User_Utterance* $_t$ may involve features of the utterance or a representation of the potential word sequences resulting from speech recognition.

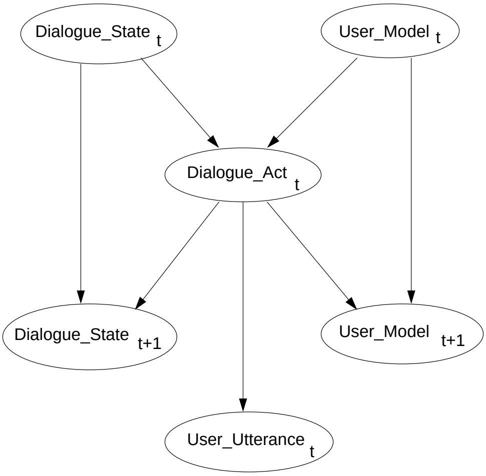


Figure 5.6: A schematic dynamic Bayesian network for dialogue act modelling.

Although all these different aspects are integrated in one Bayesian network, which may eventually lead to a very complex and even intractable model, the Bayesian network formalism supports *modularity* through the possibility of specifying conditional independencies. In this case for example, the dialogue state before the user utterance is made (*Dialogue_State* $_t$) is conditionally independent of the features of that user utterance (*User_Utterance* $_t$), given the complete dialogue act performed in that utterance (*Dialogue_Act* $_t$).

A final remark concerns the updating of the dialogue state (or information state update) on the basis of a dialogue act (or dialogue move). Using separate parts in the Bayesian network for the dialogue state before and after the performance of the dialogue act – hence the term *dynamic* – the updating process can be represented. The new dialogue state $\mathit{Dialogue\_State}_{t+1}$ is dependent on the current $\mathit{Dialogue\_State}_{t}$ and the performed $\mathit{Dialogue\_Act}_{t}$.

In the next chapter, Chapter 6, experiments concerning the assessment of different dialogue act classifiers from data in a dialogue corpus are discussed. The Bayesian networks involved in these experiments only involve a set of utterance features as a representation of user utterances ($\mathit{User\_Utterance}_{t}$) and dialogues from preceding utterances as a representation of the dialogue state ($\mathit{Dialogue\_State}_{t}$). No aspects concerning beliefs or preferences of the user are involved. The Bayesian networks neither are dynamic.

Chapter 6

Dialogue Act Classification

In which we describe the experiments that were performed concerning dialogue act classification with Bayesian networks. This includes an outline of the annotation of the SCHISMA *dialogue corpus, from which data are extracted for the machine learning of the classifiers. The experiments were aimed at both finding relevant features for classification and comparing Bayesian network classifiers with other classifier types.*

6.1 Introduction

In Chapter 2 we discussed the notion of dialogue acts, that can serve as a high-level characteristic of natural language utterances within the context of a dialogue. In the design of a dialogue agent, the task of recognising the dialogue act that was performed by a user when producing an utterance plays a central role. In Section 2.4, two approaches to dialogue act recognition were discussed, the plan-based and the cue-based approach. In the cue-based approach, the dialogue act type of an utterance is based on a set of cues or features. This means that given the representation of an utterance by means of a set of feature-value pairs, one of a set of possible dialogue act types is to be chosen as the most likely one.

As we have seen that a complete and exact mapping between features and dialogue act types can never be found, many research efforts are based on machine learning techniques, in which models are induced from the data in an annotated corpus (e.g., Nagata and Morimoto, 1994; Samuel et al., 1998; Kipp, 1998; Wright, 1998; Wright et al., 1999; Stolcke et al., 2000; Black et al., 2003). In the formulation of the task of dialogue act recognition as a classification task, the models are referred to as classifiers. In Section 3.7, we have discussed various classifier types, such as Decision Trees and Naive Bayes. In Section 4.3.1, Bayesian networks were shown to be interpretable and used as classifiers as well.

In this chapter the experiments on using Bayesian networks for dialogue act recognition and their results are presented. The combination of Bayesian networks and natural language dialogue is fairly new, although Pulman (1996) presented some preliminary experiments on using Bayesian networks for dialogue move recognition. Our first explorations on using Bayesian networks for dialogue act recognition and some preliminary experiments have been presented in (Keizer, 2001a,b; Keizer et al., 2002).

In the experiments described here, different classifiers were constructed and evaluated, using data from an annotated Wizard of Oz corpus of Dutch keyboard-typed dialogues . We compared results from using different feature sets and different classifiers. The features used for the classification task include both superficial utterance features that were derived from the output of a part-of-speech tagger, and context features in the form of dialogue act types of preceding utterances. The present experiments were primarily aimed at analysing the significance of the features in determining the most plausible dialogue act type of a given utterance.

6.2 Annotation of the corpus

6.2.1 The SCHISMA Corpus

The SCHISMA corpus is a collection of 64 Dutch dialogues, obtained through a series of Wizard of Oz experiments. The interaction between the wizard (a human simulating the system to be developed) and the human user was established through keyboard-entered utterances, so the dialogues are textual. Furthermore, the dialogues are mixed-initiative: the initiative may switch between the participants within a single dialogue. The task domain concerns information exchange and transaction: users are enabled to make inquiries about theatre performances scheduled and if desired, make ticket reservations. The user has the initiative when making inquiries about performances, but if he makes a request for ticket reservation, the system takes over the initiative in order to get information from the user that is required to make that reservation. To get an idea of the dialogues in the corpus, a small fragment of a dialogue from the corpus is given:

C : *Wanneer draait Sweeney Todd?* (Eng.: When is Sweeny Todd on?)

S : *U kunt "Sweeney Todd" zien in de Grote Zaal in de periode 28 december 1993 t/m 30 december 1993.* (Eng: You can see Sweeny Todd in the 'Grote Zaal' in the period December 28 to 30 1993.)

C : *En Under a blue roof?* (Eng.: What about Under a blue roof?)

S : *U kunt "Under a blue Roof" zien in de Grote Zaal op 19 mei.* (Eng.: You can see 'Under a blue Roof' in the 'Grote Zaal' on May 19.)

C : *Kan ik daar kaarten voor reserveren* (Eng.: Can I order tickets for that)

S : *Heeft u een reductiekaart?* (Eng.: Do you have a reduction card?)

S : *Hoeveel kaartjes wilt U?* (Eng.: How many tickets do you require?)

C : *Ik heb geen reductiekaart.* (Eng.: I don't have a reduction card.)

C : *Vier kaartjes graag.* (Eng.: Four tickets please.)

6.2.2 Dialogue Act Taxonomy

The annotation scheme we have used is based on DAMSL (Dialogue Act Markup on Several Layers, see Section 2.5.4), a standard for annotating task-oriented dialogues (Allen and Core, 1997) with dialogue act types. In this scheme, a number of layers have been defined, each of which cover a different aspect of a communicative action that can be performed during a dialogue. In addition to the original layers COMMUNICATIVE STATUS, INFORMATION LEVEL, FORWARD-LOOKING FUNCTION and BACKWARD-LOOKING FUNCTION of the DAMSL scheme, an additional layer TOPIC MANAGEMENT has been added, containing acts that deal with what the dialogue is currently about, for example, elaborating on the current topic or introducing a new topic.

In the annotation, the tags on the layers described above are assigned to parts of the dialogue text, called *segments*. Each dialogue in the corpus is first subdivided into *turns*, which may consist of one or more *utterances*. Because in an utterance more than one dialogue act may be performed subsequently, each utterance is subdivided into one or more segments. Segments are then the basic units for dialogue act labelling. Annotations of the corpus have done on the basis of general descriptions of the dialogue acts, accompanied by example utterances, collected in a manual. A more strict guidance of the annotation process may involve the use of decision trees, as is the case for the DAMSL annotation manual. Using such decision trees, annotators may assign a dialogue act type to a segment step-by-step, via a number of questions about that segment. For annotating the SCHISMA corpus, the current manual appears to be sufficient for reliable annotations.[1].

In the following subsections, the dialogue act at the various layers are described, insofar they were not already present in the DAMSL scheme. We also give typical examples of utterances from the SCHISMA corpus.

Communicative Status

This layer is has not be modified, nor has it been removed from the scheme. As the dialogues in the consist of keyboard-typed utterances, the acts on this layer are irrelevant (except maybe `uninterpretable`).

1. **uninterpretable**

[1] However, only one annotator annotated the entire corpus and therefore, no analyses such as assessing 'inter-annotator agreement' could be performed (see e.g. Carletta, 1996)

2. **abandoned**
3. **self-talk**

Information Level

No changes have been made to the original INFORMATION LEVEL. Although most utterances in the corpus perform `task` acts, some typical examples of `task-management` and `communication-management` are given.

1. **task**
2. **task-management**: "Kan ik daar ook kaartjes voor reserveren?" (Eng.: Is it possible for me to order tickets for that?)
3. **communication-management**: "Hallo" (Hello), "Tot ziens" (Bye), "pardon?".
4. **other-level**

Forward-looking Function

This layer contains dialogue acts that characterise the influence an utterance has on the future dialogue.

1. **statement**
   - 1.1 **assert**: "'Tommie' begint om 20:00u" *(Eng.: Tommie starts at 20:00h)*; also elliptical sentences such as "ja" *(Eng.: yes)*, that can be paraphrased as "ja, ik heb een kortingskaart" *(Eng.: yes, I have a discount card)*.
   - 1.2 **other-statement**
   - 1.3 **reassert**
2. **influencing-addressee-future-action**
   - 2.1 **action-directive**: "ik wil twee kaartjes" *(Eng.: two tickets please)*; "doe maar 2 mei" *(Eng.: make it the 2nd of May)*.
   - 2.2 **open-option**: "vanavond kunt u naar Herman Finkers in de Grote Zaal" *(Eng.: this evening you can see Herman Finkers performing in the Grote Zaal)*.
3. **info-request**: this act has been further specified by means of two additional acts:
   - 3.1 **query-if**: the speaker asks the hearer whether something is the case or not. "heeft u een reductiekaart?" *(Eng.: do you have a discount card?)*.
   - 3.2 **query-ref**: the speaker asks the hearer for information in the form of references that satisfy some specification given by the speaker. "welke voorstellingen zijn er vanavond?" *(Eng.: which performances are on this evening?)*.
4. **committing-speaker-future-action**
   - 4.1 **offer**: "zal ik de kaartjes voor u reserveren?" *(Eng.: shall I book the tickets for you?)*.

- 4.2 **commit**: “ik zal de kaartjes voor u reserveren” *(Eng.: I will book the tickets for you).*

5. **conventional**
   - 5.1 **opening**
   - 5.2 **closing**: “bedankt” *(Eng.: thanks)*; “tot ziens” *(Eng.: bye).*
6. **explicit-performative**
7. **exclamation**
8. **other-forward-function**

Backward-looking Function

This layer contains dialogue acts that characterise how utterances can refer back to previous parts of the dialogue.

1. **agreement**
   - 1.1 **accept**:

     context: “zal ik de kaartjes voor u reserveren?” *(Eng.: shall I book the tickets for you?)*

     “ja graag” *(Eng.: yes please)*
   - 1.2 **accept-part**
   - 1.3 **reject**: (using the same context as for `accept`)

     “nee dank je” *(Eng.: no thanks)*
   - 1.4 **reject-part**
   - 1.5 **hold**: the speaker postpones his response to a request, proposal or claim of the hearer.

     context: “ik wilde graag reserveren voor 5 personen” *(Eng.: I would like to make reservations for 5 people)*

     De Grote Zaal heeft de volgende rangen: . . . *(Eng.: The Grote Zaal has the following classes/circles: . . .)*
   - 1.6 **maybe**
2. **understanding**
   - 2.1 **signal-non-understanding**
   - 2.2 **signal-understanding**
     - 2.2.1 **acknowledge**
     - 2.2.2 **completion**
     - 2.2.3 **repeat-rephrase**
   - 2.3 **correct-misspeaking**
3. **answer**: the speaker responds to an `info_request` of the hearer.

3.1 **positive-answer**:

context: "welke voorstellingen zijn er vanavond?" *(Eng.: what performances are on this evening?)*

"Vanavond kunt u naar: ..." *(Eng.: For this evening, the following performances have been scheduled: ...)*

3.1.1 **confirm**: positive response to a `query_if`.

3.2 **no-answer-feedback**: the speaker indicates he cannot give the information requested.

3.3 **negative-answer**:

context: "welke voorstellingen zijn er vanavond?" *(Eng.: what performances are on this evening?)*

"Vanavond zijn er helaas geen voorstellingen" *(Eng.: Unfortunately, no performances have been scheduled for this evening)*

3.3.1 **disconfirm**: negative response to a `query_if`.

Topic Management

This layer is an extension of the original DAMSL scheme. It contains acts concerning ways to control what the dialogue is currently about. In the annotation of the SCHISMA corpus, the topic refers to performances under discussion, either a list of performances or an individual one.

1. **shift**: the speaker changes the topic.

   1.1 **introduce-topic**: a topic-change by introducing a new topic.

   1.2 **refer-former-topic**: a topic-change by referring to a topic that was issued earlier in the dialogue.

2. **elaborate**: the speaker maintains the current dialogue by elaborating on it.

3. **narrow**: the speaker narrows down the current topic; typically, by issuing an individual performance from the list of performances currently under discussion.

6.2.3 XML-encoding

The annotation of the corpus is encoded using XML-documents and a DTD. This DTD very closely defines the hierarchical structure of the dialogue acts in the annotation scheme. The DTD is based on the DTD for the DAMSL annotation scheme as specified in (Mengel et al., 2000, sect. 2.3/app. C). In Appendix A.1 the DTD for the SCHISMA annotation is listed, while Appendix A.2 gives the XML-encoding of the annotation of the dialogue fragment in Section 6.2.1.

The XML-based encoding has the advantage that it is a standard for markup of content and therefore, various tools and APIs that work with XML can be used in guiding the annotation and in exploiting the annotations in statistical analyses and particularly machine learning experiments. For more information on XML, see (W3C, 2003). For work on using XML in corpus annotation, see e.g., (XCES, 2002; Isard, 2001).

6.2.4 General Statistics

The 64 dialogues in the SCHISMA corpus have been manually annotated by one annotator, who used an XML-editor with DTD-support. Being a graduate student in computer science not previously familiar with the notion of dialogue acts, the annotator can be characterised as a semi-expert.

In this section, we present the tags specified in the annotation scheme – including the dialogue act types we have chosen for our classification task – together with their frequencies in the annotated corpus. Also, the separate frequencies for the server and the client are given.

In Table 6.1 the frequencies of the top-level elements are given. One can see that the corpus consists of 64 dialogues, and moreover: there are more utterances than turns – so a number of turns consist of more than one utterance – and there are more segments than utterances – so a number of utterances consist of more than one segment. As one can see from the table, dividing turns and utterances into multiple segments has occurred mostly in the case of server turns. Displaying information about performances and reservations in one turn by the server often consists of more than one utterance. As the basic unit for dialogue act annotation is the `segment`, all subsequent frequency tables – one for each layer – show the same total frequency of 2047.

TAG	CLIENT	SERVER	TOTAL
`dialogue`			64
`turn`	864	860	1724
`utterance`	910	1124	2033
`segment`	920	1127	2047

Table 6.1: Frequencies of top-level tags.

Table 6.2 shows that by far the most frequent information level category is `task`. In the opening and closing phases of a dialogue, the speaker will use more `communication management` acts; the frequency shows that on average less than 2 times per dialogue a segment has been tagged as `communication management`.

TAG	CLIENT	SERVER	TOTAL
`information_level`	920	1127	2047
`task`	857	1013	1870
`task_management`	10	55	65
`communication_management`	51	59	110
`other_level`	2	0	2

Table 6.2: Frequencies of the information_level tags.

Table 6.3 shows that the most frequent `forward-looking-functions` are `statement` and `info-request`, as was to be expected given our task domain of users enquiring the system for information. Transaction is not a required part of the dialogues and if it is, it is not expected to take many utterances compared to the information exchange part. Therefore, influencing-addressee-future-actions are less frequent. Further notice the differences in frequencies between client and server: clients use more info-requests (504 against 203), while the server uses more statements (694 against 122). Some of the acts have been used exclusively (but not by instruction) in segments of a server turn – open-option, committing-speaker-future-action– while others have been used exclusively for segments of a client turn – (conventional)opening, explicit-performative, exclamation. This could be a reason to use different sets of dialogue acts for client and server in our classification model. In the present experiments we did not make that distinction.

