Dialogue act recognition and prediction

Explorations in computational dialogue modelling

Jeroen Geertzen
Dialogue act recognition and prediction

Explorations in computational dialogue modelling

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Tilburg,
op gezag van de rector magnificus,
prof.dr. Ph. Eijlander,
in het openbaar te verdedigen
ten overstaan van een door het college voor
promoties aangewezen commissie
in de aula van de Universiteit
op woensdag 11 februari 2009 om 14.15 uur

door

Jeroen Geertzen

geboren op 25 april 1978 te Dongen
The work in this thesis is one of the results of almost five years Ph.D. research at the Department of Communication and Information Sciences of Tilburg University. The motivation for pursuing a Ph.D. followed from of a graduate programme at the same university: “Taal en Kunstmatige Intelligente”\textsuperscript{1}. Already from the very beginning of this unique programme, which —unfortunately— does not exist anymore, I got enthusiastic about several topics in computational linguistics and artificial intelligence, and when the opportunity arose to carry out research in this direction, I did not hesitate.

During the recent years of research, I am grateful to have met many persons who have helped me along the way, and presenting this thesis would not have been possible without them. In the first place, I would like to express my gratitude to Harry Bunt, my promotor, for his trust in me and for having given me the opportunity to pursue a doctorate. I thank Jacques Terken, my co-promotor, for his feedback and for making me feel welcome each time I visited the University of Technology in Eindhoven. I thank both of them for the daunting task of proofreading and commenting on earlier versions of this work.

A substantial part of the studies presented in this thesis is based on joint work with colleagues. I would like to thank Pirokska Lendvai for a pleasant and constructive collaboration on the machine learning of dialogue act tagging. This joint work forms the basis of Section 4.3, and was helped by insightful discussions with Antal van den Bosch, Sander Canisius, and Erik Tjong Kim Sang. Also thanks to the undergraduate students, who helped in obtaining the dialogue annotations for inter-annotator agreement analysis, described in Chapter 3. I thank Volha Petukhova for the collaboration on comparing the machine learning performance of dialogue act classification for different data sets, and for the annotations needed to determine inter-annotator agreement. I also would like to thank Elizabeth Shriberg for sharing the ICSI-MRDA data set, Johanneke Caspers for sharing her Dutch Maptask corpus, and Mary Swift, Joel Tetreault, and Amanda Stent for recovering and providing the Monroe audio data.

Thanks are due to the members of the dialogue group: Harry Bunt, Hans van Dam,\textsuperscript{1}En.: Language and Artificial Intelligence
Yann Girard, Geke Hootsen, Simon Keizer, Roser Morante, Mandy Schiffrin, Barbara Suijkerbuijk, Ielka van der Sluis, and Rintse van der Werf. The dialogue meetings offered an excellent opportunity for discussing dialogue behaviour and literature, for doing data analysis, and for getting feedback on test talks.

During my stay at Tilburg University, I always enjoyed the atmosphere of the university and the interaction with my colleagues: Peter Berck, Toine Bogers, Antal van den Bosch, Harry Bunt, Bertjan Busser, Sander Canisius, Rein Cozijn, Federico Divina, Yann Girard, Iris Hendrickx, Charlotte van Hooijdonk, Geke Hootsen, Steve Hunt, Simon Keizer, Emiel Krahmer, Piroska Lendvai, Erwin Marsi, Roser Morante, Reinhard Muskens, Hans Paijmans, Volha Petukhova, Martin Reynaert, Erik Tjong Kim Sang, Mandy Schiffrin, Barbara Suijkerbuijk, Ko van der Sloot, Caroline Sporleder, Herman Stehouwer, Marc Swerts, Elias Thijsse, Marieke van Erp, Rintse van der Werf, Menno van Zaane, Tanja van Zaane, Paul Vogt, Sander Wubben, and the other people of the Faculty of Arts.

I thank Sander Canisius, Yann Girard, and Volha Petukhova for their pleasant company while sharing an office room with me. I owe special thanks to those colleagues and friends whom I spend quite some enjoyable time with: Menno van Zaane, Yann Girard, and Herman Stehouwer. I found Menno's enthusiasm in research refreshing and contagious, and I am grateful for the opportunity he offered me to spend some months carrying out research in Sydney, Australia. Thanks to our early geocaching adventures, I developed a slight obsession for geographical navigation. I enjoyed the many philosophical discussions I had with Yann, often accompanied with excellent Belgian beers. Our summer juggling sessions in the ‘Oude Wariande’ were excellent distractions from reading papers and working behind a computer screen. Herman Stehouwer turned out to be a worthy collaborator in improving coffee drinking standards, and while avoiding the public coffee machine we started to brew Italian espresso using freshly ground coffee beans and a stove-top coffee maker. This resulted in a daily afternoon coffee break session with other colleagues.

During the public defence of this thesis, I will not be standing entirely alone in front of the committee: I am happy to have Roser Morante and Volha Petukhova as paranymphs.

I thank my friends and my family for their interest and support. Above all, I thank my parents, who always supported and encouraged me in pursuing my interests. My deepest gratitude goes to my dearest friend and partner, Francesca, for the wonderful feelings of joy she brought into my life: Francesca, grazie per essere presente nella mia vita!

Jeroen Geertzen

Tilburg, January 2009
# Contents

## 1 Introduction
1.1 Motivation ........................................ 1  
    1.1.1 Recognition .................................. 3  
    1.1.2 Prediction .................................. 4  
1.2 Research questions ................................ 6  
1.3 Approach ........................................ 7  
1.4 Limitations ...................................... 7  
1.5 Thesis outline ................................... 8  

## 2 Modelling dialogue
2.1 Introduction ...................................... 9  
2.2 Speech act theory ................................ 10  
    2.2.1 Austin & Searle ................................. 10  
    2.2.2 Issues ........................................ 12  
2.3 Dialogue acts ..................................... 13  
    2.3.1 From speech acts to dialogue acts .................. 13  
    2.3.2 Dialogue acts in the context of collaborative joint activity 13  
    2.3.3 From dialogue system act sets to systematic theoretical accounts 15  
    2.3.4 Conversational Act Theory ....................... 17  
    2.3.5 Dynamic Interpretation Theory .................... 18  
    2.3.6 DAMSL taxonomy ............................... 22  
    2.3.7 DIT taxonomy ................................ 24  
2.4 Practical models of dialogue ...................... 30  
    2.4.1 Frame-based and finite state based .................. 31  
    2.4.2 Dialogue grammars .............................. 31  
    2.4.3 Plan-based approaches ........................... 33  
    2.4.4 Conversational Game Theory ...................... 34  
    2.4.5 Information-state update approaches ............... 34  
    2.4.6 Collaborative-agent based approaches ............... 35
### 6.3.2 Dialogue act recognition
- 136

### 6.3.3 Dialogue act prediction
- 136

### 6.4 Perspectives & future research
- 136
  - 6.4.1 Dialogue act scheme evaluation
    - 136
  - 6.4.2 Dialogue act recognition
    - 137
  - 6.4.3 Dialogue act prediction
    - 138
  - 6.4.4 From dialogue acts to states
    - 138

### A Guidelines for applying DIT dialogue act tags
- 141

### Samenvatting
- 167

### TiCC Ph.D. series
- 169
Chapter 1

Introduction

1.1 Motivation

Dialogue modelling in the broadest sense can be defined as describing, formalising, and explaining the dynamics of dialogue. A motivation for modelling dialogue is the construction of a theory of dialogue that answers fundamental questions such as: What enables an agent to participate in dialogue? How is linguistic behaviour expressed and interpreted and what kind of information plays a role there? In which way is information presented, and how can structure in dialogue be explained?

Apart from theoretical motivations, dialogue modelling can also be motivated as prerequisite for designing spoken dialogue systems: computer systems that interact with humans using spoken natural language. Such systems allow users to interact with computer-based applications in a natural and interactive way. Spoken dialogue systems, in turn, can be used to test and refine theories of dialogue: a theory can be used to implement a dialogue system, and human-machine interaction with this dialogue system can be studied to evaluate the theory.

For an agent (human or machine) that participates in dialogue, three activities are essential. First, the agent needs to interpret what other participants mean by what they say. Second, he should determine what actions to perform and when to perform them. This activity is usually called dialogue management. Third, he should determine a linguistic or other form for the actions and generate them. The research that is presented in this thesis relates to the former two activities and addresses the following general questions: “How can a machine detect meaning in what interlocutors say?”; and “How can a machine determine what appropriate response can or should be given to continue the dialogue?”.

The question of what meaning is and how it should be described is an important issue in the philosophy of language. Meaning in communication originates by group-wise interaction of agents with a shared environment that does not have meaning by itself. Likewise, the meaning of a spoken utterance in dialogue depends on the particular context it occurs in. In this respect, an important contribution in describing spoken
utterances based on their use is the conception that the purpose of an utterance in a certain context is to express a particular action, what is called a *speech act* [Austin, 1962].

An example of a speech act is the offering of help in the utterance “How may I help you?”; another example is the apology in the utterance “Sorry, I misunderstood”. Traditionally, speech acts concern single utterances, but in dialogue there are all kind of actions which have specific functions for participating in dialogue. As dialogue is a joint activity, several specific aspects need to be coordinated, such as the structuring of contributions, the change of topics, the taking of turns (determining who speaks next), and social conventions. To describe meanings of utterances in dialogue, the concept of *dialogue act* has been introduced. In analogy to speech acts expressing a particular action in single utterances, dialogue acts are sometimes loosely defined as entities expressing a particular action in the context of dialogue. More formally, a dialogue can be defined as “a unit in the semantic description of communicative behaviour produced by a sender and directed at an addressee, specifying how the behaviour is intended to influence the context through understanding of the behaviour” [Bunt, 2005a], and two different fundamental aspects of action can be distinguished: the *semantic content* and the communicative function. The semantic content describes what the utterance is about; the communicative function indicates the purpose that the utterance has in the dialogue.

An important aspect of dialogue acts is that of multifunctionality: a single utterance can express multiple dialogue acts, addressing different aspects of communication at the same time. For example, when looking at the following consecutive utterances by participants A and B: A: “...We should start the sales now. Moreover...” B: “Why should we start now?” where B interrupts A, the dialogue act TURN GRABBING can be assigned to B’s utterance; at the same time that B is managing the communication by interrupting, he is also asking a QUESTION.

Several schemes of dialogue acts have been proposed for both theoretical and practical purposes, and some of them are part of a theory of conversation, most notably DAMSL [Allen and Core, 1997] and DIT++ [Bunt, 2005a]. Both take characteristics such as the multifunctionality of utterances explicitly into account. DAMSL aims to facilitate dialogue act comparison across corpora, and provides a generic set of dialogue acts that can be used as supertypes in specific-purpose tagsets. DAMSL has been evaluated and used in a range of studies. DIT++ also aims to provide a generic dialogue act tagset, but differs from DAMSL in several respects. Functions in DIT++ are more consistently defined by the intention conveyed. For instance, where DAMSL has the act INFO-REQUEST as Forward Looking Function, being defined as “…any utterance that creates an obligation for the hearer to provide information, using any form of communication…”, DIT++ defines Information-seeking functions as having the basic precondition that “the speaker wants to know the truth of a given proposition”. Functions in DIT++ are also grouped in well-defined dimensions, each of which addresses

1The term ‘dialogue act’ was introduced by Bunt [1979] as the smallest functional unit of dialogue.
a different aspect of communication. For instance, where DAMSL does not identify different kind of social obligations and feedback regarding different levels of linguistic processing (PERCEPTION, INTERPRETATION, EVALUATION, etc...), DIT++ groups these actions in dimensions defined by the aspect of dialogue. 2

When we consider the dialogue acts that are performed by an utterance to be the meaning that is intentionally conveyed, modelling dialogue at the level of dialogue acts can be depicted as in Figure 1.1 and involves first the recognition of dialogue acts to derive the meaning of the utterances. Sequences of dialogue acts can be used to predict dialogue acts that may be performed as appropriate subsequent contributions in a dialogue. The two central questions, how machines can detect meaning and how machines can participate in dialogue, can be stated more precisely: “How can dialogue acts be recognised automatically?”, and “How can dialogue acts be predicted automatically?”. These questions are addressed in the chapters four and five, respectively.

1.1.1 Recognition

Recognising dialogue acts directly from sensory signals such as sound wave recordings is difficult. In finding out what kind of linguistic information could potentially be helpful (and to what extent), various kinds of linguistic features and abstractions from several levels of linguistic processing (acoustics, phonetics, prosody, syntax, et cetera) can be used (see Figure 1.2). Examples of linguistic features are intonation, syntactic construction, and the use of domain-specific words.

Much of the research on dialogue act recognition has focussed on determining the dialogue act(s) expressed by the utterances in dialogue. This task concentrates on the classification of dialogue acts and assumes that the spans of communication, the utterances, are known beforehand. A fully automated act recognition process, however, requires both locating the dialogue act (segmentation) and determining the correct dialogue act type (classification). An important question that can be raised here is how

---

2 A detailed overview of both schemes is provided in 2
1.1. MOTIVATION

Dialogue is best segmented into utterances, manually or automatically, taking into account the multifunctionality that is found in dialogue behaviour.

Dialogue act recognition is commonly viewed as segmentation followed by classification. Segments are, however, difficult to define independently of the interpretation of linguistic behaviour. Therefore it can be argued that dialogue act recognition is rather a unitary process that should not be subdivided into segmentation and classification, as to segment implies to identify borders according to the very nature of the span that is being delimited. It is the question if this assumption is confirmed by dialogue act recognition experiments.

1.1.2 Prediction

Obtaining the meaning of utterances by recognition of dialogue acts as outlined in the previous section does not directly address the question of what to do with dialogue acts once they are successfully recognised. To place dialogue acts into context, dialogue needs to be conceived as a process in which interlocutors proceed from one state of a dialogue to the next by performing dialogue acts. The prediction of the dialogue acts that are likely to be produced next can be based on the dialogue history of previously produced dialogue acts. This is depicted in Figure 1.3.

Figure 1.2: Dialogue act recognition.

Figure 1.3: Dialogue act prediction.
A simple and straightforward way to predict dialogue acts would be to consider the sequence of dialogue acts a stochastic process with a Markov property: the probability of a future event depends on a finite number of events preceding it. This assumption is used in \textit{n}-gram language models, in which the probability of the last event of a sequence of \textit{n} events depends on the probability of the subsequence of the previous \textit{n} – 1 events. Such models have been applied frequently in dialogue act prediction (see e.g. [Stolcke et al., 2000]) as they are robust and often give acceptable performance. A disadvantage is that \textit{n}-grams are limited by their local nature: regularities that can be found on a more global level cannot be captured.

An approach that aims to predict dialogue acts while also considering global objectives, such as task completion and user satisfaction, is to automatically (using reinforcement learning [Sutton and Barto, 1998]) derive a decision model that prescribes what actions should be taken in various dialogue states to maximise global user-satisfaction and task-completion. This approach has gained increasing attention over the recent years. Alternatively and complementary, the disadvantage of the local nature of \textit{n}-grams could be addressed by looking for more global structures that are characteristic to dialogue.

Much of the existing work in dialogue act prediction is actually rather dialogue act type prediction: only the dialogue act types are predicted, and the propositional content of the dialogue act is ignored. Prediction with both dialogue act type and propositional content come closer to full dialogue management, but would also be considerably more difficult.
1.2 Research questions

The previous discussion gives rise to the formulation of the research questions that this dissertation aims to answer. The first central question is *How and how well can dialogue acts be recognised automatically*; a second question is *How and how well can sequences of successfully recognised dialogue acts be used to predict future dialogue acts*. 

Dialogue acts as concepts are in essence based on speech act theory, but there are various ways to define acts and the exact role dialogue acts have in the whole activity of participating in a dialogue. Since the DIT++ taxonomy defines communicative functions in terms of intended effects, features a clear conception of multidimensionality, and has not yet been systematically applied in automatic dialogue act recognition and prediction, it would seem interesting to focus on the use of this taxonomy. In this context, the following two related research questions are addressed:

1. How reliably can the communicative functions in a multidimensional scheme, such as the DIT++ taxonomy, be applied and how can this be measured accurately, taken into account that the taxonomy features hierarchical pragramsemantic relations between functions?

2. In estimating the reliability of using an annotation scheme, is it useful to use different kinds of annotators?

The main research question concerning dialogue act recognition can be decomposed into the following questions:

3. How is communicative behaviour best segmented and specified accurately, while taking into account the multifunctionality that is explicitly modelled in multidimensional taxonomies?

4. How well can the communicative functions in DIT++ and other multidimensional schemes be recognised automatically?

5. Much work has focussed on the automatic dialogue act classification of presegmented utterances. What is the performance on automatic dialogue act segmentation and how does this affect the classification score?

6. How does a two-step approach of act recognition, that first segments and subsequently classifies the segments, compares to an approach where segmentation and classification are carried out simultaneously?

Once the dialogue acts are successfully identified, they may be used in dialogue management, which is a research area in itself. The specific questions explored here are the following:
7. To what extent can the communicative functions of DIT++ dialogue acts be predicted based on a dialogue history of previous communicative functions? What is the effect on the prediction performance of additionally taking the semantic content of dialogue acts into account?

8. Can prediction of dialogue acts be based on patterns in dialogue that are more flexible and powerful than n-gram language models?

1.3 Approach

For answering the abovementioned questions, dialogue corpora for mapping speech utterances to meaning are used. A new corpus of task-oriented information-seeking dialogues (the DIAMOND corpus) was compiled and annotated with DIT++ dialogue acts [Geertzen et al., 2004], as no such recorded and annotated dialogue material existed. The evaluation of this corpus and the characteristics of the dialogue acts evoked an elaboration of certain methodological aspects, more specifically the assessment of reliability and applicability of a tagset for annotation.

To obtain models that allow automatic dialogue act recognition and dialogue act prediction, and take the characteristics of available corpus data into account, machine learning methods are used. Applying machine learning techniques raises the question which algorithms to use and which kinds of information to use in the learning process. These two considerations are not explored ad fundum, but in sufficient detail to give a representative indication of performance, using a variety of learning algorithms and including information that is considered to be relevant (and important) for the tasks.

1.4 Limitations

From a practical point of view, not all kinds of conversations and all modalities that could be considered for answering the abovementioned questions can exhaustively be explored, and corpus study depends much on what is available to the research community. The type of conversation that will be studied in this thesis is mostly task-oriented, and conversations which are held with no other purpose than to converse on anything whatsoever are not considered. Furthermore, there is a focus on spoken conversation, excluding written discourse or text-based interaction. The speech modality plays a central role in this work and non-verbal modalities of communication such as gesticulation and use of facial expressions are not taken into account.
1.5 Thesis outline

Chapter 2 is concerned with approaches to the study of meaning as expressed in language, and with concepts, tools, and systems relevant for computational dialogue modelling. The meaning of language as action according to speech act theory is described, as well as how speech act theory is fundamental to the notion of dialogue act. The dialogue act taxonomies that are most relevant for this thesis are introduced and discussed, together with their underlying theories. The chapter ends with a concise overview of practical dialogue models.

Chapter 3 discusses the data sets that have been used in the experiments that are reported. More importantly, annotation issues are discussed that can have serious impact on the machine learning of dialogue act recognition. A novel evaluation metric is proposed for measuring inter-annotator agreement in multidimensional dialogue act taxonomies (like DAMSL and DIT++), and the consequences of the annotator policy are discussed.

In Chapter 4, the automatic recognition of dialogue acts is discussed. The first part involves the introduction of a more refined way of segmenting dialogue behaviour. To be able to evaluate dialogue segmentation, dialogue act classification, and the combination of both, these tasks are formulated as classification tasks on token level. Using several machine learning classifiers, the performance for each of the tasks is assessed for different datasets. Finally, the performance of the machine learners is compared with that of human annotators.

Given the automatically recognised dialogue acts, the question how to behave as an interlocutor is discussed in Chapter 5. In this chapter, dialogue management is formulated as a dialogue act prediction task based on sequences of dialogue acts. Familiar statistical dialogue act prediction using $n$-gram language models is compared to dialogue act prediction based on a novel algorithm that uses grammatical inference to capture regularities in input sequences.

Chapter 6 summarises and discusses the main findings and their impact on computational dialogue modelling. A brief discussion of limitations, loose ends, and emerging new questions as topics for future research ends this chapter.
Chapter 2
Modelling dialogue

This chapter provides a brief introduction to models of dialogue. A historical overview is presented of approaches to the notation of meaning in communication, characterising communication as action (speech act theory). Two theories are discussed that describe dialogue dynamics in terms of dialogue acts: conversational act theory and dynamic interpretation theory. The taxonomies that are based on these theories, and which will be used in future chapters, are described in detail. Finally, the most common models of dialogue based on the notion of dialogue act are outlined.

2.1 Introduction

There is a wide variety of questions to be addressed when attempting to account for human linguistic interaction: what is meaning, how is it constituted, how is dialogue structured, what mechanisms exist to maintain conversation and recover from errors, ambiguities, and misunderstandings, and how, what, when, and why do participants contribute to the conversation. The research presented in subsequent chapters is mostly concerned with the automatic encoding and decoding of meaning of dialogue utterances, and the task of predicting how human conversation is likely to continue given the dialogue context.

In this chapter, a concise overview of approaches to dialogue modelling will be presented in three parts. The first part (Section 2.2) deals with the notion of ‘meaning’ in natural language generally and dialogue in particular. It aims at concisely contextualising the research presented in subsequent chapters, addressing the fundamentals of speech act theory. The second part (Section 2.3) addresses what kinds of meaning could be distinguished in dialogue. It discusses two general theories of describing
dialogue dynamics, Conversational Act Theory [Traum and Hinkelman, 1992] and Dynamic Interpretation Theory [Bunt, 2000], and the taxonomies of dialogue acts that are based on these frameworks. The discussion of the theories and taxonomies provides the background for understanding the dialogue act tagsets and annotations that are used in Chapter 3, 4, and 5. The third part (Section 2.4) introduces the most important paradigms and formalisations in dialogue modelling and supports the background for the dialogue modelling in Chapter 5.

2.2 Speech act theory

In contemporary philosophy of language two major lines of thought can be distinguished when it comes to the question how meaning and language are related. One line of thought, which is known as logical positivism, assumes that the meaning of an utterance can be adequately described by logical formulae that describe the facts that the utterance corresponds to. It focusses on how language relates to the physical world. It has been founded by Frege and Russell, and has been further explored by Montague, Carnap, Tarski, Kripke, and others. The other line of thought is based on the use of language, and has resulted in what is known as speech act theory. It has been founded by Wittgenstein and Austin, and has further been explored by Searle and Grice. Where the former line assumes the unit of meaning to be isolated propositions, the latter line assumes it to be a combination of a proposition and the way it is contextualised in language.

In the line of thought of logical positivism, a sentence is meaningful if and only if it can be tested for truth or falsity, either in a purely formal logical framework (logical verification) or by observation (empirical verification). Unverifiable utterances are considered nonsense. With logical formalisms as basis, logical positivism has greatly contributed to the theory of sentence meaning. Most notably, the development of logics of sense — including temporal connectives and quantification — has allowed to express aspects of propositional content adequately. However, because an account of meaning in natural language based on propositions only is not sufficient to account for certain contextual and performative uses of language, the focus in this section is on speech act theory.

2.2.1 Austin & Searle

Austin’s ideas, put forward in [Austin, 1962], contrast with the assumptions that were brought forward in a logical positivistic way of describing language. These assumptions are that the basic type of language use is declarative and that the meaning of utterances can be described by propositional truth or falsity. Austin opposes these assumptions and makes two important observations. First, not all utterances are statements and much of conversation involves non-statements such as exclamations, questions, expressions of desire, et cetera. Second, even utterances that have a grammatical form
corresponding to that of declaratives are not all used to make statements. Austin argues that these kinds of utterance contain no truth-conditional statements and are in themselves a kind of action. For instance, by uttering “I promise to help you tomorrow” the speaker is not just describing a promise but is actually making a promise. He named this kind of utterances performative utterances or performatives; other utterances were named constative utterances or constatives. Typically, performative utterances exhibit a specific linguistic structure: they tend to be realised in the first-person present-tense and ‘perform’ the action specified by the main verb. This main verb belongs to a class of verbs describing verbal activities, such as to promise, to warn, et cetera.

In a later stage of his analysis, Austin concludes that there is no theoretically sound way to distinguish between performatives and constatives, and he eliminates the distinction by considering statements just another type of speech act, which he called ‘stating’. As a consequence, Austin arrives at the view that all utterances constitute speech acts. In some utterances the type of act is marked by a main verb indicating the action performed; in others the action is implicitly signalled.

In describing an utterance as a kind of action being performed by the speaker, Austin distinguishes between the intention of the speaker and the effect on a listener. More specifically, a speech act involves a senseful message being expressed in a particular language (the locutionary act), an action being intended (the illocutionary act), and the effects on a listener (the perlocutionary act). Examples of illocutionary acts are: asking, asserting, and promising. Examples of perlocutionary acts are: convincing, and reassuring. The focus of attention in Austin’s studies, and in pragmatic studies on language use in general, is the illocutionary act, and the terms ‘speech act’ and ‘illocutionary act’ are often used synonymously.

The assumption that all utterances constitute speech acts, and that speech acts encode the meaning of an utterance facilitates the view that a speech act can be decomposed in two parts: an illocutionary force and a propositional content (which need not to be realised). For example, we can consider the following three similar utterances with associated speech acts:

1. “Eve is pressing a button.” (statement)
2. “Is Eve pressing a button?” (question)
3. “Eve, press a button” (order)

For each of the utterances above, the propositional content can be described by the proposition $\text{Press}(\text{eve}, \text{button})$. However, the type of illocutionary force is different for each utterance: the first utterance is of a declarative type, the second utterance is of an interrogative type, and the third utterance is of an imperative type.

Austin explored the space of illocutionary forces on the basis of performative verbs in the vocabulary of a language and proposed a set of classes to group them in. However, this classification was inadequate, which led Searle to propose an alternative, more precise, classification.

An important point that Searle makes is that illocutionary acts often conflate with the verbs that convey them. Hence, a taxonomy of illocutionary acts such as the one
proposed by Austin in his early work is useful as such, but is limited in applicability for several reasons. An important limitation is that such a taxonomy would be language dependent. Moreover, a taxonomy based on verbs only may not be sufficient to address all utterances that are not explicitly performativve. To deal with these limitations, Searle proposes to focus on illocutionary forces, rather than on verbs, and introduces several classes of illocutionary force to characterise illocutionary acts [Searle, 1969, 1979; Searle and Vanderveken, 1985].

By placing the illocutionary force central, Searle argues —like Austin—, that illocutionary force is an essential aspect of meaning that cannot be explained adequately by means of truth and falsity only. For an adequate account of meaning, it is necessary to know what the addressee is to do with the proposition (if any) conveyed in an utterance, and should therefore be explained by a theory of action rather than a theory of truth-conditional meaning.

2.2.2 Issues

Concerning the assignment of speech acts there are mainly two distinct complicating issues that will not be treated at full length at this point but which will be addressed later in this chapter. First, a one-to-one relationship between utterance and speech act is not always possible. Depending on the definition of utterance and the fine-grainedness of the act taxonomy a speaker may use several utterances to perform a single act but could also perform multiple acts with a single utterance. In the latter case, the utterance is said to be multifunctional. Multifunctionality is exemplified in the following QUESTION-ANSWER pair:

(2.1) \[ U: \text{“Can you tell me the departure time of the first train on Sunday morning?”} \]

(2.2) \[ S: \text{“On Sunday morning the first train to the airport leaves at 5.32.”} \]

Interlocutor \( S \) produces an utterance that answers \( U \)’s question, and simultaneously provides positive feedback on \( S \)’s understanding of the question. Second, the relationship between the surface form of an utterance and the illocutionary act(s) of the utterance is not always straightforward. Searle [1975] gives as example the utterance “Can you pass the salt?”. This utterance is an interrogative, which usually is motivated by the speaker’s goal to get an answer. The utterance has, plausibly, a different purpose: to get the addressee to pass the salt. Searle calls the act that is produced here an indirect speech act. Indirect speech acts introduce a problem in assigning a speech act to an utterance because it requires an analysis of what the speaker intends, which is not always easy. Several accounts have been proposed for how to determine indirect speech acts and for how linguistic form relates to illocutionary force, from treating indirect speech acts as simple idioms (e.g. [Sadock, 1974]) to using pragmatic inference and context evaluation (e.g. [Gordon and Lakoff, 1975]).
2.3 Dialogue acts

2.3.1 From speech acts to dialogue acts

Speech act theory aims to describe the effect of an utterance as such, but often insufficiently allows to express the meaning of the utterance in the context of the dialogue it occurs in. In an ongoing dialogue, this context is dynamic in that beliefs, desires, and intentions change over time. Given that dialogue involves multiple participants issues such as the coordination of communication and the establishment of mutual understanding (i.e. *grounding*) have to be considered. Dialogue being a collaborative social activity also involves joint action plans and social attitudes being communicated.

In more recent work on dialogue modelling and spoken dialogue systems the Searlean notion of speech act is incorporated in a more elaborate type of act that can also address other conversational functions an utterance can play. These acts are often called *dialogue acts* and in many dialogue act taxonomies also cover domain-specific information. Other terms that are used synonymously or relating to a more or less similar concept, depending on framework or theory, are *conversational act* [Traum and Hinkelman, 1992], *communicative act* [Allwood, 1976], *conversational move* [Power, 1979], or simply *speech act*.

2.3.2 Dialogue acts in the context of collaborative joint activity

A dialogue may be described as a sequence of contributions by the interlocutors. To explain why certain contributions are followed by others, several researchers have argued that dialogue is best regarded as a form of joint activity: participants act in coordination with each other.

Collaboration and grounding

A prominent account of interaction based on joint activity is the collaborative model of Clark [1996]. This model extends the traditional sender/receiver models of communication, which assign the speaker the task of producing understandable utterances, and the listener the task of understanding these utterances. In the traditional models, conversation proceeds utterance by utterance. In Clark’s collaborative model, the speaker and addressee additionally attempt to establish the *mutual belief* that the addressee has understood what is uttered. In Clark’s conception, conversation does not proceed utterance by utterance, but contribution by contribution, in which a contribution consists of a presentation and acceptance of an utterance. The process of establishing mutual understanding of each others’ intentions and actions is called *grounding*; the common set of beliefs is called *common ground*. Grounding is understood to be functional in nature and efficient: interlocutors tend to minimalise collaborative effort in communicating (c.f. [Clark and Wilkes-Gibbs, 1986; Sperber and Wilson, 1995]). Clark et al. [Clark and Schaefer, 1989; Clark and Brennan, 1991] defined a contributions as hav-
ing two phases: a presentation and acceptance phase. In the presentation phase, the speaker presents an utterance to the addressee in the assumption that the utterance has been understood when the addressee gives evidence for it. In the acceptance phase, the addressee accepts the utterance by giving evidence that he believes that he understands what the utterance means, in the assumption that once the other registers the evidence he will also believe that the addressee understands.

Clark’s model gives an appealing collaborative account of dialogue, and explains why acknowledgements occur. Nevertheless, there are some problematic aspects. For instance, the messages that have a function of acceptance are themselves a presentation that needs acceptance, which leads to an infinite regression and raises the question of how much acceptance is sufficient to safely assume that information is grounded. Such considerations have been addressed in more recent accounts of grounding.

Traum [1999] goes a step further and axiomatises the grounding principles by proposing in addition to dialogue acts called grounding acts. The motivation is that in the model of grounding of Clark et al grounding is often established only after analysis of later utterances, and grounding acts offer an intermediate level and can be performed by a discourse unit that may contain one or more dialogue acts.

In the context of DIT++, Bunt et al. [2007] propose grounding mechanisms that operate on the belief state of the interlocutor, on a higher level than the intermediate grounding acts proposed by Traum.

Feedback

Linguistic feedback relates directly to grounding, as grounding can only be successful by means of positive feedback, be it explicit or implicit. Positive feedback acknowledges that (an aspect of) communication works well; negative feedback signals a problem in communication.

Important in the grounding process is the level of understanding of an utterance. To communicate, interlocutors should establish contact with each other. The speaker may produce an utterance, but this does not guarantee that the addressee is actually successfully hearing it. Apart from hearing the utterance, the addressee should also be able to extract and understand its meaning, and to successfully incorporate it in his own belief state. The levels of understanding and their hierarchical relations have been extensively explored by Allwood et al. [1992] and Clark [1996], who both distinguish the four similar levels depicted in Table 2.1.

The first level is about whether the interlocutors have contact (a communicative channel) and pay attention to each other. The second level is about whether the addressee has perceived the utterance. The third level is about whether the addressee understands what has been said, and the fourth level is about whether the addressee has integrated the information.

Bunt [2000] generally adopts the four levels described above, but distinguishes two

---

1See also Section 2.3.4.
Allwood Clark Bunt

<table>
<thead>
<tr>
<th></th>
<th>Allwood</th>
<th>Clark</th>
<th>Bunt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>contact</td>
<td>attention</td>
<td>attention</td>
</tr>
<tr>
<td>2</td>
<td>perception</td>
<td>identification</td>
<td>perception</td>
</tr>
<tr>
<td>3</td>
<td>understanding</td>
<td>understanding</td>
<td>interpretation</td>
</tr>
<tr>
<td>4</td>
<td>reaction to main evocative function</td>
<td>consideration</td>
<td>evaluation</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>execution</td>
</tr>
</tbody>
</table>

Table 2.1: Levels of feedback according to Allwood et al. [1992], Clark [1996], and Bunt [2000].

levels, ‘evaluation’ and ‘execution’, where Allwood and Clark distinguish one level. Evaluation compares the information that the utterance encodes with the belief state and becomes problematic when accepting the new information would put the addressee in an inconsistent belief state. Execution, also called ‘application’ or ‘dispatch’, is the performance of the requested or instructed action. For instance, the execution of a question is the gathering of the information to answer; executing an answer is integrating its semantic content with the belief state.

Another aspect of feedback functions is direction [Allwood, 2001]. The speaker gives feedback when he wants to let the addressee know about the status of his linguistic processing; the speaker elicits feedback when he wants to know about the status of linguistic processing of the addressee. Bunt [1995] distinguishes also between the speaker giving feedback to the addressee about his own linguistic processing (auto-feedback) and that of the addressee (allo-feedback). Allo-feedback is either elicited (e.g. “Do you understand?”) or spontaneously provided (e.g. “You haven’t understood.”).

2.3.3 From dialogue system act sets to systematic theoretical accounts

As a result of the diversity of dialogue types and the range of underlying motivations to model them, many researchers construct and use their specific set of dialogue acts according to considerations of what kind of dialogues are desired or expected, or what kind of analysis is to be performed. For instance, take a scenario in which a dialogue system (S) needs to be developed that is able to assist in car travel planning and should be able to have the interaction with a human user (U) as depicted in Figure 2.1.

Assuming a minimal but sufficient number of acts, a set of the following dialogue acts could be suggested: QUESTION-PLACE-FROM, ANSWER-PLACE-FROM,
1. **S** From where do you want to start your travel?
2. **U** Warandelaan . . . Tilburg
3. **S** Where do you want to go?
4. **U** Museumplein . . . Amsterdam

Figure 2.1: Short information-seeking human-machine dialogue.

**QUESTION-PLACE-TO**, and **ANSWER-PLACE-TO**. Such a set is sufficient to deal with simple interactions as the one illustrated above. When the set of dialogue acts should allow future extension in domain complexity or coverage of multiple domains, a tight coupling between dialogue acts and domain-specific concepts should be avoided. Even with wider coverage of an application domain, or with coverage of multiple domains, the exemplified set lacks acts that, in addition to acts related to application domains, have to do with the activity of having a dialogue. The decomposition of an act in illocutionary force and propositional content as proposed in speech act theory is ignored here. An example of a dialogue act set that had the aim to be flexible in application domain and dialogue is the VerbMobil I dialogue act set (see Figure 2.2, taken from [Alexandersson and Reithinger, 1995]) that was is used to model appointment scheduling dialogues in the beginning of the VerMobil project. The taxonomy has been constructed in such a way that it is domain independent at the highest levels and
more domain-dependent at lower levels. In a later stage of the project, a hierarchical
taxonomy that is less domain-dependent has been favoured by separating dialogue act
type and semantic content (cf. [Alexandersson et al., 1998]).

Except for dialogue act sets that originate from dialogue system engineering, there
is a substantial body of research that aims to describe the illocutionary acts that can
be distinguished in conversation by means of dialogue act taxonomies. One way of
ordering and categorising dialogue acts is by means of a hierarchy, where dialogue
acts lower in the hierarchy are more specific than dialogue acts higher in the hierarchy.
Furthermore, dialogue acts can be grouped according to the domain they belong to;
some dialogue acts are specific for an application domain (e.g. making appointments)
whereas other dialogue acts deal with general aspects of communication.

In order to formulate dialogue as meaning construction, there have been several ap-
proaches that aim to provide a systematic account, almost all of which elaborate on the
core notion of speech act in speech act theory. Two of these accounts that have been
worked out in considerable detail are Conversational Act Theory (CAT) [Traum and
Hinkelman, 1992] and Dynamic Interpretation Theory (DIT) [Bunt, 2000]. Both the-
ories have directly or indirectly resulted in dialogue act taxonomies that are claimed
to be domain independent. Moreover, the observation made by several researchers
that speech act theory is problematic in that it assumes an utterance to have a single
dialogue act while various acts in dialogue may occur simultaneously, is taken into
account in both theories by allowing multiple dialogue acts to be assigned to a single
utterance in several layers (CAT) or dimensions (DIT). Conversational act theory is im-
plemented within the TRAINS dialogue system [Allen et al., 1994] and has aspects that
have been incorporated in the DAMSL dialogue act taxonomy [Allen and Core, 1997].
Dynamic Interpretation Theory is the theoretical framework for the DIT++ dialogue
act taxonomy and has been implemented in the PARADIME dialogue management
module [Keizer and Bunt, 2006]. Both theories and the above-mentioned dialogue act
taxonomies will be described in more detail.