TAG	CLIENT	SERVER	TOTAL
forward_looking_function	920	1127	2047
statement	123	694	817
assert	122	694	816
reassert	1	0	1
other_statement	0	0	0
influencing_addressee_future_action	239	122	361
action_directive	239	11	250
open_option	0	111	111
info_request	504	203	707
query_if	70	38	108
query_ref	433	165	598
committing_speaker_future_action	0	70	70
offer	0	66	66
commit	0	4	4
conventional	29	31	60
opening	1	0	1
closing	19	30	49
explicit_performative	2	0	2
exclamation	4	0	4
other_forward_function	19	7	26

Table 6.3: Frequencies of the forward_looking_function tags.

Table 6.4 shows that answers are the most frequent backward-looking functions; these are of course typical reactions to the frequently occurring info-requests. Most of the answers are positive-answers. The very high frequency of no-blf can be explained by the observation that first utterances in a dialogue *by definition* have no backward-looking function. Furthermore, there are many question-answer sub-dialogues, where segment in which the question was asked does not really have a backward-looking function. The high frequency of no-blf does not suggest that the dialogues in the corpus are not very coherent, because in many cases the coherence is maintained more implicitly and not explicitly as required in the annotation scheme for other dialogue acts on the backward-looking function layer. If an answer is given to a question for example, this implicitly means that the utterance in which the question was asked was understood.

TAG	CLIENT	SERVER	TOTAL
backward_looking_function	920	1127	2047
agreement	125	223	348
accept	69	54	123
accept_part	1	0	1
maybe	1	0	1
reject_part	2	1	3
reject	13	7	20
hold	39	161	200
understanding	11	20	31
signal_non_understanding	3	20	23
signal_understanding	7	0	7
correct_misspeaking	1	0	1
answer	195	505	700
positive_answer	162	399	561
negative_answer	30	42	72
no_answer_feedback	3	63	66
correction_feedback	0	1	1
no_blf	589	379	968

Table 6.4: Frequencies of the backward_looking_function tags.

From the `topic-management` frequencies in Table 6.5, one can see that the client is more dominant in changing the topic: `shift`s and `narrow`s are mostly performed by the client, while most topic management acts by the server are `elaborate`s, maintaining the current topic.

TAG	CLIENT	SERVER	TOTAL
topic_management	920	1127	2047
shift	154	14	168
introduce_topic	118	4	122
refer_former_topic	36	10	46
elaborate	595	1025	1620
narrow	122	5	127
no_tm	49	83	132

Table 6.5: Frequencies of the topic_management tags.

6.3 Specification of the Classification Task

Our experiments concern a classification task: given a number of features of a case, select the most likely class value for the case. In our domain of natural language dialogue, we have to find the most likely dialogue act type performed in an utterance in the context of a dialogue, so each case is an utterance-in-context. The experiments are subdivided into classifying the backward-looking function (class variable **blf**) and forward-looking function (class variable **flf**) of a client utterance. The features we use concern both superficial utterance features and contextual features. The contextual features involve dialogue act types of previous utterances. The utterance features are obtained directly from the (textual) utterance or from the output of a Part-Of-Speech (POS) tagger.

6.3.1 Selection of Dialogue Act Types

From the hierarchies of backward- and forward-looking functions, we selected two sets of dialogue act types as class-values for the classification tasks, see Table 6.6. These selections were based on some initial intuition concerning what types a dialogue system should primarily distinguish between for dialogues in the domain of information-exchange and transaction. At the same time, the number of classes was kept limited for technical reasons. Therefore, types with very low frequency in the corpus for both client and server utterances were not selected.

BLF-selection
accept
hold
reject
signal-non-understanding
signal-understanding
positive-answer
negative-answer
no-answer-feedback
no-blf
other

FLF-selection
statement
action-directive
open-option
query-if
query-ref
committing-speaker-future-action
conventional
no-flf
other

Table 6.6: Class values for BLF- and FLF-dialogue act types.

The selection procedure ensured that no two types could be chosen that are engaged in a supertype-subtype relation, ruling out the combination of, for example, `accept` and `agreement.` Furthermore, in each selection an additional class value with label 'other' was introduced, covering the remaining dialogue act types from the original hierarchy that were not yet covered by the initially selected dialogue act types. In Figures 6.1 and 6.2, the selections are indicated within the dialogue act hierarchies from which the types have been selected: the initially selected dialogue act types are given in **boldface**, while the dialogue act types covered by the class-value 'other' are given in *italics*.

6.3.2 Utterance Features

Based on our initial intuitions on what features are most informative with respect to identifying the forward- and backward-looking function (flf and blf respectively) of an utterance, restricted by the information the available tagger can give us, we formulated an initial set of utterance features used for the present experiments. Below, the features are given with short descriptions and motivation.

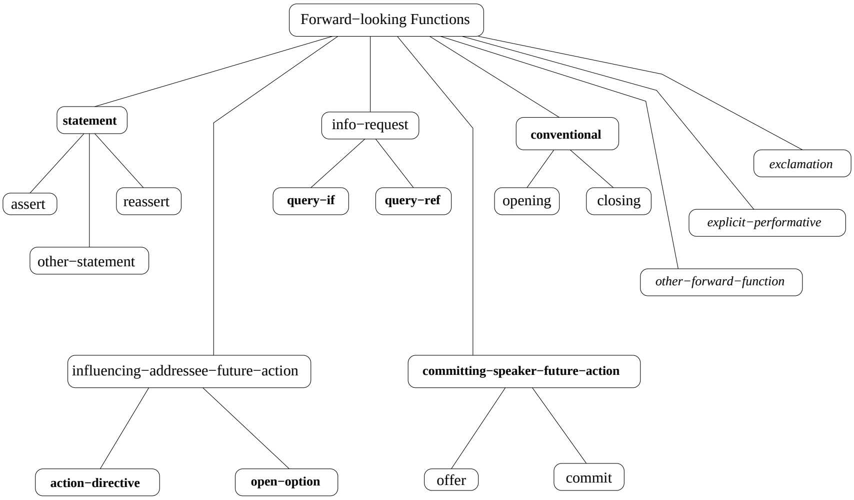


Figure 6.1: Selected class values from the dialogue act hierarchy on the Forward-looking Function Layer.

- **lenq**: qualitative account of the number of words in the utterance; 'one' for 1 word, 'few' for 2, 3 or 4 words, and 'many' for 5 or more words. Can be an indicator for short answers like "yes" or "no" consisting of only one word; for short utterances consisting of 4 words at most, like "nee bedankt", "4 kaartjes graag" or "van wie is Othello?". The boundary of 4 words, separating 'few' from 'many' words is still open: we can experiment with different values.

- **isContinue**: 'true' if the utterance shows a 'continuation pattern', i.e. starts with the (Dutch) word "en" ("and ...", "what about ..."), "toch" ("but still ...") or "maar" ("but ..."). This feature might be a good indicator if we also take into account previous flf of the same speaker; if an utterance shows a continuation pattern, it will more probable that it had the same flf as the previous utterance.

- **startsWithWHExpr**: 'true' if the utterance starts with a 'wh-expression', i.e. either a wh-word ("wie", "wat", "welke", etc.) or a preposition followed by a wh-word ("voor wie", "op wat"). Clearly, this feature is a strong indicator for the syntactic sentence type 'wh-question' and this is expected to be closely related to the flf-type `info-request` and `query-ref` in particular.

- **endsWithQuestionMark**: 'true' if the utterance ends with a question mark ("?"). This feature is particularly expected to separate utterances with flf-type `statement` from those with flf-type `info-request`. Utterances of type `action-directive` can either occur as 'questioning' ("wilt u voor me reserveren?") or not ("ik wil reserveren"). This feature might also be relevant in distinguishing utterances of type `offer` from those of type `commit`, although the present experiments do not include this distinction (both types merge into the more general type `committing-speaker-future-action`).

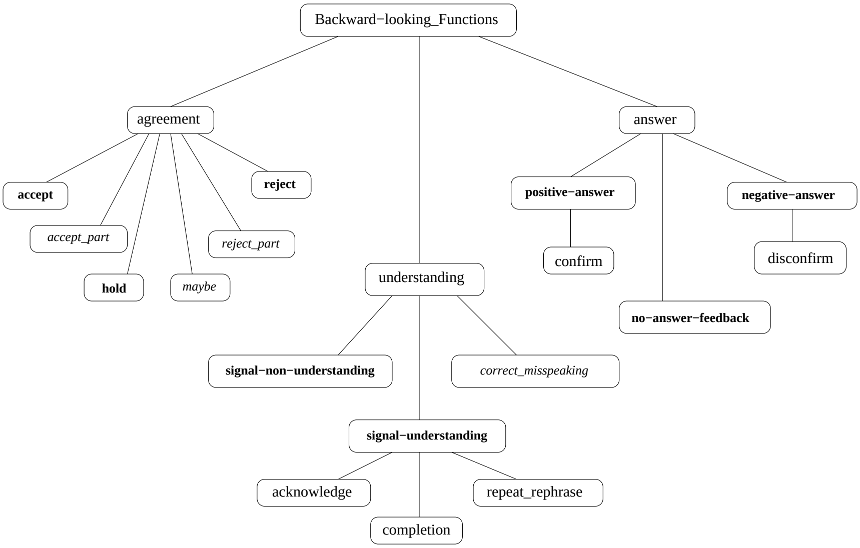


Figure 6.2: Selected class values from the dialogue act hierarchy on the Backward-looking Function Layer.

- **startsWithCanYou**: 'true' if the utterance starts with (Dutch) phrases like "kunt u", "kun je", "zou je", "wilt u". This feature seems a good indicator for identifying requests: `action-directive`, `open-option`, `query-if` and `query-ref`.
- **startsWithCanI**: 'true' if the utterance starts with any of the (Dutch) phrases "kan ik", "mag ik", "moet ik". This feature seems a good indicator for identifying information requests `query-if` and `query-ref`.
- **startsWithIWant**: 'true' if the utterance starts with any of the (Dutch) phrases "ik wil", "ik zou", "ik moet". This feature is also meant for identifying requests.
- **containsPositive**: 'true' if the utterance contains a (Dutch) word like "ja", "jazeker", "inderdaad". ("yes", "yes, indeed", "exactly", etc.) This feature should be informative with respect to identifying positive responses: `positive-answer` and `accept`.
- **containsNegative**: 'true' if the utterance contains a (Dutch) word like "nee", "niet", "geen". ("no", "not", "none", etc.) This feature should be informative with respect to identifying negative responses: `negative-answer` and `reject`.
- **containsOkay**: 'true' if the utterance contains a (Dutch) word like "ok", "akkoord", "ja", "goed". ("ok", "alright", etc.) This feature should also be informative with respect to identifying positive responses, especially the blf's of type `accept`.
- **containsTell**: 'true' if the utterance contains a (Dutch) verb like "vertel", "zeg", "noemen". ("tell", "say", "name") Utterances containing verbs like these – particularly in questions – are more probably information requests than statements for example.

- **containsDo**: ′true′ if the utterance contains a (Dutch) word like "breng", "doe", "brengen". ("take", "bring", etc.) These verbs clearly point into the direction of the utterance being an `action-directive`.

- **containsLocativePrep**: ′true′ if the utterance contains a (Dutch) locative prepositions like "van", "naar", "langs". ("from", "to", "via", etc.)

- **containsLocativeAdverb**: ′true′ if the utterance contains a (Dutch) locative adverb like "hier", "daar", "voor". ("here", "there", etc.)

The last two features are actually ′relics′ from earlier experiments with a very small collection of made-up dialogues with a navigation agent. Clearly, they were chosen with the specific task underlying these navigation dialogues in mind and seem to be of less relevance concerning the SCHISMA dialogues. In the initial model they have been left in the set of utterance features; in a subsequent model with modified utterance features, they have been replaced by their ′temporal′ counterparts, see Section 6.4.2.

6.3.3 Contextual Features

In our classification model, the context is modelled by features of previous parts of the dialogue. In the annotations, the features are attributed to segments, and therefore we considered sequences of segments when preparing the datasets for training. In each dataset, the features consisted of the utterance features corresponding to the current segment, i.e., the segment of which the expression of the speaker (the client) is to be interpreted, and the forward- and backward-looking functions corresponding to preceding segments. The number of preceding segments is measured by the *window-size*, i.e., a window-size of 1 means that we only consider the current segment and do not take into account any contextual features, while a window-size of 2 means that we also take into account the features of the preceding segment. Generic feature names have been used for the contextual features: the forward- and backward-looking functions of the i-th segment preceding the current segment are referred to as **flf-i** and **blf-i** respectively. So, if a window-size of 2 is taken, the contextual features are **flf-1** and **blf-1**. For any contextual features included, one additional value ′null′ had to be introduced to account for instances in which a previous segment did not exist – for example, the first segment in a dialogue does not have a previous segment.

In our initial model, sequences of segments are generated from n-grams of turns. In this way, it could be ensured that the current segment stemmed from a client turn and preceding segments alternatingly stemmed from server and client turns respectively (in Section 6.4.3, a different approach to generating sequences of segments is presented). To deal with the fact that a turn may consist of more than one segment, we chose to generate several sequences of segments from a single n-gram of turns.

6.4 Results

The experiments presented should be seen as preliminary explorations in finding appropriate models for dialogue act classification. This includes, on the one hand, finding relevant utterance features and contextual features, and on the other hand, comparing different types of classifiers (in particular, Bayesian networks) and machine learning techniques. The presented experiments consist of four series, every other series containing some modifications to the model used in the preceding series. In Section 6.3, we presented the initial model; in the second model (Section 6.4.2), some modifications have been made to the original selection of linguistic features; in the third model (Section 6.4.3), the way in which context is represented has been modified. In Section 6.4.4, some further modifications have been made to both model and the set-up of the experiments.

From the annotated corpus, six different datasets were created for each model. First, we created a dataset for each of the three window-sizes 1, 2 and 3 (see Section 6.3.3). After that, from each of those three datasets a dataset for classification of the backward-looking function (with class variable **blf**) and a dataset for classification of the forward-looking function (with class variable **flf**) was produced: in the dataset for **blf**-classification, the **flf** had been removed, and in the dataset for **flf**-classification, the **blf** had been removed.

For each of these datasets, 10-fold cross-validation of three different classifiers was performed *for all possible subsets of the features specified*. From the evaluation results, the feature selection with the highest classification accuracy was derived, for each of the three classifiers. Three classifiers were used: the Naive Bayes classifier, the Bayesian Network classifier, and the Decision Tree classifier (see Section 3.7). In the setup of our testing environment, we made use of the Java-classes of the WEKA toolbox for machine learning implementations (see Witten and Frank, 1999).

6.4.1 Initial Model

The results of the analyses of all six datasets are presented in the twelve tables in Appendix B.1. For each dataset two tables have been generated: one table presenting the frequencies (COUNT) and relative frequencies (FRACTION) in the dataset of the different values of the class variable, and one table presenting the maximum accuracy found for each classifier (with 95% confidence intervals) and the selection of features that resulted in this accuracy. Besides the classifier results, also the selection of features yielded by an *attribute selection algorithm* is given. This algorithm evaluates subsets of features (in this case based on correlations among the features) and thus searches (using best first search) for an optimal set of features.

In Tables 6.7 and 6.8 we have summarised the accuracy results. The results show that in all cases, all classifiers show a significant improvement w.r.t. the chosen *baseline*: the relative frequency of the most frequent class value. Because of the high frequency of `no-blf` in the corpus, the baselines for blf-classification become much higher in comparison to the average relative frequency of $1/10 = 10\%$. The accuracies for the different classifiers do not differ significantly, although the Naive Bayes classifier performs worst in all 6 cases. This may be explained by the fact that the Naive Bayes classifier has serious problems when applied to sparse data.

The accuracies increase significantly when including blf and flf of the previous segment: when going from 1-grams to 2-grams, blf-classification gains more than 10% for all 3 classifiers. When including blf and flf from the previous *client* turn (i.e. in the case of 3-grams), the classification accuracies do not change significantly. The accuracy of the Naive Bayes classifier has even decreased for both flf- and blf-classification; the Decision Tree classifier only gains accuracy for blf-classification; the Bayes Net is the only classifier that shows improved accuracy for both flf- and blf-classification.

CONTEXT	Naive Bayes	Bayes Net	Decision Tree	BASELINE
1-grams	70.89 (±3.06)	71.01 (±3.06)	70.65 (±3.07)	64.38
2-grams	82.09 (±2.25)	83.44 (±2.19)	82.81 (±2.22)	54.37
3-grams	81.83 (±2.17)	84.15 (±2.06)	83.24 (±2.10)	54.91

Table 6.7: Maximum accuracies for blf-classification.