2.3.4 Conversational Act Theory

Traum and Hinkelman [1992] argue that speech act theory is not adequate in describing
dialogue and some of its assumptions are too simple, and present a new framework
which they claim to be more general.

Traum and Hinkelman distinguish four classes or ‘levels’ of acts in dialogue:

1. *Turn-taking acts*, which allow participants in conversation to control or influence
who is speaking. There are four acts in this level: ASSIGN TURN, KEEP TURN,
TAKE TURN, and RELEASE TURN.

2. *Grounding acts*, which are used to create and manipulate a common ground
between the participants in the conversation and as such signify the part an utter-
ance plays in the conversation. The acts in this level are: INITIATE, CONTINUE,
ACKNOWLEDGE, REPAIR, REQUEST-ACKNOWLEDGE, REQUEST-REPAIR, and CANCEL.

3. **Core speech acts**, which are like the classical illocutionary acts in Speech Act Theory but are reinterpreted as collaborative achievements that are only fully realised after having been acknowledged. Examples are: INFORM, YN-QUESTION, WH-QUESTION, SUGGEST, REQUEST, ACCEPT, REJECT.

4. **Argumentation acts**, which specify more comprehensive argumentational action and may be composed of a combination of speech acts in a dialogue game or rhetorical discourse structure. Examples of acts are: SUMMARISE, CONVINCE, CLARIFY.

The four levels are structured hierarchically in that turn taking acts and grounding acts are sub-units of core speech acts, and that core speech acts are part of the argumentation plan specified by argumentation acts. A more detailed taxonomy has been presented by Poesio and Traum [1997], who introduce one additional level for representing locutionary acts containing the full utterance. In this taxonomy, the level of core speech acts consists of two groups of acts: acts that have a forward-looking function and acts that have a backward-looking function. The forward-looking acts introduce new social attitudes in the conversation that have to be addressed in the ongoing dialogue. As a result, such acts constrain the future discourse in attempting the addressee to perform a particular action. Examples of such acts are: OFFER, INFO-REQUEST, and ASSERT. Backward-looking acts are responses to previous acts, and as such are usually related to the domain or to the grounding process. For instance, a dialogue participant may accept or reject a previous request for information. The backward-looking acts indicate how the current utterance relates to the previous discourse. Examples are: ACCEPT, REJECT, and ANSWER. The grounding acts, turn-taking acts, and argumentation acts are roughly the same compared to what has been presented in [Traum and Hinkelman, 1992].

Fundamental to Conversational Act Theory is an account of grounding by means of acts. Traum and Poesio model the state of the conversation, which they call the conversational score, as a discourse representation structure (as in Discourse Representation Theory [Kamp and Reyle, 1993]), which includes a record of grounded and ungrounded information during conversation. The formalisation of the conversational score is based on Compositional DRT [Musken, 1996] and the dynamics of the score is based on the use of dialogue acts, which are specified in terms of conditions on the update and the effects of an update on the score.

### 2.3.5 Dynamic Interpretation Theory

In Dynamic Interpretation Theory (DIT, see e.g. [Bunt, 1989, 1995, 2000]), dialogue acts are considered to be operators that update the context (i.e. the information state)
of a dialogue participant. Context is defined as the totality of conditions that may influence understanding and generation of communicative behaviour. It is characterised along five dimensions (linguistic, semantic, cognitive, physical, and social) and contains a state of beliefs of each dialogue participant including assumed beliefs about other dialogue participants.

Similar to speech acts being composed of propositional content and illocutionary force as proposed by Austin and Searle, a dialogue act in DIT consists of two components: a semantic content (SC) and a communicative function (CF). The SC contains information with which the context can be updated and can for instance be expressed in first-order logic; the CF specifies the way in which the context is updated with the SC.

A CF in DIT is considered an operator (on the context model), and is defined in terms of applicability conditions, called preconditions, which enable its performance and contain intentions and assumptions that the speaker (S) has with regard to himself and with regard to the addressee (A). Take for instance the utterance in Example 2.3:

\begin{equation}
\text{(2.3) “Does that book has a red cover?”}
\end{equation}

This utterance has as CF a PROPOSITIONAL-QUESTION, which is defined as follows: Given the proposition, $p$, $S$ wants to know the truth value of $p$, and $S$ assumes that $A$ knows the truth value of $p$. In DIT terminology, the dialogue act performed with utterance 2.3 has a SC which may be expressed as a first-order logic formula like in Example 2.4:

\begin{equation}
\text{(2.4) } \text{Book}(x_1) \land \text{Color}(x_1, \text{‘red’})
\end{equation}

where $x_1$ is a variable to be bound through the resolution of the anaphor “that book”.

The various aspects of communication that can be addressed independently are called dimensions [Bunt and Girard, 2005; Bunt, 2006]. The DIT++ tagset distinguishes 10 dimensions. There is a distinction between CFs that have to do with the task or the domain (called task-related CFs) and CFs that have to do with the communication as such (called dialogue control CFs). For instance, when dialogues are about meetings, a set of functions containing OPEN-MEETING, CLOSE-MEETING, and MAKE-APPOINTMENT could be part of the Task dimension. The other dimensions all contain a number of communicative functions that are specific to each of them, such as for Turn Management, for Time Management, and for the management of social obligations. The CFs in these dimensions are called dimension-specific communicative functions.

Besides dimension-specific communicative functions, DIT also distinguishes a layer of communicative functions that are not specific to any particular dimension but that can be used to address any aspect of communication. These functions, which include questions, answers, statements, and commissives as well as directive acts, are called general purpose functions. These functions can be grouped according to the classes depicted in Figure 2.3. A dialogue act falls within a specific dimension if it has a
2.3. DIALOGUE ACTS

Figure 2.3: Organisation of general purpose functions in DIT++.  

communicative function specific for that dimension (e.g. utterance 2.5) or if it has a 
general-purpose function and a semantic content relating to that dimension (e.g. utter-
ance 2.6).

(2.5) “Thanks.”

(2.6) “I am most grateful for your help.”

Dialogue utterances can in principle have a function (but never more than one) in 
each of the dimensions, so annotators using the DIT++ scheme can assign at most one 
tag for each of the 10 dimensions to any given utterance. Both within the set of general-
purpose communicative function tags and within the sets of dimension-specific tags, 
tags can be hierarchically related in such a way that a label lower in a hierarchy is 
more specific than a label higher in the same hierarchy. A tag $F_1$ is more specific than 
a tag $F_2$ if $F_1$ inherits all the preconditions of $F_2$ and defines a context update operation 
that includes the update operation corresponding to $F_2$. For instance, a CHECK is more 
specific than a PROPOSITIONAL QUESTION, because it has an additional precondition 
concerning the speaker’s expectations, namely that the proposition is probably true.

Apart from precondition, formal descriptions of operators usually also define the 
(post) effects and the body. The effects specify the results that are obtained by apply-
ing the operator; the body describes how the effects are achieved. In DIT the latter 
two aspects are not specified for each CF separately as the effects and bodies are a 
consequence of general communication principles (responsible for certain effects) and 
context update principles (mechanisms to manipulate) that work on the belief state 
[Morante Vallejo, 2007]. The general communicative effects are specified as follows:
• **Understanding effects:** the effects of the addressee understanding the utterance of the speaker and its implicit positive feedback effects on the previous utterance of the addressee, unless the utterance has an explicit feedback function. In particular, the addressee upon understanding the utterance believes that the corresponding preconditions apply for the speaker;

• **Expected understanding effects:** the effects of the speaker expecting (in terms of a weak belief) that the understanding effects for the addressee take place. The assumption that both interlocutors have expected understanding leads for both interlocutors to the mutual belief that the speaker expects that the addressee understand the utterance;

• **Adoption effects:** the effects of the addressee incorporating the information presented by the speaker in his own information state;

• **Expected adoption effects:** effects of the speaker expecting the adoption effects for the addressee to take place. The assumption that both interlocutors have expected adoption leads for both interlocutors to the mutual belief that the speaker weakly believes that the addressee adopted the information presented.

The update mechanisms for the context are the following:

• **Creation:** An interlocutor introduces a belief as the effect of assigning an interpretation to what has been said by another interlocutor;

• **Adoption:** An interlocutor incorporates beliefs of other interlocutors as beliefs of his own;

• **Cancellation:** A belief or goal is cancelled because a belief does not apply anymore or a goal has been achieved or has been understood to be unachievable;

• **Strengthening:** An expectation, or ‘weak belief’ becomes a firm belief because sufficient supporting evidence for the belief becomes available.

Using the effects dialogue acts have in the context of the general communication principles, the update mechanisms provide an exact specification of how the belief states of the interlocutors evolve during a dialogue. In this sense DIT has much in common with the information-state update approach of Traum and Larsson [2003] (see also 2.4.5), and the formulation of dialogue acts as having the function of conveying intention and being operators on belief dynamics makes the approach similar in spirit to the Belief-Desire-Intention model of practical reasoning agents [Allen and Perrault, 1980].
2.3.6 DAMSL taxonomy

DAMSL (Dialogue Act Markup using Several Layers), like most other schemes, defines a set of communicative actions that can be used to analyse dialogues. Rather than trying to capture the meaning conveyed in an utterance with a single layer, the scheme allows multiple labels in multiple layers to be applied to an utterance. The dialogue acts defined in DAMSL are intended to be applicable to various types of dialogues. The scheme has been developed by the Multiparty Discourse Group with the aim of providing a framework of generic dialogue acts, to be used as such or as super types in other schemes, which allows comparative studies across corpora.

Drawing on the work of Allwood et al. [1992]; Allwood [2000], the DAMSL scheme features a distinction between the relations an utterance may have with the preceding discourse (called backward looking functions) and the effects the utterance may have on the future discourse (forward looking functions). Apart from these two dimension, two other dimensions or ‘layers’ are distinguished in DAMSL: communicative status and information level. Each dimension refers to (more or less) independent aspects of dialogue and contains a set of acts:

1. **Communicative Status** indicates whether the utterance is intelligible and whether it was successfully completed. It contains the following elements:

   - UNINTERPRETABLE: The utterance is not comprehensible.
   - ABANDONED: The utterance is a result of the speaker making an error or changing his mind and abandoning the utterance, and in doing so not providing any content to the dialogue.
   - SELF-TALK: The utterance is a result of the speaker talking to himself.

2. **Information Level** characterises the semantic content of the utterance, and contains the following elements:

   - TASK: Utterances that involve performing the task that is the reason for the dialogue.
   - TASK-MANAGEMENT: Utterances that explicitly address the problem solving procedure that is not part of the task but is about the task.
   - COMMUNICATION-MANAGEMENT: Utterances that are aimed at maintaining the communication, such as those that are related to turn management and maintaining contact and understanding.
   - OTHER-LEVEL: Utterances with semantic content that do not belong to any of the other categories of the Information Level layer.

3. **Forward Looking Function** indicates how the current utterance constrains the future beliefs and actions of the participants and affects the discourse. The following classes of functions are distinguished:
• **statement**: Utterances that make explicit claims about the world. In general, the content of statements can be evaluated as being either true or false, although weaker forms such as suggesting that something might be true are also considered assertions.
  - **ASSERT**: $S$ is trying to change a belief of $A$ by making a claim.
  - **REASSERT**: $S$ thinks that the claim has already been made.
  - **OTHER-STATEMENT**: any other statement that is not an ASSERT or REASSERT.

• **influencing-addressee-future-action (influence on listener)**: Utterances that aim to directly influence the addressee’s future non-communicative actions (e.g. a request). The following functions are distinguished:
  - **ACTION-DIRECTIVE**: Utterances in which $S$ creates an obligation that $A$ does a certain action unless $A$ indicates otherwise.
  - **OPEN-OPTION**: Utterances in which $S$ proposes options to $A$ without creating any obligation.

• **INFO-REQUEST**: An aspect that is marked when the utterance is a request for information.

• **committing-speaker-future-action (influence-on-speaker)**: Utterances that potentially commit $S$ to some future course of action. The possible functions in this class are:
  - **OFFER**: $S$ conditions his commitment on future agreement of $A$.
  - **COMMIT**: $S$ does not condition his commitment on future agreement of $A$.

• **conventional**:
  - **OPENING**: If the utterance is a conventional phrase used to summon $A$ or start the dialogue.
  - **CLOSING**: If the utterance is a conventional phrase used to dismiss $A$ or close the dialogue.

• **EXPLICIT-PERFORMATIVE**: If an action is performed by virtue of producing the utterance.

• **EXCLAMATION**: If the utterance is an exclamation.

• **OTHER-FORWARD-FUNCTION**: If $S$ is performing an action not captured by any other Forward Looking Function.

4. **Backward Looking Function** indicates how the current utterance relates to previous discourse by another interlocutor. The following classes of functions are distinguished:

• **agreement**
  - **ACCEPT**: $A$ accepts an offer made by $S$.
  - **ACCEPT-PART**: $A$ partly accepts an offer made by $S$. 
○ MAYBE : A refuses to make a judgement, e.g. expressing doubt, about an offer made by S.
○ REJECT : A rejects S’s offer.
○ REJECT-PART : A partly rejects S’s offer.
○ HOLD : utterances such as clarification questions that delays A’s reaction to S’s question or proposal.

• understanding : Concerns the action that interlocutors take in order to keep understanding each other during conversation.
  ○ SIGNAL-NON-UNDERSTANDING : A signals (e.g. by clarification request) that he did not understood S.
  ○ signal-understanding
    – ACKNOWLEDGE : A acknowledges what S said.
    – REPEAT-REPHRASE : A repeats some of S’s utterance.
    – COMPLETION : A completes S’s sentence.
  ○ CORRECT-MISSPEAKING : A understands S and attempts to correct S.

• ANSWER : Indicates that an utterance is supplying information explicitly requested by a previous INFO-REQUEST act.

• INFORMATION-RELATION : Describes how the information in the current utterance relates to the previous utterances. The idea is to assign here a relation like those that can be found in Rhetorical Structure Theory [Mann and Thompson, 1988]. However, the set of possible relations has still to be defined.

### 2.3.7 DIT taxonomy

The DIT taxonomy distinguishes 10 dimensions that each address a particular aspect in communication. The DIT++ tagset [Bunt, 2005a] was designed to combine the communicative functions of dialogue acts in DIT and many of those in DAMSL [Allen and Core, 1997] and other annotation schemes. An important characteristic is the elaborate and fine-grained set of functions for feedback and other aspects of dialogue control that is available in DIT, partly inspired by the work of Allwood et al. [1993].

The dimensions in DIT are: addressing information about the task domain (Task); providing information about the speaker’s utterance processing (Auto-feedback), eliciting information about or addressing the addressee’s utterance processing (Allo-feedback); managing communicative difficulties of the speaker (Own Communication Management) or of the addressee (Partner Communication Management), dealing with the allocation of the speaker role (Turn Management), with contact maintenance (Contact Management) and with the use of time (Time Management), addressing the structure of the dialogue (Dialogue Structure Management), and dealing with social conventions (Social Obligations Management). The taxonomy, as mentioned, contains two types of CF: those linked to a particular dimension (‘dimension-specific functions’) an those which can be applied in any dimension (‘general-purpose functions’).
The conditions of applicability of a CF (preconditions) are about the beliefs and goals of the speaker. In this context, the speaker holds a belief when the speaker considers a proposition to be true. The speaker holds a weak belief when he considers a proposition as likely to be true but is uncertain about this. Such uncertain beliefs typically motivate an interlocutor to perform a check question. In the definitions of CFs, the speaker is said to know a proposition when he holds this proposition to be true, regardless whether it is actually true or not.

**General Purpose Communicative Functions**

General-purpose CFs are functions that can be applied to any dimension. They can be grouped in information transfer functions (related to seeking or providing information) and action discussion functions (related to performing actions):

1. Information-seeking functions

   - INDIRECT PROPOSITIONAL-QUESTION (IND-PROP-Q) : $S$ wants to know the truth-value of $P$, and $S$ does not know if $A$ knows the truth-value of $P$.
     - PROPOSITIONAL-QUESTION (PRO-Q) : $S$ believes $A$ knows the truth-value of $P$.
       - CHECK : $S$ weakly believes $P$.
         - POSI-CHECK : $S$ weakly believes $A$ believes $P$.
         - NEGA-CHECK : $S$ weakly believes $A$ believes not $P$.

   - INDIRECT SET-QUESTION (IND-SET-Q) : $S$ wants to know which elements of a given set have a given property; $S$ believes there is at least one such element; $S$ does not know whether $A$ knows which elements of that set have that property.
     - SET-QUESTION (SET-Q) : $S$ believes $A$ knows which elements of the set have the property.

   - INDIRECT ALTERNATIVES-QUESTION (IND-ALT-Q) : $S$ wants to know which one from a given list of alternative propositions is true; $S$ believes exactly one of the alternatives is true; $S$ does not know whether $A$ knows which of the alternative propositions is true.
     - ALTERNATIVES-QUESTION (ALT-Q) $S$ believes $A$ knows which of the alternative propositions is true.

2. Information-providing functions

   - UNCERTAIN INFORM (UNC-INF) : $S$ wants to make the information that forms the semantic content known to $A$; $S$ weakly believes the information he provides is correct.
     - INFORM : $S$ believes the information he provides is correct.
2.3. DIALOGUE ACTS

- AGREEMENT: $S$ believes $A$ weakly believes the SC to be true.
- DISAGREEMENT: $S$ believes $A$ weakly believes the SC to be false.
  - CORRECTION: $S$ wants the SC, which he believes to be correct, to replace a belief by $A$ that $S$ believes to be incorrect.
- INFORMS with a rhetorical function: $S$ is motivated by a semantic relation of the kind defined in Rhetorical Structure Theory (RST) [Mann and Thompson, 1988].

- UNCERTAIN SET-ANSWER (UNC-SET-A): $S$ believes $A$ wants to know which elements of a given set $C$ have a given property; $S$ believes $A$ believes $S$ knows that; $S$ is not certain.
  - SET-ANSWER (SET-A): $S$ is certain.
- UNCERTAIN PROPOSITIONAL-ANSWER (UNC-PRO-A): $S$ believes $A$ wants to know $P$; $S$ believes $A$ believes $S$ knows that; $S$ weakly believes $P$.
  - UNCERTAIN DISCONFIRM (UNC-DISCONF): $S$ believes $A$ weakly believes not $P$.
    - CONFIRM: $S$ believes $A$ believes $P$.
    - DISCONFIRM: $S$ believes $A$ believes not $P$.

3. Action discussion functions

- Commissives: $S$ is committed to perform a certain action, possibly dependent on $A$’s consent that $S$ does so.
  - OFFER: $S$ is committed to perform the action if $A$ would like $S$ to do so.
    - PROMISE: $S$ is committed to perform the action.
    - ADDRESS-REQUEST: $S$ knows $A$ wants $S$ to perform the action; $S$ is committed to conditionally perform the action.
      - ACCEPT-REQUEST: $S$ is committed to unconditionally perform the action.
      - DECLINE-REQUEST: $S$ is committed to not perform the action.
    - ADDRESS-SUGGESTION: $S$ knows $A$ believes the action is potentially promising for achieving a certain goal; $S$ knows $A$ believes $S$ is able to perform the action (possibly together with $A$); $S$ is committed to conditionally perform the action.
      - ACCEPT-SUGGESTION: $S$ is committed to unconditionally perform the action.
      - DECLINE-SUGGESTION: $S$ is committed to not perform the action.
CHAPTER 2. MODELLING DIALOGUE

- Directives: S wants A to consider a certain action which A might carry out (possibly together with S), potentially wanting to put pressure on A to do so.
  - INSTRUCT: S wants A to perform the action; S assumes A is able to do so.
  - ADDRESS-OFFER: S believes A is committed to perform the action on S’s consent that A does so.
    - ACCEPT-OFFER: S wants A to perform the action.
    - DECLINE-OFFER: S wants A to not perform the action.
  - INDIRECT-REQUEST: S wants A to perform the requested action, conditional on A’s consent.
    - REQUEST: S wants A to perform the requested action, conditional on A’s consent; S assumes that A is able to do so.
  - SUGGEST: S wants A to be aware that the action mentioned is potentially promising for achieving a certain goal which is either contextually salient or named explicitly.

Dimension-Specific Communicative Functions

Each dimension listed in Section 2.3.7 has a number of CFs that are typical for that particular dimension. A so-called dimension-specific function can only be assigned in a particular dimension. Dimension-specific functions can be grouped into functions for a particular task domain (task-specific functions) and functions for dialogue control (dialogue-control functions).

Task specific functions denote specific actions in the task domain that the conversation is about. For instance, for conversations in the VERBMOBIL domain, which is about making appointments, there are domain-specific functions such as ACCEPT-DATE, ACCEPT-LOCATION, and MOTIVATE-APPOINTMENT [Alexandersson et al., 1998]. The dialogue-control functions address aspects that have to do with the communication as such and as a consequence are relevant to any conversation, no matter the domain. This closed set contains the following functions per domain:

1. The two feedback dimensions Auto-feedback and Allo-feedback signal information about the processing of the previous utterances of the speaker himself (Auto-feedback) or the addressee (Allo-feedback). The feedback functions in both dimensions have in common that they address two aspects: polarity and processing level. The polarity expresses if feedback is positive or negative. Positive feedback signals successful processing; negative feedback signals problems in processing. The processing level indicates which level of processing is addressed. Ordered from the lowest to the highest level, these are:
   - Attention, i.e. paying attention to the speaker (e.g. listening, looking);
   - Perception, i.e. the recognition of the auditive, visual, or tactile components of communicative behaviour;
2.3. DIALOGUE ACTS

- **Interpretation**, i.e. the assignment of meaning to the recognised communicative behaviour (which is the assignment of semantic content and communicative functions to utterances);

- **Evaluation**, i.e. comparing the information that an utterance encodes, due to its communicative functions and semantic content, with what was already known;

- **Execution**, i.e. performing an action that an utterance intended to evoke. For instance, execution of a request or instruct is performing the requested or instructed action; execution of a question is gathering the information to answer; executing an answer is integrating its semantic content with the belief state.

The processing levels are related in such a way that positive feedback on a higher processing level implies successful processing at all lower processing levels. Negative feedback at a lower processing level implies problematic processing at all higher processing levels.

Taking into account that when there is feedback, the processing level may be unspecified, both feedback dimensions have the same 12 functions (the combination of five levels and two polarities, plus two polarities without a level specified). For instance, with a NEGATIVE-INTERPRETATION function in the Allo-feedback dimension (ALF-NEG-INT), S signals his belief that A misunderstood the utterance of S.

In addition to the above-mentioned 12 functions, the Allo-feedback dimension has four functions which address elicitation of feedback on the four highest levels. For instance, with a PERCEPTION-ELICITATION function in the Allo-feedback dimension (ALF-ELIC-PERC), S wants to know if A’s perception of S previous utterance was successful or not.

2. Turn-management functions are involved in those dialogue acts that are performed in order to keep or to reallocate the speaker role.

A turn ends either because the current speaker assigns the speaker role to the addressee, or because he offers the speaker role without putting any pressure on the addressee to take the turn, or because the addressee interrupts the speaker and ’grabs’ the speaker role. As an utterance in a turn can address aspects of both managing the receiving a turn (e.g. grabbing a turn) and letting the turn go (e.g. assigning a turn to an addressee), a turn-management function may be composed of an initial function, a final function, or a pair of both.

- **initial functions**:
  - TAKE: S wants to have the turn, which is available;
  - ACCEPT: S agrees to take the turn, which A has given to him;
CHAPTER 2. MODELLING DIALOGUE

○ GRAB: $S$ wants to have the turn, which $A$ currently has, before $A$ assigns the turn to him or releases it.

• final functions:
  ○ KEEP: $S$ wants to keep the turn;
  ○ ASSIGN: $S$ wants $A$ to take the turn;
  ○ RELEASE: $S$ wants to make the turn available to any participant.

Taking into account that a Turn-management function can be an initial function, a final function, or a pair of both, there are 15 dimension-specific CFs in the Turn-management dimension.

3. Time management functions address aspects of timing in the communication:

  • STALLING: $S$ needs a little bit of time, and stalls his speech;
  • PAUSING: $S$ needs some time to do something, and suspends the conversation.

4. Contact management functions address aspects of establishing and maintaining contact:

  • CHECK: $S$ wants to know whether $A$ is (still) available for communication;
  • INDICATION: $S$ wants to indicate that $S$ is (still) available for communication.

5. Own-communication management functions aim to update the addressee on the speaker’s difficulties in producing the utterances that he is currently contributing:

  • ERROR-SIGNALLING: $S$ wants to signal that he has made a mistake in speaking;
  • RETRACTION: $S$ wants to withdraw something that he just said.
  • SELF-CORRECTION: $S$ wants to correct an error that he made in speaking;

6. Partner-communication management functions aim to support the addressee in producing a contribution to the dialogue:

  • COMPLETION: $S$ wants to help $A$ to complete an utterance that $A$ is struggling to complete;
  • CORRECT-MISSPEAKING: $S$ wants to correct (part of) an utterance by $A$, assuming that $A$ made a speaking error.

7. Discourse Structure Management functions have the purpose to structure the communication:

  • OPENING: $S$ wants $A$ to know $S$ is ready an willing to engage in a dialogue with $A$;
• PRE-CLOSING: S wants to end the current dialogue shortly;
• TOPIC-INTRODUCTION: S wants to introduce the topic mentioned in the semantic content;
• TOPIC-SHIFT-ANNOUNCEMENT: S wants to change the topic.
  ◦ TOPIC-SHIFT: S wants to shift the topic to the one mentioned in the semantic content.

8. Social obligation management functions address communication involving social obligations:

• INITIAL GREETING: S wants A to be aware of S’s presence; S believes S and A are in the position to communicate;
• RETURN GREETING: S wants A to be aware of S’s presence; S believes S and A are in the position to communicate; S is pressed to respond to a self-introduction by A;
• INITIAL SELF-INTRODUCTION: S wants to make himself known to A;
• RETURN SELF-INTRODUCTION: S wants to make himself known to A; S is pressured to do so by an initial greeting by A;
• APOLOGY: S wants A to know S regrets having made an communication error or problems in communicating;
• APOLOGY-DOWNPLAY: S wants to mitigate A’s feeling of regret;
• THANKING: S wants A to know S is grateful for what A has done;
• THANKING-DOWNPLAY: S wants to mitigate A’s feeling of gratitude;
• INITIAL GOODBYE: S wants to say A farewell;
• RETURN GOODBYE: S wants to say A farewell in response to A saying farewell to S;

2.4 Practical models of dialogue

Both from the point of view of theoretical analysis and of practical application (spoken dialogue systems), there is the desire to describe the underlying principles that are needed to predict how a dialogue may develop and what considerations of interlocutors play a role at which point. Typically, the perspective is that of one of the interlocutors; the aim is that of describing as accurately as possible how to act as an interlocutor algorithmically. The resulting models should account for the structure of the dialogue flow. In the case of simple task-oriented dialogues, the dialogue flow is likely to reflect the task structure. However, when the domain is more complex and the interaction is expected to be flexible (more natural), and to cope with misunderstandings, subtle
timing of speech, and disagreements, it becomes necessary to use models that consider
the information state of the interlocutor, involving beliefs, expectations, and intentions.
Such a model can said to be based on agency, as it uses reasoning processes.

In this section several approaches to model dialogue (and an interlocutor) as a gen-
erative activity are reviewed: frame-based, dialogue grammars, plan-based, conversa-
tional game based, and information-state based. Moreover, the recent use of reinforce-
ment learning to obtain optimal dialogue strategies is discussed.

2.4.1 Frame-based and finite state based

Frame-based models, also called form-based or slot-based methods, are typically in-
tended to facilitate information-seeking dialogue in which one interlocutor (usually the
dialogue system) provides information on request of the other interlocutor (usually a
user). The possible interaction for the information-providing interlocutor is defined
by means of a template, which is represented by means of attribute-value pairs. The
dialogue is usually driven by the goal of the information provider to obtain the neces-
sary attribute values from the information seeker in order to look up and provide the
information that the information seeker requested. The set of attribute values needed
to fulfil the information need can be seen as a form that needs to be filled out for which
each field (i.e. attribute, frame, or slot) may be a prompt for the information provider
to the information seeker. In its simplest form, the model dictates for each slot to be
filled a separate prompt from the information provider towards the information seeker
and the sequence of prompts that characterises the dialogue is fixed (usually ordered
by priority of information needed). This kind of model is also known as finite-state
based. For example, if in the route-planning domain the empty slot is DESTINATION,
the system prompt could be “where do you want to go?”.

More sophisticated frame-based models take into account that the information seeker
may provide information to fill multiple slots in a single prompt, are flexible in the
order of filling the slots given the dependencies between them, or work with more
complex templates that include type hierarchies or task-structure graphs.

With the recent use of the VoiceXML markup language (cf. [Abbott, 2002]), par-
ticularly in the e-commerce and transport information domains, frame-based models
have gained popularity. In this context, the templates are specified by VoiceXML doc-
uments, which can be exchanged as input for a core dialogue system.

Due to a combination of simplicity and flexibility, many frame-based systems have
been created. Example are: Bobrow et al. [1977], Seneff and Polifroni [1996], Kellner
et al. [1997], Strik et al. [1997], Aust and Schröer [1998], Lamel et al. [1999], and
Seneff [2002].

2.4.2 Dialogue grammars

The approach of dialogue grammars is based on the observation that dialogue exhibits
reoccurring patterns or sequences of acts. The assumption is that a set of grammar
rules allows to formulate sequential and hierarchical constraints on acceptable dialogues. For instance, consider the short dialog between interlocutors A and B depicted in Figure 2.4.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>u1</td>
<td>A</td>
<td>Hello!</td>
</tr>
<tr>
<td>u2</td>
<td>B</td>
<td>Hi</td>
</tr>
<tr>
<td>u3</td>
<td>A</td>
<td>What is your name?</td>
</tr>
<tr>
<td>u4</td>
<td>B</td>
<td>Kevin</td>
</tr>
</tbody>
</table>

Figure 2.4: Short dialogue with GREETING, QUESTION, and ANSWER.

Some dialogue acts are typically followed by others. For instance, an initial GREETING (u1) is typically followed by another GREETING (u2) and a QUESTION (u3) is often followed by an ANSWER (u4). Such bigrams are also known as adjacency pairs [Schegloff, 1968] and this kind of patterns could be modelled using phrase-structure grammar rules (cf. [Reichman, 1981]) much like the way such grammar rules constrain sentence generation to construct grammatically acceptable productions. In the grammar that defines the possible dialogue structures, the terminals are the dialogue acts that can be performed; the non-terminals are the possible stages in the dialogue. A simplified example of such a grammar is depicted in Figure 2.5. This grammar

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>dialogue ⇒ adjacency-pair</td>
<td></td>
</tr>
<tr>
<td>dialogue ⇒ adjacency-pair dialogue</td>
<td></td>
</tr>
<tr>
<td>adjacency-pair ⇒ GREETING GREETING</td>
<td></td>
</tr>
<tr>
<td>adjacency-pair ⇒ QUESTION ANSWER</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.5: A simplified dialogue grammar.

can produce the dialogue in Figure 2.4 and dialogues which comprise one or more adjacency-pairs (the first two rules). An adjacency-pair can either be realised by two consecutive GREETINGS (third rule) or by a QUESTION followed by an ANSWER.

In many respects, dialogue grammars have much in common with finite state automata (FSAs). When using FSAs, the states in the dialogue correspond to the states in the FSA, and the dialogue acts that are recognised or produced correspond to the state transition labels.

There are at least three major drawbacks of using dialogue grammars. First, a dialogue grammar typically allows to realise only a single state (non-terminal) at a time. This may pose problems for utterances that are multifunctional, as the dialogue grammar would need to be in multiple states at the same time. Second, the model lacks clear criteria for how to choose at a particular point in the dialogue amongst the possible grammar rules. Arguably, a dialogue grammar should generate dialogue contributions that are not only possible in human-human dialogue but that are also purposeful. Third, as a descriptive model dialogue grammars do not explain the underlying motivations of the behaviour.
An example is the SUNDIAL system (Andry et al. [1990]; Bilange [1991]), which uses a dialogue grammar and speech acts to model spoken dialogues about travel reservations. The same approach is used in the LINLIN system ([Ahrenberg et al., 1990]).

2.4.3 Plan-based approaches

Plan-based models assume that the dialogue acts that interlocutors perform are part of a plan with the aim of achieving certain goals. For instance, in an information seeking dialogue, the information seeker typically has a plan of obtaining certain knowledge and the information provider aims to get to know the plan(s) underlying the communicative behaviour of the information seeker, and to respond accordingly. The model assumes conversation to be a cooperative, joint activity. As conversation is a communicative process, the most elementary and abstract goal an interlocutor can have is that of changing the mental state of the addressee.

The advantage of plan-based models is that they are potentially very powerful. Furthermore, these models are able to deal with the interpretation of indirect dialogue acts rather effectively, as the plan and context of the dialogue may make apparent the indirect act [Perrault and Allen, 1980].

A disadvantage is that such models require decision making and goal and plan construction and inference (cf. [Allen and Perrault, 1980; Sidner and Israel, 1981; Carberry, 1988, 1990]), activities which can easily get so complex that they become computationally intractable or undecidable. Another disadvantage of plan-based approaches is that they have difficulties in dealing with (positive and negative) feedback and the repair of communication problems, since these activities are not part of a preconceived plan. Furthermore, a model of the mental states of the interlocutors, and expectations of goals and actions that are likely because of the context need to be specified. Just as with dialogue grammars, the explanatory aspect of the early plan-based approaches is problematic: events such as clarification questions are not explained as dialogue is modelled by means of plan recognition and plan generation. More sophisticated approaches (e.g. [Grosz and Sidner, 1990; Cohen and Levesque, 1991]) consider conversation as a joint activity in which all interlocutors have a joint commitment for achieving and maintaining understanding.

The disadvantage of high complexity is not the only aspect that makes plan-based approaches problematic. Problematic issues in plan-inference are mainly the susceptibility for noise in the input and the effective disambiguation among competing plan hypotheses (see [Carberry, 1990, 2001] for a discussion), two issues that also make it difficult to scale up from small, well-defined domains to large domains.

An example of a dialogue system in which plan recognition plays a major role is TRAINS ([Allen et al., 1996]), which has route planning as domain. More recently, a plan-based architecture of dialogue systems has been proposed in ([Allen et al., 2001]). Even though VerbMobil [Wahlster, 2000] is not a dialogue system but a speech-to-speech translation system in the domain of appointment scheduling, it is worth mentioning for plan inference similar to that of plan-based models in dialogue systems.
2.4.4 Conversational Game Theory

Conversational Game Theory (CGT) [Power, 1979] is a framework that combines ideas from both dialogue grammars and plan-based models that assume rational and collaborative interaction. In the model, conversations consist of exchanges called games. A game is composed of a sequence of dialogue moves (usually dialogue acts) that is the result of a game-specific set of rules similar to a dialogue grammar. The various kinds of games that are possible are assumed to be shared knowledge of the interlocutors. Furthermore, the interlocutors keep (shared) beliefs and goals (which determine the game to choose) during the dialogue.

The planning of games coincides with the realisation of goals and manifests structure on a higher level such as that of the task-structure in task-oriented dialogue (cf. [Grosz and Sidner, 1986]) and sub-goals may be realised by means of nesting of games.

One application of CGT is the markup of the Map Task Corpus [Carletta et al., 1996] where 128 task-oriented Map Task dialogues have been annotated with conversational moves, start and endings of games, and potential embedding. The set of conversational moves was the following: QUERY-YN, QUERY-W, REPLY-W, QUERY-YN, INSTRUCT, EXPLAIN, ALIGN, CHECK, ACKNOWLEDGE, CLARIFY, and READY. Evaluation of the annotations in terms of annotator agreement shows that whereas conversational moves can be assigned reasonably reliably, the annotation of games and nesting is more difficult [Carletta et al., 1997]. Usually there is agreement in determining the game category, but especially where utterances of interlocutors are overlapping and where there are situations of misunderstanding it appears difficult to agree on the placement of game boundaries.

In dialogue systems that use CGT, the model of conversation is usually conceived as outlined by [Kowtko et al., 1992]. For instance, the system described by Williams [1996], called MailSec, is an automated directory enquiry system that keeps a stack of conversational games in addition to the dialogue history (modelled by a list of previous utterances). Each conversational game has a beginning and an end move that in the minimal case roughly corresponds to adjacency pairs. Because games can be nested, the conversational games are maintained on a push-down stack. Another example of a system in which CGT has a central role is a travel planner presented by Lewin et al. [1993].

2.4.5 Information-state update approaches

In the information state update (ISU) approach [Traum et al., 1999; Traum and Larson, 2003], utterances are analysed in terms of their update effects on the so-called information state of the interlocutor. The information state refers to the information an interlocutor keeps track of during the conversation, and may contain diverse kinds of information, which may be structured in a variety of ways (e.g. by stacks, queues, sets,
et cetera). Furthermore, the ISU approach makes use of a set of dialogue moves that will trigger the update of the information state once they are recognised or generated. A set of update rules is defined, allowing the updating of the information state based on the moves observed and the current information state. An update strategy is used to decide which rules to select at a given point.

The information state itself resembles in essence often the dialogue game board proposed by Ginzburg [1996]. There is a division of information which is private to the interlocutor (beliefs and goals) and information which is shared (shared beliefs, questions under discussion, dialogue move history). A simple example of such an information state is depicted with the following attribute-value structure, in which QUD denotes the question under discussion and LM denotes the last move:

```
PRIVATE
  Beliefs Set(Proposition)
  Agenda Stack(Action)

SHARED
  Beliefs Set(Proposition)
  QUD Stack(Question)
  LM Move
```

An advantage of the ISU approach is that it is easy to incrementally increase the complexity of the state representation and dialogue strategy. However, the more complex the state representation, the more difficult the evaluation of the system behaviour will be. Bos points out in [Bos et al., 2003] that even for a relatively small set of rules, system developers tend to lose the overview of the intended behaviour of their system.

An application of the ISA approach is Trindikit [Larsson and Traum, 2000], which allows to specify, test, and compare various information-state update approaches. An example of a system that has been developed using this toolkit is GoDiS [Larsson et al., 2000]. Another toolkit that allows the construction of ISA based dialogue systems is DIPPER [Bos et al., 2003], an architecture that is intended for prototyping spoken dialogue systems and comes with a dialogue management component based on the ISU approach.

### 2.4.6 Collaborative-agent based approaches

In collaborative agent-based approaches, a dialogue is explicitly modelled as a collaborative process between intelligent agents that work together to achieve a mutual understanding of intentions and goals of the interlocutors.

Where dialogue grammars and plan-based approaches generally concentrate more on the structure of the tasks in the application domain, the collaborative agent-based approach focuses on inferring the motivations behind communication based on communication principles of rational agency. In doing so they model the beliefs and belief dynamics of each interlocutor and include a model of grounding to keep track of shared beliefs and goals.
An important contribution to rational agency in communication is the work of Cohen and Levesque [1990a,b], who developed a formal framework for expressing intention and communication. This approach is based on theories of intentionality in which the agent’s mental structures and attitudes concern beliefs and goals, most notably on the BDI (beliefs, desires, intentions) model [Bratman, 1987].

The ARTIMIS system [Sadek, 1999] implements a rational agent-based approach based on a formal theory of interaction. This theory involves a set of generic axioms which models, in a homogeneous logical framework, principles of rational behaviour, communication, and cooperation. It thus supports the rational aspect of an autonomous communicating agent, which, in our case, is the kernel of a cooperative spoken dialogue system. It is expressed in a first-order (multi)modal logic of mental attitudes (belief, uncertainty, and intention) and actions. The implementation includes an inference engine, specifically designed to reason using the axioms and rules of this kind of logic in their specific syntactical form.

Collaborative agent-based approaches have the advantage of being able to deal with more complex communication problems and dialogues that involve collaborative problem solving or negotiations. A disadvantage is the complexity of the system: engineering a robust agent-based collaborative dialogue system takes much more time and effort than most of the other approaches.

Examples of systems that use an agent-based approach are a system that facilitates conversation between Air Traffic Control and a pilot on landing procedures [Novick and Ward, 1993], the TRAINS-93 dialogue manager [Traum and Geneve, 1996], ARTIMIS [Bretier and Sadek, 1997], COLLAGEN [Rich and Sidner, 1998], and TRIPS [Ferguson and Allen, 1998]. Collaborative agent-based models can also be information-state based, such as the model described by [Keizer and Morante, 2006]. Other recent work that aims to deal with more complex dialogue is the collaborative problem solving model of Blaylock and Allen [2005].

### 2.4.7 Probabilistic dialogue models

Many dialogue models, such as the early frame-based approaches and the information state approach, rely on hand-crafted deterministic rules for interpreting dialogue acts and updating the dialogue state. More recently, there is an increasing interest in using probabilistic dialogue models: models that are obtained by statistically learning an optimal dialogue strategy from dialogue data. A dialogue system can use this strategy to determine, for example, when to generate a confirmation or when to take initiative. Such models assume the dialogue to be a Markov Decision Process (MDP) [Levin and Pieraccini, 1997; Levin et al., 1998], or a Partially Observable MDP (POMDP) [Roy et al., 2000; Zhang et al., 2001]. MDPs require the dialogue state to be fully observable; POMPDPS take into account that it is also possible that only part of the state may be observable (see e.g. Bui [2008]).

To learn an optimal dialogue strategy, reinforcement learning [Sutton and Barto, 1998] is often used. It is a machine learning method in which the machine learner
interacts with a dynamic environment and receives feedback. This feedback comes in the form of positive or negative reward and allows the machine learner to optimise its actions in order to maximise the overall reward.

2.5 Discussion

In speech act theory it is proposed that a speech act, or illocutionary act, contains a propositional content (a truth-conditional expression) and an indicator of the illocutionary force, specifying what to do with the propositional content in a communicative game. Apart from the classifications of speech acts proposed by Austin and Searle, there have been numerous proposals for alternative classifications that seem equally plausible. As a result, also the variety of proposed dialogue act taxonomies is high, and the question can be raised what the criteria are for considering one taxonomy better than the other. Arguably, the only sensible way to evaluate theories of linguistic communication is to test them empirically and comparatively by adopting them in a dialogue system and evaluate user interaction.

Important to realise is that most accounts of speech acts, being in the line of the work of Austin and Grice, present a speaker-centric conception of meaning in communication in which acts are defined and explained in terms of intentions. This has epistemological implications for assigning speech acts to utterances of dialogue corpora. As no one but the speaker has direct access to the illocutionary acts that are being conveyed, communication researchers functioning as passive observers can occasionally identify illocutionary acts only a posteriori, based on the perlocutionary effect in subsequent interaction. Even then, perlocutionary effects which the speaker intends not to produce or not intends to produce may not be addressed by the speaker later on in the conversation, keeping the observer unaware of, or even misguided about, the true intention of the speaker. Amongst assumptions in speech act theory that have received critical response\textsuperscript{2}, the speaker-centric notion of meaning in early speech act theory that Austin proposed is prominent.

Most of the taxonomies of speech acts or dialogue acts rely on the assumption that there is a correlation between form and function and that the occurrence of a speech act is evidenced by linguistic marking, or by so-called illocutionary force indicating devices (IFIDs). Examples of IFIDs are intonational pitch, performativity of verbs, word order, stress, et cetera. However, identification by means of IFIDs is not reliable in the sense that utterances do not always have clear indicators. For example, the sentence “It is hot in here.” could be understood as a complaint, a request, an assertion, or even an order. This aspect should be taken into account when machine learning is applied in the automatic identification of speech acts or dialogue acts, as using only surface features from utterances may not be sufficient. In this context, dialogue acts could be conceived in two ways: as units encoding the full intentional meaning of utterances in dialogue,

\textsuperscript{2}For example, see [Levinson, 1981] and [Marcu, 2000].
or as derived or intermediate concepts within a more comprehensive account of rational communicative behaviour. This difference in conception is to some extent typical for the major difference between Conversational Acts Theory and Dynamic Interpretation Theory. Where CAT accounts for grounding by dedicated grounding acts that can be subsumed in higher-level acts (speech acts and argumentation acts), the aspects of grounding in DIT are not modelled on the level of communicative functions or dialogue acts, but are a product of the interaction of context-update mechanisms, triggered by dialogue acts, with the belief state (cf. [Bunt et al., 2007; Morante Vallejo, 2007]). Another notable difference between CAT and DIT is that the communicative functions in DIT are grouped in dimensions, which themselves are not related hierarchically like in CAT.

The taxonomies of dialogue acts that have been discussed so far have a substantial theoretical basis. There are many other proposed dialogue act taxonomies and tagsets which are motivated by specific research questions. Additionally, many of them do not describe acts only, but also include the denotation of events that are useful to describe in conversation, such as abandoned utterances or whether the utterance is a response to a previous utterance.

The use of acts, act taxonomies, or sets of tags that describe certain phenomena in conversation, other than only giving a speaker-centric intentional account of meaning in conversation is peripherally relevant, but is not within the scope of the research presented in this dissertation.³

The various practical models for dialogue that have been described all have their strengths and weaknesses depending on the kind of dialogues that are considered. For instance, information-seeking task-oriented dialogues on simple domains, such as dialogues on train travel planning, work very well with frame-based models as the task is central and the interaction is mostly, if not always, single-initiative (from the information provider). If the interaction is mixed-initiative and should also be able to deal in a flexible way with communication problems, frame-based models are less suitable, and other models, such as dialogue grammars, plan-based approaches, and information-state approaches, become more suitable. The most powerful models are a synthesis of collaborative agency with information state modelling, as they are flexible, adaptable, and can interact in a more natural way by elaborate feedback mechanisms, belief modelling, and reasoning.

There are few approaches that combine characteristics of the dialogue-planning and information-state update approaches. For example, the ARTIMIS system [Sadek, 1999] is based on a framework that allows the representation of behaviour principles (rationality, communication, cooperation, etc.) and performs reasoning about mental attitudes (beliefs, uncertainty, intentions) and communicative acts.

³For a discussion of the construction and use of taxonomies, see e.g. Traum [2000].
2.6 Summary

In this chapter, three aspects of modelling dialogue have been addressed: meaning in conversation, the kinds of meaning that could be distinguished in dialogue, and the most important paradigms and formalisations in dialogue modelling.

It has been argued that language can be characterised as action, as characterised by illocutionary forces or communicative functions and that an utterance has at least one communicative function, i.e. performs at least one dialogue act, that brings about a change in the belief states of the interlocutors. The determination of the communicative functions of an utterance does not depend on form alone, but requires taking into account a rather complex interaction between form and context. This context has several dimensions, such as a linguistic one (the dialogue history), the task domain(s) being discussed, the roles of the interlocutors, et cetera.

This context is dynamic in that beliefs and intentions change over time. Moreover, utterances may have multiple functions at the same time that address aspects of communication which are characteristic for conversation, such as maintaining contact and understanding, and hence motivates extending classical speech act theory to a theory of dialogue acts. The multifunctionality of utterances is taken into account in multidimensional dialogue act schemes such as DAMSL and DIT++.

The most prominent approaches to model dialogue as a generative activity were reviewed, of which agent-based and information-state based systems allow the most natural and rich interaction.
Chapter 3
Annotating dialogue with dialogue acts

In this chapter, the annotation is discussed of conversation with dialogue acts. This discussion includes an overview of the variety in dialogue corpora that may be considered, and describes the data sets that are used in subsequent chapters. Furthermore, the notion of “reliability” is identified as an essential aspect of dialogue act annotation and a prerequisite for interpreting automatic dialogue act recognition, and a proposal is developed to assess reliability using different kinds of annotators. A reliability analysis of annotating with the DIT taxonomy is presented, and a refined taxonomically weighted way of measuring inter-annotator agreement is proposed.

3.1 Introduction

The systematic analysis of dialogue relies on the availability of annotated corpora. Annotating the utterances in dialogue corpora with the communicative functions of the dialogue acts that they express, may serve three purposes.

First, dialogue acts can be studied for their appearance in empirical data: statistical analysis (frequency, cooccurrence patterns) and the identification of contextual properties of dialogue acts support systematic research of phenomena. Second, corpora of dialogues with associated dialogue acts can be used to train machine learning algorithms to recognise dialogue acts automatically, or to automatically find those surface features of dialogues and utterances that appear to be the most important for the recognition task. Third, dialogue act annotated corpora can be used to probabilistically learn optimal dialogue strategies.

In this chapter, the practical and methodological aspects of collecting data to support empirically-based analysis of characteristics of dialogue acts are addressed. Sec-
tion 3.2 describes the range of characteristics of dialogue corpora and tagsets that will be considered. This contextualises the datasets that are used in the subsequent chapters. The corpora of spoken dialogue that are used in next chapters are described in Section 3.3. Most of these corpora have been the basis for layers of linguistic annotation such as part-of-speech, coreference, semantic roles, speech acts, dialogue acts, et cetera. In this chapter the focus is on those corpora that are used throughout this dissertation with particular attention for a corpus that has been compiled to allow the analyses and experiments that are presented in the following sections and subsequent chapters. Section 3.4, based on joint work that has appeared as [Geertzen and Bunt, 2006], deals with evaluating the applicability of tagsets in general and that of DIT++ in particular. Another methodological consideration of dialogue act annotation is what type of annotators to use and the perspective annotators should take with respect to the interlocutors of the conversation (Section 3.5, which is based on joint work that has appeared as [Geertzen et al., 2008]). The applicability of tagsets is revisited in Section 3.6, which looks into the gain of simplifying tagsets that feature hierarchical relations and group fine-grained dialogue into higher concepts. Finally, the considerations put forward in this chapter are discussed and summarised (Section 3.7).

### 3.2 Aspects of corpora and tagsets

In characterising or building a dialogue corpus and obtaining annotations of dialogue acts there are decisions to be made on the kind of data and the kind of annotation that is desired: As for the kind of dialogue data, there are at least the following considerations:

- **Domain**: what are the dialogues about. Familiar domains are making appointments (e.g. VERBMOBIL, [Alexandersson et al., 1998]), planning a travel (e.g. OVIS, [Strik et al., 1997]), how to operate devices (e.g. DIAMOND, [Geertzen et al., 2004]), or to solve a collaborative task (e.g. MONROE, [Stent, 2000] or MAPTASK, [Anderson et al., 1991]). Usually, dialogue is about a single, restricted domain, but more recently there is more attention for multiple-domain or open-domain (unrestricted) conversation.

- **Modalities**: what modalities are recorded. Is there one modality, such as text (e.g. IRC logs) or speech, or are also non-verbal modalities captured, such as gestures, body and head movements, gaze, and facial expression.

- **Role of interlocutors**: what role do interlocutors play and how do they relate. E.g. in information-seeking dialogues one interlocutor provides information and the other requests information; in instructional dialogue there usually is an instructor and a follower.

- **Participant relationships**: what relationship do the participants have with respect to each other given the context of the dialogue. Clark [1996] proposed a dynamic
model of participant relationships in which the participant that produces the illocutionary act is the speaker and the participant who is the dialogue partner in the joint action that is performed is the addressee. Other participants that take part in the conversation but are not currently being addressed are called side participants. Last, there may be others, called the overhearsers, that do not take part in the conversation. Overhearsers can be openly present (bystanders) or listening without the awareness of the speaker (eavesdroppers).

- **Type of interlocutor:** are the interlocutors all human, all machine, a mix of both, or is there simulation of one kind by the other.

Elaborating on the above-mentioned type of interlocutors, the main consideration is if the interaction to be studied is completely human (e.g. human-human dialogue), or if it does involve human interaction with a machine (e.g. human-machine dialogue). Again, the motivations of the research determine what kind of interactions to collect and the underlying question here is if human-machine interaction should be like human-human interaction.\(^1\)

To record human-machine interaction when there is no dialogue system available, the system is often simulated by a human. This way of data acquisition in interaction experiments is called the ‘Wizard of Oz’ (WoZ) method [Kelley, 1984], called after one of the characters in a similarly named book by Lyman Frank Baum. In this method, the behaviour of the dialogue system is simulated by one of the experimenters while the subjects that interact with the ‘system’ assume the dialogue system to be genuine. In this way, the effect of diverse dialogue management strategies can be tested without the availability of a full-fledged dialogue system. WoZ studies have been widely used in designing and testing dialogue systems (cf. [Fraser and Gilbert, 1991] and [Dahlbäck et al., 1993]).

Finally, it could be insightful to study machine-machine interaction. Particularly in the evaluation of spoken dialogue systems, full user simulation might be beneficial because it allows to avoid costly and time-consuming human interaction during prototyping of the system.

As for the dialogue act tagset, there are other considerations to take into account:

- **Purpose:** what is the purpose of the tagset. Study of a specific phenomenon in dialogue may not require the annotation of all dialogue acts of a taxonomy. The purpose of the annotations also determines the annotation guidelines to be followed. For instance, when the purpose is to obtain learning data for training a dialogue system that will function as an interlocutor, depending on the learning algorithm, it may be necessary for an annotator to put himself in the position of the interlocutor at whom the utterance was addressed.

\(^1\)Additionally, people tend to adapt their behaviour and speaking style to the machine due to e.g. awareness of the artificial agent. Hence, the assumption that human-human dialogue is adequate to serve as a generative model for human-machine interaction is questionable.
• **Granularity:** how detailed are the dialogue acts to be distinguished. Is it sufficient to know if an utterance is QUESTION or STATEMENT, or is it desired to distinguish more fine-grained differences (e.g. PRO-QUESTION and SET-QUESTION instead of QUESTION).

• **Reliability:** what minimum level of reliability of the assigned dialogue acts is desired or needed.

For theoretical analysis of dialogue, it is desirable to have a tagset which expresses fine-grained differences in intentions of interlocutors. For practical systems, by contrast, it might not be useful to discriminate between some of these differences. In these cases it makes sense to reduce the granularity of the tagset by grouping fine-grained dialogue acts into more general dialogue acts as long as the tagset is rich enough to be useful for the application. There may be two advantages for such a reduction. First, the inter-annotator agreement for the reduced tagset is usually higher, as fine-grained differences that may cause confusion and disagreement are eliminated. Second, due to the reduction the remaining dialogue acts occur more frequently, which is helpful for automatic classification using machine learning. Moreover, as a result of the reduction, there are more instances of occurrences of the same dialogue act, which reduces data sparsity and as such is also helpful for machine learning an automatic classifier.

The dialogue act tagset of the taxonomies introduced in Section 2.3.6 (DAMSL) and Section 2.3.7 (DIT++) are both intended to be domain-independent and generally applicable to any dialogue. As the dialogue corpora that are used in this work are annotated either with DAMSL or DIT++, it is relevant to assess the reliability and applicability of dialogue act annotations based on these tagsets.

### 3.3 Dialogue corpora

Various corpora of spoken human-human dialogue have been compiled and are available for research. The most notable are: HCRC Maptask (giving directions on a map)\(^2\), Switchboard (telephone speech)\(^3\), Verbmobil (appointment scheduling)\(^4\), TRAINS (arranging railroad freight)\(^5\), and ATIS (flight reservation)\(^6\). More recently, there has been an increasing interest in multiparty interaction, as large corpora such as the ICSI Meeting Corpus (meetings of work teams)\(^7\) and the AMI Meeting corpus (meetings of industrial design teams)\(^8\) have come available to the research community.

The speech corpora from which data sets are drawn that are used throughout this thesis are explained in the following sections.

\(^2\)[Anderson et al., 1991]
\(^3\)[Godfrey et al., 1992]
\(^4\)[Alexandersson et al., 1998]
\(^5\)[Allen et al., 1994]
\(^6\)[Dahl et al., 1994]
\(^7\)[Janin et al., 2003]
\(^8\)[Carletta, 2006]
3.3.1 **MONROE (2000)**

The MONROE Corpus\(^9\) [Stent, 2000] consists of human-human, mixed-initiative, task-oriented dialogues about disaster handling. The interlocutors are engaged in collaborative problem-solving, mixed initiative interaction, which involves a scenario at an emergency control centre: an instructor (U) receiving incoming information about a disaster, and a remote subject (S) initially knowing nothing about the task. An example dialogue is shown in Figure 3.1.

```
147 s so that we can send it over to <SIL> jefferson <SIL> and pick this person up <SIL> while the first ambulance still picks up <SIL> s <SIL> people as <SIL> uh <SIL> planned earlier
148 u ee <SIL> yeah <SIL> okay <SIL>
149 s do <SIL> do you think that will work
150 u yeah that’ll work too
151 i mean <SIL> d <SIL> i
152 s <SIL> okay so let’s do that
153 right
154 so we’ll <SIL> send the helicopter twice to the airport
155 u right
156 s and the ambulance <SIL> one ambulance there right now
157 u mm hm <SIL>
```

Figure 3.1: Conversation excerpt from the MONROE corpus.

For eight dialogues speech has been manually transcribed, segmented into utterances and turns, and annotated with the DAMSL tag set, resulting in a data set of 2,897 speaker utterances that are segmented into 1,701 turns (on average 189 turns per dialogue). Each utterance can have multiple communicative functions in four layers [Allen and Core, 1997]; there is almost always at least one function assigned to an utterance. The MONROE corpus is annotated with 13 main DA types that can further contain arguments.

The reliability of the dialogue act annotations have been tested by using two coders, resulting in kappa scores ranging from .6 to .9 depending on the DAMSL dimension [Stent, 2000, p. 9].

3.3.2 **ICSI Meeting (2002)**

The ICSI Meeting Corpus\(^10\) (Janin et al. [2003]) is a collection of 75 meetings collected at the International Computer Science Institute in Berkeley during the years 2000-

---

\(^9\)For more information and availability, see the project website (http://www.cs.rochester.edu/research/speech/monroe/).

\(^10\)For more information and availability, see the project website (http://www.icsi.berkeley.edu/Speech/mr/) and the Linguistic Data Consortium (http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2004T04).
2002. The meetings included are ‘natural’ meetings in the sense that they would have occurred anyway: they are generally regular weekly meetings of various ICSI working teams, including the team working on the ICSI Meeting Project. In recording meetings of this type, the aim was to capture meeting dynamics and speaking styles that are as natural as possible given that speakers are wearing close-talking microphones and are fully aware of the recording process. The speech files range in length from 17 to 103 minutes, but generally run just under an hour each.

This corpus consists of 75 word-level transcripts (one transcript file per meeting), time-synchronised to digitised audio recordings. There are approximately 795K words and 13K unique words in the transcripts.

The sample in Figure 3.2 illustrates an interaction with three dialogue participants.

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>c1</td>
<td>um ... so far I have thought of it as sort of adding it onto the modeller knowledge module</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>c0</td>
<td>that is the d-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>c3</td>
<td>hmm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>c0</td>
<td>ok</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>c0</td>
<td>yeah</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.2: Conversation excerpt from the ICSI corpus.

The meetings were recorded with close-talking and far-field microphones. The transcripts were based mostly on the close-talking microphones, either separately or blended together in a so-called ‘mixed’ channel. The focus of the transcripts was on capturing the flow of audible events, especially the words which were spoken, and who spoke them.

There are a total of 53 unique speakers in the corpus. Meetings involved anywhere from three to 10 participants, averaging six. The corpus contains a significant proportion of non-native English speakers, varying in fluency from nearly-native to challenging-to-transcribe.

3.3.3 MRDA (2004)

The MRDA corpus [Shriberg et al., 2004] is a companion set of segmentations and annotations on the ICSI Meeting Corpus, which consists of 75 non-scenario based meetings that each are roughly an hour in length. On average, there are about six English speakers, native and non-native, per meeting. Most of the meetings were group discussions about the ICSI meeting recorder project itself or on topics in natural language processing.

The utterances in the MRDA corpus have been annotated with a modified version of the SWBD-DAMSL tagset [Jurafsky et al., 1997], in which a dialogue act is a combination of at least one general tag, with a variable number of possible specific tags.
attached. There are 11 general tags. Tags in this dataset are thus mutually exclusive, which is a major difference from the MONROE material. The MRDA data contains 51,452 turns (on average 826 turns per dialogue).

Reliability scores in assigning the MRDA tags has been tested using three coders, obtaining an average agreement score of .77 kappa.

### 3.3.4 DIAMOND (2005)

The DIAMOND corpus consists of recorded interactions in Dutch and has been compiled to support development of dialogue theory (most notably to refine and elaborate the DIT framework), to study dialogue phenomena, to provide test material for dialogue system evaluation and dialogue system prototyping, and to provide material for some of the experiments reported in this dissertation (cf. [Geertzen et al., 2004]).

The corpus consists of two kinds of video-recorded interaction: human-machine (12 sessions of approximately 45 minutes) and human-human (four sessions of approximately 30 minutes). The first kind is obtained in a Wizard-of-Oz setting; the second kind is on exactly the same domain with the same scenario. For both kinds, the subjects had to get familiar with a fax device by completing several tasks related to the fax device, such as sending a fax message, programming the list of telephone numbers, or changing the settings of the device. Depending on the kind of interaction, during the sessions subjects could query a dialogue system (human-machine setting) or a help desk (human-human setting). The fax device and the scenario were sufficiently complex for the subjects to make interaction with the dialogue system or help desk necessary.

The sample in Figure 3.3 illustrates a representative conversation.\(^\text{11}\)

\begin{verbatim}
1 a hmm..
2 a let me have a look
3 a I already have a question now
4 b yes how can I help you?
5 a eh I would like to eh make also a copy also from the back side
6 a and does he have to on the same document or does it have to go on another?
7 b ehm..
8 a a new sheet of paper I mean
9 b ehm..
\end{verbatim}

Figure 3.3: Dialogue excerpt from the DIAMOND corpus.

The audio recordings from the experiments are transcribed orthographically on utterance level using PRAAT\(^\text{12}\) and are annotated with dialogue acts (communicative functions and semantic content type information) using the DIT++ tagset.

---

\(^{11}\)Translated freely from Dutch.

\(^{12}\)A speech analysis tool, available at: \url{http://www.fon.hum.uva.nl/praat/}.
3.4 Applicability of dialogue act tagsets

3.4.1 Introduction

The reliability of applying a tagset for annotation is usually expressed by the agreement between multiple annotators on the same dialogue data. Agreement is sometimes measured as a percentage of the cases on which the annotators agree, but more often expected agreement is taken into account in using the kappa statistic [Cohen, 1960; Carletta, 1996], which is given by:

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$  \hspace{1cm} (3.1)

where $p_o$ is the observed proportion of agreement and $p_e$ is the proportion of agreement expected by chance. The value of $\kappa$ expresses agreement taken into account expected agreement. There are a few proposals in literature about agreement, for how to interpret $\kappa$ related scores. Two influential scales for interpreting $\kappa$ scores that are proposed in literature are those by Landis and Koch [1977] and Krippendorff [1980]. These scales are listed in Table 3.4.1.

<table>
<thead>
<tr>
<th>$\kappa$</th>
<th>score</th>
<th>agreement</th>
<th>$\kappa$</th>
<th>score</th>
<th>agreement</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>$\leq \kappa \leq 0.20$</td>
<td>slight</td>
<td>$\leq \kappa \leq 0.67$</td>
<td>not tentative</td>
<td></td>
</tr>
<tr>
<td>0.20</td>
<td>$0.20 &lt; \kappa \leq 0.40$</td>
<td>fair</td>
<td>$0.67 &lt; \kappa \leq 0.80$</td>
<td>tentative</td>
<td></td>
</tr>
<tr>
<td>0.40</td>
<td>$0.40 &lt; \kappa \leq 0.60$</td>
<td>moderate</td>
<td>$0.8 &lt; \kappa \leq 1.00$</td>
<td>reliable</td>
<td></td>
</tr>
<tr>
<td>0.60</td>
<td>$0.60 &lt; \kappa \leq 0.80$</td>
<td>substantial</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.80</td>
<td>$0.80 &lt; \kappa \leq 1.00$</td>
<td>almost perfect</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The scales provided in Table 3.4.1 are indicative. As pointed out by Krippendorff [1980], what makes an acceptable level of agreement depends on what one intends to do with the coding. Carletta [1996] argues that the scale proposed by Krippendorff, being intuitive and acceptable to researchers in content analysis, might be too optimistic for use in dialogue, as:

“... coding discourse and dialogue phenomena, and especially coding segment boundaries, may be inherently more difficult than many previous types of content analysis ...”
Important to note is that comparison of the $\kappa$ scores across various studies should be done very carefully keeping in mind that the comparison is likely not to be reliable since distributional characteristics of the data within a class can affect the scores for this class considerably.

In the following section (3.4.2), annotator agreement for DAMSL and DIT++ is given. For DAMSL, reliability measures have been published in several studies (c.f. [Core and Allen, 1997; Stent, 2000]). For DIT++, such a reliability evaluation is presented in this chapter. Based on certain problems with the de facto reliability metric and characteristics of the DIT taxonomy, a derived reliability metric is proposed (Section 3.4.3 & 3.4.4) and applied (Section 3.4.5).

### 3.4.2 Annotator agreement for DAMSL and DIT++

Inter-annotator agreements have been calculated with the purpose of qualitatively evaluating tagsets and individual tags. For DAMSL, the first agreement results were presented in [Core and Allen, 1997], based on the analysis of TRAINS 91-93 dialogues [Gross et al., 1993; Heeman and Allen, 1995]. In this analysis, 604 utterances were tagged by mostly two annotators. Following the suggestions by Carletta [1996], Core et al. consider kappa scores above 0.67 to indicate significant agreement and scores above 0.8 reliable agreement. Another more recent analysis was performed for 8 dialogues of the MONROE corpus [Stent, 2000], counting 2897 utterances in total, processed by two annotators for 13 DAMSL dimensions. The resulting agreement scores are listed in Table 3.2. Other analyses apply DAMSL derived schemes (such as SWBD-DAMSL) to various corpora (e.g. [Di Eugenio et al., 1998; Shriberg et al., 2004]).

Existing work on annotator agreement analysis has mostly involved only two annotators. It may be argued that especially for annotation of concepts that are rather complex, an odd number of annotators is desirable. First, it allows having majority agreement unless all annotators choose entirely different. Second, it allows to deal better with the undesirable situation that one annotator chooses quite differently from the others.

In calculating the agreement scores for DIT++, the above-mentioned consideration is taken into account by using the annotations of three annotators, using the method proposed by Davies and Fleiss [1982]. For this purpose, a set of task-oriented dialogues in Dutch is annotated. To account for different complexities of interaction, both human-machine and human-human dialogues are considered. Moreover, the dialogues are drawn from various corpora: 193 human-machine utterances from OVIS [Strik et al., 1997], 131 human-machine utterances and 114 human-human utterances from DIAMOND (see Section 3.3.4), and 120 human-human utterances from a collection of Dutch Map Task dialogues [Caspers, 2000], summing up to 558 utterances in total.

Six undergraduate students annotated the selected dialogue material. They had been introduced to the DIT++ annotation scheme and the underlying theory while

---

13See [http://dit.uvt.nl/](http://dit.uvt.nl/).
participating in a course on pragmatics. During this course they were exposed to approximately four hours of lecturing and few small annotation exercises. For all dialogues, the audio recordings were transcribed and the annotators annotated pre-segmented utterances for which agreement was established on segmentation beforehand. During the annotation sessions the annotators had — apart from the transcribed speech — access to the audio recordings, to the on-line definitions of the communicative functions in the scheme and to a very brief, 1-page set of annotation guidelines (specified in Appendix A). The task was facilitated by the use of an annotation tool (cf. [Geertzen, 2007]) that had been built for this occasion; this tool allowed the subjects to assign each utterance one DIT++ tag for each dimension without any further constraints. In total 1,674 utterances were annotated. The resulting agreement scores are listed in Table 3.3.

When the agreement statistics of Table 3.3 are evaluated, and κ scores above 0.67 are considered to be significant and scores above 0.8 considerably reliable, it can be concluded that the dimensions Turn-management and Social-obligations-management can reliably be annotated. For some dimensions (contact, own/partner com. management), the occurrences of functions in these dimensions in the annotated dialogue material were too few (n < 10) to draw any conclusions.

<table>
<thead>
<tr>
<th></th>
<th>MONROE</th>
<th></th>
<th></th>
<th>TRAINS</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>p_o</td>
<td>p_e</td>
<td>κ</td>
<td>p_o</td>
<td>p_e</td>
<td>κ</td>
</tr>
<tr>
<td>influence-on-listener</td>
<td>0.97</td>
<td>0.77</td>
<td>0.88</td>
<td>0.88</td>
<td>0.60</td>
<td>0.70</td>
</tr>
<tr>
<td>influence-on-speaker</td>
<td>0.95</td>
<td>0.73</td>
<td>0.83</td>
<td>0.88</td>
<td>0.87</td>
<td>0.15</td>
</tr>
<tr>
<td>info-request</td>
<td>0.98</td>
<td>0.81</td>
<td>0.90</td>
<td>0.00</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>abandoned</td>
<td>0.94</td>
<td>0.82</td>
<td>0.66</td>
<td>0.98</td>
<td>0.93</td>
<td>0.62</td>
</tr>
<tr>
<td>agreement</td>
<td>0.96</td>
<td>0.60</td>
<td>0.89</td>
<td>0.78</td>
<td>0.61</td>
<td>0.43</td>
</tr>
<tr>
<td>answer</td>
<td>0.98</td>
<td>0.85</td>
<td>0.86</td>
<td>0.95</td>
<td>0.72</td>
<td>0.81</td>
</tr>
<tr>
<td>conventional</td>
<td>1.00</td>
<td>0.99</td>
<td>0.88</td>
<td>0.00</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>exclamation</td>
<td>0.99</td>
<td>0.98</td>
<td>0.70</td>
<td>0.00</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>info-level</td>
<td>0.88</td>
<td>0.70</td>
<td>0.59</td>
<td>0.82</td>
<td>0.56</td>
<td>0.59</td>
</tr>
<tr>
<td>other-forward-function</td>
<td>0.99</td>
<td>0.90</td>
<td>0.88</td>
<td>0.92</td>
<td>0.85</td>
<td>0.48</td>
</tr>
<tr>
<td>performative</td>
<td>1.00</td>
<td>0.99</td>
<td>0.45</td>
<td>0.00</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>response-to</td>
<td>0.90</td>
<td>0.40</td>
<td>0.83</td>
<td>0.83</td>
<td>0.28</td>
<td>0.77</td>
</tr>
<tr>
<td>statement</td>
<td>0.93</td>
<td>0.41</td>
<td>0.88</td>
<td>0.82</td>
<td>0.47</td>
<td>0.67</td>
</tr>
<tr>
<td>understanding</td>
<td>0.96</td>
<td>0.56</td>
<td>0.91</td>
<td>0.83</td>
<td>0.59</td>
<td>0.58</td>
</tr>
<tr>
<td>unintelligible</td>
<td>0.98</td>
<td>0.98</td>
<td>0.31</td>
<td>0.99</td>
<td>0.98</td>
<td>0.14</td>
</tr>
</tbody>
</table>
### 3.3.3 Problems with standard kappa

Ever since its introduction in general [Cohen, 1960] and in computational linguistics in particular [Carletta, 1996], many researchers have pointed out that there are quite some problems in using $\kappa$ (e.g. [Di Eugenio and Glass, 2004]), one of which is the discrepancy between $p_o$ and $\kappa$ for skewed class distribution. Another is that the degree of disagreement is not taken into account, which is relevant for any non-nominal scale. To address this problem, a weighted $\kappa$ has been proposed [Cohen, 1968] that penalises disagreement according to their degree rather than treating all disagreements equally. It would be arguable that in a similar way, characteristics of dialogue acts in a particular taxonomy and possible pragmatic relatedness between them should be taken into account to express annotator agreement.

For example, a **PROPOSITIONAL QUESTION** and a **CHECK QUESTION** are more similar than an **OFFER** and a **CHECK QUESTION**. Consequently, disagreement involving the former two acts can be considered less than disagreement involving the latter two acts. For dialogue act taxonomies which are structured in a meaningful way, such as those that express hierarchical relations between concepts in the taxonomy, the taxonomic structure can be exploited to express how much annotators disagree when they choose different concepts that are directly or indirectly related. Recent work that accounts for some of these aspects is a metric for automatic dialogue act classification [Lesch et al., 2005] that uses distance in a hierarchical structure of multidimensional labels. Along similar lines, a kappa metric for partial agreement has been discussed in the Dialogue Group at the University of Tilburg [Bunt, 2005b], which formed the basis of the work presented in the following sections.

If the standard $\kappa$ statistic is applied to DIT++ annotations, as in the previous sec-

---

<table>
<thead>
<tr>
<th>Dimension</th>
<th>$p_o$</th>
<th>$p_e$</th>
<th>$\kappa$</th>
</tr>
</thead>
<tbody>
<tr>
<td>task</td>
<td>0.52</td>
<td>0.09</td>
<td>0.47</td>
</tr>
<tr>
<td>auto feedback</td>
<td>0.32</td>
<td>0.14</td>
<td>0.21</td>
</tr>
<tr>
<td>allo feedback</td>
<td>0.53</td>
<td>0.19</td>
<td>0.42</td>
</tr>
<tr>
<td>turn management</td>
<td>0.90</td>
<td>0.42</td>
<td>0.82</td>
</tr>
<tr>
<td>time management</td>
<td>0.91</td>
<td>0.79</td>
<td>0.58</td>
</tr>
<tr>
<td>contact management</td>
<td>1.00</td>
<td>0.53</td>
<td>1.00</td>
</tr>
<tr>
<td>own com. management</td>
<td>1.00</td>
<td>0.50</td>
<td>1.00</td>
</tr>
<tr>
<td>partner com. management</td>
<td>1.00</td>
<td>1.00</td>
<td>-</td>
</tr>
<tr>
<td>dialogue struct. management</td>
<td>0.87</td>
<td>0.48</td>
<td>0.74</td>
</tr>
<tr>
<td>social obl. management</td>
<td>1.00</td>
<td>0.19</td>
<td>1.00</td>
</tr>
</tbody>
</table>
tion, an important aspect of the annotation scheme concerning the differences between alternative tags would be ignored, and consequently the possible differences in disagreement between annotators using these tags. An aspect in which the DIT taxonomy differs from many other taxonomies for dialogue acts is that, as noted in Chapter 2 Section 2.3.5, communicative functions (CFs) within a dimension as well as general-purpose CFs are often structured into hierarchies in which a difference in level represents a relation of specificity. An example of one hierarchy in the group of general-purpose CFs is depicted in Figure 3.4.

![Figure 3.4: Two hierarchies of information-seeking general purpose functions.](image)

When annotators differ in that they assign different tags which both belong to the same hierarchy, then they differ in the degree of specificity that they want to express, but they agree to the extent that these tags inherit the same elements from tags higher in the hierarchy.¹４ Inter-annotator disagreement is in such a case much less than if they would choose two unrelated tags. This is for instance obvious in the example of the annotations of two utterances by two annotators as depicted in Figure 3.5.

<table>
<thead>
<tr>
<th></th>
<th>ann. 1</th>
<th>ann. 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A what do you want to know?</td>
<td>SET-Q PRO-Q</td>
</tr>
<tr>
<td>2</td>
<td>B can I print now?</td>
<td>PRO-Q CHECK</td>
</tr>
</tbody>
</table>

![Figure 3.5: Two utterances annotated with DIT++.](image)

For utterance 1, the annotators should be said simply to disagree (in fact, annotator 2 incorrectly assigns a PRO-Q function). Concerning utterance 2 the annotators also disagree, but from Figure 3.4 and the definitions of the dialogue acts given in Section 2.3.5 it can be concluded that the disagreement in this case is quite small, as a CHECK inherits the properties of a PRO-Q. Therefore, a black-and-white measure of agreement like the standard $\kappa$ would be less suitable and what is needed is a measure for partial annotator agreement.