CONTEXT	Naive Bayes	Bayes Net	Decision Tree	BASELINE
1-grams	62.01 (±3.27)	66.98 (±3.17)	67.10 (±3.17)	46.75
2-grams	69.13 (±2.72)	70.93 (±2.67)	71.83 (±2.65)	40.59
3-grams	68.95 (±2.61)	71.43 (±2.54)	70.85 (±2.56)	40.05

Table 6.8: Maximum accuracies for flf-classification.

We also evaluated the classifiers using the data corresponding to the features selected by the attribute selection algorithm. The results are given in Tables 6.9 and 6.10. A first thing to notice is that all accuracies significantly outperform the baselines. The most important conclusion to be drawn however, is that the accuracies for flf-classification (Table 6.10) are significantly worse than the optimal ones (in Table 6.8), except for the Naive Bayes classifier. The accuracies for blf-classification (Table 6.9) however, are not significantly worse with respect to the optimal ones (in Table 6.7).

CONTEXT	Naive Bayes	Bayes Net	Decision Tree	baseline
1-grams	70.18 (± 3.08)	70.18 (± 3.08)	70.41 (± 3.08)	64.38
2-grams	81.10 (± 2.30)	81.55 (± 2.28)	81.19 (± 2.30)	54.37
3-grams	79.60 (± 2.27)	82.82 (± 2.12)	82.49 (± 2.14)	54.91

Table 6.9: Accuracies using the feature selections from the attribute selection algorithm for blf-classification.

The resulting feature selections did not show very clear which features are most significant in the classification tasks. However, some things can be noticed when comparing the various selections. In Table 6.11 a summarisation of

CONTEXT	Naive Bayes	Bayes Net	Decision Tree	baseline
1-grams	56.80 (± 3.34)	57.63 (± 3.33)	57.63 (± 3.33)	46.75
2-grams	59.14 (± 2.89)	59.95 (± 2.88)	60.13 (± 2.88)	40.59
3-grams	57.89 (± 2.78)	57.64 (± 2.78)	57.47 (± 2.78)	40.05

Table 6.10: Accuracies using the feature selections from attribute selection for flf-classification.

the selection results is presented. For each of the three window-sizes, the number of times a feature occurred in an optimal feature selection for a classifier is given for both blf- and flf-classification (with a maximum of 3) and in total (maximum 6). For example, **endsWithQuestionMark** occurred in two of the three selections in the case of blf-classification and in all three selections in the case of flf-classification.

	WINDOW-SIZE 1			WINDOW-SIZE 2			WINDOW-SIZE 3			
	blf	**total**	flf	blf	**total**	flf	blf	**total**	flf	TOTAL
blf-2							1	**2**	1	
flf-2							1	**4**	3	
blf-1				3	**6**	3	3	**4**	1	
flf-1				3	**6**	3	3	**6**	3	
lenq	2	**5**	3	2	**5**	3	3	**6**	3	16
isContinue	0	**3**	3	2	**4**	2	1	**4**	3	11
startsWithWHExpr	0	**3**	3	3	**6**	3	3	**6**	3	15
endsWithQuestionMark	2	**5**	3	3	**6**	3	2	**5**	3	16
startsWithCanYou	0	**0**	0	1	**2**	1	0	**0**	0	2
startsWithCanI	0	**2**	2	1	**4**	3	0	**3**	3	9
startsWithIWant	2	**3**	1	3	**6**	3	2	**5**	3	14
containsPositive	1	**4**	3	2	**5**	3	2	**5**	3	14
containsNegative	3	**6**	3	3	**6**	3	3	**5**	2	17
containsOkay	3	**6**	3	3	**6**	3	3	**6**	3	18
containsLocativePrep	1	**3**	2	3	**3**	0	0	**0**	0	6
containsLocativeAdverb	0	**0**	0	1	**2**	1	0	**1**	1	3
containsTell	1	**1**	0	0	**1**	1	2	**4**	2	6
containsDo	2	**5**	3	0	**3**	3	2	**4**	2	12

Table 6.11: Number of times the features are in the optimal selections for the 3 classifier types.

Let us first consider the results based on only utterance features, i.e., the window-size is 1. Here, the features **containsNegative** and **containsOkay** are in all optimal selections, while **startsWithCanYou** and **containsLocativeAdverb** are in none of the selections.

In general, the pattern of occurrences of the features is maintained when contextual features are added. The features **containsNegative** and **containsOkay** are very frequent, where one might add the additional frequently occurring features **lenq**, **startsWithWHExpr** and **endsWithQuestionMark**. The features **startsWithCanYou** and **containsLocativeAdverb** are in only a few of the selections.

Now let us consider the differences in the selection results between those for blf-classification and those for flf-classification. In general, we can say that for flf-classification, more utterance features are included in optimal selections than for blf-classification. For example, if we consider the window-size 1 case, the features **isContinue** and **startsWithWHExpr** are in none of the selections for blf-classification, but in all of the selections for flf-classification when considering window-size 1. Concerning **startsWithWHExpr**, this confirms our hypothesis that it would be a significant feature for distinguishing the flf `info-request` from other flf-types. One possibly reasonable explanation of the fact that **isContinue** is specifically significant for flf-classification, even without taking into account the previous flf of the same speaker as suggested in Section 6.3.2, could be that continuations in utterances are often used for (continuing their) `info-requests` and much less frequently for other flf-types.

When taking into account the flf and blf of the previous (system) utterance (a window-size of 2), besides the features that were in all selections for the case of window-size 1, also the features **startsWithWHExpr**, **endsWithQuestionMark** and **startsWithIWant** are in all classifier selections. Where **endsWithQuestionMark** was already in all but one selection in the window-size 1 case, the significance of **startsWithWHExpr** which was in none of the selections for blf-classification, is more remarkable. The contextual features seem to 'activate' utterance features in blf-classification, which is an indication of them being dependent. The feature **containsTell** on the other hand is in only one of the selections, like it was in the case of window-size 1. It is also very clear that the contextual features **blf-1** and **flf-1** are in all selections, which already seems plausible from the significant gain in classifier accuracies (Tables 6.7 and 6.8).

The result that the accuracies for window-size 3 were not significantly higher is partly illustrated by the fact that the additional contextual features **blf-2** and **flf-2** are not in all selections, where the contextual features for window-size 2 did appear in all selections. The fact that **blf-2** is the least informative of all contextual features, can be explained from the definitions of blf and flf in the annotation scheme: the backward-looking function characterises how the utterance refers to the previous dialogue, and therefore **blf-2** only refers to parts of the dialogue not taken into account in this model; however, the forward-looking function characterises how the utterance influences the future dialogue, and therefore **flf-2** refers to parts of the dialogue that *are* taken into account in this model.

The remark in Section 6.3.2 that the features **containsLocativePrep** and **containsLocativeAdverb** are expected to be less important in the SCHISMA dialogues than in dialogues concerning navigation and should be modified, is confirmed by the experimental results. These features are in only few of all optimal selections. Besides these two features, **containsTell** and **startsWithCanYou** are in few of all selections as well. These four features are therefore subject to either modification or removal in our process of finding a set of features that results in high accuracies for blf- and flf-classification.

In summary, we may draw the following tentative conclusions:

Conclusions concerning classification accuracies

1. For all obtained classifiers, the accuracies are significantly higher compared to the baseline;
2. in the 2-gram model the accuracies are significantly better than in the 1-gram model, especially in the case of blf-classification;
3. the 3-gram model does *not* yield significant improvement in accuracy compared to the 2-gram model (Black et al. (2003) have presented similar results concerning the window-sizes);
4. although not significantly, the Naive Bayes classifiers all show slightly lower accuracies compared to those of the Bayesian Network and Decision Tree classifiers;
5. the Bayesian Network and Decision Tree classifiers do not show significant differences in terms of accuracy.

Conclusions concerning feature selection

1. The accuracies obtained with the result from the feature selection algorithm are not significantly worse than the maximum accuracies in the case of blf-classification;
2. in the case of flf-classification however, the accuracies using the feature selection result are significantly worse for the Bayesian Network and Decision Tree classifiers;
3. based on the number of times the various features occur in the selections that yielded maximal accuracies, the features **lenq**, **startsWithWHExpr**, **endsWithQuestionMark**, **containsNegative** and **containsOkay** are the most informative ones for classification;
4. in the same sense, **startsWithCanYou** and **containsLocativeAdverb** are the least informative features;
5. for flf-classification, more utterance features are in the optimal selections than for blf-classification;
6. utterance features that did not seem very informative for classification, gain informativeness when contextual features are included, especially in the case of blf-classification;
7. the contextual features **blf-1** and **flf-1** are relevant in both flf- and blf-classification in the case of window-size 2;
8. in the window-size 3 model, the contextual features are also important; the feature **blf-2** however is the least informative, which was to be expected from the definition of the backward-looking function and its distance from the current segment;

9. the results confirm the expected irrelevancy of the features **containsLocativePrep** and **containsLocativeAdverb**.

6.4.2 Modification of Utterance Features

The experiments have been repeated after some modifications to the set of linguistic features. In the new model we have tried out a modified version of the feature **lenq**: in stead of three possible values, the feature now has four possible values: 'one' (1 word), 'few' ($2 \leq \#words < 5$), 'several' ($5 \leq \#words < 10$), and 'many' (10 or more words). The features **startsWithCanYou** and **containsTell** were removed, because the results in Section 6.4 revealed that they were not very informative features for classification. Finally, the features **containsLocativePrep** and **containsLocativeAdverb** were replaced by 'temporal counterparts': **containsTemporalPrep** – in the specification of which prepositions such as "naar", "achter" and "langs" were replaced by prepositions such as "tot" *(Eng.: until)* and "na" *(Eng.: after)* – and **containsTemporalAdverb** – in the specification of which adverbs such as "hier", "daar" and "rechts" were replaced by adverbs such as "nu" *(Eng.: now)*, "vandaag" *(Eng.: today)* and "morgen" *(Eng.: tomorrow)*.

Unfortunately, the results for the modified set of linguistic features did not show significant improvements of the maximum accuracies (see Tables 6.12 and 6.13).

CONTEXT	Naive Bayes	Bayes Net	Decision Tree	BASELINE
1-grams	70.89 (±3.06)	71.12 (±3.06)	70.41 (±3.08)	64.38
2-grams	82.45 (±2.56)	83.53 (±2.50)	82.72 (±2.55)	54.37
3-grams	82.08 (±2.59)	83.98 (±2.47)	82.66 (±2.55)	54.91

Table 6.12: Maximum Accuracies for blf-classification.

CONTEXT	Naive Bayes	Bayes Net	Decision Tree	BASELINE
1-grams	62.25 (±3.27)	67.69 (±3.15)	67.57 (±3.16)	46.75
2-grams	69.49 (±3.10)	71.56 (±3.04)	71.29 (±3.05)	40.59
3-grams	68.62 (±3.13)	71.59 (±3.04)	70.85 (±3.06)	40.05

Table 6.13: Maximum Accuracies for flf-classification.

The detailed results with the optimal feature selections are given in Appendix B.2. Table 6.14 gives a summarisation of the selection results. What can be noticed is that the new features **containsTemporalPrep** and **containsTemporalAdverb** occur more frequently in the optimal selections than their locative counterparts in the previous model. In this respect we may conclude that our expectation that temporal features would be more informative that locative features in the kind of dialogues we are dealing with in the corpus, has been confirmed.

It is clear that the (pre-)selection of linguistic features was rather intuitive and was not accounted for in a systematic and elaborate way. In this light,

	WINDOW-SIZE 1			WINDOW-SIZE 2			WINDOW-SIZE 3			
	blf	**total**	flf	blf	**total**	flf	blf	**total**	flf	TOTAL
blf-2							2	**3**	1	
flf-2							1	**3**	2	
blf-1				3	**6**	3	3	**4**	1	
flf-1				3	**6**	3	3	**6**	3	
lenq	1	**4**	3	3	**6**	3	3	**6**	3	16
isContinue	1	**4**	3	2	**4**	2	2	**4**	2	12
startsWithWHExpr	1	**4**	3	3	**6**	3	3	**6**	3	16
endsWithQuestionMark	3	**6**	3	3	**6**	3	2	**5**	3	17
startsWithCanI	1	**3**	2	0	**2**	2	2	**5**	3	10
startsWithIWant	2	**2**	0	2	**4**	2	1	**4**	3	10
containsPositive	0	**3**	3	2	**5**	3	2	**5**	3	13
containsNegative	3	**6**	3	3	**6**	3	3	**6**	3	18
containsOkay	3	**5**	2	3	**6**	3	3	**6**	3	17
containsTemporalPrep	0	**2**	2	2	**3**	1	1	**3**	2	8
containsTemporalAdverb	1	**1**	0	1	**4**	3	2	**4**	2	9
containsDo	2	**4**	2	1	**4**	3	2	**5**	3	13

Table 6.14: Number of times the features are in the optimal selections for the 3 classifier types

further attention to the pre-selection of linguistic features seems worthwhile in order to attain higher classification accuracies.

In some other empirical approaches to dialogue act recognition, selecting the most plausible dialogue act type was not based on derived linguistic features, but merely on sequences of words and additional information, such as prosodic cues or dialogue history (Reithinger and Klesen, 1997; Stolcke et al., 2000). Because our dataset is quite small, this seems to be a less appealing approach and the use of prior (syntactic) information to arrive at a relatively small set of linguistic features is more appropriate. Besides that, our goal of finding more higher level linguistic features that are informative for dialogue act recognition would have to be put aside in the case of using merely word sequences.

6.4.3 Alternative Ngram Generation

In the method of extracting the appropriate data from the corpus in the previous sections, the notion of 'previous flf/blf' is taken as 'the blf/flf of the previous turn'. In generating the data, the n-grams of segments – to which the dialogue act types have been assigned – are actually based on n-grams of turns (see Section 6.3.3). Because a turn may consist of several segments, several sequences of n segments from a n-gram of turns are possible. We have chosen to collect all of those possible sequences for the data (this explains why the datasets for larger window-sizes are larger and the associated baselines for classification accuracy are different).

As an alternative, a corpus dialogue may be considered as one sequence of segments in which the turns and utterances are disregarded, and generate n-grams of segments (i.e. sequences of n segments that are actually subsequent), ending with a segment from a client turn. So in that case, the speaker in all but the last segment will no longer be fixed, i.e., previous flf/blf's may be either one performed by the server or by the client. So every data-entry now represents a 'current' segment together with a number of preceding segments, in which only the 'current' segment is ensured to be part of a client turn. The segment directly preceding the current segment may be either part of the same client turn, or part of the previous system turn. This also means that the number of instances generated is the same, regardless the window-size.

The results in Tables 6.15 and 6.16 show a significant improvement in comparison with the results in Section 6.4.

CONTEXT	Naive Bayes	Bayes Net	Decision Tree	BASELINE
1-grams	70.89 (±3.06)	71.12 (±3.06)	70.41 (±3.08)	64.38
2-grams	87.69 (±2.22)	87.81 (±2.21)	88.28 (±2.17)	64.38
3-grams	87.69 (±2.22)	87.93 (±2.20)	88.40 (±2.16)	64.38

Table 6.15: Maximum Accuracies for blf-classification.

CONTEXT	Naive Bayes	Bayes Net	Decision Tree	BASELINE
1-grams	62.25 (±3.27)	67.69 (±3.15)	67.57 (±3.16)	46.75
2-grams	68.88 (±3.12)	72.66 (±3.01)	73.73 (±2.97)	46.75
3-grams	70.41 (±3.08)	72.66 (±3.01)	73.73 (±2.97)	46.75

Table 6.16: Maximum Accuracies for flf-classification.

Some further experiments were done concerning larger context-sizes, see Tables 6.17 and 6.18. Window-sizes up to 5 were examined, where the linguistic features were not taken into account. Although we have not seen any improvement for context sizes of 3 and more, one interesting point can be made. In the case of blf-classification, we had a baseline of 64.38% and improved that to a maximum accuracy of 71.12% for the Bayesian Network classifier, when taking into account only linguistic features of the current segment. When taking into account only the blf and flf of the previous segment, a maximum accuracy of 83.08% is obtained for that classifier. For all 3 classifiers, adding the previous segment to the model yields over 11% higher accuracy than adding linguistic features. In the case of flf-classification however, adding context in stead of linguistic features produces results that are over 4% worse for the Naive Bayes classifier and over 9% worse for the Bayesian Network and Decision Tree classifier.