¹４For instance, the IND-PROPQ in Figure 3.4 has the precondition that $S$ wants to know the truth of a given proposition, and the precondition that $S$ does not know whether $A$ knows the truth of that proposition. All the functions under IND-PROPQ share these preconditions.
To measure partial (dis-)agreement between annotators in an adequate way, it should not just be taken into account whether two tags are hierarchically related or not, but also how far they are apart in the hierarchy, to reflect that two tags which are only one level apart are semantically more closely related than tags that are several levels apart. This additional requirement will be taken into account when designing a weighted disagreement statistic in the next section.

### 3.4.4 Agreement based on structural taxonomic properties

The agreement coefficient needed should in the first place be *weighted* in the sense that it takes into account the magnitude of disagreement. Two such coefficients are weighted kappa ($\kappa_w$, [Cohen, 1968]) and alpha [Krippendorff, 1980]. For the purpose at hand, $\kappa_w$ is adopted for its property to take into account a probability distribution typical for each annotator, generalise it to the case for multiple annotators by taking the average over the scores of annotator pairs, and define a function to be used as distance metric.

Assuming the case of two annotators, let $p_{ij}$ denote the proportion of utterances for which the first and second annotator assigned categories $i$ and $j$, respectively. Then Cohen defines $\kappa_w$ in terms of disagreement rather than agreement where $q_o = 1 - p_o$ and $q_e = 1 - p_e$ such that Equation 3.1 can be rewritten to:

$$\kappa = 1 - \frac{q_o}{q_e}$$  \hspace{1cm} (3.2)

To arrive at $\kappa_w$, the proportions $q_o$ and $q_e$ in Equation 3.2 are replaced by weighted sums over all possible category pairs:

$$\kappa_w = 1 - \frac{\sum v_{ij} \cdot p_{oij}}{\sum v_{ij} \cdot p_{eij}}$$  \hspace{1cm} (3.3)

where $v_{ij}$ denotes the disagreement weight of judgements $i$ and $j$. To calculate this weight, a distance function must be specified as metric.

Defining a function to calculate the difference between a pair of categories requires to determine semantic-pragmatic relatedness between the two categories in the taxonomy. For any annotation scheme, whether it is hierarchically structured or not, an assignment could be made for each possible pair of categories to a value that expresses the semantic-pragmatic relatedness between the two categories compared to all other possible pairs. However, it is difficult to find universal characteristics for CFs to express relatedness on a rational scale. When a taxonomy is considered that is structured in a meaningful way, in this case one that expresses hierarchical relations between CF based on their effect on information states, the taxonomic structure can be exploited to express in a systematic fashion how much annotators disagree when they choose different but related concepts.

The assignment of different CFs to a specific utterance by two annotators represents full disagreement in the following cases:
1. the two CFs belong to different dimensions;

2. one of the two CFs is general-purpose; the other is dimension-specific;\(^{15}\)

3. the two CFs belong to the same dimension but not to the same hierarchy;

4. the two CFs belong to the same hierarchy but are not located in the same branch.
   Two CFs are said to be located in the same branch when one of the two CFs is an ancestor of the other.

If, by contrast, the two CFs take part in an ancestor-offspring relation within a hierarchy (either within a dimension or among the general-purpose CFs), then the CFs are related and this assignment represents partial disagreement. A distance metric that measures this disagreement, which is denoted $\delta$, should have the following properties:

1. $\delta$ should be a real number normalised in the range $[0...1]$;

2. Let $C$ be the (unordered) set of CFs.\(^{16}\) For every two CFs $c_1, c_2 \in C$, $\delta(c_1, c_2) = 0$ when $c_1$ and $c_2$ are not related;

3. Let $C$ be the (unordered) set of CFs. For every communicative function $c \in C$, $\delta(c, c) = 1$;

4. Let $C$ be the (unordered) set of CFs. For every two CFs $c_1, c_2 \in C$, $\delta(c_1, c_2) = \delta(c_2, c_1)$.

Furthermore, when $c_1$ and $c_2$ are related, it should be specified how distance between them in the hierarchy should be expressed in terms of partial disagreement. For this, the following aspects must be taken into account:

1. The distance in levels between $c_1$ and $c_2$ in the hierarchy is proportional to the magnitude of the disagreement;

2. The magnitude of disagreement between $c_1$ and $c_2$ being located in two different levels of depths $n$ and $n + 1$ might be considered to be more different than that between two levels of depth $n + 1$ and $n + 2$. If this is the case, the deeper two levels are located in the tree, the smaller the differences between the nodes on those levels. For the hierarchies in DIT, the magnitude of disagreement has been kept linear with the difference in levels, and independent of level depth;

---

\(^{15}\)This is in fact a simplification. For instance, an INFORM act of which the semantic content conveys that the speaker did not understand the previous utterance forms an act in the Auto-Feedback dimension, and a tagging to this effect should perhaps not be considered to express full disagreement with the assignment of the dimension-specific tag NEGATIVE-INTERPRETATION.

\(^{16}\)Strictly speaking, in DIT a dialogue act annotation tag is either (a) the name of a dimension-specific function, or (b) a pair consisting of the name of a general-purpose function and the name of a dimension.
Given the above-mentioned considerations, the following metric can be proposed:

\[
\delta(c_i, c_j) = a^{\Delta(c_i, c_j)} \cdot \beta(c_i, c_j) \cdot b^{\Gamma(c_i, c_j)}
\]  

(3.4)

where:

- \(a\) is a constant, with \(0 < a < 1\), expressing how much distance there is between two adjacent levels in the hierarchy;
- \(\Delta\) is a function that returns the difference in depth between the levels of \(c_i\) and \(c_j\);
- \(\beta(c_i, c_j) = 1\) if \(c_i\) and \(c_j\) are in the same branch of the hierarchy and \(0 < \beta(c_i, c_j) < 1\) otherwise;
- \(b\) is a constant for which \(0 < b \leq 1\), expressing in what rate differences should become smaller when the depth in the hierarchy gets larger. If there is no reason to assume that differences at greater depth in the hierarchy are smaller than differences at a lower depth, then \(b = 1\);
- \(\Gamma(c_i, c_j)\) is a function that returns the minimal depth of \(c_i\) and \(c_j\).

A matter of concern in this way of calculating \(\delta\) is the assignment of values to the parameters \(a\), \(b\), and \(\beta\). Plausible values for \(a\), \(b\), and \(\beta\) could be \(0.75\), \(1\), and \(0.5\), respectively.\(^{17}\)

To provide some examples of how \(\delta\) would be calculated, the general purpose functions in Figure 3.4 can be considered with using the values that are suggested above:

\[
\begin{align*}
\delta(\text{IND-PRO-Q, CHECK}) &= 0.75^2 \cdot 1 \cdot 1 = 0.563 \\
\delta(\text{PRO-Q, CHECK}) &= 0.75^1 \cdot 1 \cdot 1 = 0.75 \\
\delta(\text{POSI, NEGA}) &= 0.75^0 \cdot 0.5 \cdot 1 = 0.5
\end{align*}
\]

When the taxonomic distance is used as the weighting in Cohen’s \(\kappa\), a coefficient is obtained which is called taxonomically weighted kappa, denoted by \(\kappa_{tw}\):

\[
\kappa_{tw} = 1 - \frac{\sum (1 - \delta(i, j)) \cdot p_{oij}}{\sum (1 - \delta(i, j)) \cdot p_{eij}}
\]  

(3.5)

\(^{17}\)These values are somewhat arbitrary. To tune the constants, an experiment could be administered in which subjects rate the pragma-semantic similarity between instances of several functions in the hierarchies. The averaged scores per function-pair over all subjects could then be used to tune the constants or to calculate partial disagreement directly.
3.4.5 κ_{tw} statistics for DIT

Considering the DIT taxonomy, it may be argued that due to the many hierarchies in the taxonomy of the general-purpose functions, this is the part where most is to be gained by employing κ_{tw}.

Table 3.4 shows the statistics for each dimension, averaged over all annotation pairs. With annotation pair is understood the pair of assignments an utterance received by two annotators for a particular dimension. The figures in the table are based on those cases in which both annotators assigned a function to a specific utterance for a specific dimension. Cases where either one annotator does not assign a function while the other does, or where neither annotator assigns a function, are not considered. Scores for standard κ and κ_{tw} are given in the first two columns. The column pairs indicates on how many annotation pairs the statistics are based. The last column shows the ap-ratio. This figure indicates which fraction of all annotated functions in that dimension is represented by annotation pairs and as such gives an indication how much agreement there is in addressing a particular dimension in the first place. When #ap denotes the number of annotation pairs and #pa denotes the number of partial annotations (annotations in which one annotator assigned a function and the other did not), then the ap-ratio is calculated as #ap/(#pa + #ap).

Table 3.4: Scores for corrected κ and κ_{tw} per DIT dimension. p_o denotes observed agreement, p_e denotes expected agreement, ‘pairs’ indicates on how many annotation pairs the statistics are based, and ap-ratio indicates the agreement on annotating in a dimension.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>standard</th>
<th></th>
<th>weighted</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>p_o</td>
<td>p_e</td>
<td>κ</td>
<td>p_o</td>
</tr>
<tr>
<td>task</td>
<td>0.52</td>
<td>0.09</td>
<td>0.47</td>
<td>0.76</td>
</tr>
<tr>
<td>auto feedback</td>
<td>0.32</td>
<td>0.14</td>
<td>0.21</td>
<td>0.87</td>
</tr>
<tr>
<td>allo feedback</td>
<td>0.53</td>
<td>0.19</td>
<td>0.42</td>
<td>0.58</td>
</tr>
<tr>
<td>turn</td>
<td>0.90</td>
<td>0.42</td>
<td>0.82</td>
<td>0.90</td>
</tr>
<tr>
<td>time</td>
<td>0.91</td>
<td>0.79</td>
<td>0.58</td>
<td>0.91</td>
</tr>
<tr>
<td>contact</td>
<td>1.00</td>
<td>0.53</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>own communication</td>
<td>1.00</td>
<td>0.50</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>partner communication</td>
<td>1.00</td>
<td>1.00</td>
<td>-</td>
<td>1.00</td>
</tr>
<tr>
<td>dialogue structure</td>
<td>0.87</td>
<td>0.48</td>
<td>0.74</td>
<td>0.87</td>
</tr>
<tr>
<td>social obligations</td>
<td>1.00</td>
<td>0.19</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

We can observe that due to the use of the taxonomic weighting both feedback dimensions and the task dimension gain substantially in annotator agreement. When looking at the agreement statistics and κ scores above 0.67 are considered to be significant and scores above 0.8 to be considerably reliable, as is usual for κ statistics, the annotations in the dimensions Turn-management, Contact-management, and Social-
obligations-management are found to be reliable and those in Dialogue structure management to be significant. For some dimensions, the occurrences of functions in the annotated dialogue material were too few to draw conclusions. When also the ap-ratio is taken into account, only the dimensions Task, Time-management, and Social-obligations-management combine a fair agreement on functions with fair agreement on whether to annotate in these dimensions. For the other dimensions especially, the question should be raised for which cases and for what reasons the ap-ratio is low. This question asks for further qualitative analysis, which is beyond the scope of this thesis.18

3.4.6 Discussion

In the previous sections, it was argued that the proposed taxonomically weighted $\kappa_{tw}$ can express partial disagreement for taxonomies that contain hierarchical structures, like the DIT++ taxonomy. However, there are some issues that deserve further attention.

A question that might be raised in using $\kappa_{tw}$ as opposed to ordinary $\kappa$, is whether the interpretations of $\kappa$ proposed in literature also hold for $\kappa_{tw}$ statistics. This is ultimately an empirical issue, to be decided by which $\kappa_{tw}$ scores researchers judge to correspond to fair or near agreement between annotators.

Another point of discussion is the arbitrariness of the values of the parameters that can be chosen in $\delta$. Choosing different values may change the disagreement of two distinct CFs located in the same hierarchy considerably. Still, it may be argued that by interpolating smoothly between the intuitively clear cases at the two extreme ends of the scale, it is possible to determine reasonable values for the parameters that scale well, given the average hierarchy depth.

A more general problem, inherent in almost any (dialogue act) annotation activity is that when the possible factors that influence the agreement scores are considered, it is clear that they can be numerous. Starting with the tagset, unclear definitions and vague concepts are a major source of disagreement. Other factors are the quality and extensiveness of annotation instructions, and the experience of the annotators. These were kept constant throughout the reported experiment, but clearly the use of more experienced or better trained annotators could have a great influence. Then there is the influence that the use of an annotation tool can have. Does the tool give hints on annotation consistency (e.g. an ANSWER should be preceded by a QUESTION), does it enforce consistency, or does it not consider annotation consistency at all? Are the possible choices for annotators presented in such a way that each choice is equally well visible and accessible? Clearly, when these factors are not controlled sufficiently, there is the risk that what is measured does not express what is attempted to quantify: (dis)agreement among annotators about the description of what happens in a dialogue.

In the previous sections the agreement scores for Cohen’s unweighted $\kappa$ have been

---

presented and it is claimed that for annotation schemes with hierarchically related tags, a weighted $\kappa$ gives a better indication of (dis)agreement than unweighted $\kappa$. The $\kappa$ scores for some dimensions do not seem particularly high, but they become more interesting when looking at semantic-pragmatic differences between dialogue acts or CFs. Even though there are somewhat arbitrary aspects in weighting, when parameters are carefully chosen, a weighted metric gives a more accurate representation of the inter-annotator agreements. More generally, it is proposed that semantic-pragmatic relatedness between taxonomic concepts should be taken into account when calculating inter-annotator (dis)agreement. While the DIT$^{++}$ tagset was used, the weighting function that is proposed can be employed in any taxonomy containing hierarchically related concepts, since only the structural properties of the taxonomy are used.

Additionally, the DIT$^{++}$ tagset is evaluated quantitatively\textsuperscript{19}, per dimension an indication of its usability is obtained. The focus has been on agreement per dimension, but when a global indication of the difference in semantic-pragmatic interpretation of a complete utterance is desired, other aspects must be considered as well. A truly multidimensional study of inter-annotator agreement should not only take intra-dimensional aspects into account but also relate the dimensions to each other. Bunt and Girard [2005] argue that dimensions should be orthogonal, meaning that an utterance can have a function in one dimension independent of functions in other dimensions. This is a somewhat idealistic condition, since there are some functions that show correlations and dependencies across dimensions. For this reason it makes sense to try to express the effect of the presence of strong correlations, dependencies and possible entailments in a multidimensional notion of (dis)agreement. Additionally, it may be desirable to take into account the importance that a CF can have. It is widely acknowledged that utterances are often multifunctional, but it could be argued that in many cases an utterance has a primary function and secondary functions; for instance, if an utterance has both a task-related function and one or more other functions, the task-related function is typically felt to be more important than the other functions, and disagreement about the task-related function is therefore felt to be more serious than disagreement about one of the other functions. This might be taken into account by adding a weighting function when combining agreement measures over multiple dimensions.

Other future work on this specific analysis is more methodological in nature, quantifying the relative effect of the factors that may have influenced the scores that are found. This would bring an overview that provides insight in what exactly is evaluated. As for evaluating the tagset, a further analysis of cooccurrence matrices to identify frequent misannotations has been carried out in [Geertzen, 2006], and has been used to improve some of the definitions of the tags and to improve the clarity of the annotation guidelines.

In Artstein and Poesio [2008], objections to using a weighted metrics, such as $\kappa_{tw}$, are raised. In their thorough overview of inter-coder agreement measures used in com-

\textsuperscript{19}Kappa statistics are indicative. To get a full understanding of what the figures represent, qualitative analysis by using e.g. cooccurrence matrices is required.
putational linguistics, it is concluded that weighted metrics are not easy to interpret. However, while it is true that the absolute value of the weighted kappa is not easy to interpret, for the analyses presented only the differences between \( \kappa_w \)-values for different annotators are essential. Moreover, it should be stressed that quantitative indicative figures such as agreement scores should be complemented with qualitative analyses including cooccurrence matrices.

## 3.5 Naive and Expert annotators

### 3.5.1 Introduction

An issue in determining inter-annotator agreement is what kind of annotators to use. In [Carletta, 1996] it is argued that in annotating with schemes such as those in discourse and dialogue analysis there are no real experts, and that what counts is how totally naive annotators manage, based on written instructions. When totally naive annotators are used, however, factors such as the clarity of the written instructions and the interface of the annotation tool have a bigger impact on performance than when annotators are used who are familiar with the tagset and have a good overview of the annotation concepts that can be used. Moreover, when the aim is to obtain annotations that are as accurate as possible and the dialogue act tagset is rather complex, the use of expert annotators is more obvious. It can be argued that both evaluation based on naive annotators and evaluation based on expert annotators can provide indications of the usability of the tagset, but that evaluation based on naive annotators provides more insight in the clarity of the concepts in the tagset, whereas evaluation based on expert annotators provides an indication of how reliably the tagset can be applied when errors are ruled out that are due to deficiencies in conceptual understanding, to a lack of experience in using the annotation tool, or too little experience in annotation more generally.

When inter-annotator agreement scores for data annotated with a particular tagset indicate high reliability\(^{20}\) it is not guaranteed that there is high agreement on the assignment of the right concept. Even though it is not likely to happen often, annotators could agree in assigning a certain concept but disagree with an expert on what would be the correct concept to assign. Therefore, to obtain a reliable evaluation, agreement scores should ideally be complemented with scores on annotation performance, such as annotation accuracy: the proportion of correctly annotated utterances or functions in relation to the total number of utterances or functions. However, annotation accuracy can only be calculated when there is a reference annotation, a so-called gold standard, that is completely correct. An annotation can considered to be a gold standard when domain experts unanimously agree on each single assignment. Unanimous expert agreement on all items may at times not be achievable, certainly not with rather abstract, complex, or vague concepts, but by extensively discussing differences in opin-

\(^{20}\)In case of Cohen’s kappa, this is often taken to be between 0.8 and 1.0. For a general discussion, see e.g. Landis and Koch [1977]; Krippendorff [1980].
ion on what tag to assign, consensus may often be reached and those cases for which disagreement remains could be left out of the gold standard.

In recent work, a first assessment of reliability of annotation with the DIT++ tagset has been presented based on annotations by naive annotators [Geertzen and Bunt, 2006]. As the tagset is being used — directly or indirectly — in several projects, a more accurate and elaborate tagset evaluation using annotations by both naive and expert annotators is an insightful and useful endeavour.

In this section a study will be presented in which the difference in inter-annotator agreement of naive and expert annotators on task-oriented dialogue for the DIT++ dialogue act tagset are compared, and the accuracy of naive and expert annotation against a gold standard is assessed.

### 3.5.2 Contrastive experiment outline

To get an understanding how the annotations by naive annotators differ from those by expert annotators, the results of both annotator groups can be analysed in terms of inter-annotator agreement and tagging accuracy. The difference between these two aspects, already mentioned in the previous section, can be depicted as in Figure 3.6. Five imaginary annotators are scored on tagging accuracy, illustrating combinations of low and high values for agreement and accuracy.

![Figure 3.6: Relation between inter-annotator agreement (IAA) and tagging accuracy (Acc).](image)

The group’s inter-annotator agreement (IAA) is inversely proportional to the group’s standard deviation on tagging accuracy. There are two rather obvious hypotheses that can be formulated as far as differences between the two annotator groups are concerned: inter-annotator agreement and accuracy scores are both expected to be higher for expert annotators.
Naive annotators can be characterised as subjects that have not been linguistically trained but that have participated in an introductory session explaining the dialogue data, the dialogue act tagset, and the use of annotation tools. Expert annotators can be characterised as linguistically trained subjects that have experience in annotating dialogue and are thoroughly familiar with the tagset and the tools. In the role of naive annotators, six undergraduate students annotated the selected dialogue material. They had been introduced to the annotation scheme and the underlying theory as part of a course in pragmatics. During this course they had approximately four hours of lecturing and a few small annotation exercises. Two PhD students annotated as experts. They have been actively working with the annotation scheme and have annotated substantial parts of dialogue corpora. To calculate accuracy scores, i.e. to assess to what extent the annotators in both groups have annotated correctly, a gold standard is required. To obtain such a gold standard annotation, the two expert annotators have analysed and discussed the available annotations with a third expert and have established full agreement.

For all dialogues, the audio recordings were transcribed and the annotators annotated pre-segmented utterances for which full agreement had been established on segmentation beforehand. During the annotation sessions the annotators had, apart from the transcribed speech, access to the audio recordings, to the on-line definitions of the communicative functions in the scheme and to a very brief, 1-page set of annotation guidelines\(^\text{21}\). The task was facilitated by the use of an annotation tool that had been built for this occasion [Geertzen, 2007]. This tool allowed the subjects to assign each utterance one tag for each dimension without any further constraints. Both the naive and expert annotators could provide comments with each utterance for indicating problems, explaining the decision to choose a particular tag, or indicating that none of the available dimensions was addressed. The last mentioned case did not happen for the expert annotators and happened two times for the naive annotators.

The set of dialogues that were annotated for this experiment is the same as the one described and used in Section 3.4.2: 193 human-machine utterances from OVIS [Strik et al., 1997], 131 human-machine utterances and 114 human-human utterances from DIAMOND (see Section 3.3.4), and 120 human-human utterances from a collection of Map Task dialogues [Caspers, 2000], summing up to 558 utterances in total. On average, naive annotators needed 23.2 seconds to annotate each utterance where expert annotators needed 11.8 seconds.

### 3.5.3 Quantitative comparative results

Table 3.5 shows the inter-annotator agreement statistics for each dimension, averaged over all annotation pairs. With *annotation pair* is meant a pair of assignments an utterance received from two annotators for a particular dimension. The kappa figures in

\(^{21}\)Both the definitions and guidelines have been used and tested in earlier annotation sessions and have been improved over time as a result of feedback and analysis of disagreement. The dialogue act definitions and guidelines can be found at Section 2.3.7 and Appendix A, respectively.
Table 3.5: Inter-annotator agreement for naive and expert annotators, per dimension, drawn from the set of all annotation pairs. $p_o$ denotes observed agreement, $p_e$ denotes expected agreement, ‘pairs’ indicates on how many annotation pairs the statistics are based, and $ap$-ratio indicates the agreement on annotating in a dimension.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>naive annotators</th>
<th>expert annotators</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$p_o$</td>
<td>$p_e$</td>
</tr>
<tr>
<td>task</td>
<td>0.63</td>
<td>0.17</td>
</tr>
<tr>
<td>auto feedback</td>
<td>0.67</td>
<td>0.48</td>
</tr>
<tr>
<td>allo feedback</td>
<td>0.53</td>
<td>0.29</td>
</tr>
<tr>
<td>turn</td>
<td>0.67</td>
<td>0.44</td>
</tr>
<tr>
<td>time</td>
<td>0.87</td>
<td>0.84</td>
</tr>
<tr>
<td>contact</td>
<td>0.80</td>
<td>0.66</td>
</tr>
<tr>
<td>own comm.</td>
<td>1.00</td>
<td>0.50</td>
</tr>
<tr>
<td>partner comm.</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>dialogue struct.</td>
<td>0.80</td>
<td>0.30</td>
</tr>
<tr>
<td>social oblig.</td>
<td>0.95</td>
<td>0.28</td>
</tr>
</tbody>
</table>

The table is based on those cases in which both annotators assigned a function to a specific utterance for a specific dimension. For each annotator group, scores for observed agreement ($p_o$), expected agreement ($p_e$), and kappa ($\kappa_{tw}$) are listed in the first, second, and third column, respectively. These statistics are taxonomically weighted and as such take into account semantic-pragmatic relatedness of concepts. Table 3.6 is included to have an idea of what the disagreement scores are when standard kappa is used instead of $\kappa_{tw}$.

The column ‘pairs’ indicates on how many annotation pairs the statistics are based. The last column shows the $ap$-ratio: the fraction of all annotated functions in that dimension which are present in annotation pairs.

From Table 3.5, it is obvious that for almost all dimensions, expert annotators obtain substantially higher agreement than naive annotators, as was to be expected. Considering the $ap$-ratio’s for both annotator groups, it can be observed that for most dimensions expert annotators agree more on whether or not to assign a communicative function.

The scores for tagging accuracy are found in Table 3.7. Accuracy was calculated for both groups of annotators in two ways: by taxonomically weighted kappa with the gold standard (column $\kappa_{tw}$), and by means of taxonomically weighted percentage agreement with the gold standard (column $p_o$). For each annotator a taxonomically weighted kappa score is calculated with the gold standard. The resulting scores are averaged to obtain a single score for each group. This is done for each dimension in the tagset. Second, for each annotator group the percentage agreement is calculated by
similarly averaging individual percentage agreements. Note that both accuracy scores are slightly higher than the corresponding average scores for inter-annotator comparison. When we generalise over all dimensions and calculate a single accuracy score for each group, naive annotators score 0.67 and experts score 0.92. The considerably higher score for experts is not unexpected considering the per-dimension scores. It is particularly interesting to see if there are annotators that deviate substantially in accuracy from all the others in the group. For if this is the case this tells us more if the tagging accuracy per dimension is positively or negatively biased. The accuracy scores of individual annotators are visualised in Figure 3.7.

From this figure, we see that for the naive annotators (N1 until N6), there is more deviation from the group mean than for experts (E1 and E2). More importantly, annotator N6 deviates considerably from the other annotators in the group, causing the performance of the naive annotators to be biased positively. The two expert annotators

---

**Table 3.6:** $\kappa$ scores for dimensions where $\kappa$ and $\kappa_{tw}$ differ. $p_o$ denotes observed agreement, and $p_e$ denotes expected agreement.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>naive annotators</th>
<th>expert annotators</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$p_o$</td>
<td>$p_e$</td>
</tr>
<tr>
<td>task</td>
<td>0.45</td>
<td>0.09</td>
</tr>
<tr>
<td>auto feedback</td>
<td>0.31</td>
<td>0.14</td>
</tr>
<tr>
<td>allo feedback</td>
<td>0.26</td>
<td>0.10</td>
</tr>
</tbody>
</table>

**Table 3.7:** Tagging accuracy for naive and expert annotators, per dimension, drawn from the set of all annotation pairs.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>naive annotators</th>
<th>expert annotators</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$p_o$</td>
<td>$p_e$</td>
</tr>
<tr>
<td>task</td>
<td>0.64</td>
<td>0.16</td>
</tr>
<tr>
<td>auto feedback</td>
<td>0.74</td>
<td>0.46</td>
</tr>
<tr>
<td>allo feedback</td>
<td>0.58</td>
<td>0.19</td>
</tr>
<tr>
<td>turn</td>
<td>0.67</td>
<td>0.52</td>
</tr>
<tr>
<td>time</td>
<td>0.92</td>
<td>0.81</td>
</tr>
<tr>
<td>contact</td>
<td>1.00</td>
<td>0.60</td>
</tr>
<tr>
<td>own comm.</td>
<td>1.00</td>
<td>0.52</td>
</tr>
<tr>
<td>partner comm.</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>dialogue struct.</td>
<td>0.89</td>
<td>0.36</td>
</tr>
<tr>
<td>social obl.</td>
<td>0.96</td>
<td>0.26</td>
</tr>
</tbody>
</table>
have a high tagging accuracy and deviate only little from the group mean.

### 3.5.4 Qualitative comparative results

To get a better understanding of the differences between naive and expert annotators as indicated by the statistics presented in the previous section, we can consider the cooccurrence matrices of dialogue acts in the actual annotations.\(^{22}\)

For the task and feedback dimensions, which are relatively rich in dialogue acts, the expectation that naive annotators show more diversity in the dialogue act pairs that are involved in disagreements, is confirmed. There are some cases in which both naive annotators and expert annotators show disagreement, with the difference that the magnitude of disagreement is less for the expert annotators. For instance, typical cooccurrences of dialogue acts of disagreements in the dimension Task are `INFORM`
with ELABORATE and INFORM with SET-ANSWER, which occur for naive annotators 8.6 and 4.2 percent, respectively, and for expert annotators 1.7 and 1.3 percent, respectively, of all annotation pairs. Even though the experts do better than the naive annotators, this kind of pattern motivates action to be taken in improving the tagset with respect to the concept definitions involved.

There are also cooccurrences for which the naive annotators show considerable disagreement, and the experts do (almost) not. An example in the Task dimension is the cooccurrence of the communicative function INFORM with EXPLAIN. This may be due to the experts having a better understanding of the rhetorical functions involved than the naive annotators. Sometimes, it occurred that naive annotators show, relative to the number of annotation pairs, less disagreement than the experts. For instance, for naive annotators 0.7 percent of all annotation pairs involved the cooccurrence SET-ANSWER with INSTRUCT whereas for experts this was 2.0 percent. The reason why this happened becomes apparent when we take a look at the annotations that have been made in this context, for which the following dialogue excerpt\(^2\)\(^3\), annotated for the Task dimension, is illustrative:

<table>
<thead>
<tr>
<th>utterance</th>
<th>expert 1</th>
<th>expert 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>S(_1) do you want an overview of the codes?</td>
<td>PRO-Q</td>
<td>PRO-Q</td>
</tr>
<tr>
<td>U(_1) yes</td>
<td>PRO-A</td>
<td>PRO-A</td>
</tr>
<tr>
<td>S(_2) press function</td>
<td>INSTRUCT</td>
<td>SET-A</td>
</tr>
<tr>
<td>S(_3) press key 13</td>
<td>INSTRUCT</td>
<td>SET-A</td>
</tr>
<tr>
<td>S(_4) a list is being printed</td>
<td>INFORM</td>
<td>SET-A</td>
</tr>
</tbody>
</table>

Where naive annotators stayed close to question-answer adjacency pair patterns, the two experts sometimes disagreed, in that expert 1 almost consistently annotated responses that were instructions as an INSTRUCT where expert 2 annotated them as a SET-ANSWER. Expert 1 bases in this case the annotation foremost on the form of the utterance, but the utterance having the form of an instruction is something different than the utterance being intended as an answer.

There are other systematic differences between naive and expert annotators, most notably in Turn Management. As can be seen in Table 3.5 and Table 3.7, both naive and expert annotators failed to reach substantial agreement on assigning turn management functions. In dialogue, especially in multiparty interaction, interlocutors often signal eagerness to obtain the turn by interrupting the partner (TURN GRAB), to take the turn if available (TURN TAKE), to accept the turn when it was assigned to them (TURN ACCEPT), after finishing the contribution to explicitly assign the speaker role to an addressee (TURN ASSIGN), to drop the speaker role without putting any pressure on the addressee to take the turn (TURN RELEASE), or decide to continue as a speaker.

\(^2\)This excerpt originates from the human-machine part of the DIAMOND corpus.
(TURN KEEP). Very often, however, interlocutors just start to speak if they want to say something and stop speaking if they are finished with their contributions. In these cases it is the question whether the first utterance in a turn as having a TURN TAKE function, and the last utterance in the turn as having a TURN RELEASE function. The DIT++ annotation guidelines state\textsuperscript{24} that there is no turn management when the speaker does not signal an intention to address the turn allocation explicitly and when the annotator does not have sufficient evidence in terms of utterance features (such as intonational cues). The lack of agreement was caused by a lack of such evidence. For example, to signal the intention to keep the turn the speaker may use, besides fillers such as *um* or *uh*, pauses, rising intonation, and the slowing down of speech rate. In particular the latter may be expressed subtly, which makes the annotator’s decision rather subjective. Nevertheless, the experts annotators showed a more reliable intuition by reaching an agreement of 76.7 percent where naive annotators reached 66.7 percent. An example where prosodic rather than lexical cues address turn management is the following\textsuperscript{25}:

<table>
<thead>
<tr>
<th>utterance</th>
<th>naive</th>
<th>expert</th>
</tr>
</thead>
<tbody>
<tr>
<td>S\textsubscript{1}</td>
<td>from which station to which station do you want to travel?</td>
<td>TAS:SET-Q</td>
</tr>
<tr>
<td>U\textsubscript{1}</td>
<td>from...</td>
<td>TIM:STALL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Another source of disagreement on turn management originates from dealing with multifunctionality. For instance, discourse markers such as *and*, *or*, or *but* are known to have multiple functions in dialogue, and as a rule link dialogue units and signal speaker identification (TURN TAKE) or speaker continuation (TURN KEEP).\textsuperscript{26} For analysis of di For instance, consider the following excerpt\textsuperscript{27}:

<table>
<thead>
<tr>
<th>utterance</th>
<th>naive</th>
<th>expert</th>
</tr>
</thead>
<tbody>
<tr>
<td>A\textsubscript{1}</td>
<td>to the left...</td>
<td>TAS:SET-A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TUM:KEEP</td>
</tr>
<tr>
<td>A\textsubscript{2}</td>
<td>and then slightly around</td>
<td>TAS:WH-A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TUM:KEEP</td>
</tr>
</tbody>
</table>

The expert annotators fully exploited the phenomenon of multifunctionality in their annotations and assigned all functions they thought applicable, whereas the naive annotators did not make use of this.

\textsuperscript{24} And the annotators were instructed accordingly.
\textsuperscript{25} This excerpt originates from the OVIS corpus (H-M).
\textsuperscript{26} For analyses on discourse markers in dialogue, see for example [Heeman et al., 1998; Louwerse and Mitchell, 2003; Carota, 2007; Petukhova and Bunt, 2009].
\textsuperscript{27} This excerpt originates from the map task corpus (H-H).
3.5.5 Discussion

The data show that inter-annotator agreement for naive coders may be considerably low where for expert annotators agreement it is high (mostly > 0.8). When looking at annotation accuracy it was found that calculating reliability based on inter-annotator agreement only results in an indication of reliability that is too low. A general conclusion is that both inter-annotator agreement and annotation accuracy statistics are informative in determining how reliably a scheme can be used for annotation. Calculation of the latter indicator presupposes that on expert level a ground truth can be established, meaning that the concepts in the scheme should not be too subjective and should be sufficiently well-defined. The expectations that inter-annotator agreement and accuracy scores are both higher for expert annotators are confirmed.

Remarkably, it occurred that naive annotators showed higher inter-annotator agreement (for the dimension Social-obligations Management) and higher tagging accuracy (for the dimension Contact Management). For both cases this difference is largely explained by the interaction of the score with the ap-ratio. Naive annotators disagree more (with each other and with the gold standard) whether or not to annotate in a specific dimension, but the cases in which there is agreement are mostly the easy ones to annotate. Conversely, expert annotators show more agreement on when to annotate in a specific dimension, but as a result also address more complicated cases.

3.6 Tagset granularity reduction

From the number of annotation pairs in Table 3.5 of Section 3.5 (column #pairs) it can be concluded that six dimensions were addressed much more often than others: Task, Auto-feedback, Allo-feedback, Turn Management, Time Management and Dialogue Structuring. Of these, both feedback dimensions and the Turn Management dimension have low agreement scores for the naive annotators, while Turn Management has a low agreement score for both groups of annotators. It was found that it is often difficult for annotators to determine the level of feedback (attention, perception, understanding, evaluation or execution), while for Turn Management the annotation guidelines were found to be unclear, as already mentioned. (Note the low ap-ratios for this dimension for both groups.)

These and other more detailed findings were used for designing a revised tagset as well as improving the annotation guidelines within the European project LIRICS\textsuperscript{28} (see: [Schiffrin and Bunt, 2007]). Within this project, a test suite was developed of dialogues in several European languages which were annotated with the revised tagset. For English and Dutch the test suite dialogues were all annotated by two expert annotators. An analysis of the agreement between their annotations reveals that in all of the frequently addressed dimensions a very high agreement was reached (weighted kappa

\textsuperscript{28}Linguistic Infrastructure for Interoperable Resources and Systems. See http://lirics.loria.fr/.
scores well above 0.9). By applying a mapping from the original DIT++ tagset to the revised LIRICS tagset the effects can be calculated that this revision should have on the agreement scores for both groups of annotators. The effect of the improvement of the annotation guidelines cannot be calculated in this way, but an estimation of that effect can be obtained by comparing the calculated improved agreement scores for the expert annotators with the scores that were found in the LIRICS project.

In DIT++ some of the dimensions contain one or multiple hierarchies of dialogue acts. The dialogue acts in such hierarchies are related in such a way that an act lower in a hierarchy is more specific than an act higher in the same hierarchy. For instance, in Figure 3.8 a CHECK is more specific than a PRO-QUESTION, which is in turn more specific than a INDIRECT-PROPOSITIONAL-QUESTION.

![Figure 3.8: Two hierarchies of information-seeking general purpose functions.](image)

Using the existing hierarchical structure, a hierarchy could partially (or fully) be ‘collapsed’, resulting in the grouping of acts under a least specific parent act, flattening the hierarchy and making the tagset less fine-grained. There are two major motivations of doing so. First, by grouping dialogue acts together, disagreement that is the result of considering fine-grained distinctions is eliminated. Second, grouping dialogue acts can make inter-annotator agreement analysis less susceptible to very infrequently occurring, fine-grained dialogue acts which occur too infrequently to draw significant conclusions from in evaluation. It should be remarked that collapsing a hierarchy to a general dialogue act would only be justified when the general dialogue act is sufficiently fine-grained for the intended purpose of applying the tagset.

There are various ways in which hierarchies can be collapsed into general dialogue acts. The dialogue acts proposed in the LIRICS project are largely based on acts in the DIT++ tagset but exhibit lower granularity, making it interesting to collapse DIT++ hierarchies to LIRICS dialogue acts to assess the performance of both annotator groups. Additionally, this provides indicative inter-annotator agreement scores for dialogue acts in LIRICS. Because almost all hierarchies in the DIT++ tagset are either in the set of general-purpose communicative functions or in the feedback dimensions, only these parts of the tagset are considered. The grouping and mapping used for LIRICS are depicted in Figure 3.9.

As was predicted, the scores for both annotator groups improve after recalculating inter-annotator agreement and accuracy for the LIRICS dialogue acts. The differences
Figure 3.9: Grouping and mapping of dialogue acts, where lines indicate hierarchical relations.