CONTEXT	Naive Bayes	Bayes Net	Decision Tree	BASELINE
2-grams	82.96 (±2.54)	83.08 (±2.53)	84.02 (±2.47)	64.38
3-grams	83.55 (±2.50)	83.55 (±2.50)	84.02 (±2.47)	64.38
4-grams	83.55 (±2.50)	83.55 (±2.50)	84.02 (±2.47)	64.38
5-grams	83.55 (±2.50)	83.55 (±2.50)	84.02 (±2.47)	64.38

Table 6.17: Maximum Accuracies for blf-classification.

CONTEXT	Naive Bayes	Bayes Net	Decision Tree	BASELINE
2-grams	58.11 (±3.33)	58.11 (±3.33)	58.11 (±3.33)	46.75
3-grams	58.11 (±3.33)	60.00 (±3.30)	58.93 (±3.32)	46.75
4-grams	58.11 (±3.33)	60.00 (±3.30)	59.41 (±3.31)	46.75
5-grams	58.11 (±3.33)	60.00 (±3.30)	59.41 (±3.31)	46.75

Table 6.18: Maximum Accuracies for flf-classification.

Appendix B.3 gives the detailed results for the experiments described in this section so far.

6.4.4 Further Modifications

In the case of working with segments from subsequent turns, it was clear which of the speakers (i.e., client or server) was associated with a particular segment, knowing that the current segment always came from a client turn. In the alternative n-gram generation model, only the current segment was known to be a client segment, but the speakers of preceding segments are not alternately server, client, server, client, and so on, anymore. The segment directly preceding the current client segment can be another client segment. In order to take into account the speakers associated with the different segments, they need to be made explicit by means of an additional attribute.

After adding new contextual features that make explicit the speaker for each preceding segment, we repeated the experiments. The feature **sp-n** refers to the speaker of the n-th previous turn, so **sp-1** is the speaker of the segment directly preceding the current segment. Naturally, these new features all have three possible values: 'C' (the client), 'S' (the server) and 'null' (no such segment in n-gram). The results in Tables 6.19 to 6.22 show that in the optimal accuracies, only very small improvements could be achieved in comparison to the previous model, in which the speaker in previous segments was not made explicit.

CONTEXT	Naive Bayes	Bayes Net	Decision Tree	BASELINE
1-grams	70.89 (±3.06)	71.12 (±3.06)	70.41 (±3.08)	64.38
2-grams	87.93 (±2.20)	88.28 (±2.17)	88.88 (±2.12)	64.38
3-grams	87.93 (±2.20)	88.28 (±2.17)	89.11 (±2.10)	64.38

Table 6.19: Maximum Accuracies for blf-classification.

CONTEXT	Naive Bayes	Bayes Net	Decision Tree	BASELINE
1-grams	62.25 (±3.27)	67.69 (±3.15)	67.57 (±3.16)	46.75
2-grams	68.99 (±3.12)	73.02 (±2.99)	74.08 (±2.95)	46.75
3-grams	70.65 (±3.07)	73.02 (±2.99)	74.08 (±2.95)	46.75

Table 6.20: Maximum Accuracies for flf-classification.

Another modification in the experiments we introduced recently, is not in the models themselves, but in the collection of the results. In the experiments

CONTEXT	Naive Bayes	Bayes Net	Decision Tree	BASELINE
2-grams	83.67 (±2.49)	84.85 (±2.42)	85.44 (±2.38)	64.38
3-grams	83.79 (±2.49)	85.21 (±2.39)	85.80 (±2.35)	64.38
4-grams	84.02 (±2.47)	85.21 (±2.39)	85.80 (±2.35)	64.38
5-grams	84.02 (±2.47)	85.21 (±2.39)	85.80 (±2.35)	64.38

Table 6.21: Maximum Accuracies for blf-classification.

CONTEXT	Naive Bayes	Bayes Net	Decision Tree	BASELINE
2-grams	58.11 (±3.33)	58.70 (±3.32)	58.82 (±3.32)	46.75
3-grams	58.11 (±3.33)	60.59 (±3.29)	59.65 (±3.31)	46.75
4-grams	58.11 (±3.33)	60.95 (±3.29)	60.47 (±3.30)	46.75
5-grams	58.11 (±3.33)	60.95 (±3.29)	60.47 (±3.30)	46.75

Table 6.22: Maximum Accuracies for flf-classification.

so far, we have only collected single feature selections to be optimal in the sense of yielding maximum accuracies for a classifier. However, it may very well be the case that there are *several* selections of features that yield the same maximal classifier accuracy. In retrospective, this is not very surprising as we are using very small datasets and therefore even smaller datasets for testing in a cross-validation fold. Therefore, we have now modified the software to collect *all* feature selections that yield the maximum accuracy. With this in mind, we have chosen to modify the detailed result tables (see Table 6.25) in such a way, that each classifier is now associated with two feature selection columns: one column to indicate which features were in ALL of the optimal selections, and one column to indicate which features were in ANY of the optimal selections. Unfortunately, were are forced to leave extensive experiments in this modified approach and analyses of the results for future work.

As discussed in Chapter 4, learning the network structure of a Bayesian network using the K2-algorithm requires an ordering on the variables (Section 4.5.1); different orderings may therefore lead to different network structures. In our final experiment presented in this thesis, we have repeated the experiments for blf-classification with window-size 2 as before, but with five different variable orderings. In Table 6.23 these orderings have been listed. In all orderings, the linguistic features have consecutive positions and only appear in two variants: one corresponding to the original ordering (**lenq**, ..., **containsDo**) and one in the reversed ordering (**containsDo**, ..., **lenq**). Further variations are based on the position of the contextual features and the class variable.

From the results in Table 6.24 one can conclude that there is hardly any difference in the maximal accuracies obtained for the different orderings. On the other hand, the results concerning the optimal selections show hardly any agreement. In Table 6.25, the results of the ordering ORD04 are given for our three classifiers.

ORD01:	**blf,**	**lenq, . . . , containsDo,**	**blf-1, flf-1, sp-1**	
ORD02:	**sp-1, flf-1, blf-1,**	**blf,**	**containsDo, . . . , lenq**	
ORD03:	**blf,**	**sp-1, flf-1, blf-1,**	**containsDo, . . . , lenq**	
ORD04:	**sp-1, flf-1, blf-1,**	**blf,**	**lenq, . . . , containsDo**	
ORD05:	**blf,**	**sp-1, flf-1, blf-1,**	**lenq, . . . , containsDo**	

Table 6.23: Variable orderings used in BayesNetK2 trainings (Table 6.25).

Classifier	ORD01		ORD02		ORD03		ORD04		ORD05	
Max. Accuracy	87.574		88.0473		88.0473		88.284		88.284	
Feat. Selections	ALL	ANY	ALL	ANY	ALL	ANY	ALL	ANY	ALL	ANY
sp-1	−	−	+	+	−	−	+	+	−	−
flf-1	−	−	+	+	+	+	+	+	+	+
blf-1	−	−	−	+	+	+	−	+	+	+
lenq	+	+	−	+	−	+	−	+	−	+
isContinue	−	+	−	−	−	−	−	+	−	+
startsWithWHExpr	+	+	−	−	−	−	−	+	−	+
endsWithQuestM	−	−	−	−	−	−	−	+	−	+
startsWithCanI	+	+	+	+	+	+	−	+	−	+
startsWithIWant	−	−	−	−	−	−	−	+	−	+
containsPositive	+	+	−	−	−	−	+	+	+	+
containsNegative	−	+	−	+	−	+	+	+	+	+
containsOkay	−	+	−	−	−	−	+	+	+	+
containsTempPrep	−	−	−	−	−	−	−	+	−	+
containsTempAdv	+	+	−	+	−	+	−	+	−	+
containsDo	−	+	+	+	+	+	−	+	−	+

Table 6.24: Results from blf-classification with window-size 2: using different variable orderings for BayesNetK2 learning.

In Figure 6.3, the Bayesian network (structure) is given that resulted from applying the K2 algorithm on one of the optimal selections in the case of ordering ORD03. In this network, various dependencies among the features are assumed, in contrast to a Naive Bayes classifier that considers all features to be independent given the class value. For example, the features **okay** and **containsDo** are assumed to be conditionally independent given the class variable **blf**, but **startsWithWHExpr** and **endsWithQuestionMark** are not.

CLASSIFIER	NaiveBayes		BayesNetK2		J48		Attribute Selection
MAXIMUM ACCURACY	87.929		88.284		88.8757		
FEATURE SELECTIONS	ALL	ANY	ALL	ANY	ALL	ANY	
sp-1	+	+	+	+	+	+	−
flf-1	+	+	+	+	+	+	+
blf-1	−	−	−	+	+	+	−
lenq	+	+	−	+	−	−	−
isContinue	+	+	−	+	−	+	−
startsWithWHExpr	−	−	−	+	+	+	−
endsWithQuestionMark	−	−	−	+	+	+	−
startsWithCanI	−	−	−	+	−	+	−
startsWithIWant	+	+	−	+	+	+	−
containsPositive	−	−	+	+	−	−	−
containsNegative	+	+	+	+	+	+	−
containsOkay	−	−	+	+	−	−	−
containsTemporalPrep	−	−	−	+	−	−	−
containsTemporalAdverb	−	−	−	+	−	+	−
containsDo	−	−	−	+	−	+	−

Table 6.25: Results from blf-classification with window-size 2.

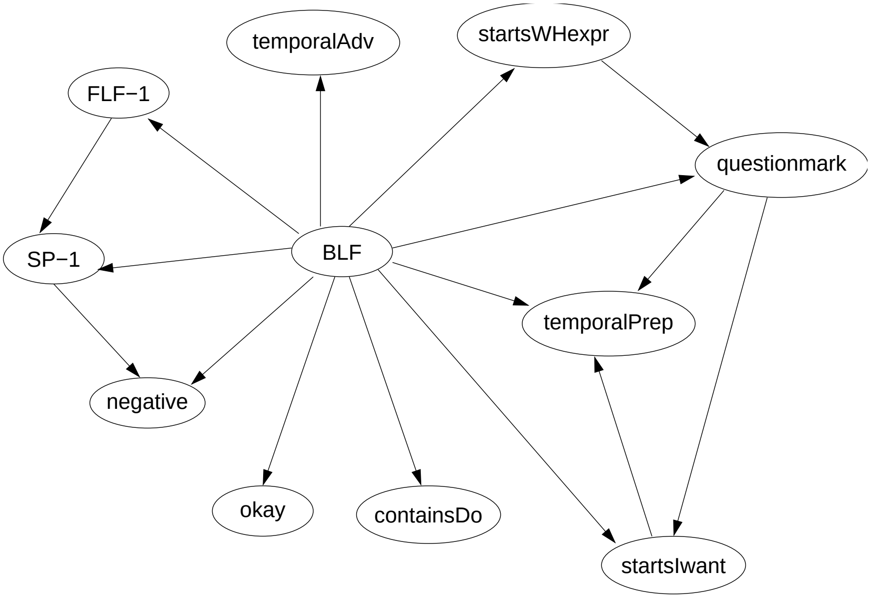


Figure 6.3: Bayesian network resulting from K2 algorithm for an optimal feature selection.

6.5 Conclusion

With the primary aim to be able to perform some initial experiments with Bayesian networks for dialogue act classification, we developed a dialogue act annotation scheme for the SCHISMA corpus to extract training data from. This annotation scheme is based on a standard for annotation of task-oriented dialogues, the DAMSL scheme (see Section 2.5.4). Our scheme contains multiple layers, including layers of forward- and backward-looking functions and a layer of topic-management acts. Manually annotating the corpus has resulted in 64 XML-files, each containing one annotated dialogue from the corpus.

Software has been developed to extract data files from the annotated corpus for various classification models. These models are given by a set of utterance features with possible values, two sets of dialogue act types to distinguish between, i.e., one set of forward-looking functions and one set of backward-looking functions, and the size of the dialogue history to be taken into account. In addition, the set-up of the experiments on dialogue act classification, involving different classifier types and different subsets of the features used, has been implemented.

The results from the experiments as presented in this chapter are subdivided into four sections, that can be seen as phases that were passed through during the experiments, each phase involving a modification or improvement of the models used. From the results in the initial model (Section 6.4.1), the most important conclusion to be drawn was that using the dialogue act type of the previous utterance significantly improved the performance of the classifiers in terms of their classification accuracies. However, taking into account further preceding utterances did not improve the accuracies. In comparing the three classifiers evaluated, Naive Bayes, Bayesian network and Decision Tree, the Naive Bayes classifiers performed somewhat worse than the other two classifiers.

The utterance features in the models were selected on the basis of merely an initial intuition on their significance to the dialogue act performed. From the results concerning evaluation of subsets of this initial feature selection, no definite conclusions could be drawn. It should be clear that much improvement may be expected in finding appropriate features to be used in classifying dialogue act types. Nevertheless, one conclusion is that the utterance features were more relevant for classification of forward-looking functions than for classification of backward-looking functions, whereas contextual features were more relevant for classification of backward-looking functions than for classification of forward-looking functions. This result confirms our prior intuition regarding the definition of forward- and backward-looking functions.

The modification of the classification models as presented in Section 6.4.3 resulted in significantly improved accuracies. The modification involved a change in the sense of context, i.e., in stead of looking at sequences of turns (with the complication of the possibility of multiple subsequent dialogue acts) the new model concerns sequences of segments (each being associated with a single dialogue act). As the sequences of dialogues considered in the new model were not alternatingly associated with client and server, the speaker of a segment was made explicit by an additional feature, a modification introduced in Section 6.4.4.

Another modification described in Section 6.4.4 concerns the analysis of subsets of features in the classification experiments. This modification involves an improved account of the fact that in several cases, multiple subsets of features resulted in the maximal classifier accuracy, in stead of the single first subset recorded and analysed in the previous models.

Variations in the ordering of the variables (features and class variable) that could result in different results for the Bayesian network classifier, did not show significant differences for our data-sets. On the other hand, the feature selections for the different orderings did not show much agreement.

6.5.1 Future Recommendations

Future work will have to consist of specifying additional utterance features and modifying and/or removing existing ones, in order to obtain higher dialogue act classification accuracies. Inspection of more detailed information about the classification results might be useful, for example, the results per class value as stored in a confusion matrix and in the form of precision/recall results.

If we are able to obtain better classification results, we might also be better able to pick one set of the most significant features to do further experiments with. These further experiments will be aimed more at the comparison of different classifiers with different parameter sets.

Another suggestion for improving classification results might be the specification of two different sets of dialogue acts for client and server utterances. In classifying client utterances we might not be interested in distinguishing between two dialogue act types, say, A and B, but this distinction might be relevant for any (previous) server utterances taken into account in the classification. The differences in frequencies of the dialogue act types for client and server in the SCHISMA corpus also suggest this (see Section 6.2.4).

In the conclusions concerning window-sizes (related to the number of preceding utterances to take into account), it was noted that including one preceding utterance in the model (a window-size of 2) yielded a significant improvement in the classification accuracies, but including further preceding utterrances did not. Experiments with models including context features only, where window-sizes up to 5 were explored, suggested that looking back any further in the dialogue seems useless, because no improved accuracies were found beyond window-size 2. However, one factor that could give rise to improved accuracies for larger window-sizes, has not been taken into account. If the dialogues are heavily structured with subdialogues, such as subdialogues for verification or repair, or subdialogues related to the task at hand, then it may occur that the current utterance is produced after completion of a subdialogue and elaborates on the utterance preceding the subdialogue. Therefore, the size of possible subdialogues should be taken into account when deciding on the window-sizes to be explored in the classification experiments.

Finally, we would like to mention the option of extending the annotation scheme with more specific dialogue act types and extend the annotated corpus accordingly. This may result in data leading to higher classification accuracies, but only for this specific type of dialogue. This reveals a trade-off in developing dialogue act classifiers. On the one hand, specific dialogue acts can be specified for specific types of dialogue, where the classification task is relatively simple, but the resulting classifiers do not work well for other types of dialogue. On the other hand, task-independent dialogue act types can be used, where the classification task is more complex and more data will be needed, but the resulting classifiers perform equally well in various types of dialogue.

Chapter 7

Conclusion

In the introduction of this thesis (Chapter 1), we showed that *uncertainty* is an important problem a dialogue agent has to deal with when interacting with another agent using natural language dialogue. The uncertainty stems from the fact that the agent has only partial information when interpreting an utterance produced by a dialogue partner. Information may be lost in the communication channel and moreover, generally more is communicated than actually said, which is due to certain tacit conventions underlying natural interaction. These conventions are enforced under the particular circumstances of the interaction. The true meaning of an utterance is therefore determined by countless aspects of the situation that cannot all be taken into account.