As can be seen from the table, the improvement for naive annotators is higher than that for expert annotators. It is not difficult to see why: for less trained annotators it is easier to annotate general concepts. For instance, in quite some cases of feedback — most notably those with feedback not being realised lexically — it is difficult to determine the feedback level, especially for naive annotators. By grouping all levels of feedback, this substantial source of disagreement is eliminated. The gain in accuracy turns out to be proportional to the relative gain in inter-annotator agreement, both for naive and expert annotators.
3.7 Summary

In this chapter, the considerations that come with compiling dialogue corpora for dialogue act annotation are addressed, motivated by the assumption that successful systematic analysis of dialogue depends on the availability, design, and quality of corpora.

The practical and methodological aspects of collecting data to support empirically based analysis of dialogue characteristics related to dialogue acts are addressed. A first aspect that is described is the range of characteristics of dialogue corpora and tagsets that can be considered. Furthermore, an overview has been presented of the dialogue corpora that are used in the thesis.

A substantial part of the work presented in this chapter deals with evaluating the applicability of tagsets in general and that of DIT++ in particular. In many studies, applicability and consistency of a dialogue act taxonomy or tagset is measured by means of inter-annotator agreement. The annotator agreement for the dialogue corpora that are used in this thesis are discussed, and in the case of DIT dialogue acts a first account of annotator agreement is given.

It is argued that for taxonomies of semantic and pragmatic concepts, such as dialogue acts, using the popular kappa metric does not take into account partial disagreement. A new metric, which is called taxonomically weighted kappa, is proposed that is ‘weighted’ in that it takes into account semantic and pragmatic distance between concepts, and is taxonomically motivated in that it takes advantage of the taxonomic structure (hierarchical relationships between dialogue acts) to give a better account of inter-annotator agreement between dialogue acts. The difference between the regular kappa and the weighted kappa is substantial for the most important aspects of dialogue, at least for the concepts in the DIT++ tagset.

Another methodological consideration of dialogue act annotation that has been discussed is what type of annotators to use. It is argued that for an insightful analysis of the annotation scheme it is recommendable to evaluate with both expert annotators and naive annotators, as they provide feedback on different aspects of the applicability of the tagset. Differences in both inter-annotator agreement and tagging accuracy between naive and expert annotators against the gold standard are considerable, and the annotations of both groups provide insights in reliability complementary to each other concerning clarity and accessibility of the tagset and problematic conceptual issues. In comparing the two annotator groups, it turns out that for multidimensional dialogue act taxonomies it is essential to distinguish agreement on whether or not to annotate in a dimension, from agreement on the dialogue act assignment within a dimension. However, calculation of tagging accuracy presupposes that on expert level a ground truth can be established, meaning that the concepts in the scheme should not be too subjective and should be sufficiently well-defined. Depending on the tagset, this might not always be possible.

Additionally, it is investigated how a part of DIT++ maps to the simpler LIRICS tagset, which essentially involves the simplification of DIT++ by collapsing certain hierarchical structures of fine-grained dialogue acts to more generic, higher concepts.
This implicitly provides insight in how well the LIRICS dialogue act tagset can be applied, and shows how the complexity of a hierarchical tagset can be reduced in a sensible way. The simplification is only permissible as long as the simplified tagset is adequate enough for the purposes it has been designed for.
In the last decennium, numerous studies have been published on the automatic recognition of dialogue acts. Most of these studies have assumed presegmented utterances to be available for act recognition.

This chapter begins with manual segmentation and the notion of ‘functional segment’ is introduced as the encoding of a dialogue act. Multidimensional segmentation is presented as an alternative to conventional segmentation, and the former is shown to allow segments to be described more accurately.

To evaluate performance on utterance segmentation, act classification, and the combination of both, tasks are formulated as classification tasks on token level. Using several machine learning classifiers, the performance for each of the tasks is assessed for various datasets.

Finally, the performance of machine learners is compared with that of human annotators.

4.1 Introduction

An essential prerequisite for successful communication is the identification of the speaker’s intention by the addressee. An addressee has to recognise the dialogue acts performed by the speaker. A fundamental question is how an addressee accomplishes this in a given dialogue context. This chapter is about a related and more practical question: how can dialogue acts that are performed in a given dialogue context be recognised automatically. Answering this question may be useful for several reasons.
First, it could help in testing and verification of theories of dialogue. Second, it allows to find relationships between properties of communicative behaviour and intention that is conveyed, improving existing models. Third, it can be used to design an intention recognition component in a dialogue system.

A simple way of conceiving the task of act recognition is to assign to each meaningful utterance a dialogue act. In the example of Figure 4.1 this would mean to identify the first utterance as a REQUEST and the second utterance as a CONFIRM. However, 

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S</td>
<td>could you make ... ehm... a copy of this?</td>
</tr>
<tr>
<td>2</td>
<td>U</td>
<td>yes</td>
</tr>
</tbody>
</table>

Figure 4.1: Example of a request.

is not accidentally slowing down his speech; the first utterance is not only a REQUEST, but also exhibitsSTALLING. This consideration makes the task more complicated, as it is necessary to take into account that a single utterance may have multiple functions; this means that the task becomes to assign to each meaningful utterance the set of acts that are performed by uttering it. Moreover, multiple discontinuous spans of speech may constitute a single dialogue act, and two dialogue acts, $DA_1$ and $DA_2$, may be hierarchically related: performance of $DA_2$ also implies that of $DA_1$, but performance of $DA_1$ does not imply that of $DA_2$. For instance, in Figure 4.2, the first utterance can be described as an INFORM. The second utterance can be considered as a CORRECT, which implies a DISAGREE.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>S</td>
<td>now you should press the green button</td>
</tr>
<tr>
<td>2</td>
<td>U</td>
<td>you mean the grey button</td>
</tr>
</tbody>
</table>

Figure 4.2: One dialogue act implies another.

The observation that acts can span multiple fragmented utterances and some acts can subsume others introduces challenges both for theories of dialogue acts and for applications such as dialogue systems.

To recognise dialogue acts, it is essential to find those properties of the utterance and its context that are characteristic for specific acts. This information can be acquired by corpus studies and psycholinguistic research, and put into a model that a system can use to automatically identify dialogue acts. Alternatively and additionally, the relation between utterance characteristics and dialogue acts can be learned automatically by means of machine learning techniques, which at the same time function as an automatic dialogue act classifier by applying their automatically learned models to new input. In this way, machine learning algorithms on the one hand provide insight into the extent to which characteristics of dialogue are important for recognising specific acts, and on the other hand can be used as a component of a dialogue system.

For the purpose of automatic dialogue act recognition, a machine learning algorithm learns a model and may function as classifier. This task may be described as
follows: given a pre-defined set of utterance properties (the features), determine the dialogue acts being performed (a class, usually one of multiple) using a set of examples to learn from (the training set). For example, a machine learning algorithm that constructs a simple probabilistic model may decide to suggest the dialogue act GREETING on the basis of the occurrence of the word-feature ‘hello’, as in the training set this word occurred the most in association with that particular act. The features that may be predefined for the machine learning classifier may include any information that is available, such as word usage, syntactic information, acoustic-prosodic information such as speech rate and pitch, non-verbal information such as gestures and facial expressions, and contextual dialogue features.

In selecting the features for the algorithm to use, there are at least two important considerations. First, the aim for which the algorithm is employed puts restrictions on what features to use. If the algorithm is to be part of an on-line dialogue system, dialogue properties that have been used in the training set should also be available during interaction with users. For example, information on what happens next in a dialogue could be valuable in training a classifier, but is not available in on-line interaction. Second, one can simply encode as many features as possible and leave it to the algorithm to select and use those features that are most useful in classification. A drawback of this approach is that some algorithms are more prone to noise (i.e. redundant or irrelevant features) than others, while they might perform substantially better with a carefully preselected feature set.

The last ten years there have been numerous studies into the automatic recognition of dialogue acts in which several techniques, conversation types, and tagsets have been used. Several researchers have been inspired by POS tagging and have used algorithms that have proven to be successful there. Thus Hidden Markov Models for tagging ([Church, 1988]) have been used in various studies (e.g. in [Ries, 1999] and [Stolcke et al., 2000]), and other POS tagging algorithms have been tried, such as Tranformation-Based Learning ([Brill, 1995]) in [Samuel et al., 1998]. Other work adopts a more statistical approach, e.g. using Bayesian Networks ([Pulman, 1996] and [Keizer et al., 2002]), language models ([Reithinger and Kleesen, 1997]), Latent Semantic Indexing ([Serafin and Di Eugenio, 2004]), or k-nearest-neighbour classification ([Lendvai et al., 2003]). Apart from using diverse algorithms, some studies have focussed more on specific aspects of communication in recognising acts. One direction is concerned with the acoustic-prosodic correlates of dialogue acts, e.g. [Jurafsky et al., 1998], [Shriberg et al., 1998], and [Fernández et al., 2007].

In most of the above-mentioned work, the task of dialogue act recognition is conceived as a high-level classification task: classify for each utterance the acts that are performed. In doing so, it is assumed that utterances have been optimally (manually) segmented prior to classification, the exceptions being [Warnke et al., 1997] and [Zimmermann et al., 2005]. However, fully automated dialogue act recognition in the context of this task would first require automated dialogue act segmentation and subsequently, based on the segments found, involve dialogue act classification. As such, the results presented in studies that classify utterances based on presegmented dialogue
should be interpreted as being the classification performance when segmentation is perfect, something that will rarely, if ever, be the case when segmentation is expected to be done automatically. Alternatively to the conception of dialogue act recognition as a sequential two-step process, it may be possible not to split the recognition process in two phases but to search for the best segmentation and dialogue act assignment at the same time. The advantage of the two-step approach may be that focussing on each of the sub-problems (which appear to be less complex than the entire problem) leads to better results than the joint approach. The disadvantage is that the classification depends on the segmentation: errors that are made in the segmentation stage will be propagated to the classification stage.

This chapter investigates a variety of approaches to the recognition of dialogue acts. The little attention for the segmentation of dialogue acts raises the question of how dialogue can be segmented accurately, taking into account the multifunctionality of dialogue behaviour. Traditional utterance-based segmentation as a basis for dialogue act classification is discussed, and a more accurate way of segmenting dialogue is proposed and evaluated (Section 4.2). The presence of multifunctionality in dialogue and the use of another way of dialogue segmentation motivates a study into how well the communicative functions organised in a multidimensional system can be recognised automatically (Section 4.3). The work described in Section 4.2 and Section 4.3 is based on joint work presented as [Geertzen et al., 2007]. Part of the work in Section 4.3 is based on joint work presented as [Lendvai and Geertzen, 2007].

In the dialogue act recognition experiments, two types of machine learning algorithms can be used: local learners and sequence learners. The latter type is specifically designed to learn from sequential input and is expected to work better for the kind of experiments at hand. This expectation is verified by using various algorithms.

Having obtained measures for the automatic classification of dialogue acts, it is interesting to look how the obtained machine performance compares to that of humans. This comparison gives an indication of the difficulty of the task. To give an indication, the machine learning performance of dialogue act classification is compared with the performance of naive and expert annotators (Section 4.4).

4.2 TOWARDS MORE ACCURATE DIALOGUE SEGMENTATION

4.2.1 Introduction

The assignment of appropriate meanings to ‘dialogue units’ presupposes a way to segment a dialogue into meaningful units. This turns out to be a complex task in itself. Previous studies in the area of automatic dialogue act assignment were mostly carried out at the level of ‘utterances’ or that of ‘turns’. A turn can be defined as a stretch of communicative behaviour produced by one speaker, bounded by periods of inactivity of that speaker or by activity of another speaker ([Allwood, 2000]). Turn boundaries
can be recognised relatively easily, but segmentation into turns is unsatisfactory because a turn may contain several smaller meaningful parts. An utterance, on the other hand, may be defined as a stretch of communicative behaviour, produced by a single interlocutor, that has a single or multiple communicative functions. Utterances may coincide with turns but are usually smaller.

The detection of utterance boundaries is not a simple task. First of all, the notion of ‘utterance’ can be defined in various ways, depending on the continuity of the speech, depending on syntactic or semantic completeness, or depending on the encoding of dialogue acts\(^1\). Usually, syntactic features (e.g. part-of-speech, verb frame boundaries of finite verbs) and prosodic features (e.g. boundary tones, phrase final lengthening, silences, etc.) are used as indicators of utterance endings ([Shriberg et al., 1998; Stolcke et al., 2000; Nöth et al., 2002]).

One of the problems with dialogue segmentation into utterances is that utterances may be discontinuous. Spontaneous speech in dialogue usually includes filled and unfilled pauses, self-corrections and restarts; for example, the speaker of the utterance in (4.1) corrects himself two times.

(4.1) “About half...about a quar-...th....third of the way down I have some hills”

Dialogue utterances may be interrupted by even more substantial segments than repairs and stallings. For example, the speaker of the utterance in (4.2) interrupts his Inform with a SET-QUESTION:

(4.2) “Because twenty five Euros for a remote...how much is that locally in pounds?...is too much money to buy an extra remote or a replacement remote”

Examples such as (4.1) and (4.2) show that the segmentation of dialogue into utterances that have a communicative function requires these units to be potentially discontinuous. In some cases a dialogue act may be performed by an utterance formed by parts of more than one turn. This often happens in dialogue with more than two participants in which participants may interrupt each other or talk simultaneously. For example in 4.3, where \(B\) interrupts \(A\):

(4.3) “Well we can chat away for...um...for five minutes or so I think at... [ mm-hmm ]\(B\) ...at most”

Another case of a dialogue act that is spread over multiple turns occurs when the speaker is providing complex information and divides it up into parts in order not to overload the addressee, as exemplified in Figure 4.3. The first part of the discontinuous segment that expresses \(S\)’s answer also has a feedback function (making clear to \(U\) what \(S\) understood).

The material in the three turns contributed by \(S\) together constitute the ‘utterance’ expressing \(S\)’s answer to \(U\)’s question. Examples such as these show that the units in

\(^1\)See [Traum and Heeman, 1997] for a more elaborate discussion.
dialogue that carry communicative functions are often different from the traditional linguistically defined notion of an utterance. It is therefore preferable to give these units a different name, that of functional segment, and to define these units as “(possibly discontinuous) stretches of communicative behaviour that have one or more communicative functions”. In essence, this is similar to the conception of utterance as being the encoding of dialogue acts, such as used in [Mast et al., 1996].

A functional segment often corresponds to an ‘utterance’ as defined by certain linguistic properties, but in other cases it does not; and so the question arises how functional segments can be recognised.

When the task is to segment a dialogue into functional segments, one complication is that of discontinuous segments, either within a turn or spread over several turns as was discussed already. An even greater challenge is posed by those cases where different functional segments overlap, as in the example shown in 4.4.

The first part of S’s turn repeats most of the preceding question, displaying what the system has heard, and as such has a feedback function. The turn as a whole minus the part “...ehm...” has the communicative function of a SET-ANSWER, and the part “...ehm...” has a stalling function. So the segments corresponding to the SET-ANSWER and the feedback function share the part “The first train to the airport on Sunday”. This means that in this turn there are two functional segments starting at the same position but ending at different ones; in other words, no single segmentation of this turn exists that gives us all the relevant functional segments.

To resolve this problem adequately, it is proposed not to maintain a single segmentation, but to use multiple segmentations to allow multiple functional segments that are associated with a specific utterance to be identified more accurately. This approach is compatible with dialogue act taxonomies that address several aspects (‘dimensions’)
of the interactive process simultaneously (e.g. DAMSL [Core and Allen, 1997] or DIT [Bunt, 2006]), such as the task or activity that motivates the dialogue, the management of taking turns, of using time, and of attention. This multidimensional view of dialogue naturally leads to the suggestion of approaching dialogue segmentation in a similarly multidimensional way, allowing the segmentation of a dialogue *per dimension* rather than in one fixed way. In the case of example (4.4), this means that S’s turn is segmented in the three dimensions addressed by the functional segments in this turn, as depicted in Figure 4.5.

![Figure 4.5: Single utterance segmented per dimension.](image)

To describe the functional segments in DIT++ terminology, the discontinuous segment “the first train to the airport on Sunday is at 6.17” has a function *SET-ANSWER* in the Task dimension; the contiguous segment “the first train to the airport on Sunday” has a function *POSITIVE-FEEDBACK* in the Auto-feedback dimension; the contiguous segment “…ehm…” including the drop in speech rate prior to it has the function *STALLING* in the dimension Time management.

In the following sections a study is described which addresses the question whether using a separate segmentation for each of the dimensions in multidimensional dialogue acts taxonomies will allow a more accurate labelling of communicative functions than when using a single segmentation for all dimensions.

### 4.2.2 Contrastive experiment outline

Any segmentation of dialogue (or multiparty interaction) into meaningful units, such as functional segments, is motivated by the meaning that is conveyed. As a result, the segmentation strongly depends on the definitions of the dialogue acts that are used in the taxonomy. The multidimensional tagset used in this experiment allows several aspects of communicative behaviour for an utterance to be addressed simultaneously. However, the functions of a segment do not necessarily address the same span in the communicative channels. Separate segmentation for each dimension may be expected to allow for a more accurate identification of spans associated with certain communicative functions. If this is the case, it would be expected that classification of communicative functions based on per-dimension segments is more successful than classification based on a single segmentation for all dimensions.
For testing this hypothesis, two machine learning classification tasks on exactly the same dialogues with exactly the same kind of features and annotated communicative functions were performed. The only difference was that in one task one segmentation that fits all dimensions (OSFAD) was used, whereas in the other task per-dimension segmentation (PDS) was used. The basic assumption is that classification scores will improve when the segmentation is more accurate.

The dialogues used in the classification tasks were drawn from the DIAMOND corpus (see 3.3.4), which contains human-machine and human-human Dutch dialogues of an assistance-seeking nature. The dataset used for classification contains orthographically transcribed dialogues consisting of 952 utterances, representing 1,408 functional segments from the human-human subset of the corpus. The functional segments were annotated with DIT++ tags (see 2.3.7) to obtain PDS data. From the annotations of the functional segments, the OSFAD segmentations have been generated by extending the span of each function to the full utterance span. To avoid effects that distributional differences could have on the machine learning, the functions have not been joined into composite tag labels. The equivalent for OSFAD data of the depiction of PDS data in Figure 4.5 is illustrated in Figure 4.6.

Ripper [Cohen, 1995], a state-of-the-art rule inducer, was used as machine learning classifier. Ripper learns sets of classification rules from the training data, where each rule is a conjunction of conditions on attribute values. To reduce the effect of imbalances in the dialogue data, the results of the classifier were obtained using stratified 4-fold cross-validation.²

The features available to the algorithm are related to dialogue context, prosody, and word occurrence. Word occurrence was represented by a bag-of-words vector indicating the presence or absence of words in the segment. The dialogue context was specified by bag-of-words vectors for the previous three turns. Prosodic features that were included are minimum, maximum, mean, and standard deviation of pitch (F0 in Hz), energy (RMS), voicing (fraction of locally unvoiced frames and number of voice breaks), and duration.

²Each of four times another 25% of the data is used as test set and the remaining 75% as training set. Subsequently, the results from each of the fold are averaged to obtain a single estimation of the performance.
4.2.3 Comparative results

Running Ripper with default parameters for both tasks resulted in the scores presented in Table 4.1.

Table 4.1: Accuracy scores for communicative functions with one segmentation that fits all dimensions (OSFAD) and per-dimension segmentation (PDS). Results that are significant at $p < .05$ (one-tailed $z$-test) are marked with $\ast$.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>OSFAD</th>
<th>PDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>task</td>
<td>66.1±0.03</td>
<td>72.8±0.11 *</td>
</tr>
<tr>
<td>auto feedback</td>
<td>80.4±0.02</td>
<td>86.3±0.05 *</td>
</tr>
<tr>
<td>allo feedback</td>
<td>98.4±0.02</td>
<td>99.6±0.03</td>
</tr>
<tr>
<td>turn</td>
<td>88.3±0.06</td>
<td>90.0±0.16</td>
</tr>
<tr>
<td>time</td>
<td>72.6±0.05</td>
<td>82.1±0.04 *</td>
</tr>
<tr>
<td>contact</td>
<td>97.3±0.01</td>
<td>97.3±0.01</td>
</tr>
<tr>
<td>own communication</td>
<td>85.9±0.03</td>
<td>87.1±0.04</td>
</tr>
<tr>
<td>partner communication</td>
<td>64.5±0.04</td>
<td>64.5±0.04</td>
</tr>
<tr>
<td>dialogue structuring</td>
<td>74.3±0.07</td>
<td>74.3±0.07</td>
</tr>
<tr>
<td>social obligations</td>
<td>93.2±0.02</td>
<td>93.3±0.03</td>
</tr>
</tbody>
</table>

From Table 4.1 it can be observed that for the most important dimensions, PDS results in better classification performance: the functions related to the dimensions Task, Auto Feedback, and Time Management show significant improvement. The reason becomes clear by considering Figure 4.7, which shows how much a dimension is addressed without any other dimensions being addressed (e.g. Task for 29.9% of the cases) or in combination with other dimensions being addressed (e.g. Turn Management and Time Management simultaneously for 8.1% of the cases).

Relating the classification scores with the figure shows that the three dimensions for which a significant improvement is found are also often addressed in a multifunctional perspective.\(^3\)

For other dimensions, classification does not take advantage of PDS, mainly because of two reasons: in the dataset some dimensions are rarely addressed at all (e.g. Partner Communication Management) and some dimensions are addressed without any other dimension being addressed around the same time (e.g. Contact Management). Both aspects depend on the kinds and characteristics of the interaction and to some extent on the limited size of the dataset.

Although not all dimensions benefit significantly, multidimensional segmentation clearly helps to classify communicative functions more accurately.

\(^3\)About 38 % of all utterances in the dataset are multifunctional.
4.2.4 Discussion

Whereas it is common practice to assign dialogue acts to a single segmentation, it has been shown that for dialogue act taxonomies that allow assignment of multiple functions to dialogue units, human communication can be described more accurately by using per-dimension segmentation instead.

There are some issues that are to be taken into account. A first issue concerns segmentation reliability. While per-dimension segmentation is shown to be more accurate, reliability in obtaining it should also be sufficient. This calls for extending the study with a reliability analysis of per-dimension segmentation.

In this context, ‘sufficiently reliable’ means that inter-annotator agreement for per-dimension segmentation is not lower than that for OSFAD. A direct comparison between the two approaches is difficult to make, but to test whether segmentation agreement is —within reasonable bounds— similar, a dataset of multiparty interaction from the AMI corpus\(^4\) was segmented and annotated. These segmentations and annotations were carried out by the same expert annotators that annotated the DIAMOND dialogues. The results of the comparison are given in Table 4.2.

The two annotators identified 176 and 180 segments, respectively. From these segments, 248 functions were evaluated, of which in 96% of the cases the boundaries matched within a 200 millisecond window (percentage agreement on segmentation). When combined agreement of segmentation and labelling is evaluated (corresponding segmentation and corresponding labelling in the same dimension with the same communicative function), the kappa score is 0.83. This score is lower than the weighted averaged expert agreement reported in Section 3.5.3, (0.88), but it should be considered

\(^4\)Augmented Multi-party Interaction. See: http://www.amiproject.org/.
Table 4.2: Agreement statistics on various aspects of multidimensional segmentation for 248 functions in AMI data.

<table>
<thead>
<tr>
<th>Aspect</th>
<th>$p_o$</th>
<th>$p_e$</th>
<th>$\kappa$</th>
</tr>
</thead>
<tbody>
<tr>
<td>segment boundaries</td>
<td>0.96</td>
<td>0.16</td>
<td>0.95</td>
</tr>
<tr>
<td>segment boundaries + dimension</td>
<td>0.90</td>
<td>0.22</td>
<td>0.87</td>
</tr>
<tr>
<td>segment boundaries + dimension + function</td>
<td>0.85</td>
<td>0.11</td>
<td>0.83</td>
</tr>
</tbody>
</table>

that the latter figure is not only obtained with less complicated interaction (interaction between two interlocutors), but also with prior established agreement on segmentation.

Another issue is the segmentation and annotation cost. The gain of more accurately identified functions comes at the cost of a slightly more complex (and hence labour-intensive) segmentation procedure. Where in conventional segmentation practices there is only one segmentation tier to consider, per dimension segmentation has multiple tiers. Segmentation and annotation cost may be quantified by the average time needed to process one utterance, which turned out to be 11.8 seconds per utterance for OSFAD and 17.1 seconds per equivalent functional segments for PDS on the DIAMOND dialogues, making the latter about one and a half times as costly.

4.3 Token-based act segmentation & classification

4.3.1 Introduction

An important aspect of machine-learning dialogue act recognition is that the training data, coming from speech corpora, are usually based on transcribed speech. It is common practice to group the dialogue participants’ token stream (typically containing words, but also disfluent elements, non-speech events, symbols for overlapping speech, etc.) into syntactically or semantically complete units, which are then further grouped into turns based on change of speaker over time. In spontaneous, stressful, or multiparty situations, spoken dialogue does not fully proceed in sequence, but often contains simultaneously occurring events, since speakers may react to each other’s (incomplete) utterances in a dynamic way. These properties motivate a modelling of dialogue or multiparty interaction as a stream of words, and other vocal productions such as laughs and sighs, and disfluent elements (henceforth: tokens) for each interlocutor parallel to those of other interlocutors.

In the following section, the global aim is to perform sequential tagging based on token sequences, exploring the effect of various data encodings and supervised machine learning algorithms on the performance of segmenting and labelling the token streams in the interaction. By applying machine learning classification on the level of
tokens, instead of utterances, the task of the classifier becomes to identify each element of the token stream as being part of a specific dialogue act type (classification), and also whether the token is an initial, final, or an internal element of a functional segment (segmentation). This approach is rather novel in the field, and allows the use of state-of-the-art sequence learning algorithms.

### 4.3.2 Data encoding for token-based classification

The encoding of the training and testing data for automatically learning a classifier can vary given the formulation of the segmentation and classification problem. For instance, we could provide a functional segment to the classifier and let the algorithm decide what the communicative functions are. The training examples for the machine learning algorithm contain a feature vector based on properties of the functional segment and its immediate context, and the associated communicative functions. As discussed earlier, this setting presupposes presegmented input.

One possible step towards segmenting dialogue into functional segments is to look at changes of speaker and postulate at each change of speaker the ending of a functional segment and the start of another. In other words, the speech is segmented into turns and it is assumed that by doing so a part of the functional segment boundaries are identified (functional segment boundaries that coincide with turn boundaries).\(^5\)

Looking for functional segments within turns may recover most of the functional segment boundaries, but the boundaries of functional segments that span multiple turns cannot be identified successfully. Nevertheless, for the sake of simplicity and explaining the token-based encoding, an example dialogue (Figure 4.8) is taken from the ICSI Meeting Corpus, annotated with general SWBD-DAMSL tags.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>what do we have?</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>not much</td>
</tr>
<tr>
<td>3</td>
<td>A</td>
<td>okay</td>
</tr>
<tr>
<td>4</td>
<td>A</td>
<td>like what?</td>
</tr>
<tr>
<td>5</td>
<td>B</td>
<td>we have models and</td>
</tr>
<tr>
<td>6</td>
<td>A</td>
<td>hhmh</td>
</tr>
<tr>
<td>7</td>
<td>B</td>
<td>training data</td>
</tr>
</tbody>
</table>

Figure 4.8: A short dialogue with questions (Q), statements (S), and a back-channel (B).

The segments that are obtained for interlocutor B after segmenting into turns correspond to the level labelled ‘turns’ in Figure 4.9.

As can be seen in the middle level of Figure 4.9, the turn boundaries of two of the three turns already correspond to utterance boundaries. Only the second turn has a

---

\(^5\)Turns by different speakers may partially overlap. In [Levinson, 1983] it is reported that less than 5% of words are overlapping, but more recent studies like [Shriberg et al., 2001] suggest that this figure is rather conservative if multiple speech corpora are investigated, and varies between 5% and 15%.
CHAPTER 4. DIALOGUE SEGMENTATION AND DIALOGUE ACT RECOGNITION

Figure 4.9: Dialogue represented for interlocutor B on turn, utterance, and token level.

turn-internal boundary, segmenting the turn into a STATEMENT and a QUESTION. To specify within the turn the position of any turn-internal boundary, it is necessary to look at the token level (the bottom level in the figure). In the context described above, the boundaries of segment turns are relatively straightforward to find by looking at speaker changes. Finding the turn-internal boundaries, however, is more difficult. The task of finding the turn-internal boundaries that separate subsequent turn-internal dialogue acts can be formulated as the task of predicting after each token if there is a boundary (marked by the bar symbol) or if there is no boundary (marked by a period). This encoding is proposed by Warnke et al. [1997] and Zimmermann et al. [2005], and it is typically suitable to train an n-gram language model or Hidden Markov Model in the prediction of segmentation markers and class labels.

To allow instance-based classification on the token level possible, similar encodings that are proposed in chunking tasks [Ramshaw and Marcus, 1995] may be adopted. To indicate to which dialogue act a token belongs, each token receives a communicative function symbol as depicted in the token-level of Figure 4.9. Additionally, for each communicative function label, a prefix marks whether the token is at the start (B-) or inside (I-) the functional segment corresponding to the dialogue act. This kind of tagging will be referred to as BI-tagging, and is illustrated in Figure 4.10.

In this way, segmentation is encoded by the BI prefixes and the classification is encoded by the communicative function label, with the possibility to evaluate the segmentation separately (only the prefix), classification separately (only the communicative function label), or both at the same time (a combination of prefix and communicative function label).

The training samples for the machine learning algorithm would now contain the token, a feature vector for this token based on contextual properties and properties of the token itself, and the associated BI tag. As can also be observed from Figure 4.10, this task would require the machine learning algorithm to solve a multiclass classification problem, where the number of classes could get quite large for detailed tagsets. As
some classifiers tend to perform better when the number of classes in the classification task is small, it is worth to reduce the number of classes by learning for each communicative function separately if the token is at the start (B-), inside (I-), or outside (O-) the corresponding segment. These tags are referred to as IOB tags (see: [Ramshaw and Marcus, 1995]). The contributions of interlocutor B in the earlier figures with IOB encoding looks as depicted in Figure 4.11\(^6\).

---

As a matter of fact, the IOB encoding proposed by Ramshaw and Marcus only assigns a B tag instead of an I tag when a token is beginning an act immediately following an act of the same type.
Section 4.3.8.

4.3.3 Evaluation metrics

There are several ways in which the performance on a classification task can be expressed. The error metrics and performance measures that will be used to assess the results reported above are in the first place those that are commonly used in information retrieval and machine learning.

For a single class \( C \), the number of cases in which an item belongs to \( C \) are called **positives** and are denoted by \( p \), whereas the number of cases in which an item does not belong to \( C \) are called **negatives** and are denoted by \( n \). The measure \( tp \) (true positives) expresses the count of cases in which an item belongs to class \( C \) and is correctly classified as such. The measure \( fp \) (false positives) expresses the count of cases in which an item is incorrectly classified as \( C \). The measure \( fn \) (false negatives) expresses the count of cases in which an item belongs to class \( C \), but is not classified as such. The measure \( tn \) (true negatives) expresses the count of cases in which an item does not belong to class \( C \) and is classified as such. Note that \( n = fp + tn \) and \( p = tp + fn \).

These abovementioned measures are illustrated in Figure 4.12. In the literature, also some derived measures are proposed that are specific to dialogue act classification and segmentation, but can be expressed in basic performance measures.

![Confusion matrix of measures for a single class](image)

Given the four measures specified, a number of commonly used performance metrics can be defined. A well known metric is **accuracy**, which is defined as the proportion of correctly classified instances from all instances in the test set (Equation 4.4).

\[
\text{accuracy} = \frac{\# \text{ correctly classified instances}}{\# \text{ total instances}} = \frac{tp + tn}{p + n} \tag{4.4}
\]

Accuracy is a useful metric, but if there is not only an interest in task performance but also in classifier performance, other metrics can better be used as accuracy does not take probability into account: when accuracy on a class label is less than or equals the probability of the class label occurring, the classifier does not performs better than chance.
The proportion of correctly classified positive instances from all classified positive instances is known as \textit{precision} (Equation 4.5).

\[
\text{precision} = \frac{tp}{tp + fp} \tag{4.5}
\]

The True Positive Rate (TPR) or \textit{recall} is the proportion of correctly classified positive instances from all positive instances (Equation 4.6).

\[
\text{recall} = \frac{tp}{p} \tag{4.6}
\]

The $F$-score [van Rijsbergen, 1979] is a metric that balances precision and recall (Equation 4.7).

\[
F_\beta = \frac{(\beta^2 + 1) \cdot \text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}} \tag{4.7}
\]

By assigning parameter $\beta$ the value 1, the $F$-score expresses the harmonic mean between precision and recall.

The metrics are computed per class; to obtain a single outcome for a set of classes, the scores can be averaged in two ways. With \textit{micro-averaging}, the score of each class is weighted proportionally to the frequency of the class in the dataset; with \textit{macro-averaging}, the scores of each class are not weighted.

Apart from these common performance metrics, several error metrics have been proposed for segmentation in general or for dialogue act segmentation and classification in particular, such as \textit{NIST sentence-like unit error} (NIST-SU, [NIST, 2003]), \textit{Dialogue-act Segmentation Error} (DSER) and \textit{Dialogue-act Error Rate} (DER) [Zimmermann et al., 2005], and \textit{Strict} and \textit{Lenient} [Ang et al., 2005].

For expressing performance in segmenting dialogue, DSER is used; for expressing performance in classifying dialogue acts, DER is used. On the level of tokens, token accuracy is used to indicate the fraction of tokens for which the communicative function or dialogue act is classified correctly, which is inversely proportional to the Lenient error metric. For clarity, metrics for segmentation have subscript $s$, metrics for classification have subscript $c$, and metrics for combined segmentation and classification have subscript $sc$. The counts that are used in the metrics are obtained by comparing the tokens and (non) boundary events suggested by the machine learning classifier with the correct ones (the ‘gold standard’). For dialogue acts this involves comparing gold standard segments (GSSs) with suggested segments. DSER is then defined as the inverted recall on the correct segmentation of dialogue (Equation 4.8).

\[
\text{DSER}_s = \frac{\# \text{GSSs with wrongly identified boundaries}}{\# \text{GSSs}} \tag{4.8}
\]

The DER is like DSER, but represents the inverted recall on the correct segmentation \textit{and} dialogue act classification of the segment (Equation 4.9).
<table>
<thead>
<tr>
<th>metric</th>
<th>counts</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSER&lt;sub&gt;s&lt;/sub&gt;</td>
<td>c</td>
<td>e</td>
</tr>
<tr>
<td>DER&lt;sub&gt;sc&lt;/sub&gt;</td>
<td>c</td>
<td>e</td>
</tr>
<tr>
<td>lenient&lt;sub&gt;c&lt;/sub&gt;</td>
<td>c</td>
<td>c</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>gold standard</th>
<th>predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>Q</td>
</tr>
</tbody>
</table>

Figure 4.13: Error metrics for segmentation and classification. Presence or absence of communicative function or segment boundaries is denoted with symbols | or ., respectively, and correct or erroneous classification is denoted with letters c or e, respectively.

\[ \text{DER}_{sc} = \frac{\# \text{GSSs with wrongly identified boundaries or CFs}}{\# \text{GSSs}} \]  

(4.9)

The Lenient error metric, or the inverted token-based accuracy on communicative functions, is the proportion of incorrectly identified communicative functions the token belongs to, from all tokens (Equation 4.10).

\[ \text{Lenient}_c = \frac{\# \text{tokens with wrongly identified CFs}}{\# \text{tokens}} \]  

(4.10)

The calculation of these metrics is illustrated in Figure 4.13. A dialogue fragment from a test set is shown with tokens that belong to a statement (S), question (Q), back-channel (B), or disruption (D) dialogue act. Presence or absence of communicative function or segment boundaries is denoted with symbols | or ., respectively, and correct or erroneous classification is denoted with letters c or e, respectively.

For example, in Figure 4.13 there are five segments in the gold standard. For three of these five segments, the boundaries are wrongly identified, resulting in a DSER<sub>s</sub> score of $3/5 = 0.60$.

In the above example, a simplified dialogue act scheme is used that is not multidimensional. When multiple functions or dialogue acts cooccur, each function is scored separately and the overall score expresses the average of the scores for each of the functions. In practice, this means that for each dimension in DIT<sup>++</sup>, a separate sequence of IOB tags is used.\(^7\)

\(^7\)As in each dimension only one communicative function can occur at the time.
4.3.4 Machine learning classifiers

As described in the introduction of this chapter, a wide variety of machine-learning techniques has been used for dialogue act recognition. For dialogue processing, it is still an open issue which techniques are the most suitable for which specific task.

In the machine learning studies various types of classifiers have been used that can roughly be grouped in those that are not specifically designed for learning to map input sequences to output sequences, and those that are. In the former case, three different types of classifiers were used: probabilistic, rule-based, and memory-based.

For a probabilistic classifier Naive Bayes was used. This algorithm assumes class-conditional independence, which does not always respect the characteristics of the features used. However, Naive Bayes often works well for complex real-world situations and is particularly suitable for problems where the dimensionality of the input is rather high. Moreover, this classifier requires relatively little computation and can be efficiently trained.

For rule induction, Ripper [Cohen, 1995] was used. The advantage of a rule induction algorithm is that regularities discovered in the data can be represented as human-readable rules. The memory-based algorithm that was used is IB1, which is a successor of the k-nearest neighbour (k-NN) algorithm. It stores a representation of all training examples in memory. When classifying new instances, it searches for the k most similar examples (nearest neighbours) in memory according to a similarity metric, and extrapolates the target class from this set to the new instances. The algorithm may yield more precise results given sufficient training data, because it does not abstract away low-frequent phenomena during the learning [Daelemans et al., 1999]. The results of all experiments were obtained using 10-fold cross-validation.

For the joint learning of segmentation and labelling, two different sequence-based machine-learning techniques were used: conditional random fields (CRFs) and memory-based tagging (MBT). Both of these have been shown to be particularly suitable for sequential natural language processing tasks such as part-of-speech (POS) tagging.

CRFs [Lafferty et al., 2001] are probabilistic learners for labelling and segmenting structured data. The algorithm defines a conditional probability distribution over label sequences given a particular observation sequence (in our case a sequence of tokens), rather than a joint distribution over both label and observation sequences. The main advantage of CRFs over e.g. hidden Markov models (HMMs) is their conditional nature, resulting in the relaxation of the independence assumptions that are required by HMMs to remain computationally feasible. For implementation the CRF++ package\(^8\) with default settings was used.

MBT is a memory-based tagger-generator that generates a sequence tagger on the basis of a training set of labelled sequences, and can subsequently tag new sequences [Daelemans et al., 2003]. It has been used to generate POS taggers and various chunkers. MBT can make use of the algorithmic parameters of TiMBL 5.2, a memory-based

\(^8\)CRF++ is publicly available at: http://crfpp.sourceforge.net/
In the experiments that are reported, a learner classifies a token from a dialogue (the token under consideration, which is called the focus token) in its context of other tokens (the context tokens). It depends on the features being used how much of a context a sequence learner will consider during classification, and the default token context initially assumed is 1. For all classifiers, the default settings will be used. It is possible to provide the learners additional information about tokens and context by means of a vector of features.

To have an idea what performance is like when using a simple heuristic, for most machine learning studies a baseline is set. When setting a baseline it is common practice to use the majority class tag, but for most of the data sets such a baseline is not very useful because of the relatively low frequencies of the tags in most dimensions. Instead, a baseline is computed that is based on a single feature, namely the tag of the previous dialogue utterance (see [Lendvai et al., 2003]).

### 4.3.5 Recognising DIT++ functions in task-oriented dialogue

**Outline**

The objective of this experiment is to see how machine learnable the recognition of the communicative functions in the DIT++ tagset is for task-oriented dialogues. The experiment is divided into three tasks. For the first task, the segment boundaries are known a priori (manually placed and corrected) and the machine learning algorithms are to predict the communicative function(s) giving features related to the segment and its context. The second task is more difficult: recognition of communicative functions by jointly predicting boundaries and the functions based on features of the previous tokens. The third task is similar to the second task: predicting boundaries only from the BI-encoding (without considering the kind of communicative function involved).

**Data**

The data set used in this classifications task is a superset of the data used in Section 4.2. It contains 1,214 utterances representing 1,592 functional segments from the human-human part of the DIAMOND corpus. The representation of dimensions, depicted in Figure 4.7, is roughly the same. The ten most frequent DIT++ functions in the dataset are listed in Table 4.3.

**Features**

Every communicative function is required to have some reflection in observable features of communicative behaviour, i.e. for every communicative function there are devices which a speaker can use to allow its successful recognition by the addressee such

---

9MBT and TiMBL are publicly available at: [http://ilk.uvt.nl/](http://ilk.uvt.nl/)
as linguistic cues, intonation properties, dialogue history, etc. State-of-the-art automatic dialogue understanding uses all available sources to interpret a spoken utterance.

The features available are related to dialogue context, prosody, and word occurrence. Word occurrence was represented by a bag-of-words vector indicating the presence or absence of words in the segment. The dialogue context was specified by bag-of-words vectors for the previous three turns. Prosodic features that were included are minimum, maximum, mean, and standard deviation of pitch (F0 in Hz), energy (RMS), voicing (fraction of locally unvoiced frames and number of voice breaks), and duration.

### Results of the first task

The resulting scores from using the three different classifiers introduced in Section 4.3.4 are presented in Table 4.4.

Table 4.3: Distribution of the ten most frequent DIT++ functions

<table>
<thead>
<tr>
<th>function</th>
<th>%</th>
<th>function</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. TUM: none-keep</td>
<td>14.8</td>
<td>6. TAS: wha</td>
<td>7.4</td>
</tr>
<tr>
<td>2. TAS: inform</td>
<td>13.1</td>
<td>7. TAS: whq</td>
<td>5.7</td>
</tr>
<tr>
<td>3. TIM: stall</td>
<td>11.9</td>
<td>8. TAS: elaborate</td>
<td>4.6</td>
</tr>
<tr>
<td>4. AUF: pos-exec</td>
<td>11.0</td>
<td>9. TAS: ymq</td>
<td>3.7</td>
</tr>
<tr>
<td>5. TAS: instruct</td>
<td>07.9</td>
<td>10. TAS: yna</td>
<td>3.7</td>
</tr>
</tbody>
</table>

From Table 4.4 it can be seen that Ripper outperforms Naive Bayes (NBayes) and
IB1 for almost all dimensions. The scores of all three systems is higher than those of the baseline (BL). Furthermore, it should be noted that some dimensions are frequently addressed whereas others are barely used, as can be seen in the distribution of functions over dimensions depicted in Figure 4.14.

![Figure 4.14: Distribution of functions over dimensions in the DIAMOND corpus.]

For scores of dimensions that are infrequently addressed, e.g. Own-communication and Partner communication management, it is therefore not possible to draw definite conclusions based on the scores presented.

The output of the rule inducer, Ripper, shows that for the prediction of the Task dimension, the bag-of-words vector representing word occurrence in the segment is important. For example, the presence of ‘hear’ in a segment is a good indicator for identifying feedback on perception; the occurrence of ‘like’, or ‘for example’, or ‘maybe’ and ‘might’ for the function SUGGESTION. Also the duration of the segment is usually longer than for example segments which address the Time or Turn Management dimensions. For the prediction of questions, word occurrence (e.g. wh-words in SET-QUESTIONS, and ‘or’ for ALTERNATIVE QUESTIONS) and prosodic features like standard deviation in pitch are essential. For the segments which are identified as having Information-Providing functions, important features were detected in the dialogue history, e.g. CONFIRM about the task is often response to a previous CHECK question about the task. The segments addressing the Auto-Feedback dimension are classified successfully on the basis of their word occurrence and dialogue history. The occurrence of words like ‘okay’ and ‘hmhm’ is an important clue for their recognition. As for the dimensions Turn and Time Management, the duration of the segment is a key feature, because the duration of these segments tends to be shorter than that of others. Moreover, these utterances are pronounced less loudly and are less voiced (e.g. about 44% of unvoiced frames). They usually occur inside larger segments, mostly in the beginning or in the middle. If they appear in clause-initial position, they usually have turn-initial functions (TAKE, ACCEPT, GRAB) and the function STALLING in the Time Management dimension; if they occur in the middle of the main segment they are used to signal that the speaker has some difficulties in completing his/her utterance, needs some time and wants to keep the turn. Of course, usage of words like ‘um’, ‘well’,
but also lengthening the words indicates the speaker’s hesitation and/or difficulties in utterance completion. Important cues for retracts (in the dimension Own Communication Management) are the relation to what is actually retracted (“reply to” feature), and the energy with which they are spoken (i.e. they are pronounced louder than the retracted reparandum).

**Results of the second task**

The resulting scores for identifying dialogue acts without prior segmentation are presented in Table 4.5.

Table 4.5: Scores for each DIT++ dimension without prior segmentation.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>BL</th>
<th>NBayes</th>
<th>CRF</th>
<th>MBT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$F_1$</td>
<td>DER$_{sc}$</td>
<td>$F_1$</td>
<td>DER$_{sc}$</td>
</tr>
<tr>
<td>task</td>
<td>14.8</td>
<td>88.3</td>
<td>27.1</td>
<td>77.8</td>
</tr>
<tr>
<td>auto feedback</td>
<td>27.1</td>
<td>75.6</td>
<td>37.4</td>
<td>67.1</td>
</tr>
<tr>
<td>allo feedback</td>
<td>19.1</td>
<td>81.7</td>
<td>32.2</td>
<td>69.5</td>
</tr>
<tr>
<td>turn</td>
<td>40.3</td>
<td>68.1</td>
<td>31.7</td>
<td>69.9</td>
</tr>
<tr>
<td>time</td>
<td>34.5</td>
<td>72.8</td>
<td>30.9</td>
<td>73.1</td>
</tr>
<tr>
<td>contact</td>
<td>33.8</td>
<td>73.3</td>
<td>34.6</td>
<td>72.4</td>
</tr>
<tr>
<td>own communication</td>
<td>16.7</td>
<td>89.0</td>
<td>29.8</td>
<td>75.0</td>
</tr>
<tr>
<td>partner communication</td>
<td>14.5</td>
<td>91.6</td>
<td>19.0</td>
<td>83.6</td>
</tr>
<tr>
<td>dialogue structuring</td>
<td>07.3</td>
<td>95.2</td>
<td>21.5</td>
<td>80.3</td>
</tr>
<tr>
<td>social obligations</td>
<td>11.0</td>
<td>91.3</td>
<td>44.0</td>
<td>56.2</td>
</tr>
</tbody>
</table>

The lower scores for each dimension for simultaneous segmentation and classification with the token-based representation (Table 4.5) shows that the task is considerably more difficult. All algorithms perform better than the majority class baseline, with MBT showing the best overall performance. In general, we see that the magnitude of performance for the dimensions is in the same range for the two tasks.

**Results of the third task**

In the third task, the aim is to evaluate what strategy is preferred: to do segmentation and classification at the same time, or one after the other. To investigate this, the highest score in each dimension obtained from the experiment in the second task, both for segmentation and segmentation+classification is taken. For simplicity we express the performance only by the $DSER_c$ and $DER_{sc}$ metric, respectively. With almost the same data the second task is repeated, the difference being that now the class labels consist of $IOB$-prefixes only. The highest scores for both tasks are listed in Table 4.6.
Table 4.6: Scores for DIT++ dimensions with two-step learning (left) and joint learning (right).

<table>
<thead>
<tr>
<th></th>
<th>two-step</th>
<th></th>
<th>joint</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DSERs</td>
<td>DERs</td>
<td>DSERs</td>
<td>DERs</td>
</tr>
<tr>
<td>task</td>
<td>56.4</td>
<td>-</td>
<td>49.1</td>
<td>63.2</td>
</tr>
<tr>
<td>auto feedback</td>
<td>41.8</td>
<td>-</td>
<td>35.5</td>
<td>59.4</td>
</tr>
<tr>
<td>allo feedback</td>
<td>44.9</td>
<td>-</td>
<td>38.0</td>
<td>53.1</td>
</tr>
<tr>
<td>turn</td>
<td>36.5</td>
<td>-</td>
<td>33.8</td>
<td>52.6</td>
</tr>
<tr>
<td>time</td>
<td>56.9</td>
<td>-</td>
<td>39.3</td>
<td>54.7</td>
</tr>
<tr>
<td>contact</td>
<td>44.7</td>
<td>-</td>
<td>44.9</td>
<td>60.1</td>
</tr>
<tr>
<td>own communication</td>
<td>47.8</td>
<td>-</td>
<td>45.6</td>
<td>60.9</td>
</tr>
<tr>
<td>partner communication</td>
<td>58.5</td>
<td>-</td>
<td>60.0</td>
<td>81.7</td>
</tr>
<tr>
<td>dialogue structuring</td>
<td>56.8</td>
<td>-</td>
<td>53.4</td>
<td>68.2</td>
</tr>
<tr>
<td>social obligations</td>
<td>35.4</td>
<td>-</td>
<td>32.1</td>
<td>47.7</td>
</tr>
</tbody>
</table>

Comparing the two DSER columns, it is clear that when using the same algorithms learning segmentation and classification together results —in general— in better segmentation performance than when first learning segmentation as a preparatory step for dialogue act classification.

4.3.6 Recognising DAMSL dialogue acts in task-oriented dialogue

Outline

The objective of this machine learning experiment is to see how machine learnable the dialogue acts in the DAMSL tagset are without prior segmentation (using a token-based representation). The experiment is comparable to the second task described in Section 4.3.5, but aims specifically at learning the functional segment boundaries (and CF labels) within a turn, without elaborate contextual information. In this task, sequence learners aimed specifically at mapping input sequences to output sequences are contrasted with baseline algorithms.

Data

The data used in this experiment consists of the eight dialogues (of twenty in total) from the MONROE corpus (see Section 3.3.1) that have been annotated with dialogue acts in the four DAMSL layers [Allen and Core, 1997]. The MONROE corpus is anno-
tated with 13 main dialogue act labels that can further contain arguments. From the two major layers of the annotation, the forward-looking and the backward-looking dimension, seven tags are included in the dataset. These are: STATEMENT, INFLUENCE-ON-LISTENER, INFLUENCE-ON-SPEAKER, INFO-REQUEST, AGREEMENT, UNDERSTANDING, and ANSWER.

The transcribed utterances in this dataset tend to be long, as dialogue units are segmented in a rather coarse-grained fashion. Only by listening to the audio recordings and by taking into account the overlapping speech marked by numerous turn-internal + symbols in the transcriptions, it is possible to segment the interaction further.

Features

The features that were used are automatically extracted from the dialogue transcriptions. Some attributes were derived using certain knowledge of transcribed boundaries; this has to do with limitations of (one of) the machine learning algorithms (even though sequence learners can handle a sequence of hundreds of tokens, it is not feasible to feed them entire dialogues).

All words were tokenised, dealing with capitalisation, separating and expanding clitics, etc., and subsequently stemmed with a Porter stemmer [Porter, 1980]. Also part-of-speech tags were used, which were automatically obtained by training MBT on the Wall Street Journal treebank [Marcus et al., 1993].

Lexical context was included as bag-of-words vectors: one containing the last 12 words uttered by the current speaker, one containing the most recently uttered 12 words of the interlocutor that spoke immediately before the current speaker, and one covering six tokens of right-context for the current speaker only, since it would be incorrect to assume the current speaker to have certainty about what the next speaker will contribute. A threshold on the lexicon size of the bag-of-words vectors has been set to only consider the 200 most frequent word tokens in the dialogues.

The [SIL] and + markings in the transcriptions were used as features for the machine learning, indicating whether or not an utterance starts or stops with a silence.

Results

The resulting scores from the three baseline algorithms and the two sequence learners are presented in Table 4.7.

From Table 4.7 it can be concluded that suggesting the majority class is a bad strategy (8.4% accuracy), since only one out of seven binary classifiers has a chance to score at all. The $F_1$ scores of both sequence learners improve largely over all baselines, confirming that sequential approaches are superior to local classification in the dialogue act identification task.

---

10 The DAMSL annotations as well as the transcripts can be found at: http://www.cs.rochester.edu/research/cisd/resources/monroe/annotate.html
Table 4.7: Dialogue act classification performance of seven binary classifiers for the MONROE corpus.

<table>
<thead>
<tr>
<th></th>
<th>token</th>
<th></th>
<th></th>
<th>token+others</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy</td>
<td>$F_1$</td>
<td>$\text{DER}_{sc}$</td>
<td>Accuracy</td>
<td>$F_1$</td>
</tr>
<tr>
<td>BL Maj.</td>
<td>8.4</td>
<td>8.5</td>
<td>90.1</td>
<td>8.3</td>
<td>8.8</td>
</tr>
<tr>
<td>BL NBayes</td>
<td>88.9</td>
<td>27.4</td>
<td>74.8</td>
<td>76.5</td>
<td>6.0</td>
</tr>
<tr>
<td>BL IB1</td>
<td>87.1</td>
<td>21.6</td>
<td>77.9</td>
<td>85.1</td>
<td>21.1</td>
</tr>
<tr>
<td>CRF</td>
<td>87.8</td>
<td>38.1</td>
<td>67.3</td>
<td>84.4</td>
<td>30.9</td>
</tr>
<tr>
<td>MBT</td>
<td>86.3</td>
<td>34.7</td>
<td>68.0</td>
<td>85.8</td>
<td>35.4</td>
</tr>
</tbody>
</table>

When taking other features than the word token only into account, the performance of the CRF learner is affected negatively whereas MBT shows a slight improvement.

A better understanding of what the scores represent is obtained by splitting down the scores according to dialogue act type (see Table 4.8). To get an impression of the contributions of contextual features in addition to using tokens only, also the combination of token and the bag-of-words vector from the contribution of the most recent previous other interlocutor is considered.

Table 4.8: $F_1$ scores per dialogue act type for the MONROE corpus using various feature sets and sequence learners.

<table>
<thead>
<tr>
<th></th>
<th>backward-looking</th>
<th>forward-looking</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Agr</td>
<td>Und</td>
</tr>
<tr>
<td>CRF</td>
<td>token</td>
<td>54.3</td>
</tr>
<tr>
<td></td>
<td>token+others</td>
<td>45.2</td>
</tr>
<tr>
<td></td>
<td>token+$BOW_{left,oth}$</td>
<td>47.0</td>
</tr>
<tr>
<td>MBT</td>
<td>token</td>
<td>53.2</td>
</tr>
<tr>
<td></td>
<td>token+others</td>
<td>46.1</td>
</tr>
<tr>
<td></td>
<td>token+$BOW_{left,oth}$</td>
<td>38.2</td>
</tr>
</tbody>
</table>

When splitting up the scores according to dialogue act type, the results indicate that for some dialogue act types there is indeed an improvement over the token+others approach from the additional information (although not over the token experiment).
4.3.7 Recognising SWBD-DAMSL dialogue acts in dialogues with more than two interlocutors

Outline

This section deals with how well dialogue with more than two interlocutors can be segmented into dialogue acts and how well dialogue acts can be classified. Particularly for answering the first question, the task will be to find the turn-internal boundaries that separate consecutive turn-internal dialogue acts, as outlined in Section 4.3.2. Additionally, the identification of dialogue act type is evaluated, not only as part of the joint learning, but also based on a gold standard segmentation to find out how successful classification as such is. In the dialogue act recognition experiments, various machine learning classifiers will be used to get a better indication of overall machine performance.

Data

The MRDA corpus [Shriberg et al., 2004] is a companion set of segmentations and annotations on the ICSI Meeting Corpus (see also 3.3.2), which consists of 75 non-scenario based meetings that each are roughly an hour in length. On average, there are six English speakers, native and non-native, per meeting. Most of the meetings were group discussions about the ICSI meeting recording project itself or on topics in natural language processing.

The utterances in the MRDA corpus have been annotated with a modified version of the SWBD-DAMSL tagset [Jurafsky et al., 1997], in which a dialogue act is a combination of at least one general tag, with a variable number of possible specific tags attached. There are 11 general tags and 39 specific tags. The MRDA corpus has been used in various segmentation and dialogue act classification studies and as in most of these studies the main types will have the focus. The dialogue act labels are grouped into five types: backchannels (B), disruptions (D), floorgrabbers (F), questions (Q), and statements (S), as well as two miscellaneous labels (X and Z). The MRDA data contains 51,452 turns (on average 826 turns per dialogue). In this experiment the focus will be on learning these five classes of dialogue acts. From the distribution of these five classes, depicted in Figure 4.15, it turns out that more than half of the dialogue acts in the corpus are STATEMENTS.

When the length of the turns are considered, there are 1.75 dialogue act units per turn at average. As can be seen from Figure 4.16, many of the turns express a single dialogue act.

Features

The features that are used are automatically extracted from the dialogue transcriptions and are almost identical as those mentioned in Section 4.3.6.
As with the MONROE dialogue data, all words were tokenised and subsequently stemmed with a Porter stemmer Porter [1980].

Results

The scores resulting from using the three baseline algorithms and the two sequence learners are presented in Table 4.9.

From Table 4.9 it can be concluded that suggesting the majority class (BL Maj.) is already rather accurate (81%), but recalls only a small fraction of the dialogue acts correctly, yielding a relatively low $F_1$ score (26.8%). Like in the MONROE experiments, the $F_1$ scores of both sequence learners improve over all baselines.

On the MRDA data we see a slight improvement over the token-only experiment for CRF (44.3% versus 41.2% $F_1$). In contrast, MBT’s scores seem to weaken for the large feature vector (45.1% versus 47.4 $F_1$).

\*\*\*Both stemming and lemmatising have been considered, but lemmatising did not result in a substantial improvement over stemming. For lemmatizing, MBLEM, a memory-based lemmatiser was used.
Table 4.9: Classification performance for the MRDA corpus, computed by multiclass learning of five dialogue act types.

<table>
<thead>
<tr>
<th></th>
<th>token</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy</td>
<td>$F_1$</td>
<td>$\text{DER}_{sc}$</td>
<td>Accuracy</td>
<td>$F_1$</td>
</tr>
<tr>
<td>BL Maj.</td>
<td>81.1</td>
<td>26.8</td>
<td>77.9</td>
<td>81.1</td>
<td>26.8</td>
</tr>
<tr>
<td>BL NBayes</td>
<td>81.8</td>
<td>15.6</td>
<td>78.3</td>
<td>81.2</td>
<td>16.5</td>
</tr>
<tr>
<td>BL IB1</td>
<td>79.4</td>
<td>23.0</td>
<td>81.1</td>
<td>83.1</td>
<td>36.7</td>
</tr>
<tr>
<td>CRF</td>
<td>83.2</td>
<td>41.2</td>
<td>65.0</td>
<td>84.3</td>
<td>44.3</td>
</tr>
<tr>
<td>MBT</td>
<td>83.9</td>
<td>47.4</td>
<td>56.2</td>
<td>81.8</td>
<td>45.1</td>
</tr>
</tbody>
</table>

Table 4.10: $F_1$ scores per dialogue act type for the MRDA corpus, using various feature sets and sequence learners.

<table>
<thead>
<tr>
<th></th>
<th>$B$</th>
<th>$D$</th>
<th>$F$</th>
<th>$Q$</th>
<th>$S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRF token</td>
<td>68.7</td>
<td>14.0</td>
<td>39.9</td>
<td>22.8</td>
<td>38.1</td>
</tr>
<tr>
<td>token + others</td>
<td>68.2</td>
<td>01.2</td>
<td>38.3</td>
<td>20.4</td>
<td>44.0</td>
</tr>
<tr>
<td>token + $\text{BOW}_{\text{left,other}}$</td>
<td>66.1</td>
<td>01.0</td>
<td>28.0</td>
<td>10.1</td>
<td>37.8</td>
</tr>
<tr>
<td>MBT token</td>
<td>70.0</td>
<td>15.6</td>
<td>39.1</td>
<td>33.7</td>
<td>46.2</td>
</tr>
<tr>
<td>token + others</td>
<td>63.4</td>
<td>19.3</td>
<td>40.3</td>
<td>28.2</td>
<td>46.3</td>
</tr>
<tr>
<td>token + $\text{BOW}_{\text{left,other}}$</td>
<td>63.9</td>
<td>18.0</td>
<td>37.4</td>
<td>31.0</td>
<td>41.9</td>
</tr>
</tbody>
</table>

4.3.8 Encoding issues

It has been shown that the differences in tagging performance as a result of variations of encodings that are syntactically equivalent to IOB on several other tasks, such as NP chunking, appear to be minor [Tjong Kim Sang and Veenstra, 1999]. Nevertheless, as the application domain is different, the question remains how much impact the choice of encoding can have on dialogue act tagging. This issue will be addressed by running a series of classification experiments where all experiment parameters (data, features, classification algorithm) are the same except for the encoding.

The representations that were compared are variations of encoding explicitly a specific beginning ($B$) or end ($E$), whether a token is inside ($I$) or not ($O$), or if it is the case that the unit consists of one token only ($S$):

- IOB1: The $B$ tag marks the first token of a unit that immediately follow another unit of the same type;

(see http://ilk.uvt.nl/mblem).
• IOB2: The B tag marks every first token of a unit;
• IOE1: The E tag marks the last token of a unit that immediately precedes another unit of the same type;
• IOE2: The E tag marks every last token of a unit;
• IOBES: The B tag marks every first token, the E tag every last token, and the S tag every token that makes the full unit.

The various encodings are exemplified in Figure 4.17, in which the per-class encodings are joined as they are mutually exclusive for the main classes of MRDA.

\[\ldots\]

<table>
<thead>
<tr>
<th></th>
<th>IOB1</th>
<th>IOB2</th>
<th>IOE1</th>
<th>IOE2</th>
<th>IOBES</th>
</tr>
</thead>
<tbody>
<tr>
<td>It</td>
<td>I O B O</td>
<td>I O B O</td>
<td>I O B O</td>
<td>I O B O</td>
<td></td>
</tr>
<tr>
<td>Is</td>
<td>I O I O</td>
<td>I O I O</td>
<td>I O I O</td>
<td>I O I O</td>
<td></td>
</tr>
<tr>
<td>Finished</td>
<td>I O I O</td>
<td>I O I O</td>
<td>I O E E</td>
<td>I O E E</td>
<td></td>
</tr>
<tr>
<td>What</td>
<td>O I O B</td>
<td>O I O B</td>
<td>O I O B</td>
<td>O I O B</td>
<td></td>
</tr>
<tr>
<td>Do</td>
<td>O I O I</td>
<td>O I O I</td>
<td>O I O I</td>
<td>O I O I</td>
<td></td>
</tr>
<tr>
<td>You</td>
<td>O I O I</td>
<td>O I O I</td>
<td>O I O I</td>
<td>O I O I</td>
<td></td>
</tr>
<tr>
<td>Mean?</td>
<td>O I O I</td>
<td>O E O E</td>
<td>O E O E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>It</td>
<td>O B O B</td>
<td>O I O B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Works?</td>
<td>O B O B</td>
<td>O I O B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>I O B O</td>
<td>I O E E</td>
<td>I O E E</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.17: Dialogue act tagging with various encodings

To compare the encodings a classification task using MRDA data (see 4.3.7), the MBT algorithm with an elementary feature set was used. Additional to the bag-of-word vectors that have been used in experiments on the MRDA data, also prosodic features were used: minimum, maximum, mean, standard deviation, contour of pitch (F0 in Hz), energy (RMS), voicing (fraction of locally unvoiced frames and number of voice breaks), and duration. The classification scores using various encodings are presented in Table 4.3.8.

From the results it appears that for the classification with token and bag-of-word vectors, explicitly marking spans by their starting token (the IOB encodings) results in slightly better scores than marking explicitly their ending token (the IOE encodings). This appears to be the case because segment-initial tokens usually are somewhat more predictive (e.g. dialogue act initial wh-words for questions). Adding prosodic speech features corrects this bias. However, all differences that are observed due to variation in encoding are very small and do not give reason to prefer one encoding over another.

In the series of classification experiments, all but the encoding was invariant. Study into the use of various encodings could be extended by looking to the interaction of
Table 4.11: Scores on MRDA data using various encodings.

<table>
<thead>
<tr>
<th>representation</th>
<th>tok+bow</th>
<th>tok+bow+pros</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOB1</td>
<td>47.3</td>
<td>56.8</td>
</tr>
<tr>
<td>IOB2</td>
<td>47.2</td>
<td>56.5</td>
</tr>
<tr>
<td>IOE1</td>
<td>47.0</td>
<td>56.8</td>
</tr>
<tr>
<td>IOE2</td>
<td>47.1</td>
<td>56.9</td>
</tr>
<tr>
<td>IOBES</td>
<td>47.2</td>
<td>56.7</td>
</tr>
</tbody>
</table>

the encoding choice with other parameters. For instance, a particular encoding might result in a slightly different performance depending on the choice of machine learning algorithm.

4.3.9 Discussion

In the first task in Section 4.3.5, the correct boundaries are given beforehand, and the scores represent how well the dialogue act classification works in this case. In the second experiment, the task is to recognise the dialogue acts, involving simultaneous segment boundary prediction and dialogue act type prediction. The substantially lower scores for the latter task confirm that predicting functional segment boundaries is difficult: the communicative functions in DIT++ can be identified with a weighted average of about 78% accuracy over all dimensions when the data is presegmented (utterance-based). In the token-based setting, scores for the more prominent dimensions range roughly between 40% and 60% $F_1$ score for combined segmentation and classification.

In the third task it is found that the segmentation obtained as a byproduct of the joint model is more accurate than the segmentation obtained when learning a separate model. When comparing the results of Section 4.3.6 with that of Section 4.3.7, it can be observed that the magnitude of performance is in the same range for the two dialogue datasets, despite that it may be more difficult to find boundaries of a large number of short functional segments than to identify fewer long ones. Where classification based on prior (correct) segmentation results in acceptable scores, token-based classification without prior segmentation turns out to be a difficult task (following from the experiments in Section 4.3.5).

A direct comparison between the machine learning experiments in the previous sections may not be informative, due to the differences between data, tagsets, and algorithms. Nonetheless, some general trends can be observed. The $F_1$ scores of both sequence learners used (CRF and MBT) improve largely over all baselines, indicating that sequential approaches are superior to local classification in the dialogue act identi-
CHAPTER 4. DIALOGUE SEGMENTATION AND DIALOGUE ACT RECOGNITION

The performance of CRFs deteriorates on the MONROE data when learning from rich feature vectors compared to token-only features, whereas MBT scores almost identically regardless of the features involved.

On the MRDA data we see a slight improvement over the token-only experiment for CRF. In contrast, the MBT scores seem to weaken for the large feature vector. The two sequence learners work in an inherently different way, which may explain this divergence. On the smaller dataset (MONROE) CRF performs somewhat better than MBT, especially in the token-only experiment (38.1% vs 34.7% $F_1$), but this is not the case for the larger dataset (MRDA), where MBT outperforms CRF in both experimental series.

Overall, the two sequence learning methods were able to produce $F_1$ scores in a similar range for rather differently transcribed dialogue datasets, both for the MRDA meeting transcriptions and the more traditionally transcribed MONROE scenario dialogues that feature longer turns and a giver-follower dialogue style. Comparing the utility of the lexical token-only versus an elaborate set of simple contextual features, it can be concluded that lexical items carry the best information for assigning dialogue act labels in the explorations reported here.

4.4 Human and machine performance in dialogue act classification

4.4.1 Introduction

The analysis of annotator disagreements presented in Chapter 3 gives an indication of the complexity of the task of dialogue act labelling. As with most tasks that involve semantic or pragmatic reasoning, human performance is considered the upper bound and determines how well machines could possibly perform in such tasks. Usually, human annotators will in addition to the errors caused by overlooking or misplacing something also make decisions based on conclusions for which genuine disagreement exists. In these cases, inter-annotator agreement may be considered as an indication of the upper bound for machines. However, as also shown in Chapter 3, human performance strongly depends on familiarity and experience with the task at hand, and when a ground truth can be established, the performance of annotators with varying experience can be compared. This offers also the possibility to compare the performance of machine classifiers with that of humans, which is interesting for several reasons. First, differences in learning bias and types of classification errors may provide insight into how humans learn or use specific kinds of concepts (see e.g., [Medin et al., 1987]). Second, comparing machine learning performance against that of humans may provide insight into the maturity and feasibility of the task (e.g., [Will, 1993] for information extraction).

This section describes the performance of a set of machine learning classifiers, trained on a set of dialogues similar to the ones reported on in Chapter 3, and tested
4.4. HUMAN AND MACHINE PERFORMANCE IN DIALOGUE ACT CLASSIFICATION

on the same dialogues that naive and expert annotators annotated. The purpose is to get an idea of how various machine learners perform in comparison to both annotator groups for the same data.

The machine learning algorithms that were used have been introduced in Section 4.3.4: IB1, a memory-based classifier, Ripper, a rule-based classifier, and Naive Bayes, a simple probabilistic classifier. The task for the machine learners is similar to that of the communicative function classification based on a priori segmentation in Section 4.3.5. The only difference is that instead of 10-fold cross-validation the results were obtained using 2-fold cross validation, effectively reducing the training data available to the machine learners from 90% to 50% of the dataset. In this way, the proportion of seen/unseen data between the human and machine annotators is more balanced.

4.4.2 Results

After training, the classification accuracy obtained was as follows: 68.1% IB1, 71.2% Ripper, 51.8% Naive Bayes. Figure 4.18 shows how these scores compare to those of the naive and expert annotators presented in Section 3.5.3.

As can be seen from the plot, the machine learners perform on group average less than the naive annotators. However, M1 and M2 perform better than most naive annotators whereas M3 (Naive Bayes) performs worse than any of the naive annotators and lowers the group average considerably. The biggest difference between the decision making of human and machine annotators turns out to be in the performance for low-frequency dimensions. In general, naive annotators performed better for low-frequency dimensions than machines, which can well be explained by the difficulty for the machine learners to infer useful models from sparse data. For the task-dimension, both groups scored about the same; M1 and M2 scored particularly well on the dimensions Auto-feedback and Time Management.

4.4.3 Discussion

An analysis of the annotations and the cooccurrence matrices reveals some clear differences in dialogue act classification between human and machine annotators.

A first notable difference that can be observed is in the variety of functions that is used in assignment. Consider, for example, the dialogue fragment in Figure 4.19, in which the annotations of annotator N4 and machine learner M1 are compared.

In this example, N4 assigns in the Task dimension the first utterance of the reply the function INFORM and considers utterance B_2 to B_4 elaborations of B_1, giving them the function ELABORATE. M1 (like M2 and M3) assigns consistently the function SET-ANSWER like it consistently does on most replies to a SET-QUESTION. Similarly, M1 consistently assigns the INSTRUCT function in the Task dimension, based on the
Figure 4.18: Tagging accuracy for naive (N) and expert (E) human annotators, and machines (M). M1 represents IB1, M2 represents Ripper, and M3 represents Naive Bayes.

Table: Utterance tagging accuracy

<table>
<thead>
<tr>
<th>utterance</th>
<th>N4</th>
<th>M1</th>
</tr>
</thead>
<tbody>
<tr>
<td>B₁ press the key function</td>
<td>INFORM</td>
<td>SET-ANSWER</td>
</tr>
<tr>
<td>B₂ press function key 6</td>
<td>ELABORATE</td>
<td>INSTRUCT</td>
</tr>
<tr>
<td>B₃ press a name button to</td>
<td>ELABORATE</td>
<td>INSTRUCT</td>
</tr>
<tr>
<td>B₄ now press the start button</td>
<td>ELABORATE</td>
<td>INSTRUCT</td>
</tr>
</tbody>
</table>

Figure 4.19: A dialogue excerpt annotated by a human and a machine in the Task dimension.

occurrence of the word ‘press’. This shows that the machine learners converge on regularities in surface features more than human annotators do.

To verify that there is a difference in variety of annotations between humans and machines, the purity of the assignments in each dimension for both groups can be compared. Given a set of assigned tags, the purity expresses how many different tags
have been assigned and relates this to the distribution of tags. The more variation in assignment, the lower the purity for each dimensions. The purity of each dimension can be determined by calculating its entropy. If within a dimension, all annotations have received the same communicative function, the entropy is 0. The greater the variety of the assigned CFs, the higher the entropy value. To compute the entropy, $E$, of a dimension, the distribution of assigned CFs is calculated. When a dimension counts $n$ communicative functions, $i$ denotes a dialogue act, and $p_i$ is the probability that CF $i$ was assigned, the entropy can be calculated with $E = -\sum_{i=0}^{n} p_i \cdot \log_2 p_i$.

To make a fair comparison between humans and machines, the probability distribution of machines is calculated using all annotations by the three machine learners. The probability distribution of humans is calculated by using the annotations of three human annotators that together balance human performance: N1, N6, and E1 of Figure 4.18. The entropy for the three dimensions that received the most annotations is listed in Table 4.12, together with the number of different communicative functions that have been observed in the annotations ($n$).

Table 4.12: Comparison of dimension entropy ($E$) between human and machine annotations for the most frequent dimensions, where $n$ denotes the number of different communicative functions that have been observed in the annotations.

<table>
<thead>
<tr>
<th></th>
<th>$E_{humans}$</th>
<th>$E_{machines}$</th>
<th>$n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>task</td>
<td>1.76</td>
<td>1.38</td>
<td>34</td>
</tr>
<tr>
<td>auto feedback</td>
<td>1.25</td>
<td>1.14</td>
<td>23</td>
</tr>
<tr>
<td>allo feedback</td>
<td>1.36</td>
<td>1.17</td>
<td>25</td>
</tr>
</tbody>
</table>

From Table 4.12 it can be concluded that, in general, the assignment by machines show less variety than that by humans: for all three dimensions the entropy of machines is lower, meaning that in general the machine learners make use of fewer functions.

Another notable difference between humans and machines is that the machine learners rely more on the simpler surface features in the feature set. For example, the ruleset of the rule inducer showed that in the dialogue fragment of Figure 4.19, the machine predicted an INSTRUCT when the word 'press' occurred in the beginning of the utterance, resulting in the machine learner to err when this heuristic fails.

### 4.4.4 Conclusions

In sum, it may be concluded that for annotating dialogue with communicative functions of DIT$$^+$$, machine learners and naive human annotators obtain comparable results. It should be noted that the difference in tagging accuracy between naive and expert human annotators is substantial.

Furthermore, it turns out that machines show less variety in the assignment of the
CFs, and that machine learners rely more on the simpler surface features in the provided feature set. Future research in this direction could include a more qualitative analysis to obtain more insight into differences by looking at what kind of specific errors and disagreements occur for humans and machines. Given the local nature of the features used by the machine learners, it is expected that machine learners have more difficulty with situations that require a deeper understanding of the dialogue situation. Moreover, it will be useful to see which information is being used when machine learners are successful. Also, in comparing human and machine performance, there has been no evaluation of the learning process and the data available to the machine learners was reduced to balance the proportion of seen/unseen data between the human and machine annotators. By monitoring how the performance of human annotators without any knowledge or experience and machine performance varies as a function of the training corpus size, insights may be gained in the difference of learning performance.
4.5 Summary and conclusions

The first question that was investigated in this chapter is how dialogue can be segmented accurately while taking the multifunctionality of dialogue behaviour into account. Whereas it is common practice to assign communicative functions to a single segmentation, it has been shown that for dialogue act taxonomies that allow the assignment of multiple functions to dialogue units, human communication can be described more accurately by means of multiple segmentations. Agreement scores on placing manual segment boundaries in the per-dimension segmentation did not differ much from conventional segmentation. There is, however, a tradeoff in the time needed to segment in multiple dimensions (segmenting and annotating in multiple dimensions costs between 1.5 and two times as much time).

The second question that was explored is how well the communicative functions in DIT++ can be recognised automatically. A token-based approach as opposed to a conventional utterance-based approach has been outlined, formulating both dialogue segmentation and dialogue act classification as a classification task. To explore key variables, three machine learning studies have been presented, looking into the machine-learnability of the DIT++ tagset evaluated in earlier chapters. With the utterance-based approach, it is possible to obtain a classification score higher than 70% $F_1$ for most dimensions, with for the Task dimension 70.5% $F_1$ and for the Auto- and Allo-feedback dimensions 85.1% $F_1$ and 96.6% $F_1$, respectively. The token-based approach, being inherently different, results in scores around 40-50% $F_1$, with for the Task dimension 43.4% $F_1$, and for the Auto- and Allo-feedback dimensions 46.3% $F_1$ and 50.7% $F_1$, respectively. It turns out that some of the dimensions in DIT++, like Own-Communication and Partner Communication management, are little addressed. This makes it difficult to draw firm conclusions for these dimensions based on small datasets. This is the case in both approaches, utterance-based and token-based.

Machine learning with sequence learners on tokens only already yield results that are not easy to improve by adding other features, such as prosodic ones. Nevertheless, further exploration of the feature space is expected to result in improvements for local classifiers and sequence learners able to deal with augmented token sequences. The results from Section 4.3.6 confirm that machine learners which are tailored to learning sequences perform better than local classifiers.

It was found that, when using the same algorithms, learning segmentation and classification together results in better segmentation performance than when first learning segmentation as a preparatory step for dialogue act classification. This outcome can be motivated in two ways. First, the joint approach does not have the problem of error propagation from the segmentation stage to the classification stage of the two-step approach. Second, the boundaries of the functional segment depend more on the specific communicative function than on the segment being a functional segment.

It was investigated whether the encoding of functional segment boundaries and communicative function labels would have an impact on the machine learning performance. Differences between the IOB1 encoding and other encodings turned out to be
very small.

To see how machine learners compare to naive and expert human annotators in tagging dialogue with DIT++ concepts, three machine learning classifiers were trained on the same dialogue material that was annotated manually by naive and expert annotators. It was shown that machine learners perform worse than expert annotators, and perform competitively with naive annotators.
While the previous chapter addressed the automatic segmentation and classification of dialogue acts, this chapter addresses dialogue management: the prediction of the next dialogue act(s) which could be performed in an ongoing conversation.

In this chapter, conversations are modelled as sequences of sets of dialogue acts, and a method of statistical act prediction based on grammar induction is presented. This method compares favourably to a heuristic baseline and to well known n-gram language models.

The act prediction is explored both for dialogue acts without realised semantic content (consisting of a communicative function only) and for dialogue acts with realised semantic content.

5.1 Introduction

Dialogue management is the activity of determining how to behave as an interlocutor at a specific moment of time in a conversation: which action can or should be taken at what state of the dialogue. The systematic way in which an interlocutor chooses among the options for continuing a dialogue is often called a dialogue strategy.

Any dialogue strategy aims to achieve successful communication. In an information-seeking context, a successful strategy should be based on a model of information inquiry and should support interlocutors in performing information-acquisition tasks. For example, a strategy could be to produce a series of questions according to a template, considering each reply as the content to fill a slot under question, and to repeat the question when the reply does not match any database entities (see frame-based models, Section 2.4.1). The strategy in this example is said to be single-initiative. Alternatively, a strategy could allow an interlocutor to take initiative and to decide in what
way to accomplish the task at hand (*mixed-initiative*). Such a strategy requires more sophistication. To have more natural dialogue, the strategy should include mechanisms to recover from communication problems.

Coming up with suitable dialogue management strategies for dialogue systems is not an easy task. Traditional methods typically involve manually crafting and tuning frames or rules, requiring considerable implementation time and cost. More recently, statistical methods are being used to semi-automatically obtain models that can be trained and optimised using dialogue data.¹ These methods are usually based on two assumptions. First, the training data is assumed to be representative of the communication that may be encountered in interaction. Second, it is assumed that dialogue action can be modelled as a Markov Decision Process (MDP) [Levin et al., 1998], which implies that dialogue is modelled as a sequential decision task in which each contribution (action) results in a transition from one state to another. The latter assumption allows to assign a reward for action-state pairs. These rewards can be used to determine the dialogue management strategy that results in the maximum expected reward by finding for each state the optimal action by using reinforcement learning (cf. [Sutton and Barto, 1998]). Reinforcement learning approaches to dialogue management have proven to be successful in learning dialogue management strategies in several task domains (see for example [Paek, 2006; Lemon et al., 2006]). In this process there is no supervision, but what is optimal depends usually on factors that require human action, such as task completion or user satisfaction.

This chapter presents an approach to dialogue act prediction which improves over n-gram language models and which can be used in isolation or for user simulation, without yet providing a full alternative to reinforcement learning.

After a brief introduction to the structural properties of the task-oriented kind of interaction the dialogues exhibit (Section 5.2), the way in which dialogue may be modelled as a sequence of symbols is explored in Section 5.3. The discussion focusses on the use of sequences of the communicative functions discussed in previous chapters, and outlines the specification and use of semantic content in the act prediction task. Related work on n-gram language models in dialogue act prediction is briefly discussed in Section 5.5, and in Section 5.6 a prediction algorithm based on grammatical inference is presented. Finally, experiments in comparing the two approaches are reported and discussed in Section 5.7.

### 5.2 Structural properties of task-oriented dialogue

One of the best known regularities that are observed in dialogue are the two-part structures, known as *adjacency pairs* [Schegloff, 1968], like QUESTION-ANSWER or GREETING-GREETING. An example from the DIAMOND corpus is given in Figure 5.1.

¹See e.g. [Young, 2002] for an overview.
 CHAPTER 5. DIALOGUE ACT PREDICTION

<table>
<thead>
<tr>
<th>utterance</th>
<th>function</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: what to do next?</td>
<td>WH-Q</td>
</tr>
<tr>
<td>B: press the button</td>
<td>WH-A</td>
</tr>
</tbody>
</table>

Figure 5.1: Example of an adjacency pair from the DIAMOND corpus.

A simple model of predicting a plausible next dialogue act that deals with such regularities could be based on bigrams, and to include more context also higher-order \( n \)-grams could be used. For instance, Stolcke et al. [2000] explore \( n \)-gram models based on transcribed words and prosodic information for DAMSL dialogue acts in the Switchboard corpus [Godfrey et al., 1992]. After training back-off \( n \)-gram models [Katz, 1987] of different order using frequency smoothing [Witten and Bell, 1991], it was concluded that trigrams and higher-order \( n \)-grams offer only a small gain in perplexity with respect to bigrams.

Apart from adjacency pairs, there is a variety of more complex re-occurring interaction patterns, such as in Figure 5.2, which illustrates a clarification sub-dialogue within an information request dialogue.

```
A: how do I do a fax?               SET-QUESTION
    B: do you want to send or print one? ALT-QUESTION
        A: i want to print it SET-ANSWER
        B: just press the grey button SET-ANSWER
```

Figure 5.2: Clarification sub-dialogue within an information request dialogue.

Such structures have received considerable attention and their models are referred to as discourse/dialogue grammars (e.g. [Polanyi and Scha, 1984]), or as conversational/dialogue games (e.g. [Levin and Moore, 1988]). Hulsijn [2000] argues that there is a strong link between such structures and claims they are recipes for joint action: “The smallest recipes for joint action are precisely the exchanges described by dialogue game rules. On the other hand, plans and goals may function as a semantics for dialogue game rules. They motivate the illocutionary point of initiating a game and explain various aspects of cooperativity in dialogue.”.

As also remarked by Levin et al. [1999], predicting and recognising dialogue games using \( n \)-gram models is not really successful. There are various causes for this. The flat horizontal structure of \( n \)-grams does not allow (hierarchical) grouping of symbols. This may weaken the predictive power and reduces the power of the representation since
nested structures such as exemplified above cannot be represented in a straightforward way.

A better solution would be to express the structure of dialogue games by a context-free grammar (CFG) representation. A CFG representing the last example is depicted in Figure 5.3 as a tree, in which the terminals are communicative functions and the non-terminals denote conversational games.

![CFG representing a dialogue game.](image)

Figure 5.3: CFG representing a dialogue game.

This would essentially imply the explicit specification of discourse grammar. This could be done by hand but it would be a great advantage if CFGs could automatically be induced from the data. In the latter case it is also easy to assess the frequency of typical patterns and a stochastic context-free grammar (SCFG) may be produced which can be used for parsing the dialogue data.

### 5.3 Sequencing dialogue acts

One issue with both $n$-gram language models and SCFG based models is that they work on sequences of symbols, and that these symbols can become quite complex in encoding essential information for adequate act prediction. Using more complex symbols increases data sparsity: encoding more information increases the number of unique symbols in the dataset and decreases the number of reoccurring patterns which could be used in the prediction.

In compiling the symbols for the prediction experiments, three aspects are important: identification of interlocutors, definition of dialogue acts in $\text{DIT}^{++}$, and the multidimensionality of the dialogue act taxonomy.

In the case of dialogue act prediction without taking multi-dimensionality into account, a dialogue $D$ can be represented as a sequence of dialogue acts $a$:

$$D = a_1, a_2, \ldots, a_n$$  (5.1)

where $n$ is the number of acts in the conversation and $a_i$ denotes the $i$-th act in sequence $D$. In the most elementary encoding of the dialogues, $D$ could express the $\text{DIT}^{++}$ communicative functions that are communicated, e.g.:

$$D = \text{SET-Q, PRO-Q, PRO-A, SET-A}$$  (5.2)
To be able to interpret who the dialogue act uttered, the speaker should be encoded as well. By encoding the interlocutors $A$ and $B$, sequence $D$ becomes:

$$D = A:\text{SET-Q}, B:\text{PRO-Q}, A:\text{PRO-A}, B:\text{SET-A}$$  \hspace{1cm} (5.3)

Another aspect which makes the symbols more complex is the multidimensionality in $\text{DIT}^{++}$: functions can occur simultaneously, each addressing a different dimension of communication. To get an indication of the impact of the multidimensionality of $\text{DIT}^{++}$ on the dataset, the distribution of main communicative functions with possible other functions occurring at the same time is depicted in Figure 5.4.

![Figure 5.4: Degree of multifunctionality in the dataset.](image)

It can be concluded from Figure 5.4 that for the kind of dialogue in the dataset, in 23 percent of the cases one could speak of multifunctionality, and taking this aspect into account will likely make the prediction task proportionally more difficult.

The information concerning interlocutor and multifunctionality is encoded in a single symbol and denoted by means of a $n$-tuple. Assuming that at most three functions can occur simultaneously, a 4-tuple is needed.\(^2\) A bigram of 4-tuples would then look as follows:

$$\langle A, \text{PRO-Q}, _, _, \rangle \langle B, \text{PRO-A}, \text{TURN-TAKE}, _ \rangle$$ \hspace{1cm} (5.4)

Two symbols are considered to be identical when the same speaker is involved and when the symbols both address the same functions. To make it easy to determine of two symbols are identical, the order of elements in a tuple should be fixed. A simple approach is to order the functions alphabetically on the name of the dimension they occur in. In many cases of multifunctionality, however, one function is usually the most important and can be denoted as the primary function of the utterance. For instance, if an utterance has both a task-related function and one or more other functions, the task-related function is typically considered to be more important than the other

\(^2\)Ignoring the half percent of occurrences with four simultaneous functions.
functions. This raises the question how recognition performance using multifunctional symbols compares against recognition performance using symbols that only encode the primary function, a comparison that is made in Section 5.7.3, 5.7.4, and 5.7.5. For this reason, functions that occur simultaneously are first ordered on importance of dimension, and subsequently on alphabet. The task-related functions are considered the most important, followed by feedback-related functions, followed by any other remaining functions.

5.4 Semantic interpretation

5.4.1 Including semantic content

Sequences of symbols based on DIT++ communicative functions describe only part of the meaning of dialogue utterances, as in DIT a dialogue act is defined as a pair consisting of a communicative function (CF) and a semantic content (SC): \( a = \langle CF, SC \rangle \).\(^3\) In quite some cases, particularly when dialogue control is addressed and dimension-specific functions are realised, the SC is empty. General-purpose functions, by contrast, are always used in combination with a realised SC. For the utterances in Figure 5.1, the dialogue acts would look as depicted in Figure 5.5. As can be understood

<table>
<thead>
<tr>
<th>utterance</th>
<th>dialogue act</th>
<th>function</th>
<th>semantic content</th>
</tr>
</thead>
<tbody>
<tr>
<td>A what to do next?</td>
<td>WH-Q</td>
<td>next-step(X)</td>
<td></td>
</tr>
<tr>
<td>B press the button</td>
<td>WH-A</td>
<td>press(Y) ∧ button(Y)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.5: CF and SC of adjacency pair of utterances from the DIAMOND corpus.

from Figure 5.5, the SC —if realised— describes objects, properties, and events in the domain of conversation.\(^4\)

Including the SC is important as predicting the future illocutionary force only is not sufficient for adequately updating a belief state. Dialogue acts with resolved SC also open the possibility to bootstrap a domain model from dialogue corpora only.\(^5\) By including the SC, however, the prediction task becomes considerably more difficult as—often depending on the size of the task domain—the SC may be very versatile and may make learning unfeasible because of data sparsity.

According to the encoding established in the previous section, the two dialogue acts in Figure 5.5 would be encoded as the following bigram:

\(^3\)See Section 2.3.5 for more details.

\(^4\)In many conversational settings, the conversation can usually be considered as having two domains: the interaction and the task or application.

\(^5\)The learning of the task domain is possible if the assumption can be made that the communicated semantic content is correct.
In general, using increasingly complex sequence symbols increases training data sparsity, which is likely to decrease the performance of the model. To see if this is actually the case, experiments are done on a variety of symbols.

5.4.2 Representing semantic content

Representation of semantic content are often expressed in some form of predicate logic type formula. Examples are varieties of description logics, which extend semantic frames and networks with a formal logic-based semantics that uses predicates.

In any case, the semantic representation should ideally be powerful enough to take into account complexities such as negation, quantification, a certain degree of underspecification and (complex) modifiers to be interesting for use in advanced interactive question answering systems and dialogue systems. Several semantic representations have been proposed that take these aspects into account, such as Quasi Logical Forms [Alshawi, 1990], Dynamic Predicate Logic [Groenendijk and Stokhof, 1991], Under-specified Discourse Representation Theory [Reyle, 1993], Minimal Recursion Semantics [Copestake et al., 1997], and GLUE semantics [Dalrymple, 1999]. Most importantly, the logical form should be suitable to support feasible reasoning, for which also theorem provers, model builders, and model checkers can be used.

To express the SC of the corpus data that will be used to statistically predict dialogue acts later in this chapter, a simplified first order logic is used similar to quasi logical forms. Utterances and their corresponding SC as found in the DIAMOND data are illustrated in Table 5.1.

Three types of predicate groups are distinguished: action predicates, element predicates, and property predicates. These types have a fixed order. The action predicates appear before element predicates, which appear in turn before property predicates. This allows to simplify the semantic content for the purpose of reducing data sparsity in act prediction experiments, by stripping away e.g. property predicates. For instance, if desired the SC of Example 3 in Table 5.1 could be simplified to that of Example 2, making the semantics less detailed but still meaningful.

In the domain of the DIAMOND data, i.e. operating a fax device, the predicates and arguments in the logical expressions refer to entities, properties, events, and tasks in the application domain. As explained in Chapter 3, the application domain of the fax device is complex but small: the domain model consists of 70 entities with at most 10 properties, 72 higher-level actions or tasks, and 45 different settings.

6For a survey, see [Bunt, 2007].
5.4. SEMANTIC INTERPRETATION

Table 5.1: Examples of utterances, and their corresponding SC in $\lambda$-expressions of first-order logic.

<table>
<thead>
<tr>
<th>utterance</th>
<th>semantic content</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 wat moet ik nu doen? (what do I have to do now?)</td>
<td>$\lambda x \cdot \text{next-step}(x)$</td>
</tr>
<tr>
<td>2 druk op een toets (press a button)</td>
<td>$\lambda x \cdot \text{press}(x) \land \text{button}(x)$</td>
</tr>
<tr>
<td>3 druk op de groene toets (press the green button)</td>
<td>$\lambda x \cdot \text{press}(x) \land \text{button}(x) \land \text{color}(x, \text{green})$</td>
</tr>
<tr>
<td>4 wat zit er boven de starttoets? (what is located above the starttoets?)</td>
<td>$\lambda x \cdot \text{loc-above}(x, \text{button041})$</td>
</tr>
<tr>
<td>5 wat doet de grote knop? (what does the big button do?)</td>
<td>$\lambda x \cdot \text{function}(x) \land \text{button}(x) \land \text{size}(x, \text{big})$</td>
</tr>
</tbody>
</table>

5.4.3 Semantic interpretation for utterances in the DIAMOND corpus

Semantic interpretation involves the process of 'translating' natural language to a representation of its meaning. Semantic interpretation could be understood as the task of mapping syntax to semantics, assuming that the syntactic relationships in an utterance correspond to functional relationships in the meaning representation.

Relevant work in this area using techniques from machine translation and machine learning in the mapping from natural language to meaning-representation languages is [Ge and Mooney, 2005; Wong and Mooney, 2006; Kate and Mooney, 2006]. These approaches can be robust, and thus would be useful in dealing with large quantities of utterances, but require large amounts of annotated data.

Since the syntax of natural language does not change much from domain to domain, an alternative way is to use the output of a wide-coverage syntactic parser as a basis for single, multiple, or even open-domain language processing. To obtain a sufficiently detailed semantic representation, the phrases in the parses should be linked with domain-specific knowledge concepts. This approach has been used to obtain the semantic content for the DIAMOND utterances.
CHAPTER 5. DIALOGUE ACT PREDICTION

Approach

The semantic representation is obtained in two stages. In the first stage, the utterances are syntactically parsed. In the second stage, the most probable derivation obtained in the syntactic parsing is used to construct the semantic representation.

For the syntactic interpretation of the utterances, the Alpino Parser is used [Bouma et al., 2001]. This HPSG-based\(^7\) dependency parser aims to accurately provide full parses of unrestricted Dutch text and is publicly available.\(^8\)

In the context of spoken dialogue processing, a syntactic parser has to deal with fragmented input and many syntactically less well-formed utterances in comparison to text parsing. For this reason, the utterances are additionally parsed with a shallow parser, and the resulting parse is used in case the Alpino parser fails to provide a full parse. As shallow parser, a memory based chunk parser trained for spoken Dutch [Canisius and van den Bosch, 2003] is employed.

To resolve pronouns, a simple pronoun resolution algorithm has been implemented. This algorithm is similar to the centering algorithm proposed by Tetreault [2001]. While processing the utterances, each noun phrase identified is placed on a temporary queue which is pushed on a history stack once the utterance or turn is closed. Upon encountering a pronoun, the first element on the queue that meets gender and number agreement is selected as antecedent. If no candidate is found, the previous queue on the stack is evaluated until an antecedent is found or all queues on the history stack are evaluated.

The semantic representation is constructed by traversing the dependencies in the parse and by mapping words and phrases to domain concepts. These domain concepts are events, elements, and domain tasks stored in a database. This process of semantic interpretation is depicted and exemplified with Example 3 from Table 5.1 in Figure 5.6.

The approach reported here has several aspects in common with that of Bos [2004], who uses a CCG based parser [Clark and Curran, 2004] and assigns Discourse Representation Structures (DRSs) to the lexical categories used by the parser after which semantic construction is driven by the syntactic derivation. A notable difference is that Bos first constructs a DRS representation which is subsequently translated into first-order logic. Another difference is that in the approach described in this section, syntactic representations obtained by the wide-coverage dependency parser are complemented with that of a chunk parser, which increases robustness when dealing with fragmented input, common in spoken dialogue.

Evaluation

The approach for obtaining semantic representations has been tested on a dataset of 160 utterances from the DIAMOND corpus and their corresponding semantic content.

\(^7\)The parser is based on the HPSG (Head-driven Phrase Structure Grammar) theory of grammar [Pollard and Sag, 1994].

\(^8\)See: http://www.let.rug.nl/~vannoord/alp/Alpino/.
All utterances are related to the fax domain. On complete semantic representations, an accuracy of 88.1% is achieved. The performance on identifying each of the three types of predicates in the semantic representations is specified in Table 5.2.

Table 5.2: Performance on semantic interpretation.

<table>
<thead>
<tr>
<th></th>
<th>action predicates</th>
<th>element predicates</th>
<th>property predicates</th>
<th>overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>accuracy (%)</td>
<td>92.2</td>
<td>81.4</td>
<td>94.3</td>
<td>88.1</td>
</tr>
</tbody>
</table>

Table 5.2 shows that identification of element predicates is the least successful. Where actions and properties are usually mentioned explicitly, domain elements can be described in various ways. Moreover, in a substantial number of utterances pronouns are used, which cannot always be resolved successfully.

5.5  N-gram language models

There exists a significant body of work on the use of language models in relation to dialogue management. Nagata and Morimoto [1994] describe a statistical model of discourse based on trigrams of utterances classified by custom speech act types. They
report 39.7% prediction accuracy for the top candidate and 61.7% for the top three candidates.

In the context of the dialogue component of the speech-to-speech translation system VERBMOBIL, Reithinger and Maier [1995] use n-gram dialogue act probabilities to suggest the most likely dialogue act. In later work, Alexandersson and Reithinger [1997] describe an approach which comes close to the work in this chapter: Using grammar induction, plan operators are semi-automatically derived and combined with a statistical disambiguation component. This system is claimed to have an accuracy score of around 70% on turn management classes.

Another study is that of Poesio and Mikheev [1998], in which prediction based on the previous dialogue act is compared with prediction based on the context of dialogue games. Using the Map Task corpus annotated with ‘moves’ (dialogue acts) and ‘transactions’ (games) they showed that by using higher dialogue structures it was possible to perform significantly better than a bigram model approach. Using bigrams, 38.6% accuracy was achieved. Additionally taking game structure into account resulted in 50.6%; adding information about speaker change resulted in an accuracy of 41.8% with bigrams, 54% with game structure.

All studies discussed so far are only concerned with sequences of communicative functions, and disregard the semantic content of dialogue acts.

5.6 Dialogue grammars

To automatically induce patterns from dialogue data in an unsupervised way, grammatical inference (GI) techniques can be used. GI is a branch of unsupervised machine learning that aims to find structure in symbolic sequential data. In this case, the input of the GI algorithm will be sequences of dialogue acts.

5.6.1 Dialogue Grammars Inducer

For the induction of structure, a GI algorithm has been implemented that will be referred to as Dialogue Grammars Inducer (DGI). This algorithm is based on distributional clustering and alignment-based learning [van Zaanen and Adriaans, 2001; van Zaanen, 2002; Geertzen and van Zaanen, 2004]. Alignment-based learning (ABL) is a symbolic grammar inference framework that has successfully been applied to several unsupervised machine learning tasks in natural language processing. The framework accepts sequences with symbols, aligns them with each other, and compares them to find interchangeable subsequences that mark structure. As a result, the input sequences are augmented with the induced structure.

The DGI algorithm takes as input time series of dialogue acts, and gives as output a set of SCFGs. The algorithm has five phases:

1. SEGMENTATION: In the first phase of DGI, the time series are —if necessary— segmented in smaller sequences based on a specific time interval in which no
communication takes place. This is a necessary step in task-oriented conversation in which there is ample time to discuss (and carry out) several related tasks, and an interaction often consists of a series of short dialogues.

2. ALIGNMENT LEARNING: In the second phase a search space of possible structures, called hypotheses, is generated by comparing all input sequences with each other and by clustering sub-sequences that share similar context. To illustrate the alignment learning, consider the input sequences in Figure 5.7.

A: SET-Q, B: PAUSE, B: RESUME, B: SET-A.
A: SET-Q, B: SET-A.

Figure 5.7: Example of input sequences for alignment learning.

The alignment learning compares all input sequences with each other, and produces the hypothesised structures depicted in Figure 5.8. The induced structure is represented using bracketing.

[ [ A: SET-Q, ] ]

Figure 5.8: Example of output sequences for alignment learning.

The hypothesis $j$ is generated because of the similar context (which is underlined). The hypothesis $i$, the full span, is introduced by default, as it might be possible that the sequence is in itself a part of a longer sequence.

3. SELECTION LEARNING: The set of hypotheses that is generated during alignment learning contains hypotheses that are unlikely to be correct. These hypotheses are filtered out, overlapping hypotheses are eliminated to assure that it is possible to extract a context-free grammar, and the remaining hypotheses are selected and remain in the bracketed output. The decision of which hypotheses to select and which to discard is based on a Viterbi beam search [Viterbi, 1967].

4. EXTRACTION: In the fourth phase, SCFG grammars are extracted from the remaining hypotheses by means of recursive descent parsing. Ignoring the stochastic information, a CFG of the above-mentioned example looks in terms of grammar rules as depicted in Figure 5.9.

5. FILTERING: In the last phase, the SCFG grammars that have small coverage or involve many non-terminals are filtered out, and the remaining SCFG grammars are presented as the output of DGI.
Depending on the mode of working, the DGI algorithm can generate a SCFG covering the complete input or can generate a set of SCFGs. In the former mode, the grammar that is generated can be used for parsing sequences of dialogue acts and by doing so suggests ways to continue the dialogue. In the latter mode, by parsing each grammar in the set of grammars that are expected to represent dialogue games in parallel, specific dialogue games may be recognised, which can in turn be used beneficially in dialogue management.

5.6.2 A worked example

In testing the algorithm, DGI has been used to infer a set of SCFGs from a development set of 250 utterances of the DIAMOND corpus (see also Section 5.7.1). Already for this small dataset, DGI produced, using default parameters, 45 ‘dialogue games’. One of the largest produced structures was the following:

```
S ⇒ A:SET-Q J B:SET-A
J ⇒ B:PRO-Q A:PRO-A
J ⇒ B:PAUSE A:RESUME
J ⇒ 0
```

Figure 5.9: A CFG as output of the DGI algorithm. The non-terminals are in bold.

```
4 S ⇒ A:SET-Q NTAX NTBT B:SET-A
4 NTAX ⇒ B:PRO-Q NTFJ
3 NTFJ ⇒ A:PRO-A
1 NTFJ ⇒ A:PRO-A A:CLARIFY
2 NTBT ⇒ B:PRO-Q A:PRO-A
2 NTBT ⇒ 0
```

Figure 5.10: A SCFG structure induced from the DIAMOND development set. Non-terminals are in bold.

In this figure, each CFG rule has a number indicating how many times the rules has been used. One of the dialogue fragments that was used to induce this structure is the excerpt in Figure 5.11.

Unfortunately, many of the 45 induced structures were very small or involved generalisations already based on only two input samples. To ensure that the grammars produced by DGI generalise better and are less fragmented, a post-processing step has been added which traverses the grammars and eliminates generalisations based on a low number of samples. In practice, this means that the post-processing requires the
5.7 Act prediction experiments

To determine how to behave as an interlocutor at a specific moment of time in a conversation, the DGI algorithm can be used to infer a SCFG that models the structure of the interaction. The SCFG can then be used to suggest a next dialogue act to continue the dialogue. In this section, the performance of the proposed SCFG based dialogue model is compared with the performance of the well-known \( n \)-gram language models, both trained on intentional level, i.e. on sequences of sets of dialogue acts.

5.7.1 Data

The task-oriented dialogues used in the dialogue act prediction tasks were drawn from the DIAMOND corpus (see Section 3.3.4), which contains human-machine and human-human Dutch dialogues that have an assistance seeking nature. The data set used in this task is the same as used in Section 4.3.5: It contains 1,214 utterances representing 1,592 functional segments from the human-human part of the DIAMOND corpus.\footnote{For an overview of the most frequent DIT++ functions in the dataset, please see Section 4.3.5.}

5.7.2 Methodology and metrics

Evaluation of overall performance in communication is problematic; there are no generally accepted criteria as to what constitutes an objective and sound way of comparative evaluation. An often-used paradigm for dialogue system evaluation is PARADISE

\[ N = 2 \] by default, but may increase with the size of the training data.

 Remaining grammatical structure to be presented \( N \) times or more in the data.\footnote{\( N = 2 \) by default, but may increase with the size of the training data.} The algorithm without post-processing will be referred to as DGI1; the algorithm with post-processing as DGI2.
[Walker et al., 2000], in which the performance metric is derived as a weighted combination of subjectively rated user satisfaction, task-success measures and dialogue cost. Evaluating if the predicted dialogue acts are considered as positive contributions in such a way would require the model to be embedded in a fully working dialogue system.

To assess whether the models that are learned produce human-like behaviour without resorting to costly user interaction experiments, it is needed to compare their output with real human responses given in the same contexts. This will be done by deriving a model from one part of a dialogue corpus and applying the model on an ‘unseen’ part of the corpus, comparing the suggested next dialogue act with the observed next dialogue act. To measure the performance, accuracy is used, which is defined as the proportion of suggested dialogue acts that match the observed dialogue acts.

In addition to the accuracy, also perplexity is used as metric. Perplexity is widely used in relation to speech recognition and language models, and can in this context be understood as a metric that measures the number of equiprobable possible choices that a model faces at a given moment. Perplexity, being related to entropy (introduced in Section 4.4.3) is defined as follows:

\[
\text{Entropy} = - \sum_i p(w_i|h) \cdot \log_2 p(w_i|h) \quad (5.5)
\]

\[
\text{Perplexity} = 2^{\text{Entropy}} \quad (5.6)
\]

where \(h\) denotes the conditioned part, i.e. \(w_{i-1}\) in the case of bigrams and \(w_{i-2}, w_{i-1}\) in the case of trigrams, et cetera. In sum, accuracy could be described as a measure of correctness of the hypothesis and perplexity could be described as how probable the correct hypothesis is.

For all \(n\)-gram language modelling experiments reported in this chapter, good-turing smoothing was used [Katz, 1987]. To reduce the effect of imbalances in the dialogue data, the results were obtained using 5-fold cross-validation.

To have an idea how the performance of both the \(n\)-gram language models and the SCFG models relate to the performance of a simple heuristic, a baseline has been computed which suggests a majority class label according to the interlocutor role in the dialogue. The information seeker has \(\text{SET-Q} \) and the information provider has \(\text{SET-A} \) as majority class label.

### 5.7.3 Results for communicative functions

The scores for communicative function prediction are presented in Table 5.3. For each of the three kinds of symbols, accuracy and perplexity are calculated: the first two columns are for the main CF, the second two columns are for the combination of speaker identity and main CF, and the third two columns are for the combination
of speaker identity and all CFs. The scores for the latter two codings could not be calculated for the 5-gram model, as the data were too sparse.

As was expected, there is an improvement in both accuracy (increasing) and perplexity (decreasing) for increasing $n$-gram order. After the 4-gram language model, the scores drop again. This could well be the result of insufficient training data, as the more complex symbols could not be predicted well.

Both language models and SCFG models perform better than the baseline, for all three groups. The two SCFG models, DGI1 and DGI2, clearly outperform the $n$-gram language models with a substantial difference in accuracy. Also the perplexity tends to be lower. Furthermore, model DGI2 performs clearly better than model DGI1, which indicates that the ‘flattening’ of non-terminals which is described in Section 5.6 results in better inductions.

When comparing the three groups of sequences, it can be concluded that providing the speaker identity combined with the main communicative function results in better accuracy scores of 5.9% on average, despite the increase in data sparsity. A similar effect has also been reported by Stolcke et al. [2000].

Only for the 5-gram language model, the data become too sparse to learn reliably a language model from. There is again an increase in performance when also the last two positions in the 4-tuple are used and all available dialogue act assignments are available. It should be noted, however, that this increase has less impact than adding the speaker identity. The best performing $n$-gram language model achieved 66.4% accuracy; the best SCFG model achieved 78.9% accuracy.
5.7.4 Results for dialogue acts

The scores for prediction of dialogue acts, including SC, are presented in Table 5.4. The presentation is similar to Table 5.4 in the previous section: for each of the three kinds of symbols, accuracy and perplexity were calculated. For dialogue acts that may include semantic content, computing a useful baseline is not obvious. The same baseline as for communicative functions was used, which results in lower scores.

Table 5.4: Dialogue act prediction scores for \( n \)-gram language models and SCFGs in accuracy (\( \text{acc} \), in percent) and perplexity (\( \text{pp} \)). CF\(_{\text{main}}\) denotes the main communicative function, SPK speaker identity, and CF\(_{\text{all}}\) all occurring communicative functions.

<table>
<thead>
<tr>
<th></th>
<th>DA(_{\text{main}})</th>
<th>SPK + DA(_{\text{main}})</th>
<th>SPK + DA(_{\text{all}})</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>acc</td>
<td>pp</td>
<td>acc</td>
</tr>
<tr>
<td>baseline</td>
<td>18.5±2.01</td>
<td>31.0±1.64</td>
<td>19.3±1.79</td>
</tr>
<tr>
<td>2-gram</td>
<td>31.2±1.42</td>
<td>28.5±1.03</td>
<td>34.6±1.51</td>
</tr>
<tr>
<td>3-gram</td>
<td>29.0±1.14</td>
<td>34.7±2.82</td>
<td>31.9±1.21</td>
</tr>
<tr>
<td>4-gram</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5-gram</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DGI1</td>
<td>38.8±3.27</td>
<td>25.1±0.94</td>
<td>42.5±0.96</td>
</tr>
<tr>
<td>DGI2</td>
<td>39.2±2.45</td>
<td>25.0±1.28</td>
<td>42.7±1.03</td>
</tr>
</tbody>
</table>

As can be seen from Table 5.4, the attempts to learn to predict additionally the semantic content of utterances quickly run into data sparsity problems. It turned out to be impossible to make predictions from 4-grams and 5-grams, and for 3-grams the combination of speaker and all dialogue acts could not be computed. Training the SCFGs, by contrast, resulted in fewer problems with data sparsity, as the models abstract quickly.

As with predicting communicative functions, the SCFG models show better performance than the \( n \)-gram language models, for which the 2-grams show slightly better results than the 3-grams. Where there was a notable performance difference between DGI1 and DGI2 for CF prediction, for dialogue act prediction there is only a very little difference, which is insignificant considering the relatively high standard deviation. This small difference is explained by the fact that DGI2 becomes less effective as the size of the training data decreases.

As with CF prediction, it can be concluded that providing the speaker identity with the main dialogue act results in better scores, but the difference is less big than observed with CF prediction due to the increased data sparsity.
5.7.5 Results for dialogue acts with simplified semantic content

The prediction scores of dialogue acts with full semantic content and simplified semantic content are presented in Table 5.5. For both cases multifunctionality is taken into account by including all occurring communicative functions in each symbol.

Table 5.5: Dialogue act prediction scores with full and simplified SC.

<table>
<thead>
<tr>
<th></th>
<th>SPK + DA_{all}</th>
<th>full SC</th>
<th>simplified SC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>acc</td>
<td>pp</td>
</tr>
<tr>
<td>baseline</td>
<td></td>
<td>18.2±1.93</td>
<td>31.6±1.38</td>
</tr>
<tr>
<td>2-gram</td>
<td></td>
<td>35.1±1.30</td>
<td>26.9±0.47</td>
</tr>
<tr>
<td>3-gram</td>
<td></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4-gram</td>
<td></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5-gram</td>
<td></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DGI1</td>
<td></td>
<td>42.9±2.44</td>
<td>27.3±1.98</td>
</tr>
<tr>
<td>DGI2</td>
<td></td>
<td>42.4±2.19</td>
<td>28.0±1.57</td>
</tr>
</tbody>
</table>

As can be seen from Table 5.5, the simplification of the semantic content leads to improvements in the prediction performance for both types of model. The best $n$-gram language model improved with 2.4% accuracy from 35.1% to 37.5%; the best SCFG-based model improved with 3.7% from 42.9% to 46.6%.

Moreover, the simplification of the semantic content reduced the problem of data-sparsity, making it also possible to predict based on 3-grams although the accuracy is considerably lower than that of the 2-gram model.

5.8 Discussion

Both $n$-gram language models and SCFG based models have their strengths and weaknesses. $n$-gram models have the advantage of being very robust and they can be easily trained. The SCFG based model can capture regularities that have gaps, and allow to model long(er) distance relations. Both algorithms work on sequences and hence are easily susceptible to data-sparsity when the symbols in the sequences get more complex. The SCFG approach, though, has the advantage that symbols can be clustered in the non-terminals of the grammar, which allows more flexibility.

The multidimensional nature of the DIT++ functions can be adequately encoded in the symbols of the sequences. Keeping track of the interlocutor and including not only the main communicative function but also other functions that occur simultaneously results in better performance even though it decreases the amount of data to learn from.
The prediction experiments based on main communicative functions assume that in case of multifunctionality, a main function can clearly be identified. Moreover, it is assumed that task-related functions are more important than feedback functions or other functions. For most cases, these assumptions are justified, but in some cases they are problematic. For instance, in a heated discussion, the turn management function could be considered more important for the dialogue than a simultaneously occurring domain specific function. In other cases, it is impossible to clearly identify a main function as all functions occurring simultaneously are equally important to the dialogue.

In general, \(n\)-grams of a higher order have a higher predictability and therefore a lower perplexity. However, using high order \(n\)-grams is problematic due to sparsity of training data, which clearly is the case with 4-grams, and particularly with 5-grams in combination with complex symbols as for CF prediction.

Considerably more difficult is the prediction of dialogue acts with realised semantic content, as is evidenced in the differences in accuracy and perplexity for all models. Considering that the data set, with about 1,600 functional segments, is rather small, the statistical prediction of logical expressions increases data sparsity to such a degree that from the \(n\)-gram language models, only 2-gram (and 3-grams to some extent) could be trained. The SCFG models can be trained for both CF prediction and dialogue act prediction.

As noted in Section 5.7.2, objective evaluation of dialogue strategies and behaviour is difficult. The evaluation approach used here compares the suggested next dialogue act with the next dialogue act as observed. This is done for each dialogue act in the test set. This evaluation approach has the advantage that the evaluation metric can easily be understood and computed. The approach, however, is also very strict: in a given dialogue context, continuations with various types of dialogue acts may all be equally appropriate. To also take other possible contributions into account, a rich dataset is required in which interlocutors act differently in similar dialogue context with a similar established common ground. Moreover, such a dataset should contain for each of these cases with similar dialogue context a representative set of samples.

Future work in this direction can go in several directions. First, the grammar induction approach shows potential of learning dialogue game-like structures unsupervised. The performance on this task could be tested and measured by applying the algorithm on corpus data that has been annotated with dialogue games, such as the Map Task corpus (see Section 3.3). Second, the models could also be extended to incorporate more information than dialogue acts alone. This could make comparisons with the performance obtained with reinforcement learning or with Bayesian networks interesting. Third, it would be interesting to learn and apply the same models on other kinds of conversation, such as dialogue with more than two interlocutors. Fourth, datasets could be drawn from a large corpus that covers dialogues on a small but complex domain. This makes it possible to evaluate according to the possible continuations as found in the data for situations with similar dialogue context, rather than to evaluate according to a single possible continuation. Last, the rather unexplored parameter space of the DGI algorithm might be worth exploring in optimising the system’s performance.
5.9 Summary

In this chapter, an approach to the prediction of communicative functions and dialogue acts has been presented that makes use of grammatical inference to automatically extract structure from corpus data.

An algorithm, based on alignment-based learning, has been presented and tested against a baseline and several n-gram language models. From the results it can be concluded that the algorithm clearly outperforms the n-gram models: on the task of predicting the communicative functions, the best performing n-gram model achieved 66.4% accuracy; the best SCFG model achieved 78.9% accuracy.

To be able to also include dialogue acts with realised semantic content, semantic content was computed by employing a system that first parses utterances syntactically, using a dependency parser and a chunk parser, and subsequently maps from dependency structure to a semantic content. The resulting parses were manually corrected and used in the prediction experiments.

Predicting the semantic content in combination with the communicative functions is difficult, as evidenced by moderate scores. Obtaining lower degree n-gram language models is feasible, whereas higher degree models are not trainable. Prediction works better with the SCFG models, but does not result in convincing scores. As the corpus is small, it is expected that with scaling up the available training data, scores will improve for both tasks.
This chapter is organised as follows: Section 6.1 presents an overview of the work discussed in the thesis. The implications of the findings as a result of the dialogue act recognition and prediction studies are considered in Section 6.2. The limitations of the studies are discussed in Section 6.3, and Section 6.4 describes the issues that emerge and may be addressed in future study.

6.1 Overview

Two essential activities for the participation of an agent in dialogue are the recognition of meaning in the communicative behaviour of other interlocutors and the decision of what and how to contribute to the ongoing dialogue. With the assumption that dialogue acts encode the essence of what is communicated, the former activity is conceived as the recognition of dialogue acts. The latter activity involves the decision of which dialogue acts to produce next.

From the systems of dialogue acts that are available, a focus has been placed on DIT++. The communicative functions in this taxonomy have been evaluated for applicability. In this context it is argued that by taking partial annotator (dis)agreement into account, a more precise indication of applicability of the tags is possible in comparison with using only a binary agreement value. From evaluating inter-annotator agreement with both naive and expert annotators, it can be concluded that the important dimensions in the DIT++ taxonomy can be annotated ‘reliably’ only with expert annotators. Using naive annotators provides insights on the tagset and its use that are complementary to those obtained by using expert annotators.

To automatically recognise dialogue acts and to decide which dialogue acts to produce next, machine learning techniques have been used. From the point of view of a machine learner, the process of deciding what to do next can be viewed as the prediction of appropriate dialogue acts based on what has happened in the preceding discourse.
In identifying dialogue acts in communicative behaviour, it should be considered that functions can occur simultaneously and may partially overlap. Ignoring this aspect and describing a combination of overlapping functions by a single span is less accurate and leads to inferior machine learning performance.

### 6.1.1 Evaluation of dialogue act tagsets

In many studies, applicability of a dialogue act taxonomy or tagset is measured by calculating inter-annotator agreement. Agreement can be calculated in various ways, but usually a combination of percentage agreement and Cohen’s kappa gives a fair indication of how difficult it is for annotators to apply the concepts in the tagsets.

Usually, the acts in the tagset are not completely unrelated, and by using these metrics of evaluation, partial (dis)agreement is not considered. To express partial agreement more precisely, a derived metric called *taxonomically weighted kappa* is proposed that is ‘weighted’ in that it considers semantic and pragmatic distance between communicative functions, and that is taxonomically motivated in that it takes advantage of the structure of the taxonomy (hierarchical relations between communicative functions).

For an insightful analysis of the applicability of a tagset it is commendable to evaluate with both expert and naive annotators, as they provide feedback on different aspects of the applicability. For DIT++, the scores of inter-annotator agreement with naive annotators are moderate (< 0.8). For expert annotators, scores are considerably better. Differences in both inter-annotator agreement and tagging accuracy between naive and expert annotators against the gold standard are considerable, and the annotations of both groups provide complementary insights in reliability concerning the clarity and accessibility of the tagset, and fundamental conceptual issues. In comparing both annotator groups, it turns out to be essential for multidimensional dialogue act taxonomies to distinguish agreement on whether to annotate in a certain dimension from agreement on the assignment of a communicative function *within* a dimension. However, calculation of tagging accuracy presupposes that on expert level a ground truth can be established. Establishing a ground truth requires the concepts in the scheme to be sufficiently objective and well-defined.

By collapsing the hierarchies in DIT++ to general dialogue acts corresponding to LIRICS dialogue acts, the scores on inter-annotator agreement and accuracy improve for both annotator groups. The improvement for naive annotators is greater than that for expert annotators, as for the former group it is easier to annotate general concepts. Nevertheless, the scores for naive annotators on the LIRICS dialogue acts remain too low for reliable coding.

### 6.1.2 Dialogue act recognition

It is common practice to assign dialogue acts to a single segmentation, but human communication can be described more accurately by using per-dimension segmentation. At the same time, inter-annotator agreement for per-dimension segmentation and
classification does not differ significantly from that of weighted averaged expert percentage agreement reported in the conventional setting. There is a tradeoff, however, in the time needed to segment in multiple dimensions: segmenting and annotating in multiple dimensions costs between 1.5 and two times as much time.

When using machine learning algorithms to automatically classify the communicative functions in the DIT++ taxonomy on the information seeking dialogues, $F_1$ scores ranging from 62.6 to 96.6 are obtained. For the most important dimensions, Task, Auto-feedback, and Allo-feedback, scores of 70.5, 85.1, and 96.6 are obtained, respectively.

To also learn segmentation, a token-based approach was used in which the position of each word (or other speech production) uttered relative to the segmentation and the dialogue act it expressed is labelled. The task of segmentation and subsequent classification turns out to be considerably more difficult, with $F_1$ scores ranging from 29.2 to 57.8. For the most important dimensions (Task, Auto-feedback, and Allo-feedback) scores of 43.4, 46.3, and 50.7 are obtained, respectively.

When using the same algorithms on the same data, and comparing the evaluation scores on segmentation of dialogue act recognition by subsequently segmenting and classifying with that of a joint approach, it has been shown that the latter approach results in lower segmentation error rates.

To see how machine learners compare to naive and expert human annotators in tagging dialogue with DIT++ concepts, three machine learning classifiers were trained on the same dialogue material that was annotated manually by naive and expert annotators. It was shown that, while the machine learners do not perform as well as expert annotators, they score competitively with naive annotators.

### 6.1.3 Dialogue act prediction

In predicting the next appropriate dialogue acts, dialogues are modelled as sequences of sets of dialogue acts, and a method of statistical act prediction that is based on grammar induction is presented. Act prediction has been explored both for dialogue acts without realised semantic content (consisting only of a communicative function) and for dialogue acts with realised semantic content.

The new algorithm that is proposed clearly outperforms $n$-gram models: on the task of predicting communicative functions of DIT++, the best performing $n$-gram model achieves 66.4% accuracy; the best SCFG model achieves 78.9% accuracy.

To be able to include also dialogue acts that have realised semantic content, semantic content was computed by employing a semantic parser that first parses utterances syntactically, using a HPSG-based dependency parser and a memory-based shallow parser, and subsequently maps dependency structures to semantic content representations. The resulting representations were manually corrected after which the accuracy of the system on full semantic interpretation was assessed, which turned out to be 88.1 percent on an evaluation set of 160 task-related utterances. More importantly,
the manually corrected representations could subsequently be used in the dialogue act prediction experiments.

Predicting also semantic content in addition to communicative functions is feasible with lower degree \( n \)-gram language models, but works better with SCFG models. The scores in this task are not convincing because of increasing data sparsity. Simplifying the semantic representations by reducing them to action predicates only, results in a notable improvement in prediction.

6.2 Implications

6.2.1 Dialogue act scheme evaluation

The proposal to segment dialogue acts that may occur simultaneously in separate segmentation layers may seem an obvious step to make, but in practice, per-dimension (or per-layer segmentation) has never systematically been used. This can be partly contributed to the minor importance that multifunctionality has in most dialogue act tagsets. Moreover, the segmentation of functions that occur simultaneously can frequently be described with a single segmentation, making the advantage of multiple segmentations less apparent. By using the segmentation practise also for other tagsets that include multifunctionality it is expected that the more accurate indication also results in better machine learning performance in tasks such as dialogue act recognition. In the evaluation of agreement in semantic and pragmatic schemes such as dialogue act taxonomies, taking into account partial agreement has the advantage that the resulting scores are more refined. This refinement allows a better assessment of the effect of changes that are introduced in improving dialogue act tagsets. Moreover, the taxonomic properties of tagsets can be exploited systematically in deciding how much related dialogue acts have in common. The basic assumption in this step is that the dialogue acts in the tagset are structured according to semantic-pragmatic considerations.

6.2.2 Dialogue act recognition

The dialogue act recognition scores that are presented in this work are the first results with DIT\(^{++}\) dialogue acts. With this consideration in mind, the results provide means of comparison for classification and recognition studies that use DIT\(^{++}\) dialogue acts on other kinds of dialogue data.

The classification of dialogue acts based on presegmented utterances assumes that a divide-and-conquer approach to dialogue act recognition is preferable, and conveniently ignores the necessity of locating the dialogue acts, an activity that is additionally required for a fully automated approach of act recognition and which has an influence on recognition performance, as segmentation errors are propagated to the classification task. The results presented in this thesis show that segmentation as by-product of a unitary approach to dialogue act recognition is more accurate than segmentation
as a final objective. This suggests that a divide-and-conquer approach to dialogue act recognition may not be an optimal solution.

### 6.2.3 Dialogue act prediction

The improvement that the SCFG-based models of dialogue act prediction achieve over \( n \)-gram models allows to obtain a better user simulation in the reinforcement learning of dialogue strategies, which is currently often obtained by training \( n \)-gram language models (see e.g. [Kallirroi et al., 2006]). The improvement can also be considered as another step towards a model-based alternative to reinforcement learning.

### 6.3 Limitations

In the studies that are discussed in this thesis, only the verbal part of dialogue has been addressed. However, in most circumstances people do not only communicate by using speech, but also move their eyes, make gestures, change the way they look (facial expressions), and make posture shifts. Many of these non-verbal behaviours are produced in coordination with spoken utterances and contribute to the meaning of utterances. Sometimes these behaviours do not even require verbal behaviour to communicate. A typical example of multimodal communication is an indicative gesture accompanying a deictic expression (e.g., pointing somewhere while saying ‘there’), while common examples of feedback are head nods, eyebrow movements, shrugs, and so on. The studies in this thesis do not address non-verbal aspects. For a complete account of communicative interaction, all modalities should be considered.

#### 6.3.1 Dialogue act scheme evaluation

The impact of using multiple segmentations seems rather small. The consideration that manual segmentation in more than a single tier is considerably more labour intensive raises the question whether the additional work is worth the effort. On the other hand, the multidimensional segmentation results in a more accurate indication of the associated spans of communicative behaviour.

The use of partial disagreement in e.g. \( \text{DIT}^{++} \) is not without controversy. In particular, the distance function which is used for expressing semantic-pragmatic distance between dialogue acts requires certain parameters to be set. Possible values of these parameters have been suggested, based on intuitions on how much dialogue acts in different levels of the same hierarchy are related. This makes the choice of the assigned values somewhat arbitrary. This disadvantage is less of an issue when the proposed taxonomically weighted kappa metric is used for intra-taxonomic evaluation only.

In the chapter on annotation, the inter-annotator agreement calculations for the several dimensions in \( \text{DIT}^{++} \) are based on a diverse and limited set of dialogues involving
two interlocutors. This analysis does not involve dialogues in which more than two
interlocutors play a role.

6.3.2 Dialogue act recognition

The dialogue act recognition scores that are presented in this work are limited in the
sense that the dialogue data addresses only task-oriented information seeking dialogue
in a specific domain. As a consequence, it is not certain that similar scores would also
be obtained for other kinds of dialogue.

Some of the dimensions in the DIT$^+$ tagset are rarely addressed in the kind of dia-
logue that has been used. Because of having only few instances of functions addressing
these dimensions available, it was not always possible to achieve solid figures. Another
limitation of the recognition studies presented is that there is no systematic exploration
of various kinds of machine learning algorithms, and no exhaustive evaluation of poss-
able kinds of features. Moreover, no feature selection methods have been employed.

6.3.3 Dialogue act prediction

The improvement of SCFG-based models over $n$-gram models is apparent, but it is
not yet clear how the improved scores relate to the performance of more advanced
models than $n$-gram language models, such as that of Bayesian networks on the same
task. However, approaches that do not allow to generalise over symbol strings by more
complex patterns share most of the disadvantages that also $n$-gram language models
have.

A limitation for $n$-gram language models and also SCFG-based models (in lesser
extent though) is that they describe regularities that are rather local, and as a conse-
quence may possibly lead to interactions that are not deterministic.

The SCFG-based models are based grammatical inference. The technique that has
been used involves alignment-based learning. Even though alignment-based learning
has shown to be succesful in various tasks of natural language processing, it may well
be that better performance can be obtained by considering alternative grammatical in-
ference algorithms, such as ADIOS [Solan et al., 2005].

6.4 Perspectives & future research

6.4.1 Dialogue act scheme evaluation

In the evaluation of the dialogue act taxonomy using taxonomically weighted kappa,
the distance between concepts within dimensions is considered. This means that for
an utterance that features multifunctionality, there is no single figure that expresses
overall agreement, which could be worked out in further research.
Another issue is the further refinement of a measure of partial agreement. In many cases of multifunctionality, a primary function and secondary function(s) can be identified. In such cases it makes sense to attach more value to agreement on the primary function than to agreement on secondary functions.

The rather arbitrarily chosen parameter values in the distance metric could be tuned according to empirical results from perception experiments. Semantic-pragmatic relatedness of dialogue acts can be estimated by offering a group of subjects pairs of example utterances (with dialogue context) that correspond to pairs of communicative functions or dialogue acts, and to have them indicate how differently they are perceived.

The evaluation so far has concerned only dialogues that involve two interlocutors. Extending this to also involve more than two interlocutors would offer a more complete picture. It would also be interesting to increase the size of the dataset to assure that there are also sufficiently many instances for the dimensions that are less frequently addressed, to allow drawing firm conclusions.

### 6.4.2 Dialogue act recognition

In the experiments on dialogue act recognition there are various aspects which call for further study:

- The performance of act recognition, especially with the token-based approach, could significantly benefit from scaling up the size of the training data. This could be achieved rather efficiently by using an active learning framework to semi-automatically annotate new dialogue material;

- In the recognition experiments, three basic types of machine learning algorithms are used to get an indication of average machine performance. In addition to the three particular learners, other machine learning algorithms could additionally be employed. For all learners, default parameters have been used, which makes it interesting to investigate the effect of optimising them;

- The same experiments can be redone using a richer feature set, including new core features and derived features. Moreover, the features could be grouped according to their nature: lexical, syntactic, and prosodic. The contribution of each group in the performance relative to that of the other groups could be assessed;

- When using richer feature sets, also the importance of feature selection increases, as some machine learning algorithms are more sensitive to redundant or noisy features than others. This motivates the exploration of various feature selection methods.

Concerning the comparison between human annotators and machines, two issues emerge for future research. First, qualitative analysis could provide more insight in
differences by looking at what kind of specific errors and disagreements are made by humans and machines. Second, there has been no evaluation of the learning process and the data available to the machine learners was reduced to balance the proportion of seen/unseen data between the human and machine annotators. By monitoring how the performance of human annotators without any knowledge or experience and machine performance varies as a function of the training corpus size, insights can be gained in the difference of learning performance.

6.4.3 Dialogue act prediction

An important issue for future research is the size of the training dataset. For prediction based on speaker and multifunctional communicative functions, and especially for prediction based on full-fledged dialogue acts (including full semantic content), the training dataset was sparse. Scaling up the dataset would give more representative scores, but would also require manual tagging.

On the technical side, there are a lot of implicit parameters to the SCFG-based grammatical induction algorithms. For the experiments reported in the thesis, default parameters have been used, which makes it worth to explore the parameter space and find globally optimal parameter values.

Future work in this area can go in several directions. First, the grammar induction approach shows potential of learning dialogue-game-like structures unsupervised. The performance on this task could be tested and measured by applying the algorithm on corpus data that has been annotated with dialogue games, such as the Map Task corpus. Second, the models could also be extended to incorporate more information than dialogue acts alone. This could be interesting for comparison with the performance obtained with reinforcement learning or with Bayesian networks. Third, it would be interesting to learn and apply the models on dialogue involving more than two interlocutors. Fourth, datasets could be drawn from a large corpus that covers dialogues on a small but complex domain, which makes it possible to evaluate according to the possible continuations as found in the data for situations with similar dialogue context. Finally, the rather unexplored parameter space of the DGI algorithm might be worth exploring for optimising the system’s performance.

6.4.4 From dialogue acts to states

Dialogue acts are defined as context or information state changing operators, and can be considered as intermediate functional units in an agent’s processing of another interlocutor’s contribution. This means that a part of the meaning of utterances depends also on the belief dynamics that are driven by the recognised dialogue acts: Having recognised dialogue acts there are various ways in which current beliefs and goals held by the agent can be effected and result in a new context or information state. In this process, grounding plays an important role.
From both practical and theoretical perspective, the use of dialogue acts (and speech acts) as intermediate steps in full processing of observed dialogue behaviour raises the question if dialogue acts are actually essential. If dialogue processing in agents is conceived as sequences of contexts or information states, it makes sense to map utterances to contexts or information states\(^1\) (Figure 6.1:B) instead of mapping utterances to dialogue acts (Figure 6.1:A).

Context state recognition would require a large dataset with for each utterance the resulting change of context state (or a complete new context state). Similarly, prediction of appropriate dialogue behaviour could be utterance based (Figure 6.1:D) instead of dialogue act based (Figure 6.1:C). In the latter case prediction can be directly based on sensory data.

Dialogue acts, like any other linguistic concept, are abstractions that are most of the time helpful in understanding and processing natural language. This, however, does not mean that these abstractions are necessary to enable a machine to act as interlocutor. It may also be the case that such linguistic abstractions are limiting rather than helpful to machine learners when sufficiently large datasets become available. Another transition to make when large datasets are available is to move towards encoding full information states in symbols. For very limited domains this may be feasible for average-sized corpora, for instance, Kallirroi et al. [2006] train a user simulator by using \(n\)-gram language models with complex symbols, to be used for automatic dialogue strategy learning.

---

\(^1\)What could be called information or context state recognition.
Dialogue act annotation is about indicating the kind of intention that the speaker had; what kind of thing was he trying to achieve? When participating in a dialogue, this is what agents are trying to establish. The first and most important two guidelines follow from this.

1. *First and most important guideline: "Do as the Addressee would do!"

   When assigning annotation tags to a dialogue utterance, put yourself in the position of the participant at whom the utterance was addressed, and imagine that you try to understand what the speaker is trying to do. Why does (s)he say what (s)he says? What are the purposes of the utterance? What assumptions does the speaker express about the addressee? Answering such questions should guide you in deciding which annotation tags to assign, regardless of how exactly the speaker has expressed himself. Use all the information that you could have if you were the actual addressee, and like the addressee, try to interpret the speaker’s communicative behaviour as best as you can.

2. *Second and equally important guideline: "Think functionally, not formally!"

   The linguistic form of an utterance often provides vital clues for choosing an annotation, but such clues may also be misleading; in making your choice of annotation tags you should of course use the linguistic clues to your advantage, but don’t let them fool you - the true question is not what the speaker says but what he means.

   For example, SetQuestions are questions where the speaker wants to know which elements of a certain domain have a certain property. In English, such questions often contain a word beginning with "wh", such as which as in Which books did you read on your holidays? or where in Where do your parents live?. But in other languages this is not the case; moreover, even in English not all sentences of this...
form express a SetQuestion: Why don’t you go ahead is for instance typically a Suggestion rather than a question.

Similarly, PropositionalQuestions are questions where the speaker wants to know whether a certain statement is true or false. Such sentences typically have the form of an interrogative statement, such as Is The Hague the capital of the Netherlands? or Do you like peanut butter? But not all sentences of this form express a PropositionalQuestion; for example, Do you know what time it is? functions most often as in IndirectSetQuestion (What time is it?), and Would you like some coffee? is an Offer; Shall we go? is a Suggestion.

3. **Another important general guideline is: "Be specific!"

Among the communicative functions that you can choose from, there are differences in specificity, corresponding with their relative positions in hierarchical subsystems. For instance, a CheckQuestion is more specific than a PropositionalQuestion, in that it additionally carries the expectation that the answer will be positive. Similarly, a Confirm is more specific than a PropositionalAnswer, in that it carries the additional speaker that the addressee expects the answer to be positive.

In general, try all the time to be as specific as you can. But if you’re in serious doubt about specific functions that you could choose between, then simply use a less specific function tag that subsumes the more specific tags.

4. **On indirect speech acts: "Code indirect speech acts just like direct ones."**

Standard speech act theory regards indirect speech acts, such as indirect questions, as just an indirect form of the same illocutionary acts. By contrast, the DIT++ taxonomy incorporates the idea that indirect dialogue acts signal subtly different packages of beliefs and intentions than direct ones. For example, the direct question What time is it? carries the assumption that the addressee knows what time it is, whereas the indirect question Do you know what time it is? does not carry that assumption (it does at least not express that assumption; in fact it questions it).

5. **On implicit functions: "Do not code implicit communicative functions, that can be deduced from functions that you have already assigned."

Implicit communicative functions occur in particular for positive feedback. For example, someone answering a question may be assumed to (believe to) have understood the question. So any time you annotate an utterance as an answer (of some sort), you might consider annotating it also as providing positive feedback on the interpretation of the question that is answered. Don’t! It would be redundant.

Notice also that the definition of a positive (auto-) feedback act concerning interpretation stipulates that the speaker wants the addressee to know that he (speaker)
has understood the question. A speaker who answers a question does not so much want to tell the addressee that his question was understood – that’s just a side-effect of giving an answer, that no speaker can avoid. Similarly for reacting to an offer, a request, a suggestion, etc.

6. Guidelines for the annotation of feedback functions.

Negative feedback, where the speaker wants to indicate that there was a problem in processing a dialogue utterance, is always explicit and as such mostly easy to annotate.

6.1 Implicit and explicit positive feedback.

Positive feedback is sometimes given explicitly, and very often implicitly. Examples of explicit positive auto-feedback are the following utterances by B, where he repeats part of the question by A:

A What time does the KLM flight from Jakarta on Friday, October 13 arrive?

B The KLM flight from Jakarta on Friday, October 13 has scheduled arrival time 08.50

B The flight from Jakarta on Friday has scheduled arrival time 08.50

B The KLM flight from Jakarta on October 13 has scheduled arrival time 08.50

B The flight from on October 13 has scheduled arrival time 08.50

In such cases, the utterance by B should be annotated as having, besides the general-purpose function SetAnswer in the Activity dimension, also a function in the Auto-Feedback dimension (see below).

By contrast, the short answer: At 08.50 would carry only implicit feedback information, and should therefore, following Guideline 5, not be coded in the Auto-Feedback dimension.

6.2 Levels of feedback.

The DIT++ taxonomy distinguishes 5 levels of feedback:

1 participant A pays attention to participant B’s utterance.

2 A perceives B’s utterance, i.e. A recognises the words and nonverbal elements in B’s contribution.
3 A understands B’s utterance, i.e. A assigns an interpretation to B’s utterance, including what A believes B is trying to achieve with this utterance (what are his goals and associated beliefs about the task/domain and about A).

4 A evaluates B’s utterance, i.e. A decides whether the beliefs about B that characterise his understanding of B’s utterance, can be added to A’s model of the dialogue context, updating his context model without arriving at inconsistencies.

5 A ‘executes’ B’s utterance, i.e. A performs actions which are appropriate for achieving a goal that he had identified and added to his context model. (For instance, executing a request is to perform the requested action; executing an answer is to add the content of the answer to one’s information; executing a question is to look for the information that was asked for.)

There are certain relations between these levels: in order to execute a dialogue act one must have evaluated it positively ("accepted" it); which is only possible if one (believes to) have understood the corresponding utterance; which presupposes that one perceived the utterance in the first place, which, finally, requires paying attention to what is said. So for instance positive auto-feedback about the acceptance of the addressee’s previous utterance implies positive feedback at the "lower" levels of understanding, perception, and attention. For positive feedback functions a higher-level function is more specific than the lower-level functions. (Remember that a function is more specific if it implies other functions.)

For negative feedback the reverse holds: when a speaker signals the impossibility to perceive an utterance, he implies the impossibility to interpret, evaluate and execute it. So negative feedback at a lower level implies negative feedback at higher levels.

Since, following Guideline 3, you should always be as specific as possible, you should observe the following guideline for annotating feedback functions:

Guideline 6: **When assigning a feedback function, choose the most specific level of feedback in the case of positive feedback that you feel to be appropriate, and choose the least specific level in the case of negative feedback.**

While this guideline instructs you to be as specific as possible, sometimes you’ll be in serious doubt. You may for instance find yourself in a situation where you have no clue whether a feedback signal (such as OK) should be interpreted at the level of interpretation or that of evaluation. In such a case you should use the less specific of the two, since the more specific level would mean that you "read" more into this utterance than you can justify.
In practice, it is often difficult to decide the level of feedback that should be chosen. One of the reasons for this is that the same verbal and nonverbal expressions may be used at most of the levels (with a tendency to signal feedback (positively or negatively) with more emphasis as higher levels of processing are involved). It may happen that you encounter a feedback signal and you have no clue at all at which level you should interpret that signal. In this situation the annotation scheme allows you to use the labels ‘Positive’ and ‘Negative’, which leave the level of feedback unspecified.

7. Guidelines for the annotation of Interaction Management functions.

7.1 Turn Management.

General guideline: "Code Turn Management functions only when these are not just implied."

In a spoken dialogue, the participants take turns to speak. (Their nonverbal behaviour is not organised in turns; both participants use facial expressions and gestures more or less all the time.) A turn, that is a stretch of speech by one of the participants, in general consists of smaller parts that have a meaning as a dialogue act; these parts we call "utterances". Turn Management acts are the actions that participants perform in order to manage the allocation of the speaker role. These acts are subdivided into acts for taking the turn (utterance-initial acts) and those for keeping the turn or giving it away (utterance-final acts). Usually only the first utterance in a turn has an utterance-initial function and only the last an utterance-final one. The non-final utterances in a turn do not have an utterance-final function, except when the speaker signals (for example by using a rising utterance-final intonation) that the utterance is not going to be the last one in the turn, that he wants to continue. In that case the utterance has a Turn Keeping function. Except for the first one, the utterances in the turn do not have an utterance-initial function; the speaker does not have to perform a separate act in order to continue; all he has to do is to continue speaking.

When a speaker accepts a turn that the addressee has assigned to him through a Turn Assign act, the utterance should be annotated as having the utterance-initial function Turn Accept only when the speaker performs a separate act for the purpose of accepting the turn, so don’t code this when the turn is accepted implicitly by simply starting to speak.

Similarly, an utterance should be annotated as having the utterance-initial function Turn Take only if the speaker performs a separate act to that effect.
If he just goes ahead and makes a contribution to the dialogue, without first signalling his intention to do so, then the utterance should not be marked with an utterance-initial Turn Management function.

The verbal as well as nonverbal activities that a speaker performs to seize the turn should be marked as Turn Grabbing, but the utterance that follows after he has seized the turn should not be marked as having an utterance-initial Turn Management function.

7.2 Time Management.

When a speaker is buying time, using fillers such as Well,...; Let’s see,..., then the utterance should be annotated as having the Stalling function in the Time Management dimension. There may be several reasons why a speaker wants to have more time; it may be that the speaker has trouble completing his current utterance, or that he is interrupted by some urgent event that requires his attention before he can continue the dialogue. But it may also be that he needs some time to find some information (for instance, for answering a question). So when you encounter a Stalling act, you may well pay attention to the reason why the speaker is stalling. (For instance, Stalling often goes hand in hand with turn acceptance or turn keeping.) However, don’t speculate; only code additional functions for which you have evidence.

7.3 Topic Management.

During a dialogue, the topic is often changed implicitly, simply by talking about a new topic. This happens especially if the new topic is closely related to the previous one, for instance by being a subtopic of the previous topic, or by both being a subtopic of a more general topic. Implicit topic management should not be encoded; it would be redundant. Topic Management functions should be annotated only if the speaker explicitly introduces or closes a topic, or signals his intention to do so.

7.4 Contact Management.

The management of contact in the sense of both partners being ready to send and receive messages to and from each other, is important especially in other than face-to-face situations, such as telephone conversations, video-conferencing, and internet chatting.

Note that in many languages expressions used for establishing contact can often be used for other purposes as well, for example for greeting (Hello!).
When annotating a dialogue where this happens, the utterance in question should be marked as having both a Contact Management function and a Social Obligation Management function.

7.5 *Own Communication Management.*

Own Communication Management (OCM) functions should be coded whenever a speaker signals that he made a speech error and/or wants to edit what he is saying. Since this typically requires some extra effort and time, OCM acts often go hand in hand with acts whose function is to win time, such as hesitations (Ehm...), which have a Stalling function. See also 7.2.

7.6 *Partner Communication Management.*

Partner Communication Management (PCM) functions should be coded whenever a speaker signals that he believes the addressee made a speech error or has difficulty in completing an utterance, for instance being unable to recall a name or to find the right words to express something. The use of dimension-specific PCM functions for this purpose is typically only possible by interrupting the dialogue partner or in immediate response to a partner utterance.

7.7 *Dialogue Structuring.*

These functions should be coded only when the speaker explicitly signals something about his intention to open or close the dialogue, or to continue in a particular way.

Across the board, the following guideline applies to the encoding of Interaction Management functions:

Guideline 7: "**Code only explicit Interaction Management functions.**"

8. *Guidelines for annotating Social Obligation Management (SOM) functions.*

Utterances that serve a ‘social’ purpose such as greetings, thanks, and apologies can often be used for other purposes as well. Greetings like Hello!, for example, can be used also for establishing contact (Contact Management function) and/or for opening a dialogue (a Dialogue Structuring function). Also, an expression of thanks can be used to signal that the speaker wants to soon end the dialogue (Dialogue Structuring function PRE-CLOSING), and can also be used for overall positive feedback. In such cases, the utterances should be coded with the appropriate functions in all these dimensions.
Guideline 8: "When coding an utterance as having a SOM function, look out for additional functions in other dimensions."

Note on segmentation.

The segmentation of a dialogue into utterances may present several difficulties.

First, if you’re working from a transcription of a spoken dialogue, the segmentation in the transcription is not necessarily perfect. You may run into cases where you would prefer the utterance to be segmented as a sequence of parts that each have a functional interpretation. In such a case it is best to assign the various tags that you would prefer to assign to the parts to the utterance as a whole. Or conversely, it may also happen that a turn has been segmented into certain parts, where you would want to annotate the longer utterance formed by these parts together. In such a case it is recommended to annotate all these parts with the same tags.

Second, an utterance may be self-interrupted by a part that has a different communicative function, as in the following example: When, I mean what time, does the train to ehm,... Viareggio leave? Here the utterance as a whole is a SetQuestion; it includes a Self-Correction (I mean what time) and a Stalling utterance (ehm). In such cases, again, it is best to assign the tags for the intervening parts of the utterance to the utterance as a whole.

Third, it may happen that a dialogue act corresponds to (parts of) more than one turn, as in the following example, where the utterances in turns 1 and 3 together form a SetAnswer:: 1. A: There are two flights early in the morning, at 7.45 and at 8.15,.. 2. B: Yes 3. A: and two more in the evening, at 7.15 and at 8.30.

In such a case it is best to give each of these parts the same tag (SetAnswer, in this example).


Language Learning (CoNLL-2005), pages 9–16, Ann Arbor, Michigan, USA. Association for Computational Linguistics (ACL).


Bij het voeren van een gesprek spelen twee belangrijke opeenvolgende processen een rol. Het eerste is het herkennen van wat een gesprekspartner precies bedoelt met wat hij zegt; het tweede behelst het bedenken en uitvoeren van een volgende bijdrage aan het gesprek.

In het algemeen kan worden gesteld dat de spreker door middel van een taalkundige uiting een bepaalde handeling uitvoert, een zogenaamde ‘dialooghandeling’. Het is vervolgens aan de geadresseerde om dit te herkennen. Wordt er om informatie gevraagd? Of wordt het juist gegeven? En waar gaat het dan precies over? Heeft de geadresseerde de dialooghandelingen van de spreker eenmaal met succes achterhaald, dan kan de geadresseerde besluiten een passende volgende bijdrage te leveren aan het gesprek, bijvoorbeeld door op een gestelde vraag antwoord te geven.

Het praten met elkaar is een activiteit waarbij mensen op een natuurlijke manier informatie kunnen overdragen, hetgeen het interessant maakt om machines te leren, of te laten leren, deel te nemen in een gesprek zoals mensens dat doen. De hoofdvragen die in dit onderzoek dan ook centraal staan zijn: “Hoe kan een machine achterhalen wat gespreksdeelnemers bedoelen?” en “Hoe kan een machine bepalen wat een juiste bijdrage is om een gesprek voor te zetten?”.

Het automatisch achterhalen van wat gespreksdeelnemers bedoelen met wat ze zeggen en wat een mogelijke bijdrage aan de dialoog zou kunnen zijn komt neer op respectievelijk het herkennen en voorspellen van dialooghandelingen. Met behulp van een verzameling van opgenomen dialogen in het Engels en Nederlands, voorzien van de bijbehorende dialooghandelingen zijn met behulp van machineleeralgoritmen modellen verkregen voor het automatisch herkennen van dialooghandelingen. Het schema dat gebruikt is om de dialogen te annoteren richt zich bewust op dialooghandelingen die generiek zijn. Zo zijn het doen van een vraag of het doen van een mededeling handelingen die in alle domeinen kunnen voorkomen; de inhoud van beide hangt echter af van het onderwerp waarover op dat moment gesproken wordt. Dit schema wordt in het proefschrift geëvalueerd met behulp van een nieuwe metriek, en door te kijken naar de annotatieverschillen van annotatoren met verschillende mate van training wordt een
indicatie gegeven van hoe goed het herkennen van dialooghandelingen mensen afgaat. Hierbij blijkt dat analyse van annotaties door kortgetrainde annotatoren en analyse van annotaties door langgetrainde annotatoren complementaire inzichten oplevert over het annotatiesysteem.

In veel van de eerdere studies die zich op dialooghandelingsherkenning richten werd gekeken naar welk type dialooghandeling kan worden toegekend aan een bepaalde uiting. Hierbij werd meestal nauwelijks aandacht besteed aan het noodzakelijke voorwerk: om een dialooghandeling toe te kennen aan communicatief gedrag is het allereerst nodig te bepalen waar de dialooghandeling begint, of er wellicht onderbrekingen zijn, en waar de dialooghandeling eindigt. Om zowel de locatie als de identiteit van een gerealiseerde dialooghandeling te bepalen is een aanpak verkend waarbij voor elk ‘token’ (woord of ander relevant geluid) dat een spreker geproduceert gekeken wordt of het deel uitmaakt van een specifieke dialooghandeling. Uit de experimenten blijkt dat wanneer er op deze manier tegelijkertijd wordt gekeken naar waar de uiting begint en eindigt en om welke dialooghandeling het gaat er betere resultaten worden behaald als dit afzonderlijk na elkaar gebeurt.

Het automatisch bepalen van hetgeen een gespreksdeelnemer het beste zou kunnen bijdragen aan een gesprek is complexe bezigheid. Met behulp van veelgebruikte statistische modellen als n-gram taalmodellen kan al met enige mate van succes gepoogd worden om te voorspellen wat een menselijke gespreksdeelnemer in een vergelijkbare situatie zou doen. Deze modellen geven de gelegenheid een dialooghandeling te voorspellen op basis van de n voorafgaande dialooghandelingen. Dit soort modellen worden echter beperkt door het lokale karakter: abstraheren naar algemene gedragpatronen is met n-gram taalmodellen lastig. Als een mogelijke oplossing voor deze beperking wordt in dit proefschrift een methode voorgesteld op basis van stochastische context-vrije grammatica’s die automatisch zijn verkregen. Uit experimenten blijkt dat het voorspellen van dialooghandelingen met deze modellen beter gaat dan met n-gram taalmodellen.
1. Pashiera Barkhuysen  
*Audiovisual prosody in interaction*  
Promotores: M.G.J. Swerts, E.J. Krahmer  
Tilburg, 3 October 2008

2. Ben Torben-Nielsen  
*Dendritic morphology: function shapes morphology*  
Promotores: H.J. van den Herik, E.O. Postma  
Co-promotor: K.P. Tuyls  
Tilburg, 3 December 2008

3. Hans Stol  
*A framework for evidence-based policy making*  
Promotor: H.J. van den Herik  
Tilburg, 21 January 2009

4. Jeroen Geertzen  
*Dialogue act recognition and prediction: Explorations in computational dialogue modelling*  
Promotor: H.C. Bunt  
Co-promotor: Dr. J.M.B. Terken  
Tilburg, 11 February 2009