In case of an artificial *dialogue agent* (that is, a dialogue system) interacting with a human user, some formal specification for natural language dialogue is required. Because no such specification has been found – and never will be – that is accurate and complete, uncertainty arises because of abstractions that may leave relevant phenomena uncaptured. A user may not behave according to the rules that have been specified in the model of the agent.

In order to deal with a situation in which the agent is uncertain about the interpretation of an utterance, he may decide to ask for further information, or he may take an *educated guess* and use this as a plausible assumption that may however have to be revised later in the dialogue because of newly obtained information. In Chapter 3, we have presented uncertainty as a mental attitude of an agent in interaction with its environment and argued for the use of probability theory as a convenient formalism supporting plausible common-sense reasoning. *Bayesian networks* have been defined in Chapter 4 as probabilistic models in which explicit assumptions of conditional independency have been made via a directed acyclic graph. These assumptions help making the model computationally tractable. Bayesian networks can be used to make plausible assumptions based on incomplete information. Furthermore, expert knowledge can be combined with raw empirical data to arrive at appropriate models.

Central in most approaches to modelling dialogue is some notion of *dialogue* *acts*, reflecting the fact that in producing utterances, speakers perform communicative actions. In Chapter 2, a detailed discussion on dialogue acts and its origin in Searle's speech act theory has been given. Wittgenstein pointed out earlier that there are countless different ways of language use, but Searle nevertheless attempted to arrive at a systematisation of language use by concentrating on one particular sense of language use: illocutionary acts, which consist of an illocutionary force and a propositional content. In the light of Wittgenstein's observation we may conclude that any attempt to a systematisation of ways of using language introduces abstractions from possibly very relevant aspects of language use, and hence forms an inevitable, but apparent source of uncertainty.

Dialogue acts have been introduced as an extension of illocutionary acts, reflecting that utterances are to be seen as communicative actions that are contributions to a coherent dialogue as well. Similar to illocutionary acts consisting of an illocutionary force and a propositional content, dialogue acts consist of *dialogue act type* and a semantic content. Dialogue act types can be organised in *dialogue act taxonomies*. Such taxonomies may form the basis of a scheme for annotating a corpus of dialogues. In this thesis we have discussed different existing taxonomies and annotation schemes and presented a new annotation scheme (Chapter 6) for the SCHISMA dialogue corpus, built on the DAMSL standard.

An important aspect in any dialogue system is the task of identifying what the user meant, based on the available information about the utterance and the circumstances under which the utterance was produced. Inspired by Searle's systematic work on relating Illocutionary Force Indicating Devices (IFIDs) that are claimed to be used conventionally by speakers to establish an illocutionary act, the task of *dialogue act recognition* has become an important research subject in computational approaches to dialogue. One method of finding models that relate features of utterances-in-context to dialogue act types is using raw data in an annotated dialogue corpus. Machine learning techniques can be used to induce dialogue act classifiers from the data.

We have taken this *dialogue act classification* task to do some first concrete explorations in using Bayesian networks in dialogue systems. In Chapter 6, we discussed the annotation of the SCHISMA corpus with the new annotation scheme mentioned earlier. The annotated corpus was used for extracting datafiles for training and evaluating dialogue act classifiers, in particular Bayesian network classifiers. By using XML-specifications with a DTD, the annotator was enabled to produce annotations that comply with the dialogue act hierarchy as specified in the annotation scheme. Furthermore, an existing Java API for processing XML files could be used in the software for extracting data files.

The annotation scheme consists of multiple layers, including the layers of Forward- and Backward-looking Functions. This model suggests that in each utterance a forward- and backward-looking function are performed simultaneously (except for some special cases). Therefore, our experiments involved evaluation of separate classifiers for these two sorts of dialogue act types. In the numerous experiments, we trained classifiers for the forward-looking function that performed with accuracies of ±73%, against a baseline (the most frequent dialogue act) of 47% and classifiers for the backward-looking functions that performed with accuracies of ±88%, against a baseline of 64%.

The results confirmed our intuition that utterance features were more informative to classifying forward-looking functions than to classifying backward-looking functions. The context features seemed to be more informative to classifying backward-looking functions. Another important conclusion from the results is that the performance of the classifiers improved significantly when taking the previous utterance into account. However, taking further preceding utterances into account did not result in significant improvements.

We believe that most improvements in the performance of the classifiers can be established in the selection of better utterance features. The current set of features were merely based on initial intuitions concerning their significance to identifying dialogue act types. Therefore, more elaborate research on the utterance features is recommended. Concerning the contextual features, experiments with larger window-sizes (in other words, a larger dialogue history) may yield better results because due to the occurrence of subdialogues, dialogue acts that were performed before such a subdialogue may be relevant to the dialogue act type of the current utterance after all.

Three different classifier types were used for each of the data-sets: Bayesian network, Naive Bayes and Decision Tree. In the overall results, the Naive Bayes classifiers performed somewhat worse than the Bayesian network and Decision Tree classifiers, which may be explained by the fact that the used data is sparse, which is a common problem for Naive Bayes classifiers. The Bayesian network and Decision Tree classifiers themselves hardly differed in their performance.

Concerning the experiments on dialogue act classification, there are numerous things left that can be tried out. More elaborate analyses of the results could give useful insights; for example, besides accuracies, one could also take a closer look at the confusion matrices that result from the evaluations. Also, other types of classifiers with different parameter settings could be trained and subjected to, for example, paired t-tests. Another issue might be reconsidering the set of dialogue act types to distinguish between. More specific types may lead to improved accuracies, but less data may be available for these types. It may be more useful to see how Bayesian network classifiers perform on data-sets that are larger and that are more widely used, so as to get a more reliable comparison with other classifier types and machine learning techniques.

In Chapter 5, we have discussed dialogue modelling approaches that have been applied in existing dialogue systems. Two state of the art approaches of modelling that use some notion of dialogue state have been discussed, the BDI-agent approach and the Information State approach. In our proposal to use a Bayesian network framework to incorporate various relevant aspects along the different levels of analysis into one model, the agent-based approach is taken. Bayesian networks are used as mental models of the conversational agent, or in other words, belief models of an agent reasoning under uncertainty. Dynamic Bayesian networks can be used to model the change a dialogue state undergoes due to the performance of a dialogue act. This may replace deterministic implementations in the form of information state update rules or other inference rules.

Further experiments are needed to show how Bayesian networks can be used in dialogue systems in this more general sense than just for the specific task of dialogue act classification. The Bayesian network classifiers that were discussed extensively in Chapter 6, consisted of only one unobserved class variable and the other variables, the features, were known for each instance. However, other unobserved, but relevant aspects – for example, the beliefs and preferences of the speaker – may also be incorporated into Bayesian network models. In the field of user modelling, the use of Bayesian networks is indeed becoming a trend, and it seems to be worthwhile to incorporate existing techniques into conversational systems.

Finally, experiments with dynamic Bayesian networks may lead to some interesting results, where we may start for example with dynamic Bayesian networks for the task of dialogue act classification, along the lines of the experiments described in this thesis.

Appendix A

XML-encoded Annotation

A.1 DTD

```
<!ELEMENT dialogue ( turn+ ) >
<!ATTLIST dialogue
    author  CDATA  #REQUIRED
    date    CDATA  #REQUIRED
>

<!-- A ’turn’ indicates a period of the dialogue during
     which one of the participants has the turn. -->
<!ELEMENT turn ( utterance+ ) >
<!-- the attribute ’sp’ indicates the speaker of the turn:
     ’C’ refers to the client (the user)
     ’S’ refers to the server (the wizard)
-->
<!ATTLIST turn
    sp  ( C | S )  #REQUIRED
>

<!-- An utterance is a sentence uttered by
     the current speaker. -->
<!ELEMENT utterance ( segment+ ) >

<!--
     A segment is the basic unit for
     labelling dialogue acts.
-->
<!ELEMENT segment ( communicative_status?, information_level,
  forward_looking_function+, backward_looking_function+,
  topic_management, content, surface_features? ) >
<!ATTLIST segment
    id  ID  #REQUIRED
>
```

```
<!-- First Layer: Communicative Status -->
<!ELEMENT communicative_status ( uninterpretable | abandoned |
          self_talk ) >
<!ELEMENT uninterpretable EMPTY >
<!ELEMENT abandoned EMPTY >
<!ELEMENT self_talk EMPTY >

<!-- Second Layer: Information Level -->
<!ELEMENT information_level ( task | task_management |
  communication_management | other_level ) >
<!ELEMENT task EMPTY >
<!ELEMENT task_management EMPTY >
<!ELEMENT communication_management EMPTY >
<!ELEMENT other_level EMPTY >

<!-- Third Layer: Forward-looking Functions -->
<!ELEMENT forward_looking_function ( statement? |
   influencing_addressee_future_action? | info_request? |
   committing_speaker_future_action? | conventional? |
   explicit_performative? | exclamation? |
   other_forward_function? | no_flf? ) >

<!ELEMENT statement (assert? | reassert? | other_statement?) >
<!ELEMENT assert EMPTY >
<!ELEMENT reassert EMPTY >
<!ELEMENT other_statement EMPTY >

<!ELEMENT influencing_addressee_future_action ( open_option? |
  action_directive? ) >
<!ELEMENT open_option EMPTY >
<!ELEMENT action_directive EMPTY >

<!ELEMENT info_request ( query_ref? | query_if? ) >
<!ELEMENT query_ref EMPTY >
<!ELEMENT query_if EMPTY >

<!ELEMENT committing_speaker_future_action (offer?|commit?) >
<!ELEMENT offer EMPTY >
<!ELEMENT commit EMPTY >

<!ELEMENT conventional ( opening? | closing? ) >
<!ELEMENT opening EMPTY >
<!ELEMENT closing EMPTY >

<!ELEMENT explicit_performative ( thank? ) >
<!ELEMENT thank EMPTY >

<!ELEMENT exclamation EMPTY >
<!ELEMENT other_forward_function EMPTY >
<!ELEMENT no_flf EMPTY >
```

```
<!-- Fourth Layer: Backward-looking Functions -->
<!ELEMENT backward_looking_function ( agreement? |
   understanding? | answer? | no_blf?) >

<!ATTLIST backward_looking_function
  ref  IDREFS  #IMPLIED
>   <!-- temporary -->

<!ELEMENT agreement ( accept? | accept_part? | maybe? |
   reject_part? | reject? | hold? ) >
<!ELEMENT accept EMPTY >
<!ELEMENT accept_part EMPTY >
<!ELEMENT maybe EMPTY >
<!ELEMENT reject_part EMPTY >
<!ELEMENT reject EMPTY >
<!ELEMENT hold EMPTY >

<!ELEMENT understanding ( signal_non_understanding? |
  signal_understanding? | correct_misspeaking? ) >
<!ELEMENT signal_non_understanding EMPTY >
<!ELEMENT signal_understanding ( acknowledge? |
   repeat_rephrase? | completion? ) >
<!ELEMENT acknowledge EMPTY>
<!ELEMENT repeat_rephrase EMPTY>
<!ELEMENT completion EMPTY>
<!ELEMENT correct_misspeaking EMPTY >

<!ELEMENT answer ( positive_answer? | negative_answer? |
  no_answer_feedback? | correction_feedback? ) >
<!ELEMENT positive_answer ( confirm? ) >
<!ELEMENT confirm EMPTY >
<!ELEMENT negative_answer ( disconfirm? ) >
<!ELEMENT disconfirm EMPTY >
<!ELEMENT no_answer_feedback EMPTY >
<!ELEMENT correction_feedback EMPTY >

<!ELEMENT no_blf EMPTY >
```

```
<!-- Fifth Layer: Topic Management -->
<!ELEMENT topic_management ( shift | elaborate | narrow |
   no_tm ) >

<!ELEMENT shift ( introduce_topic | refer_former_topic ) >
<!ELEMENT introduce_topic EMPTY >
<!ATTLIST introduce_topic
    id  ID  #REQUIRED
>
<!ELEMENT refer_former_topic EMPTY >
<!ATTLIST refer_former_topic
    ref  IDREF  #REQUIRED
>
<!ELEMENT elaborate EMPTY >
<!ATTLIST elaborate
    ref  IDREF  #REQUIRED
>
<!ELEMENT narrow EMPTY >
<!ATTLIST narrow
    ref IDREF  #REQUIRED
    id  ID     #REQUIRED
>
<!ELEMENT no_tm EMPTY >

<!-- Sixth Layer: Content -->
<!ELEMENT content ( #PCDATA ) >
<!-- the typed utterance itself -->

<!-- Linguistic Features of the segment -->

<!ELEMENT surface_features ( continuation?, sentence_type,
   punctuation, person_of_subject, wh_word?,
   other_features )? >

<!-- <snip> -->

<!-- end of DTD -->
```

A.2 Example Annotation

```
<dialogue>

<!-- <snip> -->

<turn sp="C"><utterance><segment id="s03">
         <forward_looking_function>
            <info_request><query_ref/></info_request>
         </forward_looking_function>
         <backward_looking_function>
            <no_blf/>
         </backward_looking_function>
         <topic_management>
            <narrow ref="t01" id="t011"/>
         </topic_management>
         <content>Wanneer draait Sweeney Todd?</content>
      </segment></utterance></turn>

<turn sp="S"><utterance><segment id="s04">
         <forward_looking_function>
            <influencing_addressee_future_action>
               <open_option/>
            </influencing_addressee_future_action>
         </forward_looking_function>
         <backward_looking_function ref="s03">
            <answer><positive_answer/></answer>
         </backward_looking_function>
         <topic_management>
            <elaborate ref="t011"/>
         </topic_management>
         <content>U kunt "Sweeney Todd" zien in de Grote
            Zaal in de periode 28 december 1993 t/m
            30 december 1993.</content>
      </segment></utterance></turn>

<turn sp="C"><utterance><segment id="s05">
         <forward_looking_function>
            <info_request><query_ref/></info_request>
         </forward_looking_function>
         <backward_looking_function>
            <no_blf/>
         </backward_looking_function>
         <topic_management>
            <narrow ref="t01" id="t012"/>
         </topic_management>
         <content>En Under a blue roof?</content>
      </segment></utterance></turn>
```

```
<turn sp="S"><utterance><segment id="s06">
         <forward_looking_function>
            <influencing_addressee_future_action>
               <open_option/>
            </influencing_addressee_future_action>
         </forward_looking_function>
         <backward_looking_function ref="s05">
            <answer><positive_answer/></answer>
         </backward_looking_function>
         <topic_management>
            <elaborate ref="t012"/>
         </topic_management>
         <content>U kunt "Under a blue Roof" zien in
            de Grote Zaal op 19 mei 1994.</content>
      </segment></utterance></turn>

<turn sp="C"><utterance><segment id="s07">
         <forward_looking_function>
            <info_request><query_if/></info_request>
         </forward_looking_function>
         <backward_looking_function>
            <no_blf/>
         </backward_looking_function>
         <topic_management>
            <elaborate ref="t012"/>
         </topic_management>
         <content>Kan ik daar kaarten voor
            reserveren</content>
      </segment></utterance></turn>

<turn sp="S">
   <utterance><segment id="s081">
         <forward_looking_function>
            <info_request><query_if/></info_request>
         </forward_looking_function>
         <backward_looking_function ref="s07">
            <agreement><hold/></agreement>
         </backward_looking_function>
         <topic_management>
            <elaborate ref="t012"/>
         </topic_management>
         <content>Heeft u een reductiekaart?</content>
      </segment></utterance>
   <utterance><segment id="s082">
         <forward_looking_function>
            <info_request><query_ref/></info_request>
         </forward_looking_function>
         <backward_looking_function>
            <no_blf/>
         </backward_looking_function>
         <topic_management>
            <elaborate ref="t012"/>
         </topic_management>
         <content>Hoeveel kaartjes wilt U?</content>
      </segment></utterance>
</turn>

<turn sp="C">
   <utterance><segment id="s091">
         <forward_looking_function>
            <statement><assert/></statement>
         </forward_looking_function>
         <backward_looking_function ref="s081">
            <answer><negative_answer/></answer>
         </backward_looking_function>
         <topic_management>
            <elaborate ref="t012"/>
         </topic_management>
         <content>Ik heb geen reductiekaart.</content>
      </segment></utterance>
   <utterance><segment id="s092">
         <forward_looking_function>
            <influencing_addressee_future_action>
               <action_directive/>
            </influencing_addressee_future_action>
         </forward_looking_function>
         <backward_looking_function ref="s082">
            <answer><positive_answer/></answer>
         </backward_looking_function>
         <topic_management>
            <elaborate ref="t012"/>
         </topic_management>
         <content>Vier kaartjes graag.</content>
      </segment></utterance>
</turn>

<!-- <snip> -->

</dialogue>
```

Appendix B

Dialogue Act Classification Results

B.1 Initial Model

VALUE	COUNT	FRACTION
accept	61	7.22
hold	37	4.38
reject	12	1.42
signal_non_understanding	3	0.36
signal_understanding	7	0.83
positive_answer	145	17.16
negative_answer	29	3.43
no_answer_feedback	2	0.24
no_blf	544	64.38
other	5	0.59
TOTAL	845	100

Table B.1: Class statistics for blf in case of 1-grams of turns

CLASSIFIER	NaiveBayes	BayesNetK2-S-ENTROPY	J48	AttribSelect
MAX. ACCURACY	70.8876	71.0059	70.6509	
FEATURES	FEATURE SELECTION			
lenq	−	+	+	−
isContinue	−	−	−	−
startsWithWHExpr	−	−	−	+
endsWithQuestionMark	+	+	−	+
startsWithCanYou	−	−	−	−
startsWithCanI	−	−	−	−
startsWithIWant	+	+	−	−
containsPositive	−	+	−	−
containsNegative	+	+	+	+
containsOkay	+	+	+	+
containsLocativePrep	−	+	−	−
containsLocativeAdverb	−	−	−	−
containsTell	+	−	−	−
containsDo	+	+	−	−

Table B.2: Results from blf-classification with window-size 1; the accuracy baseline is 64.38%.

VALUE	COUNT	FRACTION
statement	114	13.49
action_directive	218	25.8
open_option	0	0.0
query_if	68	8.05
query_ref	395	46.75
committing_speaker_future_action	0	0.0
conventional	26	3.08
no_flf	0	0.0
other	24	2.84
TOTAL	845	100

Table B.3: Class statistics for flf in case of 1-grams of turns

CLASSIFIER	NaiveBayes	BayesNetK2-S-ENTROPY	J48	AttribSelect
MAX. ACCURACY	62.0118	66.9822	67.1006	
FEATURES	FEATURE SELECTION			
lenq	+	+	+	−
isContinue	+	+	+	−
startsWithWHExpr	+	+	+	+
endsWithQuestionMark	+	+	+	+
startsWithCanYou	−	−	−	−
startsWithCanI	+	+	−	−
startsWithIWant	−	+	−	−
containsPositive	+	+	+	−
containsNegative	+	+	+	−
containsOkay	+	+	+	−
containsLocativePrep	+	−	+	−
containsLocativeAdverb	−	−	−	−
containsTell	−	−	−	−
containsDo	+	+	+	−

Table B.4: Results from flf-classification with window-size 1; the accuracy baseline is 46.75%.

VALUE	COUNT	FRACTION
accept	105	9.45
hold	55	4.95
reject	18	1.62
signal_non_understanding	3	0.27
signal_understanding	8	0.72
positive_answer	252	22.68
negative_answer	54	4.86
no_answer_feedback	3	0.27
no_blf	604	54.37
other	9	0.81
TOTAL	1111	100

Table B.5: Class statistics for blf in case of 2-grams of turns

CLASSIFIER	NaiveBayes	BayesNetK2-S-ENTROPY	J48	AttribSelect
MAX. ACCURACY	82.0882	83.4383	82.8083	
FEATURES	FEATURE SELECTION			
blf-1	+	+	+	+
flf-1	+	+	+	+
lenq	−	+	+	−
isContinue	+	+	−	−
startsWithWHExpr	+	+	+	+
endsWithQuestionMark	+	+	+	+
startsWithCanYou	+	−	−	−
startsWithCanI	−	+	−	−
startsWithIWant	+	+	+	−
containsPositive	−	+	+	−
containsNegative	+	+	+	+
containsOkay	+	+	+	+
containsLocativePrep	+	+	+	−
containsLocativeAdverb	−	+	−	−
containsTell	−	−	−	−
containsDo	−	−	−	−

Table B.6: Results from blf-classification with window-size 2; the accuracy baseline is 54.37%.

VALUE	COUNT	FRACTION
statement	186	16.74
action_directive	332	29.88
open_option	0	0.0
query_if	79	7.11
query_ref	451	40.59
committing_speaker_future_action	0	0.0
conventional	33	2.97
no_flf	0	0.0
other	30	2.7
TOTAL	1111	100

Table B.7: Class statistics for flf in case of 2-grams of turns

CLASSIFIER	NaiveBayes	BayesNetK2-S-ENTROPY	J48	AttribSelect
MAX. ACCURACY	69.1269	70.9271	71.8272	
FEATURES	FEATURE SELECTION			
blf-1	+	+	+	+
flf-1	+	+	+	−
lenq	+	+	+	−
isContinue	+	−	+	−
startsWithWHExpr	+	+	+	+
endsWithQuestionMark	+	+	+	+
startsWithCanYou	−	−	+	−
startsWithCanI	+	+	+	−
startsWithIWant	+	+	+	−
containsPositive	+	+	+	−
containsNegative	+	+	+	−
containsOkay	+	+	+	−
containsLocativePrep	−	−	−	−
containsLocativeAdverb	−	+	−	−
containsTell	−	−	+	−
containsDo	+	+	+	−

Table B.8: Results from flf-classification with window-size 2; the accuracy baseline is 40.59%.

VALUE	COUNT	FRACTION
accept	120	9.91
hold	56	4.62
reject	19	1.57
signal_non_understanding	3	0.25
signal_understanding	11	0.91
positive_answer	263	21.72
negative_answer	60	4.95
no_answer_feedback	3	0.25
no_blf	665	54.91
other	11	0.91
TOTAL	1211	100

Table B.9: Class statistics for blf in case of 3-grams of turns

CLASSIFIER	NaiveBayes	BayesNetK2-S-ENTROPY	J48	AttribSelect
MAX. ACCURACY	81.8332	84.1453	83.237	
FEATURES	FEATURE SELECTION			
blf-2	−	+	−	−
flf-2	−	+	−	+
blf-1	+	+	+	+
flf-1	+	+	+	+
lenq	+	+	+	+
isContinue	+	−	−	−
startsWithWHExpr	+	+	+	+
endsWithQuestionMark	+	+	−	+
startsWithCanYou	−	−	−	−
startsWithCanI	−	−	−	−
startsWithIWant	+	+	−	−
containsPositive	−	+	+	−
containsNegative	+	+	+	+
containsOkay	+	+	+	+
containsLocativePrep	−	−	−	−
containsLocativeAdverb	−	−	−	−
containsTell	+	−	+	−
containsDo	+	+	−	−

Table B.10: Results from blf-classification with window-size 3; the accuracy baseline is 54.91%.

VALUE	COUNT	FRACTION
statement	212	17.51
action_directive	345	28.49
open_option	0	0.0
query_if	85	7.02
query_ref	485	40.05
committing_speaker_future_action	0	0.0
conventional	41	3.39
no_flf	0	0.0
other	43	3.55
TOTAL	1211	100

Table B.11: Class statistics for flf in case of 3-grams of turns

CLASSIFIER	NaiveBayes	BayesNetK2-S-ENTROPY	J48	AttribSelect
MAX. ACCURACY	68.9513	71.4286	70.8505	
FEATURES	FEATURE SELECTION			
blf-2	−	+	−	−
flf-2	+	+	+	−
blf-1	−	−	+	+
flf-1	+	+	+	−
lenq	+	+	+	−
isContinue	+	+	+	−
startsWithWHExpr	+	+	+	+
endsWithQuestionMark	+	+	+	+
startsWithCanYou	−	−	−	−
startsWithCanI	+	+	+	−
startsWithIWant	+	+	+	−
containsPositive	+	+	+	−
containsNegative	+	+	−	−
containsOkay	+	+	+	−
containsLocativePrep	−	−	−	−
containsLocativeAdverb	−	−	+	−
containsTell	−	+	+	−
containsDo	+	+	−	−

Table B.12: Results from flf-classification with window-size 3; the accuracy baseline is 40.05%.

B.2 Modification of Utterance Features

CLASSIFIER	NaiveBayes	BayesNetK2-S-ENTROPY	J48	AttribSelect
MAX. ACCURACY	62.2485	67.6923	67.574	
FEATURES	FEATURE SELECTION			
lenq	+	+	+	+
isContinue	+	+	+	−
startsWithWHExpr	+	+	+	+
endsWithQuestionMark	+	+	+	+
startsWithCanI	+	+	−	−
startsWithIWant	−	−	−	−
containsPositive	+	+	+	−
containsNegative	+	+	+	+
containsOkay	−	+	+	+
containsTemporalPrep	+	−	+	−
containsTemporalAdverb	−	−	−	−
containsDo	+	+	−	−

Table B.13: Results from blf-classification with window-size 1; the accuracy baseline is 64.38%.

CLASSIFIER	NaiveBayes	BayesNetK2-S-ENTROPY	J48	AttribSelect
MAX. ACCURACY	62.2485	67.6923	67.574	
FEATURES	FEATURE SELECTION			
lenq	+	+	+	−
isContinue	+	+	+	−
startsWithWHExpr	+	+	+	+
endsWithQuestionMark	+	+	+	+
startsWithCanI	+	+	−	−
startsWithIWant	−	−	−	−
containsPositive	+	+	+	−
containsNegative	+	+	+	−
containsOkay	−	+	+	−
containsTemporalPrep	+	−	+	−
containsTemporalAdverb	−	−	−	−
containsDo	+	+	−	−

Table B.14: Results from flf-classification with window-size 1; the accuracy baseline is 46.75%.

B.3 Alternative Ngram Generation

VALUE	COUNT	FRACTION
accept	61	7.22
hold	37	4.38
reject	12	1.42
signal_non_understanding	3	0.36
signal_understanding	7	0.83
positive_answer	145	17.16
negative_answer	29	3.43
no_answer_feedback	2	0.24
no_blf	544	64.38
other	5	0.59
TOTAL	845	100

Table B.15: Class statistics for blf in case of 1-grams of turns

CLASSIFIER	NaiveBayes	BayesNetK2-S-ENTROPY	J48	AttribSelect
MAX. ACCURACY	70.8876	71.1243	70.4142	
FEATURES	FEATURE SELECTION			
lenq	−	+	−	+
isContinue	−	+	−	−
startsWithWHExpr	−	+	−	+
endsWithQuestionMark	+	+	+	+
startsWithCanI	+	−	−	−
startsWithIWant	+	+	−	−
containsPositive	−	−	−	−
containsNegative	+	+	+	+
containsOkay	+	+	+	+
containsTemporalPrep	−	−	−	−
containsTemporalAdverb	+	−	−	−
containsDo	+	+	−	−

Table B.16: Results from blf-classification with window-size 1; the accuracy baseline is 64.38%.

VALUE	COUNT	FRACTION
statement	114	13.49
action_directive	218	25.8
open_option	0	0.0
query_if	68	8.05
query_ref	395	46.75
committing_speaker_future_action	0	0.0
conventional	26	3.08
no_flf	0	0.0
other	24	2.84
TOTAL	845	100

Table B.17: Class statistics for flf in case of 1-grams of turns

CLASSIFIER	NaiveBayes	BayesNetK2-S-ENTROPY	J48	AttribSelect
MAX. ACCURACY	62.2485	67.6923	67.574	
FEATURES	FEATURE SELECTION			
lenq	+	+	+	−
isContinue	+	+	+	−
startsWithWHExpr	+	+	+	+
endsWithQuestionMark	+	+	+	+
startsWithCanI	+	+	−	−
startsWithIWant	−	−	−	−
containsPositive	+	+	+	−
containsNegative	+	+	+	−
containsOkay	−	+	+	−
containsTemporalPrep	+	−	+	−
containsTemporalAdverb	−	−	−	−
containsDo	+	+	−	−

Table B.18: Results from flf-classification with window-size 1; the accuracy baseline is 46.75%.

CLASSIFIER	NaiveBayes	BayesNetK2-S-ENTROPY	J48	AttribSelect
MAX. ACCURACY	87.6923	87.8107	88.284	
FEATURES	FEATURE SELECTION			
blf-1	−	−	+	−
flf-1	+	+	+	+
lenq	+	−	−	−
isContinue	+	−	−	−
startsWithWHExpr	−	−	+	−
endsWithQuestionMark	−	+	+	−
startsWithCanI	−	−	−	−
startsWithIWant	−	+	+	−
containsPositive	−	−	−	−
containsNegative	+	+	+	−
containsOkay	−	−	−	−
containsTemporalPrep	−	−	−	−
containsTemporalAdverb	−	−	−	−
containsDo	−	+	−	−

Table B.19: Results from blf-classification with window-size 2; the accuracy baseline is 64.38%.

CLASSIFIER	NaiveBayes	BayesNetK2-S-ENTROPY	J48	AttribSelect
MAX. ACCURACY	68.8757	72.6627	73.7278	
FEATURES	FEATURE SELECTION			
blf-1	+	−	−	−
flf-1	+	+	+	+
lenq	−	+	+	−
isContinue	+	+	+	−
startsWithWHExpr	+	+	+	+
endsWithQuestionMark	+	+	+	+
startsWithCanI	+	+	−	−
startsWithIWant	+	−	−	−
containsPositive	−	+	−	−
containsNegative	−	+	+	−
containsOkay	+	+	+	−
containsTemporalPrep	+	−	−	−
containsTemporalAdverb	+	+	−	−
containsDo	+	+	+	−

Table B.20: Results from flf-classification with window-size 2; the accuracy baseline is 46.75%.

CLASSIFIER	NaiveBayes	BayesNetK2-S-ENTROPY	J48	AttribSelect
MAX. ACCURACY	70.4142	72.6627	73.7278	
FEATURES	FEATURE SELECTION			
blf-2	+	−	−	−
flf-2	+	−	−	−
blf-1	−	−	−	−
flf-1	+	+	+	+
lenq	+	+	+	−
isContinue	−	+	+	−
startsWithWHExpr	+	+	+	+
endsWithQuestionMark	+	+	+	+
startsWithCanI	+	+	−	−
startsWithIWant	+	−	−	−
containsPositive	−	+	−	−
containsNegative	−	+	+	−
containsOkay	−	+	+	−
containsTemporalPrep	−	−	−	−
containsTemporalAdverb	−	+	−	−
containsDo	+	+	+	−

Table B.21: Results from flf-classification with window-size 3; the accuracy baseline is 46.75%.

CLASSIFIER	NaiveBayes	BayesNetK2-S-ENTROPY	J48	AttribSelect
MAX. ACCURACY	87.6923	87.929	88.4024	
FEATURES	FEATURE SELECTION			
blf-2	−	+	+	−
flf-2	−	−	−	−
blf-1	−	−	+	−
flf-1	+	+	+	+
lenq	+	−	−	−
isContinue	+	+	−	−
startsWithWHExpr	−	−	+	−
endsWithQuestionMark	−	+	+	−
startsWithCanI	−	+	−	−
startsWithIWant	−	+	+	−
containsPositive	−	−	−	−
containsNegative	+	+	+	−
containsOkay	−	−	−	−
containsTemporalPrep	−	+	−	−
containsTemporalAdverb	−	−	−	−
containsDo	−	−	−	−

Table B.22: Results from blf-classification with window-size 3; the accuracy baseline is 64.38%.

CLASSIFIER	NaiveBayes	BayesNetK2-S-ENTROPY	J48	AttribSelect
MAX. ACCURACY	82.9586	83.0769	84.0237	
FEATURES	FEATURE SELECTION			
blf-1	−	+	+	−
flf-1	+	+	+	+

Table B.23: Results from blf-classification with window-size 2; the accuracy baseline is 64.38%.

CLASSIFIER	NaiveBayes	BayesNetK2-S-ENTROPY	J48	AttribSelect
MAX. ACCURACY	83.5503	83.5503	84.0237	
FEATURES	FEATURE SELECTION			
blf-2	+	+	−	−
flf-2	−	−	−	−
blf-1	−	−	+	−
flf-1	+	+	+	+

Table B.24: Results from blf-classification with window-size 3; the accuracy baseline is 64.38%.

CLASSIFIER	NaiveBayes	BayesNetK2-S-ENTROPY	J48	AttribSelect
MAX. ACCURACY	83.5503	83.5503	84.0237	
FEATURES	FEATURE SELECTION			
blf-3	−	−	−	−
flf-3	−	−	−	−
blf-2	+	+	−	−
flf-2	−	−	−	−
blf-1	−	−	+	−
flf-1	+	+	+	+

Table B.25: Results from blf-classification with window-size 4; the accuracy baseline is 64.38%.

CLASSIFIER	NaiveBayes	BayesNetK2-S-ENTROPY	J48	AttribSelect
MAX. ACCURACY	83.5503	83.5503	84.0237	
FEATURES	FEATURE SELECTION			
blf-4	–	–	–	–
flf-4	–	–	–	–
blf-3	–	–	–	–
flf-3	–	–	–	–
blf-2	+	+	–	–
flf-2	–	–	–	–
blf-1	–	–	+	–
flf-1	+	+	+	+

Table B.26: Results from blf-classification with window-size 5; the accuracy baseline is 64.38%.

CLASSIFIER	NaiveBayes	BayesNetK2-S-ENTROPY	J48	AttribSelect
MAX. ACCURACY	58.1065	58.1065	58.1065	
FEATURES	FEATURE SELECTION			
blf-1	–	–	–	+
flf-1	+	+	+	+

Table B.27: Results from flf-classification with window-size 2; the accuracy baseline is 46.75%.

CLASSIFIER	NaiveBayes	BayesNetK2-S-ENTROPY	J48	AttribSelect
MAX. ACCURACY	58.1065	60	58.9349	
FEATURES	FEATURE SELECTION			
blf-2	–	+	+	–
flf-2	–	+	–	–
blf-1	–	+	–	+
flf-1	+	+	+	+

Table B.28: Results from flf-classification with window-size 3; the accuracy baseline is 46.75%.

CLASSIFIER	NaiveBayes	BayesNetK2-S-ENTROPY	J48	AttribSelect
MAX. ACCURACY	58.1065	60	59.4083	
FEATURES	FEATURE SELECTION			
blf-3	–	–	+	–
flf-3	–	–	+	+
blf-2	–	+	+	–
flf-2	–	+	–	–
blf-1	–	+	–	+
flf-1	+	+	+	+

Table B.29: Results from flf-classification with window-size 4; the accuracy baseline is 46.75%.

CLASSIFIER	NaiveBayes	BayesNetK2-S-ENTROPY	J48	AttribSelect
MAX. ACCURACY	58.1065	60	59.4083	
FEATURES	FEATURE SELECTION			
blf-4	−	−	−	−
flf-4	−	−	−	−
blf-3	−	−	+	−
flf-3	−	−	+	+
blf-2	−	+	+	−
flf-2	−	+	−	−
blf-1	−	+	−	+
flf-1	+	+	+	+

Table B.30: Results from flf-classification with window-size 5; the accuracy baseline is 46.75%.

Bibliography

Alexandersson, J., Buschbeck-Wolf, B., Fujinami, T., Kipp, M., Koch, S., Maier, E., Reithinger, N., Schmitz, B., and Siegel, M. (1998). Dialogue acts in VERBMOBIL-2 second edition. VM-Report 226, DFKI GmbH.

Allen, J. (1987). *Natural Language Understanding*. Benjamin/Cummings, second edition.

Allen, J. and Core, M. (1997). Draft of DAMSL: Dialog Act Markup in Several Layers. Dagstuhl Workshop.

Allen, J. and Perrault, C. (1980). Analyzing intention in utterances. *Artificial Intelligence*, 15:143–178.

Allen, J. and Schubert, L. (1994). The TRAINS project: A case study in defining a conversational planning agent. Technical Report TR 532, URCSD.

Allen, J., Schubert, L., Ferguson, G., Heeman, P., Hwang, C., Kato, T., Light, M., Martin, N., Miller, B., Poesio, M., and Traum, D. (1995). The TRAINS project: A case study in building a conversational planning agent. *Journal of Experimental and Theoretical AI*, 7:7–48.

Allwood, J. (1976). *Linguistic Communication as Action and Cooperation*. PhD thesis, Göteborg University.

Andersen, S., Olesen, K., Jensen, F., and Jensen, F. (1989). HUGIN - a shell for building Bayesian belief universes for expert systems. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*, pages 1080–1085, Detroit, Michigan. URL: `http://www.hugin.dk`.

Austin, J. (1975). *How to Do Things with Words*. Harvard University Press, 2nd edition.

Black, W., Thompson, P., Funk, A., and Conroy, A. (2003). Learning to classify utterances in a task-oriented dialogue. In Jokinen, K., Gambäck, B., Black, W., Catizone, R., and Wilks, Y., editors, *Proceedings of the Workshop on Dialogue Systems: Interaction, Adaptation and Styles of Management*, pages 9–16, Budapest.

Bouckaert, R. (1993). Probabilistic network construction using the minimum description length principle. In *Symbolic and Quantitative Appraches to Reasoning and Uncertainty ECSQARU'93*, volume 747 of *Lecture Notes in Computer Science*, pages 41–48.

Bunt, H. (1995). Dynamic interpretation and dialogue theory. In Taylor, M., Bouwhuis, D. G., and Neel, F., editors, *The Structure of Multimodal Dialogue*, volume 2. John Benjamins, Amsterdam.

Bunt, H. (1996). Interaction management functions and context representation requirements. In Luperfoy, S., Nijholt, A., and van Zanten, G. V., editors, *Dialogue Management in Natural Language Systems*, pages 187–198. University of Twente.

Buntine, W. (1994). Operations for learning with graphical models. *Artificial Intelligence*, 2:159–225.

Carletta, J. (1996). Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254.

Carletta, J., Isard, A., Isard, S., Kowtko, J., Doherty-Sneddon, G., and Anderson, A. (1996). HCRC Dialogue Structure Coding Manual. Technical Report TR-82, HCRC, Edinburgh.

Cohen, P. and Perrault, C. (1979). Elements of a plan-based theory of speech acts. *Cognitive Science*, 3:177–212.

Cooper, G. (1990). The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405.

Cooper, G. and Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347.

Cozman, F. (1998). *JavaBayes Version 0.346: Bayesian Networks in Java: User Manual*. University of São Paulo. URL: `http://www.cs.cmu.edu/~javabayes/Home/`.

Dechter, R. (1996). Bucket elimination: A unifying framework for probabilistic inference. In Horvitz, E. and Jensen, F., editors, *XIIth Conference on Uncertainty in Artificial Intelligence*, pages 211–219, San Francisco. Morgan Kaufmann.

Dempster, A. (1968). A generalization of Bayesian inference. *Journal of the Royal Statistical Society*, 30:205–247.

Doyle, J. (1979). A truth maintenance system. *Artificial Intelligence*, 12(3):231–272.

Eugenio, B. D., Jordan, P., and Pylkkänen, L. (1998). The COCONUT project: Dialogue annotation manual. Technical Report 98-1, ISP.

Ferguson, G. and Allen, J. (1998). TRIPS: An intelligent integrated problem-solving assistant. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, pages 567–573, Madison, WI.

Gärdenfors, P. (1992). *Belief Revision*, volume 29 of *Cambridge tracts in theoretical computer science*. Cambridge University Press.

Gärdenfors, P. and Rott, H. (1995). Belief revision. In Gabbay, D. M., Hogger, C., and Robinson, J., editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 4, pages 35–132. Oxford Science Publications.

Grice, H. (1969). Utterer's meaning and intentions. *Philosophical Review*, 78:147–177.

Grice, H. (1991). *Studies in the Way of Words*. Harvard University Press.

Heckerman, D. (1995). A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research.

Heckerman, D. and Horvitz, E. (1998). Inferring informational goals from free-text queries: A Bayesian approach. In *Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 230–237, Madison, WI. Morgan Kaufmann.

Hofs, D., op den Akker, R., and Nijholt, A. (2003). A generic architecture and dialogue model for multimodal interaction. In *Proceedings of the First Nordic Symposium on Multimodal Communication*, Copenhagen. to appear.

Horvitz, E., Breese, J., Heckerman, D., Hovel, D., and Rommelse, K. (1998). The Lumière project: Bayesian user modeling for inferring the goals and needs of software users. In *Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 256–265, Madison, WI. Morgan Kaufmann.

Hsu, W., Guo, H., Perry, B., and Stilson, J. (2002). A permutation genetic algorithm for variable ordering in learning Bayesian networks from data. In Langdon, W., Cantú-Paz, E., Mathias, K., Roy, R., Davis, D., Poli, R., Balakrishnan, K., Honavar, V., Rudolph, G., Wegener, J., Bull, L., Potter, M., Schultz, A., Miller, J., Burke, E., and Jonoska, N., editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002)*, pages 383–390, New York. Morgan Kaufmann.

Isard, A. (2001). An XML architecture for the HCRC MapTask corpus. In *BI-DIALOG 2001—Proceedings of the 5th Workshop on Formal Semantics and Pragmatics of Dialogue*, pages 280–286, Bielefeld, Germany.

Jaynes, E. (1990). Probability theory as logic. In Fougere, P., editor, *Maximum Entropy and Bayesian Methods*, Dordrecht, Holland. Kluwer. revised version 1994.

Jaynes, E. (2003). *Probability Theory: The Logic of Science*. Cambridge University Press. edited by G.L. Bretthorst.

Jensen, F. (1996). *An Introduction to Bayesian Networks*. UCL Press, London.

Jensen, F., Olesen, K., and Andersen, S. (1990). An algebra of Bayesian belief universes for knowledge-based systems. *Networks*, 20:637–659.

Jurafsky, D., Bates, R., Coccaro, N., Martin, R., Meteer, M., Ries, K., Shriberg, E., Stolcke, A., Taylor, P., and Ess-Dykema, C. V. (1997). Automatic detection of discourse structure for speech recognition and understanding. In *Proc. IEEE Workshop on Speech Recognition and Understanding*, pages 88–95, Santa Barbara.

Jurafsky, D. and Martin, J. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall Series in Artificial Intelligence. Prentice Hall.

Katsuno, H. and Mendelzon, A. O. (1992). On the difference between updating a knowledge base and revising it. In Gärdenfors, P., editor, *Belief Revision*, pages 183–203. Cambridge University Press.

Keizer, S. (2001a). A Bayesian approach to dialogue act classification. In Kühnlein, P., Rieser, H., and Zeevat, H., editors, *BI-DIALOG 2001: Proceedings of the 5th Workshop on Formal Semantics and Pragmatics of Dialogue*, pages 210–218, Bielefeld, Germany.

Keizer, S. (2001b). Dialogue act modelling using Bayesian networks. In Striegnitz, K., editor, *Proceedings of the Sixth ESSLLI Student Session*, pages 143–153, Helsinki.

Keizer, S., op den Akker, R., and Nijholt, A. (2002). Dialogue act recognition with Bayesian networks for Dutch dialogues. In *Proceedings 3rd ACL/SIGdial Workshop on Discourse and Dialogue*, pages 88–94, Philadelphia, PA.

Kim, J. and Pearl, J. (1983). A computational model for combined causal and diagnostic reasoning in inference systems. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence (IJCAI-83)*, pages 190–193, Karlsruhe. Morgan Kaufman.

Kipp, M. (1998). The neural path to dialogue acts. In Prade, H., editor, *Proceedings of the 13th European Conference on Artificial Intelligence*, pages 175–179.

Klein, M., Bernsen, N., Davies, S., Dybkjær, L., Garrido, J., Kasch, H., Mengel, A., Pirrelli, V., Poesio, M., Quazza, S., and Soria, S. (1998). Supported coding schemes. MATE Deliverable D1.1. URL: `http://mate.nis.sdu.dk/about/D1.1`.

Kowtko, J., Isard, S., and Doherty, G. (1992). Conversational games within dialogue. Technical Report RP-31, HCRC, Edinburgh.

Lam, W. and Bacchus, F. (1994). Learning Bayesian belief networks: An approach based on the MDL principle. *Computational Intelligence*, 10(4):269–293.

Larsson, S. and Traum, D. (2000). Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering: Special Issue on Best Practice in Spoken Language Dialogue Systems*, 6(3–4):323–340.

Lauritzen, S. and Spiegelhalter, D. (1988). Local computation with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B, Statistical Methodology*, 50(2):157–224.

Lemon, O., Parikh, P., and Peters, S. (2002). Probabilistic dialogue modelling. In *Proceedings 3rd ACL/SIGdial Workshop on Discourse and Dialogue*, pages 125–128, Philadelphia, PA.

Luin, J. v., op den Akker, R., and Nijholt, A. (2001). A dialogue agent for navigation support in virtual reality. In Jacko, J. and Sears, A., editors, *ACM SIGCHI Conf. CHI 2001: Anyone. Anywhere*, pages 117–118, Seattle. Association for Computing Machinery.

McCarthy, J. (1980). Circumscription: a form of non-monotonic reasoning. *Artificial Intelligence*, 13(1–2):27–39.

McDermott, D. and Doyle, J. (1980). Nonmonotonic logic I. *Artificial Intelligence*, 13(1–2):41–72.

McTear, M. (2002). Spoken dialogue technology: Enabling the conversational user interface. *ACM Computing Surveys*, 34(1).

Mengel, A., Dybkjaer, L., Garrido, J., Heid, U., Klein, M., Pirrelli, V., Poesio, M., Quazza, S., Schiffrin, A., and Soria, C. (2000). MATE dialogue annotation guidelines. Deliverable D2.1. URL: `http://www.ims.uni-stuttgart.de/projekte/mate/mdag/`.

Mitchell, T. (1997). *Machine Learning*. Computer Science Series. McGraw-Hill.

Murphy, K. (2001). The Bayes Net Toolbox for Matlab. *Computing Science and Statistics*, 33. URL: `http://www.ai.mit.edu/~murphyk/Software/BNT/bnt.html`.

Nagata, M. and Morimoto, T. (1994). First steps towards statistical modeling of dialogue to predict the speech act type of the next utterance. *Speech Communication*, 15:193–203.

Neapolitan, R. (1990). *Probabilistic Reasoning in Expert Systems: Theory and Algorithms*. Wiley, New York.

Pearl, J. (1982). Reverend Bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-82*, pages 133–136, Pittsburgh, Pennsylvania. Morgan Kaufmann.

Pearl, J. (1986). Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29:241–288.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann. revised second printing 1997.

Pearl, J. (2000). *Causality: Models, Reasoning and Inference*. Cambridge University Press.

Perrault, C. and Allen, J. (1980). A plan-based analysis of indirect speech acts. *American Journal of Computational Linguistics*, 6(3–4):167–182.

Poesio, M. and Traum, D. (1997). Conversational actions and discourse situations. *Computational Intelligence*, 13(3):309–347.

Poesio, M. and Traum, D. (1998). Towards an Axiomatization of Dialogue Acts. In Hulstijn, J. and Nijholt, A., editors, *TwenDial'98: Formal Semantics and Pragmatics of Dialogue*, number 13 in TWLT.

Pulman, S. (1996). Conversational games, belief revision and Bayesian networks. In Landsbergen, J., Odijk, J., van Deemter, K., and van Zanten, G. V., editors, *Computational Linguistics in the Netherlands*. SRI Technical Report CRC-071.

Quinlan, J. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Reiter, R. (1980). A logic for default reasoning. *Artificial Intelligence*, 13(1–2):81–132.

Reithinger, N. and Klesen, M. (1997). Dialogue act classification using language models. In Kokkinakis, G., Fakotakis, N., and Dermatas, E., editors, *Proceedings of the 5th European Conference on Speech Communication and Technology EuroSpeech-97*, volume 4, pages 2235–2238, Rhodes.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14:465–471.

Russel, S. J. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice Hall, New Jersey.

Samuel, K., Carberry, S., and Vijay-Shanker, K. (1998). Dialogue act tagging with transformation-based learning. In *Proceedings of the 36th Annual Meeting of the ACL and the 17th International Conference on Computational Linguistics (ACL-COLING)*, pages 1150–1156.

Searle, J. (1969). *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press.

Searle, J. (1975). Indirect speech acts. In Cole, P. and Morgan, J., editors, *Syntax and Semantics Vol. 3: Speech Acts*, pages 59–82. New York Academic Press.

Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press.

Shortliffe, E. (1976). *Computer-Based Medical Consultations: MYCIN*. Elsevier, North-Holland.

Stokhof, M. (2000). *Taal en Betekenis: Een Inleiding in de Taalfilosofie*. Boom, Amsterdam.

Stolcke, A., Coccaro, N., Bates, R., Taylor, P., Ess-Dykema, C. V., Ries, K., Shriberg, E., Jurafsky, D., Martin, R., and Meteer, M. (2000). Dialogue act modelling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–374.

Traum, D. (2000). 20 questions on dialogue act taxonomies. *Journal of Semantics*, 17(1):7–30.

Traum, D. and Hinkelman, E. (1992). Conversation acts in task-oriented spoken dialogue. *Computational Intelligence*, 3(8):575–599. Special Issue on Non-literal Language.

Trindi (2001). The trindi book. URL: `http://www.ling.gu.se/projekt/trindi`.

W3C (1997-2003). Extensible markup language (XML). URL: `http://www.w3c.org/XML/`.

Weizenbaum, J. (1966). ELIZA – a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.

Winograd, T. (1972). Understanding natural language. *Cognitive Psychology*, 3(1):1–191.

Witten, I. and Frank, E. (1999). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann. URL: `http://www.cs.waikato.ac.nz/~ml/weka/`.

Wittgenstein, L. (1958). *Philosophical Investigations*. Blackwell Publishers, 2nd edition. translated by G.E.M. Anscombe.

Wright, H. (1998). Automatic utterance type detection using suprasegmental features. In *Int. Conf. on Spoken Language Processing*, volume 4, pages 1403–1406, Sydney.

Wright, H., Poesio, M., and Isard, S. (1999). Using high level dialogue information for dialogue act recognition using prosodic features. In *ESCA Tutorial and Research Workshop on Dialogue and Prosody*, pages 139–143.

XCES (2000-2002). XCES corpus encoding standard for XML. URL: `http://www.xml-ces.org/`.

Zhang, N. and Poole, D. (1996). Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research*, 5:301–328.

Samenvatting

Dialoogsystemen zijn computersystemen waarbij de interactie tussen gebruiker en systeem verloopt door middel van natuurlijke taal, bijvoorbeeld in het Nederlands of het Engels. De gebruiker kan zich bedienen van gesproken taal (via een microfoon) of kan zinnen typen op een toetsenbord. Het systeem kan zo gebruikersvriendelijker worden, doordat gebruikers kunnen omgaan met het systeem op een wijze die natuurlijker is dan conventionele interactie via muis en toetsenbord. Zo kan een gebruiker door middel van een dialoog met het systeem bijvoorbeeld informatie vergaren die is opgeslagen in een database, bijvoorbeeld over reistijden in het openbaar vervoer of over theatervoorstellingen die op het programma staan. Wellicht kan daarbij ook sprake zijn van een transactie, zoals het maken van een reservering. Een gebruiker zou zich ook in een dialoog kunnen laten assisteren door het systeem bij het uitvoeren van een bepaalde taak, bijvoorbeeld bij het gebruiken van bepaalde kantoorsoftware of bij het vinden van locaties in een virtuele omgeving.

Wil zo'n dialoogsysteem goed functioneren, dan moet het adequaat kunnen reageren op uitingen van de gebruiker. Het moet de bedoeling die een gebruiker met zijn uiting had, ofwel diens intenties, trachten te achterhalen, en vervolgens daarop een reactie bepalen die in de bewuste dialoogsituatie passend is. Hierbij spelen zowel de uitvoering van de onderliggende taak (bijvoorbeeld het geven van informatie) als het volgen van zekere stilzwijgende sociale conventies die in het communicatieve proces lijken te bestaan, een rol. In de meeste geavanceerde theorieën over dialoogmodellering en ook in het ontwerp van geavanceerde dialoogsystemen worden natuurlijketaaluitingen gezien als een vorm van communicatief handelen, en als zodanig geïnterpreteerd in termen van *taalhandelingen* of *dialoogacten*. Elke uiting wordt hierbij geïnterpreteerd als bijvoorbeeld een verzoek, een vraag, een bewering, een toestemming, een correctie, een suggestie, enzovoort. Het is dus voor een dialoogsysteem zaak om op grond van de informatie die hij heeft over de uiting, de dialoogtoestand, de gebruiker en de onderliggende taak, te bepalen wat voor soort taalhandeling de gebruiker verrichtte toen hij de uiting produceerde.

In dit proefschrift wordt uitvoerig ingegaan op het probleem dat een dialoogsysteem bij het verwerken van een natuurlijketaaluiting nooit absoluut zeker kan zijn van zijn interpretatie. Enerzijds komt dat doordat er informatie verloren gaat vanaf het moment dat een spreker een uiting voortbrengt tot het moment dat een hoorder deze uiting percipiëert: de spraakverwerking levert slechts partiële informatie. Anderzijds worden bij elke poging tot het modelleren van natuurlijketaaldialoog abstracties geïntroduceerd, waardoor mogelijk relevante aspecten onbelicht blijven of over één kam geschoren worden. Zeker als het gaat om interactie door middel van natuurlijke taal zullen mensen zich vrijwel zeker niet volledig houden aan de regels die in het systeem zijn vastgelegd.

Een dialoogsysteem zal dus in het algemeen geconfronteerd worden met *onzekerheid*: hij kan nooit absolute zekerheid hebben over een interpretatie van een gegeven uiting en moet daarom ofwel via de dialoog meer informatie vergaren om zijn onzekerheid te verminderen, ofwel uitgaan van de interpretatie die hij op grond van de informatie die hij heeft het meest waarschijnlijk acht. Hij zou later moeten kunnen terugkomen op zijn eerdere aannames, mocht op grond van nieuwe informatie blijken dat deze aannames niet meer zo waarschijnlijk zijn.

Als een oplossing voor dit probleem is in het onderzoek gekeken naar de mogelijkheid om gebruik te maken van Bayesiaanse netwerken. Dit zijn kansmodellen waarin voorwaardelijke onafhankelijkheden tussen de variabelen expliciet zijn gespecificeerd, door middel van een gerichte graaf. De punten in die graaf representeren elk een variabele in het kansmodel en de pijlen tussen de punten bepalen de (on-)afhankelijkheden. Op grond van informatie, die direct kan worden vertaald in waardetoekenningen aan bepaalde variabelen in het model, kan de a posteriori kansverdeling over de overige variabelen berekend worden en kan tevens de meest waarschijnlijke combinatie van waarden bepaald worden. Bayesiaanse netwerken kunnen worden geconstrueerd op grond van expert-kennis, of uit ruwe data met behulp van statistische technieken. De beide methoden kunnen echter ook worden gecombineerd: zo kunnen de onafhankelijkheden door een expert worden bepaald en vervolgens de vereiste kansverdelingen automatisch uit data worden geïnduceerd.

De experimenten die zijn gedaan, hebben betrekking op de taak van dialoogactclassificatie. Op grond van een aantal vaste kenmerken van natuurlijketaaluitingen, is het zaak om te bepalen welk type taalhandeling, het *dialoogacttype*, er bij die uiting hoort. Daarbij moet worden gekozen uit een verzameling van dialoogacttypen die voor het systeem van belang zijn om te kunnen onderscheiden. In de experimenten zijn verschillende modellen voor deze taak, dat is, classificatoren, geconstrueerd uit data in een geannoteerd dialoogcorpus. Het gebruikte corpus bestaat uit Nederlandstalige dialogen op het gebied van theatervoorstellingen, waarin de gebruiker informatie kan inwinnen over voorstellingen en ook reserveringen kan maken. De uitingen in het corpus zijn niet gesproken, maar tekstgebaseerd. In het systeem van dialoogacttypen dat is gebruikt voor de annotatie van het corpus is onderscheid gemaakt tussen twee soorten dialoogacten: de *forward-looking functions*, die de invloed van een uiting op het verdere verloop van de dialoog karakteriseren, en *backward-looking functions*, die de manier waarop een uiting terugverwijst naar een eerder gedeelte van de dialoog karakteriseren. Voor beide soorten dialoogacten zijn aparte classificatoren geconstrueerd.

Het doel van de experimenten was tweeledig. Ten eerste is gekeken hoe Bayesiaanse netwerken presteren in vergelijking met andere modellen en technieken voor machinaal leren als het gaat om dialoogactclassificatie. Hierbij is gekeken naar de nauwkeurigheid in het classificeren, gemeten in percentages, ofwel de *accuracy* van de classificatoren. Bayesiaanse netwerken blijken niet significant beter of minder te presteren dan de andere twee onderzochte technieken, Naive Bayes en Decision tree classificatoren. Ten tweede is onderzoek gedaan naar welke kenmerken, ofwel *features*, van een uiting in context het meest informatief zijn voor het bepalen van het dialoogacttype; dit kunnen kenmerken van de uiting zelf zijn, die zijn afgeleid uit de uitvoer van een POS-tagger, maar ook kenmerken van de context, in de vorm van dialoogacttypen van voorgaande uitingen. Uitgaande van een vaste verzameling kenmerken zijn voor alle mogelijke deelverzamelingen van kenmerken classificatoren geëvalueerd. Gebleken is dat de dialoogacttypen van één direct aan de huidige uiting voorafgaande uiting verreweg de meeste informatie geven voor wat betreft de kenmerken van de context. Verder terugkijken levert geen significante verbeteringen op in de prestaties van de classificatoren. Ook wordt duidelijk dat de kenmerken van de uiting meer informatief zijn bij het classificeren van de forward-looking functions dan bij het classificeren van backward-looking functions, terwijl dit andersom is voor kenmerken van de context.

Naast het bepalen van de meest waarschijnlijke dialoogact voor een gegeven gebruikersuiting, kunnen Bayesiaanse netwerken ook voor andere taken, die in de interpretatie van natuurlijketaaluitingen van belang zijn, worden gebruikt. Wat echter wordt voorgesteld in het proefschrift, is om Bayesiaanse netwerken te gebruiken in een breder raamwerk, waarbij de verschillende relevante aspecten en taken in de analyse van uitingen in de context van een dialoog geïntegreerd worden in één model. Deze aspecten kunnen variëren van de uitvoer van de spraakherkenner, syntactische informatie, en informatie over dialoogacten, tot kenmerken van non-verbale communicatie. Een dergelijk model is modulair door de voorwaardelijke onafhankelijkheden die gespecificeerd kunnen worden; bovendien kan nog gekeken worden naar de mogelijkheid om zogenaamde *dynamische* Bayesiaanse netwerken toe te passen, waarmee de verandering van de dialoogtoestand onder invloed van een dialoogact beter gemodelleerd kan worden.

```
SIKS Dissertatiereeks

====
1998
====

1998-1  Johan van den Akker (CWI)
        DEGAS - An Active, Temporal Database of Autonomous Objects

1998-2  Floris Wiesman (UM)
        Information Retrieval by Graphically Browsing Meta-Information

1998-3  Ans Steuten (TUD)
        A Contribution to the Linguistic Analysis of Business Conversations
        within the Language/Action Perspective

1998-4  Dennis Breuker (UM)
        Memory versus Search in Games

1998-5  E.W.Oskamp (RUL)
        Computerondersteuning bij Straftoemeting

====
1999
====

1999-1  Mark Sloof (VU)
        Physiology of Quality Change Modelling; Automated modelling of
        Quality Change of Agricultural Products

1999-2  Rob Potharst (EUR)
        Classification using decision trees and neural nets

1999-3  Don Beal (UM)
        The Nature of Minimax Search

1999-4  Jacques Penders (UM)
        The practical Art of Moving Physical Objects

1999-5  Aldo de Moor (KUB)
        Empowering Communities: A Method for the Legitimate User-Driven
        Specification of Network Information Systems

1999-6  Niek J.E. Wijngaards (VU)
        Re-design of compositional systems

1999-7  David Spelt (UT)
        Verification support for object database design

1999-8  Jacques H.J. Lenting (UM)
        Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism
        for Discrete Reallocation.

====
2000
====

2000-1  Frank Niessink (VU)
        Perspectives on Improving Software Maintenance

2000-2  Koen Holtman (TUE)
        Prototyping of CMS Storage Management

2000-3  Carolien M.T. Metselaar (UVA)
        Sociaal-organisatorische gevolgen van kennistechnologie;
        een procesbenadering en actorperspectief

2000-4  Geert de Haan (VU)
```

ETAG, A Formal Model of Competence Knowledge for User Interface Design

2000-5 Ruud van der Pol (UM)
Knowledge-based Query Formulation in Information Retrieval

2000-6 Rogier van Eijk (UU)
Programming Languages for Agent Communication

2000-7 Niels Peek (UU)
Decision-theoretic Planning of Clinical Patient Management

2000-8 Veerle Coupe (EUR)
Sensitivity Analysis of Decision-Theoretic Networks

2000-9 Florian Waas (CWI)
Principles of Probabilistic Query Optimization

2000-10 Niels Nes (CWI)
Image Database Management System Design Considerations,
Algorithms and Architecture

2000-11 Jonas Karlsson (CWI)
Scalable Distributed Data Structures for Database Management

====
2001
====

2001-1 Silja Renooij (UU)
Qualitative Approaches to Quantifying Probabilistic Networks

2001-2 Koen Hindriks (UU)
Agent Programming Languages: Programming with Mental Models

2001-3 Maarten van Someren (UvA)
Learning as problem solving

2001-4 Evgueni Smirnov (UM)
Conjunctive and Disjunctive Version Spaces with Instance-Based
Boundary Sets

2001-5 Jacco van Ossenbruggen (VU)
Processing Structured Hypermedia: A Matter of Style

2001-6 Martijn van Welie (VU)
Task-based User Interface Design

2001-7 Bastiaan Schonhage (VU)
Diva: Architectural Perspectives on Information Visualization

2001-8 Pascal van Eck (VU)
A Compositional Semantic Structure for Multi-Agent Systems Dynamics.

2001-9 Pieter Jan ’t Hoen (RUL)
Towards Distributed Development of Large Object-Oriented Models,
Views of Packages as Classes

2001-10 Maarten Sierhuis (UvA)
Modeling and Simulating Work Practice
BRAHMS: a multiagent modeling and simulation language for
work practice analysis and design

2001-11 Tom M. van Engers (VUA)
Knowledge Management:
The Role of Mental Models in Business Systems Design

```
====
2002
====

2002-01 Nico Lassing (VU)
        Architecture-Level Modifiability Analysis

2002-02 Roelof van Zwol (UT)
        Modelling and searching web-based document collections

2002-03 Henk Ernst Blok (UT)
        Database Optimization Aspects for Information Retrieval

2002-04 Juan Roberto Castelo Valdueza (UU)
        The Discrete Acyclic Digraph Markov Model in Data Mining

2002-05 Radu Serban (VU)
        The Private Cyberspace Modeling Electronic
        Environments inhabited by Privacy-concerned Agents

2002-06 Laurens Mommers (UL)
        Applied legal epistemology; Building a knowledge-based ontology of
        the legal domain

2002-07 Peter Boncz (CWI)
        Monet: A Next-Generation DBMS Kernel For Query-Intensive
        Applications

2002-08 Jaap Gordijn (VU)
        Value Based Requirements Engineering: Exploring Innovative
        E-Commerce Ideas

2002-09 Willem-Jan van den Heuvel(KUB)
        Integrating Modern Business Applications with Objectified Legacy
        Systems

2002-10 Brian Sheppard (UM)
        Towards Perfect Play of Scrabble

2002-11 Wouter C.A. Wijngaards (VU)
        Agent Based Modelling of Dynamics: Biological and
        Organisational Applications

2002-12 Albrecht Schmidt (Uva)
        Processing XML in Database Systems

2002-13 Hongjing Wu (TUE)
        A Reference Architecture for Adaptive Hypermedia Applications

2002-14 Wieke de Vries (UU)
        Agent Interaction: Abstract Approaches to Modelling, Programming and
        Verifying Multi-Agent Systems

2002-15 Rik Eshuis (UT)
        Semantics and Verification of UML Activity Diagrams for Workflow
        Modelling

2002-16 Pieter van Langen (VU)
        The Anatomy of Design: Foundations, Models and Applications

2002-17 Stefan Manegold (UVA)
        Understanding, Modeling, and Improving Main-Memory Database Performance

====
2003
====

2003-01 Heiner Stuckenschmidt (VU)
```

Ontology-Based Information Sharing in Weakly Structured Environments

2003-02 Jan Broersen (VU)
Modal Action Logics for Reasoning About Reactive Systems

2003-03 Martijn Schuemie (TUD)
Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy

2003-04 Milan Petkovic (UT)
Content-Based Video Retrieval Supported by Database Technology

2003-05 Jos Lehmann (UVA)
Causation in Artificial Intelligence and Law - A modelling approach

2003-06 Boris van Schooten (UT)
Development and specification of virtual environments

2003-07 Machiel Jansen (UvA)
Formal Explorations of Knowledge Intensive Tasks

2003-08 Yongping Ran (UM)
Repair Based Scheduling

2003-09 Rens Kortmann (UM)
The resolution of visually guided behaviour

2003-10 Andreas Lincke (UvT)
Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture

2003-11 Simon Keizer (UT)
Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks