How to Do Corpus Pragmatics on Pragmatically Annotated Data

Martin Weisser

Studies in Corpus Linguistics

84

JOHN BENJAMINS PUBLISHING COMPANY

How to Do Corpus Pragmatics on Pragmatically Annotated Data

Studies in Corpus Linguistics (SCL)

ISSN 1388-0373

SCL focuses on the use of corpora throughout language study, the development of a quantitative approach to linguistics, the design and use of new tools for processing language texts, and the theoretical implications of a data-rich discipline.

For an overview of all books published in this series, please see
http://benjamins.com/catalog/books/scl

Volume 84

How to Do Corpus Pragmatics on Pragmatically Annotated Data:
Speech acts and beyond
by Martin Weisser

How to Do Corpus Pragmatics on Pragmatically Annotated Data

Speech acts and beyond

Martin Weisser
Guangdong University of Foreign Studies

John Benjamins Publishing Company
Amsterdam / Philadelphia

The paper used in this publication meets the minimum requirements of the American National Standard for Information Sciences – Permanence of Paper for Printed Library Materials, ANSI Z39.48-1984.

Cover design: Françoise Berserik
Cover illustration from original painting *Random Order*
by Lorenzo Pezzatini, Florence, 1996.

DOI 10.1075/scl.84

Cataloging-in-Publication Data available from Library of Congress:
LCCN 2017056561 (PRINT) / 2017061549 (E-BOOK)

ISBN 978 90 272 0047 1 (HB)
ISBN 978 90 272 6429 9 (E-BOOK)

John Benjamins Publishing Company · https://benjamins.com

Table of contents

List of tables

List of figures

Abbreviations

ANC	American National Corpus
ACL	Association for Computational Linguistics
ALLC	Association for Literary and Linguistic Computing
ACH	Association for Computing and the Humanities
ASCII	American Standard Code for Information Interchange
BNC	British National Corpus
CA	Conversational Analysis
CANCODE	Cambridge and Nottingham Corpus of Discourse in English
COMPGR	Comprehensive Grammar of the English Language
CAMGR	The Cambridge Grammar of the English Language
CLAWS	Constituent Likelihood Automatic Word-tagging System
CSS	Cascading Style Sheets
DA	Discourse Analysis
DAMSL	Dialogue Act Markup in Several Layers
DART	Dialogue Annotation and Research Tool
DRI	Discourse Resource Initiative
DSSSL	Document Style Semantics and Specification Language
DTD	Document Type Definition
FLOB	Freiburg Lancaster-Oslo/Bergen Corpus
FROWN	Freiburg Brown Corpus
GENAM	General American
HTML	Hypertext Markup Language
ICE	International Corpus of English
IDE	Integrated Development Environment
LINDSEI	Louvain International Database of Spoken English Interlanguage
LONGGR	Longman Grammar of Spoken and Written English
LLC	London-Lund Corpus of Spoken English
LOB	Lancaster-Oslo/Bergen Corpus
MATE	Multilevel Annotation, Tools Engineering
MICASE	Michigan Corpus of Academic Spoken English
NLP	Natural Language Processing
POS	Part of Speech
RP	Received Pronunciation
SGML	Standard Generalized Markup Language

TART	Text Annotation and Research Tool
TEI	Text Encoding Initiative
XML	eXtensible Markup Language
XSL	eXtensible Style Sheet Language
XSLT	XSL Transformations
XSL-FO	XML Formatting Objects

CHAPTER 1

Introduction

Corpus- and computer-based methods of analysis have 'revolutionised' much of the research in linguistics or natural language processing over the last few decades. Major advances have been made in lexicography (cf. Ooi 1998 or Atkins & Rundell 2008), morphology (cf. Beesley & Karttunen 2003, or Roark & Sproat 2007), (morpho-) syntax (Roark & Sproat 2007), and genre-based text-linguistics (cf. Biber et al. 1998), to name but the most important areas. These advances were in many cases linked to, or dependent upon, advances in creating and providing suitably annotated resources in the form of corpora. However, apart from the efforts made on the SPAAC project (cf. Leech & Weisser 2003), the creation of the SPICE-Ireland corpus (Kallen & Kirk 2012), or my own research into improving the automated annotation of pragmatics-related phenomena (Weisser 2010), to date very few linguistically motivated efforts have been made to construct annotated corpora of spoken language that reflect the different facets of language involved in creating meaning on the level of human interaction – in other words, on the level of pragmatics.

One aim of this book is to rectify this shortcoming and to demonstrate how it is possible to create corpora that can be annotated largely automatically on the levels of syntax, (surface) polarity (positive or negative mood of the unit), semantics (in the sense of representing major *topic* features of a textual unit), semantico-pragmatics (in the form of capturing interactional signals), and, finally, pragmatics (in the shape of *speech acts*). In contrast to current trends in computer-based, and here especially computational linguistics, this is done relying purely on linguistic surface information in conjunction with appropriate inferencing strategies, rather than employing probabilistic methods. Thus, e.g. the 'utterance' *i'd like to go from Preston to London* can be 'recorded' inside a corpus as:

a. being of *declarative* sentence type,
b. having positive surface polarity,
c. containing topic information about some locations and movements between them,
d. signalling an intent or preference on the semantico-pragmatic level, as well as
e. pragmatically, in its particular context of occurrence inside the dialogue it was taken from, representing a directive that also informs the interlocutor about the speaker's intentions.

The exact format in which such information can best be stored in order to facilitate usability and exchangeability will be presented and discussed in the relevant sections below. The main emphasis here will be on the automatic determination of speech acts.

The advantages of being able to produce pragmatically annotated corpora efficiently and thereby creating resources for many areas of linguistic research concerned with human (or human-computer) interaction should be self-evident, as this could not only greatly facilitate research about the interplay of the different linguistic levels, and in so doing also increase our understanding of how communication works, but also make it possible to use these resources in more applied areas, such as language teaching, textbook creation, or the training of other 'language professionals', for example as interpreters or even call centre personnel. Some of the ways in which this can be achieved, for instance through the pragmatic profiling of speakers or speaker groups/populations, will be discussed in the more research-oriented chapters of this book, Chapters 4 and 7, where I shall attempt to demonstrate the different forms of *appliability* of the approach.

At the same time, creating such a(n) annotation/corpus-creation methodology obviously does not constitute a purely mechanical process because, as, apart from devising appropriate algorithms, data structures and storage mechanisms for processing language on the computer, such an endeavour already involves active research into the interplay – or *interfaces*, as some researchers prefer to refer to them – of the different linguistics levels mentioned above. Another aim of this book, therefore, is to provide a substantial contribution to the practical and theoretical underpinnings of how to analyse, explain, and categorise the individual elements of language that contribute towards the generation of pragmatic meaning. Here, I will especially focus on developing the theory of speech acts – originally established by Austin (1962) and Searle (1969) – further, and present a generically applicable speech-act taxonomy that goes far beyond the limited categories proposed by Searle (ibid.) that are generally used in pragmatics research. Before going into how this is achieved in any detail, though, I will first contextualise the research discussed here by providing a brief overview of existing studies into general and computer-based pragmatics, and the analysis of spoken discourse.

1.1 Previous approaches to pragmatics and discourse

With regard to contemporary pragmatics, one can see a rough division into two different factions, or what Huang (2007: 4; cf. also Horn & Ward 2004: x) refers to as the "Anglo-American" and the "European Continental" schools. Amongst these, the former subscribes to the "**component** view" (ibid.), a view that sees pragmatics as a separate level of linguistics, such as those of phonetics/phonology, syntax and semantics, while the latter adopts the "**perspective** view" (ibid.) – following the original ideas developed by Morris in 1938 (cf. Verschueren 1999: 2–10) –, which perceives pragmatics as a function of language that influences the other levels and thus incorporates a larger situational context, including sociolinguistic factors. These different 'attitudes' towards the nature of pragmatics also to some extent manifest themselves in the two different approaches to the subject that Leech (1983: 10–11) refers to as "PRAGMA-LINGUISTICS" and "SOCIO-PRAGMATICS", respectively.

Along with pragma-linguistics and the component view generally comes an emphasis on issues in *micro-pragmatics* (cf. Mey 1993: 182), where the main topics of investigation are generally considered to be *implicature, presupposition, speech acts, reference, deixis*, as well as *definiteness and indefiniteness*. This is evident in the chapters under the heading "The Domain of Pragmatics" in Horn and Ward's (2004/2006) *Handbook of Pragmatics*, which may be seen as one of the standard references for work in pragmatics that follows the component view. These topics are also still predominantly investigated on the level of the 'sentence' – a concept which will need to be scrutinised further in Section 4.1 of this book –, rather than involving any larger context. This practice is still being adhered to, despite the fact that at least some of the emphasis in this school is now also shifting towards an analysis of contextually embedded examples, as evidenced by a number of later chapters in Horn and Ward (2006). One further feature that goes hand in hand with the 'sentence-level' analysis is that 'research' by proponents of this view still frequently involves the use of constructed ('armchair') examples (cf. Jucker 2009: 1615) and a strong adherence to philosophically-oriented, and logic-based interpretations of how meaning is created, as well as employing more formal linguistic methods. The latter also tend to stress the affinity of pragmatics to (formal) semantics, something which supporters of the component view are still struggling to resolve, as evidenced through the ongoing debate about the *Semantics – Pragmatics* distinction (cf. e.g. Szabó 2005).

In contrast, advocates of socio-pragmatics and the perspective view often concentrate on research that is more process- and context-oriented, and which focuses on *macro-pragmatics*, i.e. the investigation of larger contexts and 'meaning in use', something that also frequently involves the cultural or even extra-linguistic information that contributes to communication as a social act. In line with their more sociological orientation, socio-pragmatists seem, to some extent, also be more inclined towards employing less formal data analysis methods – reminiscent of the approaches in *conversational analysis* (CA) –, but that use a substantial amount of empirical data in a *bottom-up* strategy to draw conclusions from. There may also still be more of an emphasis on issues of sequencing of interaction, such as *turn-taking* (cf. Sacks, Schegloff & Jefferson 1974), as a means of handling or managing the social act(ion) constituted by verbal (conversational) communication, although this is beginning to play a larger role in the component view these days, too.

Supporters of the component view, on the other hand, still seem to work more along the lines of analysis methods developed in *discourse analysis* (DA), although they do not explicitly subscribe to this. DA itself developed out of the (British) Firthian linguistic tradition and is therefore essentially functional or systemic in its approach to dialogues. Its main attention was originally focussed on the identification of units and structure of interaction, as a kind of extension of the hierarchy of units employed in general linguistic analysis and description, ranging from the morpheme, word, clause, to the sentence, the micro-level referred to above. DA was initially limited to the relatively narrow scope of analysing classroom interaction (cf. Sinclair & Coulthard 1975; Coulthard 1977), in order to attempt to identify regular patterns therein. More recent approaches to DA (cf. Coulthard 1992 or Brown & Yule 1983), however, have realised that the specific conditions of classroom interaction have led to over-generalisations and incorrect labelling – and hence potentially equally incorrect interpretation – of interactional patterns, and have therefore actively sought to overcome these earlier problems. The DA approach is also more *top-down*, in the sense that categorisations are attempted earlier, and potentially also based on some degree of intuition, something that is also still clearly reflected in 'component-view' pragmatics. This does not mean, however, that DA is not an empirical approach just like CA, since both approaches are well-grounded empirically, only with a slightly different slant and emphasis on various levels of detail. I will return to these levels of detail later in the discussion of issues regarding transcription conventions or dialogue sequencing and structure.

Within the component-view school, one can also distinguish between two further subgroups, proponents of the *neo-Gricean* and the relevance-theoretical view. While the neo-Griceans have attempted to 'refine' (cf. Huang 2007: 36–54) the original approach developed by H. P. Grice that includes the *Cooperative Principle* (CP) and its associated *categories* and *maxims* (cf. Grice 1989: 26), which assumes that all communication is essentially based on the co-operative behaviour of the communication partners, supporters of *relevance theory* work on the assumption that there is only one overarching principle in communication, which is that of *relevance*:

> We share Grice's intuition that utterances raise expectations of relevance, but question several other aspects of his account, including the need for a Cooperative Principle and maxims, the focus on pragmatic contributions to implicit (as opposed to explicit) content, the role of maxim violation in utterance

> interpretation, and the treatment of figurative utterances. The central claim of relevance theory is that the expectations of relevance raised by an utterance are precise and predictable enough to guide the hearer toward the speaker's meaning. (Wilson & Sperber 2006: 608)

The concepts employed in relevance theory, though, are not 'measurable' (ibid: 610) and can hence also not be easily applied to computer-based analysis, so that they will be largely ignored in the following exposition.

As indicated above, the main emphasis of this book is on working with speech acts, so the other main issues in micro-pragmatics – *implicature, presupposition, reference, deixis*, and *definiteness and indefiniteness* – will only be referred to if they play a direct role in the identification of speech acts or are in fact constitutive thereof.

1.2 Speech acts

Research into speech acts essentially started with the 'ordinary language philosopher' Austin and his famous collection of William James lectures that was published under the title of *How to Do Things with Words* (Austin 1962). Here, Austin contradicted the idea that, unlike it had commonly been assumed by most philosophers previously since the days of Aristotle, *sentences* are only used to express *propositions*, that is facts that are either true or false.

> It was too long the assumption of philosophers that the business of a 'statement' can only be to 'describe' some state of affairs, or to 'state some fact', which it must do either truly or falsely. (Austin 1962: 1)

The concept of truth-conditionality, though, rather surprisingly, still seems to be present in many current approaches to the logic-based semantic description of language followed by at least some of the proponents of the component view. Starting from his theory of *performative verbs* (ibid.: 14ff), Austin claimed that, instead, sentences are often used to perform a 'verbal act(ion)' and distinguished between three different functions – or acts – of an utterance:

1. locution: 'what is said'
2. illocution: 'what is intended'
3. perlocution: 'what is evoked in the recipient' (cf. ibid.: 98ff.)

The explanations given in single quotation marks above represent my brief summaries of Austin's expositions. Apart from these functions, he also claims that there are a number of *felicity conditions* ("Conditions for Happy Performatives") that are necessary for such actions to become successful, amongst them that the hearer understand and accept them, such as in the act of promising (ibid.: 22f). These are often dependent upon established conventions or laws (ibid.: 14f).

Searle (1969), in his *Speech Acts: an Essay in the Philosophy of Language*, takes Austin's ideas further and defines the speech act not only as an expression of illocutionary force, but even ascribes it the most central role in communication.

> The unit of linguistic communication is not, as has generally been supposed, the symbol, the word or sentence, but rather the production or issuance of the symbol or word or sentence in the performance of the speech act. [...] More precisely, the production or issuance of a sentence token under certain conditions is a speech act, and speech acts ([...]) are the basic and minimal unit of linguistic communication. (Searle 1969: 16)

In order to distinguish between the 'locutionary elements' of a sentence, he differentiates between a "propositional" and an "illocutionary force indicator" (ibid.: 30) and introduces the notion of what has later come to simply be referred to by the acronym *IFIDs* ("illocutionary force indicating devices"). Amongst these, he lists "word order, stress, intonation contour, punctuation, the mood of the verb, and the so-called performative verbs" (ibid.). Some, but not all of these, though complemented by a few others, will later turn out to be highly relevant to our analysis methodology.

Summarising the most important points made by Austin and Searle, it becomes clear that linguistic form, (lexico-)grammar, and context or established conventions taken together, determine meaning. Since syntactic features are listed among the IFIDs, it ought to be clear that it is both the semantics and the syntax that play a role in determining the meaning of a speech act. And because analysing basic syntactic patterns is often much easier than determining the exact meaning – the (deep) semantics –, it seems only natural that one might want to begin an analysis of speech acts by testing to see how the high-level syntax may constrain the options for them, thereby also signalling high-level types of communication.

Hence it is relatively easy, although not always foolproof, to distinguish syntactically between whether someone is asking a question, making a statement or simply indicating (dis)approval/agreement, backchanneling, etc., in order to be able to limit the set of initial choices for identifying a speech-act. Once the selection has been narrowed down, one can then look for and identify further IFIDs at the semantico-pragmatic level that may reflect additional linguistic or interactional conventions, 'synthesise' the existing information, and, in a final step – as and when required – carry out some more inferencing in order to try and determine the exact primary force of the illocution as far as possible. To show how this can be done will be one of the most important aims of this book, along with demonstrating that far more in communication than has commonly been assumed to belong to the realm of *conventional implicature* (cf. Grice 1989: 25–26) – as opposed to *conversational implicature* – is indeed conventional, and can thus be investigated using methodologies similar to those traditionally employed in corpus linguistics, albeit with some major extensions.

Closely linked to the notion of conventionality is that of indirectness in meaning. For instance O'Keeffe et al. (2011), apparently subscribing to something I would like to call the 'general myth of indirectness', state that

> the utterance *I've got a headache* carries a variety of meanings according to when it is used, who uses it, who the person is talking to, where the conversation takes place, and so forth:
> – If a patient said it to a doctor during a medical examination, it could mean: *I need a prescription*.
> – If a mother said it to her teenage son, it could mean: *Turn down the music*.
> – If two friends were talking, it could mean: *I was partying last night*.
> – If it were used as a response to an invitation from one friend to another, such as *Do you fancy going for a walk?*, it could simply mean: *No*.
>
> Therefore, depending on the context it occurs in, the utterance *I've got a headache* can function as an appeal, an imperative, a complaint or a refusal, and so on.
> (O'Keeffe et al. 2011: 1–2)

Such claims to the multi-functionality and indirectness of 'utterances' – a rather vague term we shall have to evaluate in more detail later – are very common in the traditional pragmatics literature. However, it seems to me that we seriously need to question the true extent of these phenomena. Certainly, no-one, including myself, would claim that it is not possible to create meaning in highly indirect ways, and that 'locutionary facts' may assume a special meaning in context that is not expressed directly through them. Nevertheless, if we look at the above examples more closely, we can assume that the functions associated with them by O'Keeffe et al. (2011) probably do not reside in the locution *I've got a headache* itself, but in their surrounding co-text, and may therefore at best be inferred, rather than really being implicit. Thus, the first example is more likely to constitute an answering response, i.e. statement, to the doctor's query as to the ailment of the patient, and the request for a prescription would most probably follow more or less in exactly the words assumed to be the meaning of the communicative unit used as an example. Similarly, in example two, the imperative *Turn down the music* is more likely to be a kind of 'preface' to the explanatory statement *I've got a headache*, while, in the third example, some contextual information would probably be required in order to 'set the scene' for the cause of the headache, while, in the final example, the assumed refusal is more likely to be expressed by an expression of regret – in other words, a *dispreferred response* – such as *Sorry* preceding the explanation. It therefore seems that we need to perhaps adopt a more critical stance towards the notion of indirectness, and start our investigation into the meaning of functional communicative units by focussing on identifying their 'local' direct meaning first.

Yet the method referred to above, initially focussing on syntactic form and then supplementing this in an inferencing process by looking at other lexico-grammatical features, only works well for those types of verbal interaction where the individual speech act is essentially identifiable without taking too much of the surrounding context into account. However, there are also some other types of speech acts whose function is not solely determined by the propositional and illocutionary force inherent in what is said, but is rather almost exclusively related to how they function in reaction to what the previous interlocutor has said or what has been referred to within the larger context of the whole dialogue, such as answers to questions, echoing (partially or wholly repeating) something that the previous speaker has said, or confirming facts that have been established in the course of the interaction. In order to interpret these correctly, it is no longer sufficient to simply analyse the current/local textual unit itself, but necessary to look backwards or forwards within the dialogue, as well as to possibly assign multiple speech act labels that reflect the different, 'cumulative', functions on different levels. The very fact that such textual units exist that absolutely require the surrounding context for interpreting their function is yet another argument against the 'single-sentence interpretation mentality' already previously referred to in connection with logic-based traditional approaches to pragmatics.

1.3 Approaches to corpus-/computer-based pragmatics

In the preceding sections, I mainly focussed on issues and background information related to traditional, 'manual' pragmatics, but of course, 'doing pragmatics' on the computer is in many respects very different from traditional pragmatics, and this is why it is necessary to introduce this field of research separately. This difference is partly due to the nature of electronic data and the methods involved in handling it, and whose discussion will therefore form a major part of this book, but also to some extent by the aims pursued by the people who work in this area.

Computer-based pragmatic analysis has only relatively recently become a major focus of attention, most notably because of increasing efforts in creating more flexible and accurate *dialogue systems* (cf. Androutsopoulos & Aretoulaki 2003: 635–644) that allow a human user to interact with a computer system or to help human *agents* to interact and negotiate with one another if they have a different language background, such as in the German *Verbmobil* project (cf. Jekat et al. 1995). Consequently, the efforts in this field are often geared more towards the needs of the *language engineering* community, rather than attempting to improve our general understanding of communication, which is still the implicit or explicit aim of pragmatics. Although the initial efforts on the SPAAC project (cf. Leech & Weisser 2003), which provided the original basis for the research described here, were also made in order to help and improve such systems by creating annotated training materials for dialogue systems, my own emphasis has long shifted back towards a much more corpus-oriented approach, aimed at offering a wider basis for research on language and communication.

Corpus-/computer-based pragmatics, though, is still very much a developing field, and as yet there exist no real commonly agreed standards as to how such a type of research can or ought to be conducted. Having said this, there have at least been attempts to try and define the levels and units of annotation/analysis that are needed in order to create corpora of pragmatically enriched discourse data, most notably the efforts of the *Discourse Resource Initiative* (DRI). The DRI held three workshops on these issues between the years of 1995 and 1998, and, as a result of this, an annotation scheme called *DAMSL* (Allen & Core 1997) was developed. As this scheme has been fairly influential and parts of it bear some similarity to the DART (Dialogue Annotation and Research Tool) scheme used here, DAMSL will be discussed in more detail in Section 5.1 below. In the expanded title of DAMSL, *Dialogue Act Markup in Several Layers*, we can also see that the language-engineering community often prefers to use the term *dialogue act* instead of the original *speech act* (cf. Leech et al. 2000: 6), but I see no benefit in adopting this here, and will keep on using the traditional term, which is also still better-known in linguistics circles.

Other attempts at reporting on or defining best practice standards in this area have been Leech et al. (2000) within the EAGLES (*Expert Advisory Group on Language Engineering Standards*) framework and the efforts of the MATE (*Multilevel Annotation, Tools Engineering*; cf. Klein 1999) project. While these attempts at defining and possibly also standardising annotation schemes were predominantly carried out for NLP purposes, from the linguistics-oriented side, Kallen and Kirk (2012) also established a pragmatics-related annotation scheme for the *SPICE-Ireland*, based essentially on the original design of the annotation for the corpora of the *International Corpus of English* (ICE; Nelson 2002), but adding various levels of annotation, drawing mainly on Searles's speech act taxonomy. The specific issues raised through these efforts, as well as other relevant endeavours, will be discussed in more detail later.

In recent years, *corpus pragmatics*, as a specialised sub-field of corpus linguistics, has also begun to establish itself more and more. This is evidenced by such publications as the series *Yearbook of Corpus Linguistics and Pragmatics*, whose first volume appeared in 2013 (Romero-Trillo 2013), the new journal *Corpus*

Pragmatics, which was established in 2017, and, perhaps most notably, the edited collection *Corpus Pragmatics: a Handbook* (Aijmer & Rühlemann 2015). Yet, when looking through the chapters/articles in such publications, it quickly becomes apparent that much of the research conducted under this label 'only' more or less constitutes the application of relatively traditional corpus-linguistics techniques, such as concordancing or n-gram analysis, to research on highly limited features, rather than resorting to any form of annotation that would make it possible to carry out large-scale analyses of multiple communicative functions at the same time so as to be able to create communicative profiles.

As far as computer-based approaches to pragmatic analysis from a computational linguistics perspective are concerned, Jurafsky (2006: 579) identifies "[f]our core inferential problems […]: REFERENCE RESOLUTION, the interpretation and generation of SPEECH ACTS, the interpretation and generation of DISCOURSE STRUCTURE AND COHERENCE RELATIONS, and ABDUCTION." Out of these four, this book is only concerned with the latter three, with the main emphasis being on identifying speech acts through abductive (cf. Hobbs 2006), natural language-based, reasoning, instead of employing logic-based formal semantic approaches. Discourse structure and coherence relations are also treated to some, albeit slightly lesser, extent.

According to Jurafsky, "there are two distinct computational paradigms in speech act interpretation: a logic-based approach and a probabilistic approach" (ibid.) in computational pragmatics research such as it is generally conducted by computational linguists. The former approach is essentially grounded in the *BDI* (belief, desire, intention) model (cf. Allen 1995: 542–554) and the concept of *plans*. Allen (1995: 480) provides the following description for plans and their usage.

> A **plan** is a set of actions, related by equality assertions and causal relations, that if executed would achieve some goal. A **goal** is a state that an agent wants to make true or an action that the agent wants to execute. […] The reasoning needed in language understanding […] involves the plans of other agents based on their actions. This process is generally called **plan recognition** or **plan inference**. The input to a plan inference process is a list of the goals that an agent might plausibly be pursuing and a set of actions that have been described or observed. The task is to construct a plan involving all the actions in a way that contributes toward achieving one of the goals. By forcing all the actions to relate to a limited number of goals, or to a single goal, the plan-based model constrains the set of possible expectations that can be generated. (emphasis in original)

Often, the need to generate these types of constraints, and thereby limit the range of understanding of a system, is unfortunately driven by rather commercial reasons because it is obviously highly time-consuming and costly to conduct extensive research on discourse matters. This leads to a fairly limited application basis – and hence lack of generic applicability – for these plans. Plans are thus essentially only usable as, or represent, *ad hoc* methods for dealing with highly specific types of interaction, and are consequently of less concern to my discussion. Furthermore, the BDI model embodies a very complex chain of reasoning and abstract logical representation that is far removed from natural language (for an example of this, consult Jurafsky 2006: 582–586) and "requires that each utterance have a single literal meaning" (ibid. 587), something that is all too frequently not the case in real-life spoken interaction. Consequently, although there is certainly some abductive logic involved in identifying speech acts, as we shall discuss in more detail in Section 6.5, a reasoning process that is based on a complex logic-based abstraction that allows for only one single and precise interpretation seems unsuited to the task at hand. Furthermore, as its name implies, the BDI model is based almost exclusively on the assumption that it is possible to recognise intentions (along with beliefs), something that Verschueren (1999: 48) rather lucidly argues against.

> It would be unwarranted to downplay the role intentions *also* play. An important philosophical correlate of intentionality is 'directedness'. Being directed at certain goals is no doubt an aspect of what goes on in language use ([…]). But it would be equally unwise to claim that every type of communicated meaning is directly dependent on a definable individual intention on the part of the utterer. Such a claim would be patently false. Just consider the Minister who has to resign after making a stupid remark that was felt to be offensive, even if many people would agree that it was not meant offensively. Or, at a more trivial level, look at the exchange in (16).
>
> (16) 1. Dan: Como is a giant silk worm.
> Debby: Yukh! What a disgusting idea!
>
> Dan's innocent metaphor may simply be intended to mean that Como produces a large amount of silk. But that does not stop Debby from activating a meaning potential that was not intended at all And by doing so, (16)1. really *gets* the meaning Debby is reacting to. In other words, (16)1. does not simply *have* a meaning once uttered (which would be the case if meaning were determined by intentions).

One further drawback of taking a BDI approach is that it necessitates a deep semantic analysis with access to a variety of different types of linguistic – and possibly encyclopaedic – information, and thus by necessity needs to be based on ideas of relatively strict *compositionality*, a notion that would seem to contradict the basic assumption that pragmatics represents 'meaning in context', rather than 'dictionary meaning'.

The second form of analysis/identification Jurafsky identifies is what he calls the *cue-based model*.

> In this alternate CUE model, we think of the listener as using different cues in the input to help decide how to build an interpretation. […] What characterizes a cue-based model is the use of different sources of knowledge (cues) for detecting a speech act, such as lexical, collocational, syntactic, prosodic, or conversational-structure cues. (ibid.: 587–8)

In other words, in the cue-based model, we are dealing more or less exactly with IFIDs as defined by Searle, although Jurafsky claims that this approach is – unlike the plan-based one – not grounded in "Searle-like intuitions" (ibid.), but

> […] draws from the conversational analytic tradition. In particular, it draws from intuitions about what Goodwin (1996) called MICROGRAMMAR (specific lexical, collocation, and prosodic features which are characteristic of particular conversational moves), as well as from the British pragmatic tradition on conversational games and moves (Power 1979). (ibid.)

There thus clearly seems to be a misunderstanding regarding the potential origins of the cue-based approach, especially also as the term *microgrammar* never appears in the article by Goodwin referred to above. Be that as it may, the presumed origin of this approach – which is generally also the one followed in the present methodology – is not the real reason why one might want to disagree with computational linguists like Jurafsky in employing cues for the identification of speech acts. Rather, it is the way in which speech acts are in fact recognized by them, which is largely through manual labelling of examples, followed by *machine learning* to derive possible cues, and then applying probabilistic techniques to identify the latter, thereby arriving at a speech act assignment. Although probabilistic methods have relatively successfully been employed in morpho-syntactic tagging (cf. Marshall 1987) and other areas of linguistic analysis, it is well-known that they generally suffer from a *sparse data problem* (cf. Manning & Schütze 1999: 195ff.). This essentially means that they can only work reliably if trained on a fairly large amount of existing data, something which is usually not available when moving from the analysis of one particular domain to another, and using potentially relatively short stretches of text. Furthermore, probabilistic approaches also represent somewhat of a black box which is likely to conflate domain-specific patterns and generic structures induced through the machine learning techniques (cf. Weisser 2015). In other words, what should be common across a variety of different domains – and hence indicate common features of human interaction – may frequently not be easily extractable from the machine-learnt patterns to generalise from in order to re-use this information. This is where the methodology used in this book introduces some very distinct advantages. It (a) specifies the clues to be used as linguistically motivated and transparent patterns and (b) tries to identify and use as many of the generic elements that exist on the different linguistic levels as possible, so that these can then be adapted or augmented as necessary when introducing a new domain into the analysis routines.

The practical corpus-linguistic approach discussed along with the theoretical issues involved in creating pragmatically annotated corpora have also lead me to develop ideas as to which kinds of components may be useful or necessary to incorporate into a research tool that supports the types of analysis and annotation mechanisms discussed here. Incorporating such features into a research tool is also of great importance in corpus linguistics because general linguistic analysis tools like most *concordancers* typically do not provide the functionality required to investigate multiple linguistic levels at the same time. This is why, along with the general theoretical and practical sides of dialogue annotation on multiple levels, I will also introduce one particular research tool here, called DART (*Dialogue Annotation and Research Tool*; Weisser 2016b), designed by me for this specific research purpose.

In any computational analysis of communication, there are different 'mechanics' at work, and those also require different types of computational treatment. One is to do with identifying, storing, and retrieving the right types of information to make it possible to capture and look them up again in whichever data structure(s) one may choose to use for this purpose. The other is to identify the necessary patterns in order to label these units and their content appropriately, consistently and reliably. Since the latter essentially consists in pattern identification and matching operations, it can be achieved quite efficiently by using *finite-state technology* in the form of *regular expressions* (cf. Weisser 2009: 69–79 or Weisser 2016a: 82–101 for overviews of or introductions to their use in linguistic analysis).

Although the research described here is primarily based on English data, I will also occasionally draw on materials from other languages, so that it will hopefully become clear to which extent the theoretical concepts underlying the approach are transferable. Any real in-depth discussion of these other languages is beyond the scope of this book, though, so my treatment will only remain exemplary and sometimes even superficial.

In view of the complexities inherent in the construction of meaning discussed above, and the relative dearth of concrete research into these from a large-scale empirical point-of-view, I intend to investigate the following research questions in this book in order to fill these particular gaps:

1. Which levels of meaning can we distinguish within dialogues pertaining to different domains, some more restricted (e.g. call-centre interactions or problem-solving tasks), and others more open (e.g. data drawn from the Switchboard Corpus)?

2. How can we classify and describe these levels of meaning in order to relate them to the identification of pragmatic force/speaker intentions?
3. What would a suitably generic taxonomy of speech acts look like and what does it need to cover?
4. To what extent is a large-scale automated pragmatic annotation feasible and how could this incorporate different levels of (in-)directness?
5. How can such an annotation make truly corpus-based pragmatics research possible?

1.4 Outline of the book

Having outlined and put into focus the basic framework, it is now possible to proceed to looking at how the task of annotating dialogues can be achieved, which individual steps or particular resources are required for enriching corpora with pragmatics-relevant features, and, once the annotation has been completed, how such corpora can be used in order to draw important conclusions about the various communicative strategies employed by individual speakers or across different corpora from various domains. Through(out) these discussions, I will try to demonstrate why the methodology adopted here provides distinct advantages over 'traditional', and more complex, approaches, and hence represents a major step forward in empirical research into spoken corpus pragmatics.

Yet, no approach or method is completely without caveats, and this one is no exception. As the original data the approach was developed on contained no prosodic information other than pauses, and I also had no access to any of the audio data, a high degree of manual pre- or post-processing, including some interpretation of the data, was necessary in order to allow the analysis routines to recognise – and thus categorise on the different levels – all the relevant units automatically, and with a high degree of accuracy. Not doing so would have led to unnecessary inaccuracies in the annotation, especially if the particular textual units concerned were either very long, and therefore had to be broken up into smaller units of content, or – as is the case with declarative questions – it is only the prosodic characteristics that permit the analysis routines to disambiguate between the potential functions offered by the syntactic structure. Thus, to avoid these issues, the original data were, as far as it was possible without making reference to any audio, broken down into functional units, and then enriched with information pertaining to unit-final prosody, as described in Section 2.3.2.

Before actually moving on to the main chapters, a brief overview of the main parts of the discussion is in order. Chapter 2 will be concerned with linguistic data on the computer in the form of corpora to be used for pragmatic analysis, also discussing important issues in their design and handling in general, as well as basic and specific issues in text representation and annotation. The next chapter, Chapter 3, will provide a brief introduction to the corpora used for this study, the analysis tool DART, as well as the necessary computational resources required to achieve the annotation task. The syntax of spoken language, its units, and its peculiarities that necessitate special computational treatment are covered in Chapter 4. This chapter also contains a comparison of the distribution of syntactic categories and their basic communicative functions across the corpora used. Descriptions of the levels of semantics and semantico-pragmatics, which make important contributions to the realisation of speech acts, form the main substance of Chapter 5, while Chapter 6 will present a brief overview of the largely automated annotation process in DART. In Chapters 7, I shall discuss further results of the research in the form of a discussion of the DART speech-act taxonomy, again including a comparison of the distribution of the various acts across the different sets of data, thereby also illustrating the applicability of the annotation scheme towards establishing functional profiles of the corpora used for this study. Chapter 8 will then round off with a conclusion and outlook towards further potential improvements and future applications of the DART methodology.

1.5 Conventions used in this book

In this book, I use a number of conventions that have either been established in linguistics in order to help us to distinguish between different levels of analysis and/or description, or to indicate special types of textual content relevant to the presentation, for instance to distinguish between text and computer codes, etc.

Double quotes ("…") are exclusively used to indicate direct speech or short passages quoted from books, while single quotes ('…') signal that an expression is being used in an unusual or unconventional way, or that I am referring to the meaning of a word or construction on the semantic level. Curly brackets ({…}) represent information pertaining to the level of morphology, whereas angle brackets (<…>) indicate specific spellings, to contrast these with phonetic/phonological representations of words. Furthermore, they also occur as part of the linguistic annotation introduced in the book. Paired forward slashes/square brackets generally indicate phonological or phonetic representations. Within quoted material, the latter may also signal amendments to the original material made in order to fit it into the general sentence structure.

Italics are generally used in linguistics to represent words or expressions, sometimes whole sentences, that illustrate language materials under discussion. In some cases, they may also be used to indicate emphasis or highlighting. In addition to this, I use italics to indicate specific terminology and speech act labels. Small caps are used to indicate lemmas, i.e. forms that allow us to conveniently refer to all instances of a verb, noun, etc. Last, but not least, `monospaced font` indicates computer code or annotations.

CHAPTER 2

Computer-based data in pragmatics

The acquisition or representation of linguistic material in electronic form always brings with it a number of different issues. Transcribing or transforming the data into a form that meets one's research aims is often one of the most time-consuming and major parts of creating research materials, and a substantial amount of time and resources needs to be allocated to this task before one is actually in a position to analyse the data itself (see Weisser 2016a for a practical introduction to these issues). Therefore, before beginning our exploration of corpus-based pragmatics, I will provide a brief survey of existing technologies and issues surrounding the handling of the 'raw material' involved.

2.1 Linguistic corpora and pragmatics

Today, there is an abundance of electronic corpora designed for many different purposes (cf. McEnery et al. 2006: 59ff.) Evidently, not all of these are equally suitable for different types of language analysis, especially not the type of pragmatic analysis of spoken language discussed here. The most general of these corpora, *reference corpora*, cover a large amount of naturally occurring written or spoken data from a variety of different domains, which, in theory, makes them *representative* of a given language as a whole in terms of vocabulary, syntax, and also many pragmatic aspects of language use. Yet the earliest such corpora, the American BROWN (Francis & Kucera 1979) and its British counterpart, the LOB (Lancaster-Oslo/Bergen; Johannson, Leech & Goodluck 1978) corpus, were hardly representative in this sense yet, as they 'only' contained one million words of written text each. In the 1960s, when both of these corpora were collected, this seemed like a very large amount of data, and written language was still assumed to be more important than its spoken counterpart. Since then, however, it has become clear that a *balanced* corpus needs to contain suitable amounts of both written and spoken language, and that 1 million words are hardly enough to capture many of the interesting phenomena to be observed in language, especially when it comes to identifying *collocations, idioms*, and other rarer forms of language. Sinclair (2005) provides a fairly detailed exploration as to how much data may be required to account for various types of such analyses, and how the size of corpora required for investigating especially longer sequences of words may increase exponentially. To fulfil such needs, corpora of ever-growing sizes are being produced to cover these gaps, and this is why modern *mega-corpora*, such as the British National Corpus (BNC), already contain 100 million words, subdivided into 90 million words of written and 10 million words of spoken language for the BNC, where of course the latter segment is most relevant to our research. Other mega corpora for English, like the Corpus of Contemporary American (COCA; Davies 2009), are even larger, but their content does not cover the same range of spoken language as the BNC, in particular not where unconstrained natural dialogue is concerned.

Spoken electronic corpora appropriate for conducting research in pragmatics have existed at least since the publication of the London-Lund Corpus of Spoken English (LLC; see Svartvik 1990 or <http://clu.uni.no/icame/manuals/LONDLUND/INDEX.HTM>) in 1990. A number of interesting studies on spoken interaction and its properties – such as Stenström 1994 or Aijmer 1996 – have been undertaken based on it. One major advantage of this 500,000-word corpus is that it contains detailed prosodic information that makes it possible to study nuances of attitudinal stance of the individual speakers in detail, rather than 'just' presenting the syntactic, lexical and structural information of the ongoing interaction. At the same time, this detailed prosodic information makes it very difficult to work with the corpus data and to perform automatic analyses of the kind discussed in this book on it, as the prosodic information is integrated into the transcription in such a way that it becomes difficult to recognise the 'shapes' of the individual words easily, as they may contain unusual 'accented' characters to indicate tone movement or other prosodic markers. A brief example from the beginning of the first file of the LLC is shown below, but there will be more to say on these issues in Section 2.3.1.

```
1 1   1   10 1 1 B   11   ((of ^Spanish)) . graph\ology#        /
1 1   1   20 1 1 A   11   ^w=ell# .                                /
1 1   1   30 1 1 A   11   ((if)) did ^y/ou _set _that# -      /
1 1   1   40 1 1 B   11   ^well !J\oe and _I                     /
1 1   1   50 1 1 B   11   ^set it betw\een _us#
```

Figure 2.1 Sample extract from the London-Lund Corpus

One further potential drawback of the LLC corpus is that it only reflects the speech of "adult educated speakers of English" (Svartvik 1990: 11), so that some of the features of more general spoken English may be missing from the data. A corpus that implicitly seeks to redress this problem is the two million-word CANCODE

(Cambridge and Nottingham Corpus of Discourse in English)[1], as it was “targeted towards informal encounters and were made in a variety of settings, such as in people's homes, in shops, restaurants, offices, and informal university tutorial groups, all in the British Isles” (McCarthy 1998). The main disadvantage of this corpus, however, is that it is not generally available, so that it is also not possible to replicate any studies based on it.

In theory, this would then probably leave the spoken part of the BNC, due to its relatively easy accessibility, wide coverage, and size of data, as an ideal candidate for pragmatic analysis. However, in practice, even if one were to limit the selection of data chosen from such a large corpus in a sensible way, initially analysing data from such relatively unrestricted domains computationally would pose problems in terms of the lexical coverage an analysis program would have to offer. In addition, as we shall see in the next section and also Section 3.1.4, some serious problems exist in the spoken part of the BNC. Thus, it is probably best to start designing an analysis program or methodology on the basis of corpora from relatively restricted and clearly defined domains. This is in fact the approach that was taken for the original research behind this study, and also the reason why other projects or efforts aimed at designing computationally tractable methods of analysis for pragmatic data have generally been restricted to smaller corpora and limited domains. In contrast to most previous efforts, though, one of the explicit aims in the design of the methodology employed here was to allow for an extensibility to different domains right from the start by making use of generic elements (cf. Weisser 2002) to implement the core functionality, but also allowing furthers resources to be added later.

One classic exemplar of a dedicated spoken corpus is the 146,855 word HCRC Map Task Corpus.[2] According to the classification scheme established in Leech et al. (2000: 6ff.), this corpus can be categorised as *task-oriented* and *task-driven*. In other words, it represents a specific type of dialogue corpus where two or more interlocutors negotiate or interact in order to achieve a specific task. The particular task in this case consists in finding a route to a target based on two maps that contain partly identical and partly differing information. The MapTask corpus was specifically designed to investigate features of relatively informal interaction on a number of linguistic and other levels, such as speaker gaze, general communicative strategies, etc. (cf. Anderson et al. 1991), and has been *marked up* (see 2.2 below) for a number of these features.

1. See <https://www.nottingham.ac.uk/research/groups/cral/projects/cancode.aspx> for a list of publications related to this.

2. See <http://groups.inf.ed.ac.uk/maptask/> for more details.

Other corpora that have been designed and used in the context of research on the computational analysis of dialogues – mainly in the context of developing *dialogue systems* – include materials from the domains of travel information (SUNDIAL, ATIS, etc.), transport (Trains), business appointments (Verbmobil), etc. (cf. Leech & Weisser 2003: 149). However, apart from the earlier Trains corpus materials from 1991 and 1993, data from such projects is relatively difficult to obtain.

Flöck and Geluykens (2015: 9) claim that "there are no corpora available that are tagged for individual illocutions or even illocutionary types". However, this claim is certainly not true, as, despite a relative dearth of pragmatically annotated corpora, a few corpora containing speech-act related information have been in existence for a number of years. Amongst these are the SPAADIA (ver. 1 released in 2013, ver. 2 in 2015), the Trains 93, and one version of the Switchboard Corpus, data from all of which was used to some extent in this book (see Section 3.1 below), as well as the Coconut and Monroe corpora. For more details on the original annotation of these corpora, as well as a comparison of their annotation schemes, see Weisser (2015). The MapTask corpus already mentioned above also contains annotations that are somewhat similar to speech-act labels, but referred to as *moves*.

The MICASE Corpus (Simpson et al. 2002) has also been marked up with information about pragmatic features, including a sub-corpus that contains 12 pragmatic tags (Leicher & Maynard 2007: 112). However, instead of reflecting generic concepts, the tag labels used there often represent highly domain-specific functions, such as "AHW Assigning homework" or "IRM Introductory Roadmap" (ibid.: 112), and the annotated materials – to the best of my knowledge – have never been made available publicly.

2.2 Issues and standards in text representation and annotation

As already hinted at in the prior discussion, having electronic data in a suitable format is of utmost importance for anything but a cursory analysis and generation of superficial hypotheses. This is why, in this section, an overview of the most important issues that apply to the design, representation and handling of corpora in language analysis shall be provided, beginning with a 'plea for accuracy' in recording the original data used for corpus compilation, as this is a feature that is highly likely to affect any subsequent analysis to a very large extent.

Spending a considerable amount of time on producing 'clean' data – not in the Sinclairean sense of being annotation-free, but free of potential errors due to typographical or encoding issues (cf. Weisser 2016a: 4–5 & 56–57), though – may sometimes seem an unnecessary effort, just in order to conduct a small-scale project. However, it is certainly time well-spent, as one never knows to which use the data may be put later on and how badly represented data may affect the outcome of any type of analysis. For instance, many researchers use the BNC for their work on British English because it is the major reference corpus for this variety, and highly useful research can be conducted on it in many areas, especially through powerful interfaces such as BNCweb <http://bncweb.lancs.ac.uk/> or the BYU-BNC one created by Mark Davies <https://corpus.byu.edu/bnc/>. Its compilation certainly also represents a most laudable and worthy effort, but if one takes a closer look at some of the spoken data and how it was transcribed, one cannot but wonder how much of an error may be introduced into any numerical analysis conducted on it simply due to the fact that its transcribers seem to have been relatively unqualified, and thus often did not seem to know where to use an apostrophe or not, apart from generally being somewhat insecure about their spelling. For example, in BNC file D96 alone, which I re-transcribed from the audio provided in BNCweb, there is an alarmingly high amount of instances of the contraction *we're* that were simply transcribed without an apostrophe, plus a number of other rather serious transcription errors. These issues can easily be seen in the excerpts provided below, where deleted or replaced items are marked as struck through, and replacements or insertions indicated in bold italics:

> I mean, ~~a lot~~, what I can say with on ~~the~~ youth~~s~~, I mean, I think ***we're*** ~~were~~ doing, we ~~were~~, ***we're*** ~~were~~, ***we're*** ~~were~~ ***working*** ~~walking~~ with young people at the local levels of various places in the town you know, we've got, we haven't got as many resources as we want yet, but ***we're*** ~~were~~ still trying to do that, well I actually feel, on youth we're doing quite a good job you know, ***extensive*** ~~expensive~~ job you know, that we are, and, and all ***the*** ~~that~~ concerns you raise, ***we're*** ~~were~~ certainly aware of.
>
> The problem is solving all the problems, providing all the facilities in, in ***a*** ~~the~~ situation where it's diminishing resources, I mean we wouldn't be actually be carrying out this frontline review, in the way that ***we're*** ~~were~~ gonna do it, if we didn't have the problem with the money we've got, you know.
>
> […] We're ~~with~~ still not ***losing*** ~~loosing~~ ~~site~~ ***sight*** of the idea of having a cafe, bar, coffee for ***Heyham*** ~~people~~, one of the things that ***we're*** ~~were~~, ~~that~~ gonna look to ~~through and~~ explore explore actually is er setting up some kind of coffee bar ***facility*** ~~facilities~~ at Kingsmoor, with the play farming ~~barn~~, there next to them.
>
> […] At the, at the last search, at the last highway~~s~~ committee, although ***we're*** ~~were~~ not having, having, having the, the full service that we ~~at the~~ envisage in the first instance, a lot is going to be done, there'***s*** ~~is~~ going to be some more erm shelters

> ***erected*** ~~directed~~ there and one or two other facilities and somebody has even suggested that we put a toilet there which is a very good idea.
>
> […], but ***anyway*** ~~any way~~, there will be some improvements for the bus station in the future.
>
> […] Yeah, ***you're*** ~~your~~ off the hook.
>
> […] Right, ***we're*** ~~were~~ now on other reports. ***Anybody*** ~~Any body~~ got ***anything*** ~~any thing~~ else to report, with got a few minutes left? Yes, no, any other business, you can all go, ***you're*** ~~your~~ all off the hook.
>
> […] Don't go ***overdoing*** ~~over doing~~ it.

Although the extract above probably already provides a fairly striking impression of the severity of the problem, let us take another look at the overall discrepancies in numbers between the original BNC version and my corrected one, which may still contain errors, due to intelligibility issues that perhaps no transcriber can resolve.

Table 2.1 BNC D96: Discrepancies between original and edited version

Unit	Original BNC version	Corrected version
w[a]-units/words	839	902
u[b]-units/turns	40	35
s[c]-units/c[d]-units	51	162
punctuation	160	138
insertions		14
deletion(s)		1
corrections		56

a. word
b. utterance
c. sentence(-like)
d. clausal and non-clausal (Biber et al. 1999: 1070; cf. 4.1)

The information regarding the original units in the first four rows of Table 2.1 were taken directly from the header of the BNX XML file. The relatively discrepancy in terms of w-units/words is partly due to insertions of materials that were either marked as unclear in the original, but I was able to discern from the audio after all, or corrections where the marking of previous erroneously non-marked contractions resulted in two words being present, rather than just one. The latter applies, for instance, to all cases of *were* in the original transcript that should have been transcribes as *we're*. The difference in u-units/turns can be explained mainly by the fact that, occasionally, turns by the same speaker were split over multiple u-units, in particular if some event, such as background laughter, occurs in between. Without more detailed transcription guidelines, however, it is difficult to ascertain the exact reason for this.

A similar phenomenon partly explains the divergence in the number of punctuation marks, as the transcriber of this particular file seems to have added punctuation marks even after event descriptions, in other words, non-textual material, even if this does not really make any sense at all. Although the number of c-units in the corrected version is higher than that of the s-units in the original, which would normally lead us to expect a higher instance of punctuation marks in the former, there are a number of reasons why this is not the case. First of all, in the BNC, 'sentence-like' units are marked in a rather haphazard way, where often the end of functional units is marked by a comma, rather than a major punctuation mark, as can easily be seen in the first paragraph of the extract we saw earlier. Most of these were deleted in the conversion process, but were partially replaced by 'phono-pragmatic' punctuation tags (see 2.3.2 for more information) in the corrected version. In addition, multiple functional units are often conflated into one s-unit in the BNC, while the DART scheme employs a more fine-grained system of syntactic/functional units (see Chapter 4), which accounts for the considerable difference in number between s- and c-units in the two versions.

What is perhaps more important than the differences in the individual units discussed above is the number of errors presented in the final three rows of Table 2.1. Added up, insertions, deletions, and other corrections account for 71 word tokens. If we see this number relative to the original 834 tokens, we arrive at an error rate of 8.5%. Applying the customary, albeit arguably incorrect (see Weisser 2016a: 175), method for frequency norming and extrapolating to instances per 10 million words, we could then potentially expect to find 846,250 word-token errors in the whole of the spoken part of the BNC!

The problems illustrated above, occurring within such a relatively short space of text, will of course not only skew the general results of any (word) frequency counts, but also influence any such counts based on word classes, as well as the comparative counts that seek to illustrate the differences between spoken and written language.

Although, in terms of pure frequency counts of word classes, some of these errors may actually balance out each other in that a lack of an apostrophe in one place may be compensated by an additional erroneous one elsewhere, the above observations should lead us to raise serious doubts about the validity of many frequency counts obtained from large reference corpora. This especially ought to be the case if these corpora have been collected very quickly and only few people have been involved in their compilation and processing, such as may possibly be the case with the final version of the American counterpart to the BNC, the Open

American National Corpus (ANC),[3] which, from its inception, was hailed and marketed as an enterprise in 'efficient' corpus collection.

The problems highlighted above simply indicate that the issue of *homographs* or potentially occurring word forms that have been misrepresented or represented as unclear is something that mere use of a spell-checker will not eradicate, and therefore a close reading and potential manual post-editing of transcriptions cannot be avoided, unless appropriate care has been taken to ensure that the data has been transcribed extremely thoroughly in the first place. And even then, occasional errors that were either overlooked during the compilation phase or might have been introduced by unforeseen side effects of any computer programs used to process the data cannot be discounted and may always have at least a minor influence on any kind of so-called 'statistical', i.e. frequency, analysis of texts. This might not seem much of a problem if 'all' we are interested in is the frequencies of words or their distributions, but, in the context of computational dialogue analysis, it may well affect the creation of domain-specific lexica required to do the processing (cf. Section 3.3.3).

However, it is not only frequency counts that may be affected by a somewhat careless preparation of corpus materials. Perhaps more importantly, when performing a syntactic analysis of a particular unit of text, the difference between an apostrophe being present or not may in fact prevent us from recognising a declarative structure and mistaking it for an ill-formed syntactic structure – later referred to as *fragments* (cf. Section 4.3.9) – and where the latter may be much more difficult or impossible to interpret in its function, as in the case of "your off the hook" – instead of the correct *you're off the hook* – from the BNC sample above.

To summarise: the importance of using data that has been created with the utmost care and being aware of the content of this data is not only relevant for producing or extracting high-quality information from our corpora, but also in order to be able to form the right research hypotheses and come to the right conclusions about them. This is a fact that all too often seems to be ignored in the quest for ever-increasing amounts of corpus data that can be collected and prepared for dissemination in a maximally efficient and inexpensive way. In other words, quality should always remain more important than expedience.

Having demonstrated how important it is to work with clean data, as well as to have some kind of expectation about which types of issues may be encountered in electronic data, we can now move on to discussing the basic means of rendering the data in a faithful way, and adding additional useful types of structural and linguistic information to it. In this context, it will first be necessary to introduce or discuss some important terminology that enables us to describe the essential

3. <http://www.anc.org/>

concepts behind representing linguistic data on the computer, and more specifically, how it can be ensured that the method of representation employed is one that as many potential users of the data as possible will be able to understand and make use of, e.g. for interpreting and verifying the results of the analyses. Discussing these issues at this point is of vital importance because, traditionally, logic-based and philosophically oriented pragmatics, unlike CA, does not normally pay much attention to the nature of real-life data and the forms it may occur in, but rather 'abstracts away' from the 'messiness' of naturally occurring data. It does so either by constructing examples or simply leaving out 'performance' details that seem to be irrelevant to explaining the underlying problems encountered in (re)constructing the *logical form* of an 'utterance'. Nonetheless, any kind of corpus-based pragmatics definitely needs to take heed of these problems, as ignoring them may lead to incomplete, or even incorrect, analyses.

Having explicit standards in representation and annotation is not only important for handling language data in more industrial settings, such as for language engineering purposes. Setting, understanding and adhering to these standards also enables researchers to make the nature of their data maximally explicit, and the enriching annotation as far as possible self-describing. The following sections will introduce the most important concepts in the representation and annotation of language data, and make the necessity for sensible standards explicit by showing where there have been problems in interpreting inconsistent and difficult, or perhaps unnecessarily fine-grained, coding schemes in the past, thereby making it difficult to understand and interpret data produced by different researchers or research teams. Further testimony to the fact that representation and annotation in describing language data are important issues concerning the interpretation of such data is provided by the very fact that books like Edwards and Lampert's *Talking Data: Transcription and Coding in Discourse Research* (1993) even exist. In addition to demonstrating how important these features are to rendering language information in general, I will also point out how much more of a necessity for using standardised text rendering methods there is when it comes to analysing and processing language on the computer.

Amongst the first terms one is likely to come across in the context of corpora are *markup* (also *mark-up*) and *annotation*. Edwards (1993: 20) still makes a distinction between the two terms, also introducing two synonyms for annotation, *coding* and *tagging*:

> '*Coding*' (also called 'tagging' or 'annotation') differs from transcription in its content and degree of structuring. Rather than capturing the overtly observable acoustic and non-verbal aspects of the interaction, coding focuses on events which bear a more abstract relationship to each other, that is, on syntactic, semantic and pragmatic categories. […]

> '*Mark-up*' originated in the marks used by typesetters to signal the structural units and fonts of a document. As defined here, it concerns *format-relevant* specifications intended to be interpreted by a *typesetter or computer software*, for proper segmentation of the text and cataloguing of its parts, in the service of formatting, retrieval, tabulation or related processes.

As we can see here, Edwards draws a fairly clear distinction between signalling structural *segmentation* of the text (markup) and adding or enriching data by making explicit other types of information that are only implicit in the text (annotation). However, today the two terms are often used synonymously, as can be seen from the entry in the EAGLET Term database, which defines *annotation* in the following way:

> **annotation** /ænə'teɪʃən/, /{n@'teIS@n/, [N: annotation], [plural: -s]. Domain: corpus representation. Hyperonyms: description, representation, characterisation. Hyponyms: part of speech annotation, POS annotation, segmental annotation, prosodic annotation. Synonyms: labelling, markup. Def.: 1. Symbolic description of a speech signal or text by assigning categories to intervals or points in the speech signal or to substrings or positions in the text. 2. Process of obtaining a symbolic representation of signal data. (2) The act of adding additional types of linguistic information to the transcription (representation) of a text or discourse. 3. The material added to a corpus by means of (a): e.g. part-of-speech tags. (Gibbon et al. 2000: 375)

In practice, though, it probably pays to look more closely at the words that tend to collocate with *annotation, markup*, and also the third term mentioned by Edwards, *tagging*, as well as the actions that may be associated with them. As is also partly implicit in the definition from the EAGLET Term database, annotation often refers to the process of enriching corpus data in specific ways, and we thus often talk about *corpus* or *dialogue annotation*. The term tagging, however, generally tends to occur in phrases, such as *POS (part-of-speech) tagging*, which almost exclusively refers to the action or result of adding *morpho-syntactic* (or word-class) information to the words in a text/corpus. And last, but not least, the term markup is generally used in such expressions as SGML/HTML/XML markup, which refer to the 'physical' or computer-related *representation* of materials at various levels, not only at the segmental[4] level referred to by Edwards above.

Having clarified some terminological issues, I will now provide a brief introduction to the general means preferred by corpus or computational linguists to achieve the kind of markup referred to last, beginning with a brief historical – and

4. Segmental here essentially means 'structural' and is not to be confused with the term segmental as it is used in phonology.

potentially somewhat simplified – overview of the development and linguistic utility of some of the more important *markup languages*, before discussing the particular requirements and proposed standards that exist for dialogue annotation. This discussion will be split into two sections, where the first one deals with more general aspects of markup on the computer, while the second will discuss linguistics-oriented markup.

2.2.1 General computer-based representation

The original markup language of choice for linguistic purposes was SGML (Standard Generalized Markup Language). This language developed out of attempts to standardise means of exchanging information in the 1960s (Bradley 1998: 6). Anyone who has ever had to struggle with problems related to different proprietary document, sound, or graphics formats will easily understand that standardisation is an important and commendable effort because it ensures transparency and transportability. However, SGML itself, the first standard in this respect, was only fully ratified by the ISO (International Standards Organization; <http://www.iso.org/iso/home.htm>) in 1986 (ibid.), and even though it was widely adopted by various research communities, has not 'fulfilled all its promises'. Thus, these days, it has largely been replaced by XML (eXtensible Markup Language), which is more flexible, even though it still has not eliminated some of the original issues.

The basic idea in all markup languages that are related to, or derived from, SGML is that the content is stored in *plain text* format, meaning in largely human-readable, *non-binary* form, while structural and basic category information is marked up through so-called *elements*. Elements are also sometimes referred to as *tags*, but should of course not be confused with the kind of tags employed in many tagged corpora to mark up morpho-syntactic information.

So as to be able to easily distinguish elements from the raw text data, elements tend to be represented in angle brackets (<…>), where the opening bracket (<) is immediately followed by the name of the element. This name may reflect a text-level, syntactic, morpho-syntactic, etc., category or sub-category. There are essentially – and conceptually – two different types of elements, those that surround or delimit specific divisions or categories, and those which mainly represent processing instructions to a computer and may reflect particular types of formatting, such as *line breaks*, used to link in or include external content, or express non-structural or non-hierarchical content. The former tend to enclose the marked up information in paired tags, where the closing one, to indicate the end, contains a forward slash (/) between the opening angle bracket and the element name, thus yielding something like `<element name>`*element content*`</element name>`. A processing instruction, because it is a 'one off' command, consists of only a single, unpaired tag.

Elements may also contain additional *attributes*, following the element name in the start tag. These usually specify the nature of the category expressed by the element further, or may simply be used to provide a unique identifier, such as a number, for the element. They tend to consist of an attribute name and an associated value, which are joined by an equals sign. Let us exemplify this to some extent by looking at an excerpt from one of the spoken files from the original version of the BNC (KCU), also pointing out some of the problems that tended to arise with the use of SGML.

```
<u who=PS0GF>
<s n=0001><w RP>On<c YQUE>? </s>
</u>
<u who=PS0GG>
<s n=0002><w RR>Right<c YCOM>, <w PPIS1>I<w VM>'ll <w VVI>go <w CC>and <w
VVI>get <w AT1>a <w NN1>video<c YCOM>, <w RR>okay<c YQUE>? </s>
</u>
<u who=PS0GF>
<s n=0003><w UH>Yeah <w PPIS1>I <w VD0>do<w XX>n't <w VVI>know <w DDQ>what<w
VBZ>'s <w RP>on<c YSTP>. </s>
</u>
<u who=PS0GG>
<s n=0004><w RR>Alright<c YSTP>. </s>
</u>
```

Figure 2.2 Sample excerpt from the original SGML version of the BNC

As is evident from Figure 2.2, SGML uses a fairly standard notation for opening tags, but unfortunately the sample text is not always consistent in indicating the ends of textual elements, and often the start of a new element simply has to be taken as a signal that the preceding element is now to be taken as closed. Thus, the `<u>` ('utterance') and `<s>` ('sentence') elements in the example are explicitly closed, whereas `<w>` elements are not. This type of 'shortcut', which is allowed in SGML, makes processing it much more difficult than needs be and also much more error-prone. Figure 2.2 also demonstrates that SGML is organised in a hierarchical (tree) structure where certain elements can be *nested* within one another. Thus, the sentences contain a number of words (`<w>`), but are themselves embedded in `<u>` elements. The exact document 'grammar' is specified via a so-called *DTD* (Document Type Definition).

In our example, we can also see that all attributes occurring inside the start tags are either not quoted, as is the case for the *n* IDs, which could cause *parsing* problems if the attribute values contained spaces, or the attribute name is even assumed to be explicit, as we can see in the examples of the PoS tags, where the attribute name and the conjoining equals symbol are missing. Of course, this could only work if this particular element were assumed to only ever allow a single type of attribute. Thus, as soon as one might want to add e.g. a numerical ID for each <w> element, one would first need to ensure that an appropriate attribute name is added in front of the existing value.

That the SGML annotation here is used for linguistic purposes can be understood from the tags <s> and <w>, indicating 'sentences' and words respectively. This kind of markup therefore seems to be quite appropriate for general, perhaps more written language oriented rendering of linguistic material, in order to establish category sub-divisions down to the level of syntax.

Compared to its derivatives HTML and XML, SGML also has two other major disadvantages, the first being that it absolutely requires a DTD specifying the structure allowed for the document in order to allow any type of serious processing, and the fact that it is not supported by any 'standard' browser software. On the other hand, one big advantage, at least in comparison to HTML, is that a large set of tag definitions/DTDs, such as for the TEI (see 2.2.3 below), were originally designed for SGML, although nowadays more and more of these are being 'ported' to XML, too.

Although HTML is a direct descendant of SGML, it only provides a limited set of tags, which on the one hand makes it less flexible than XML, but on the other also much easier to learn. It is widely recognised by standard browser software, and the DTDs are already built into these browsers, although they can also be explicitly specified. HTML itself is largely standardised and also technically extensible via *CSS* (*Cascading Style Sheets*; see below) to some extent, so that it is already quite useful for the presentation and visualisation of linguistic content. This extensibility is somewhat limited, though, which is why it is not really flexible enough as a markup language for representing more complex linguistic data. It is, however, possible to transform complex linguistic data encoded in SGML or XML into a more simplified HTML representation for display in a standard browser.

XML is much more versatile than HTML because – as the attribute *extensible* in the name indicates – it was designed to provide the ability to the user to completely define anything but the most basic language features. It is much easier to process and far less error-prone than SGML because some of the shortcuts illustrated before are no longer allowed. All XML documents minimally have to be *well-formed*. In other words, no overlapping tags (e.g. ~~**…<i>……</i>**~~) as were possible to use in HTML are allowed, and end tags are required for all *non-empty*, paired, elements. While the express prohibition of overlapping tags makes it easier to check the well-formedness of XML documents, it may also present a distinct disadvantage for annotating linguistic documents, as e.g. speaker *overlap* – where one speaker in a dialogue starts talking while the other has not finished yet – cannot be marked up using a container element, since this would 'interfere' with the hierarchical structure of speaker *turns* and their embedded structural utterance units.

So-called *empty tags/elements* differ from their SGML equivalents in that they have to be ended by a slash before the closing bracket, e.g. `<element name />`. They provide a work-around for the problem of overlapping tags, should it be required to indicate overlap precisely because they can be given attributes that signal its start and end, along with potentially some IDs if there should be multiple concurrent overlap sequences. Unlike with older forms of HTML, where *case* did not matter, XML is also *case sensitive*, so that tags like `<turn>`, `<Turn>` and `<TURN>` are treated as being different from one another.

The representation of individual letters – or *characters*, to be more precise – on the computer may also be an important issue in linguistics, especially in dealing with multi-lingual data or data that needs to include phonological information. For dealing with 'English only', a very limited Latin-based *character set* may appear sufficient. Originally, characters in English data were encoded in a character set called ASCII (*American Standard Code for Information Interchange*) and its later derivatives, but as computing technology spread across the world, this presented problems in representing other languages that contain accented characters, etc., as well as the occasional foreign word appearing in English texts, such as *fiancée*. In order to overcome this problem, and be able to store characters from different character sets in one and the same document, a universal character encoding strategy called *Unicode* was developed. Unicode exists in a number of different formats, the most widely used and flexible of which is called UTF-8, which is the default assumed for XML files unless an alternative encoding is specified. This, along with the fact that it is the format that Perl – the programming language used for the implementation of the analysis tool discussed in the next chapter – uses it for its internal character representation, made a combination of XML and UTF-8 the most logical choice for the encoding of the data used for this study.

Leech et al. (2000: 24) still argued against the use of Unicode and recommended to use a 7-bit ASCII character set for encoding most information, as this was most widely supported in general at the time, so that, for example, the inclusion of phonetic transcription details could only be achieved by using the transliteration format SAMPA (ibid.). However, as more and more operating systems, browsers and even standard editors available on many different platforms now widely support UTF-8, such transliterations or the use of special *character entity references* for e.g. representing foreign characters like *é* (`é`) or escaping umlaut characters – e.g. writing *"u* for <ü>, as was done for the Verbmobil data – should these days no longer be necessary. Seeing the text in the way it was meant to be represented in the respective writing or representation systems makes dealing with, and processing, the data much more intuitive, and also allows researchers to use established linguistic conventions, such as proper phonetic transcription, instead of remaining caught up in unnecessary conventions that only stem from the traditional anachronistic predominance of an American influence on data representation on the computer.

XML, in contrast to HTML, describes content, rather than layout, so that the rendering, i.e. the *visual* representation of a document, needs to be specified via a *style sheet*, or otherwise the browser or application displaying it would not know how to achieve this. If no style sheet is explicitly provided, most browsers will try to render the XML content using their own default style sheets that at least attempt to represent the hierarchical tree structure and often allow the user to expand and collapse nested (embedded) structures. Other applications can at least display the plain text, provided they support the given encoding. A screenshot of what the hierarchical XML display inside a browser looks like is shown in Figure 2.3.

```
-<u who="PSOGF">
  -<s n="1">
      <w c5="AVP-PRP" hw="on" pos="ADV">On</w>
      <c c5="PUN">?</c>
   </s>
</u>
- <u who="PSOGG">
  - <s n="2">
      <w c5="AV0" hw="right" pos="ADV">Right</w>
      <c c5="PUN">, </c>
      <w c5="PNP" hw="i" pos="PRON">I</w>
      <w c5="VM0" hw="will" pos="VERB">'ll </w>
       <w c5="VVI" hw="go" pos="VERB">go </w>
      <w c5="CJC" hw="and" pos="CONJ">and </w>
      <w c5="VVI" hw="get" pos="VERB">get </w>
      <w c5="AT0" hw="a" pos="ART">a </w>
      <w c5="NNl" hw="video" pos="SUBST">video</w>
       <c c5="PUN">, </c>
      <w c5="AV0" hw="okay" pos="ADV">okay</w>
       <c c5="PUN">?</c>
   </s>
</u>
-<u who="PSOGF">
  - <s n="3">
```

Figure 2.3 Hierarchical display of XML in a browser window

Figure 2.3 contains the fragment from file KCU of the BNC depicted as SGML earlier, and it is clearly visible that the markup has been suitably adjusted to make it well-formed, with all start and end tags properly set, all attribute names given, and all attribute values quoted. Furthermore, some additional attributes have been added, where, according to the BNC User Reference Guide[5] "hw specifies the headword under which this lexical unit is conventionally grouped, where known.". "[H]eadword" here subsumes all paradigm forms associated with a particular word form (or *type*), regardless of their PoS, so it is distinct from a *lemma*, which only subsumes those forms of a paradigm that belong to the same PoS. For example, the headword *hand* subsumes the nominal forms *hand* (sing.) and *hands* (pl.), as well as the verbal forms *hand* (inf./base form), *hands* (3rd pers. sing), etc. Furthermore, the pos-attribute now indicates a simplified PoS value, whereas the c5-attribute provides more specific PoS information, based on the more elaborate CLAWS C5 *tagset* (Garside, Leech & McEnery 1997: 256–257).

Apart from the well-formedness criterion described above, the document structure of an XML document can also be more rigorously constrained by specifying either a DTD or a *schema* that it needs to conform with, in which case we talk of a *valid* document. Issues of designing DTDs or schemas will not be discussed here because they are fairly complex[6] and the data structure used for the DART annotation scheme is relatively simple, but a brief overview of some of the rendering options for XML documents using style sheets will at least be provided.

As can be seen in the illustration above, each XML document represents a hierarchical structure. The outer 'layer' for this hierarchy – not shown above – is represented by a container or 'wrapper' element that encloses all the nested elements. In the case of the DART annotation scheme, this element is aptly named <dialogue>. This wrapper element is only preceded by a single special processing instruction, the XML declaration, `<?xml version="1.0"?>`. This declaration may also contain further attributes, such as the encoding or whether the document is a standalone document or not, i.e. whether an associated external DTD exists.

Style sheets allow the author to present or publish material in a more appropriate format, for instance specifying line spacing, indentation, positioning, font and background colours, etc. What may at first seem to only be a feature to make the rendering of the textual and annotation materials look nicer does in fact have its purpose because proper layouting and colour-coding may well help to enhance the representation of the logical structure of documents, as well as to highlight

5. At <http://www.natcorp.ox.ac.uk/docs/URG/ref-w.html>

6. For more detailed information on this, see Carstensen et al. 2004: 140 ff.

certain facts about the content of a linguistic XML document. Thus, it e.g. becomes possible to highlight information about the syntax or semantics of a particular unit of text. Something similar to a style sheet is for instance used in the implementation of the analysis program discussed later to indicate the difference between syntactic units, such as declaratives and interrogatives.

Below, a short XML sample without a style sheet is shown, followed by an illustration of what the latter may look like using a simple style sheet.

```
<?xml version="1.0" encoding="UTF-8"?>
<sample>
    <sentence>
        <word pos="DET">This</word>
        <word pos="BE">is</word>
        <word pos="DET">a</word>
        <word pos="N">sample</word>
        <word pos="N">sentence</word>
        <word pos="PUN">.</word>
    </sentence>
</sample>
```

Figure 2.4 A short, illustrative, linguistic XML sample

This is a sample sentence .

Figure 2.5 A colour-coded XML sample

```
sample {display: block; margin-left: 5%; margin-top: 5%; font-size: 2em;}
[pos=DET] {display: inline; color: blue;}
[pos=BE] {display: inline; color: red;}
[pos=N] {display: inline; color: green;}
[pos=PUN] {display: inline; color: grey;}
```

Figure 2.6 A sample CSS style sheet

The first line ensures that every time a sample element is encountered, this is displayed as a *block-level* element, a text block similar to a paragraph, with spacing around it. Furthermore, just to ensure that the display is not 'crushed' against the top and left-hand side, a margin of 5% of the page width is specified and, to enlarge the text a little, a relative value (*em*) for the font-size defined for the whole page, which is effectively twice the default font-size the browser would use. The next few lines specify that each time a *pos* attribute with either the value of *DET* (for determiner), *BE* (a form of *be*), *N* (for noun), or *PUN* (for punctuation) is encountered, whatever is enclosed in the corresponding element tag is displayed *inline*, in other words, not as a separate block, and using the appropriate colour. If you observe the XML and its corresponding style sheet-controlled output closely, it will probably become evident that the browser has also automatically added a space after rendering each inline element, something which was not part of the original XML.

XSL, the style sheet language developed for use with XML, provides similar options to CSS for formatting XML display, but also much more complex selection mechanisms and allows reuse of 'text objects', e.g. for producing tables of contents from elements marked up as headings, etc., via *XSL Transformations (XSLT)*. Layout design for other (printed) media is also supposed to be enhanced through *XML Formatting Objects (XSL-FO)*. However, for rendering XML, it is not even absolutely necessary to use XSL, but a simpler, albeit less powerful, solution is to simply link in a CSS style sheet to control the display, as we saw above. None of the XML style sheet options are currently exploited in the implementation of the annotation, but links to an appropriate – maybe user-definable – style sheet can be included in the dialogues used in DART.

2.2.2 Text vs. meta-information

In valid HTML code, there are two separate sections that make up an HTML document. The first of these is represented by the `<head>` and the second by the `<body>` element. The two different types of information expressed by these elements are quite distinct from one another. The first one is somewhat similar to the *front matter* or *imprint* of a book, which contains meta-information about that book, such as its *title*, the *author*, the *typeface* used, etc., and does in fact not represent any real book content, whereas the second one contains the actual text itself.

What is called the *head* element in HTML is usually referred to as a *header* in general. Headers in corpus data may contain various types and amounts of meta information, such as the language the data is in, its encoding, where, when and how it was collected, the author, copyright situation, whether the individual file is part of a larger collection, etc. For spoken data, often some speaker information is included, as well as the recording date and quality, the number of channels, etc. Such meta information can become quite extensive, as is e.g. the case in the BNC files, and often needs to be skipped over when processing the files, either for annotation, concordancing, or other forms of processing. Although much of this may be highly useful information about the corpus files, it does not really form part of the text itself, and can be quite distracting when 'interacting' with the linguistic data in any form. Thus, perhaps a more suitable alternative to using an extensive header is some kind of external description of the data. This has the distinct advantage of keeping the text 'clean' and easier to process, even if it may necessitate distributing additional files containing such meta-information that can be consulted e.g. when selecting data on the basis of the age or sex of the speaker. Depending on how extensive or deeply structured it is, such external documentation can either be kept in a simple plain text file or in a database (cf. Leech et al. 2000: 13).

As much of the data used for this study did not actually provide any detailed information about the speakers or was relevant in any other way for the processing, the DART XML representation does not include a separate header. In general, only the most important information pertaining to the corpus, the identifier of the dialogue within the corpus, and the language, are stored as attributes inside the container tag, e.g. `<dialogue corpus="trainline" id="01" lang="en">`, although for some data, additional information about sub-corpus types, etc., may be present.

2.2.3 General linguistic annotation

Although, for the sake of simplifying our processing later, we will often specifically disregard some of the recommendations made by the wider language research community (at least initially), it is still important to discuss some of the efforts that have been made in the past in order to establish a common framework for the exchange of annotated language data, most specifically those of the Text Encoding Initiative (TEI).[7] Apart from discussing existing practices and schemes for linguistic annotation in a general way, the motivation for choosing particular representation and annotation options employed in the annotation scheme used for the data annotation in this study will also be explained as and when appropriate.

The TEI itself is a research project, organised and funded by the major associations that deal with computing in the humanities, the ACL (Association for Computational Linguistics), the ALLC (Association for Literary and Linguistic Computing), and the ACH (Association for Computing and the Humanities). The explicit original aim of this project was to devise some recommendations, as well as an associated (SGML) markup framework, that would guarantee the successful annotation and exchange of data for many diverse language-related needs, ranging from library catalogues, via standardised dictionary entries, to critical editions of literary works or large language corpora, such as the BNC. The TEI framework has developed considerably further since its inception, especially with its changeover to XML in version 4, published in 2002. The latest version of the guidelines, P5, appeared in November 2007 and is available from <http://www.tei-c.org/Guidelines/P5/>.

7. <http://www.tei-c.org/>

When talking about the TEI, however, people generally tend to refer to the TEI's *Guidelines for Electronic Text Encoding and Interchange* (Sperberg-McQueen & Burnard 1994) and their recommendations for markup. Burnard (1995) provides a short overview of these guidelines, which will be presented here in even more condensed form, at the same time relating their usefulness to the requirements for the task of pragmatic annotation. The guidelines cater for both spoken and written texts alike. In fact, the basic unit contained in a TEI-conformant document is usually a `<text>`, although it may also be a *group* (`<corpus>`) for representing collections of text (Burnard 1995: 72). Each document contains a header that specifies variable amounts of bibliographical information; additional information common to a number of documents can also be specified in a separate header document (cf. 2.2.2 above). As already pointed out above, the corpus data used here – in contrast to the TEI guidelines – do not use *text* as a basic unit, or container element, but instead the more appropriate *dialogue*.

The TEI specifications provide a number of *tagsets*, definitions of elements that may be used in a document and which can be specified via a DTD. The two core tagsets, which are automatically used, provide definitions for the header elements and a number of general elements for encoding textual features that may be common to a variety of different types of text. A list of the latter, reproduced from Burnard (1995: 75), is given below:

1. paragraphs;
2. passages of verse or drama, distinguishing for example speakers, stage directions, verse lines, stanzaic units, etc.;
3. list of various kinds, including glossaries and indexes;
4. typographically highlighted phrases, whether unqualified or used to mark linguistic emphasis, foreign words, titles, etc.;
5. quoted phrases, distinguishing direct speech, quotation, terms and glosses, cited phrases, etc.;
6. names, numbers and measures, dates and times, and similar 'data-like' phrases;
7. basic editorial changes (e.g. correction of apparent errors; regularization and normalization; additions, deletions and omissions);
8. simple links and cross-references, providing basic hypertextual features;
9. pre-existing or generated annotation and indexing;
10. bibliographic citations, adequate for most commonly used bibliographic packages, in either free or a tightly structured format;
11. simple or complex referencing systems, not necessarily dependent on existing SGML structure.

The core tagsets can then be augmented by one *base* tagset that defines the contents of what may appear inside the `<text>` element, chosen from the following ones: prose, verse, drama, transcribed speech, letters and memoranda, dictionary entries, and terminological entries (ibid.: 73). Furthermore, any number of additional tagsets may be included. Elements in these tagsets can also be redefined or augmented if necessary.

The hierarchical annotation of the textual material, e.g. a novel, may consist of such elements as `<body>`, `<chapter>`s, `<p>`aragraphs, and `<s>`entences, with different subdivisions possible for other text categories. For the general segmentation of spoken material, there are a number of levels pre-specified, starting with divisions `<div>` at the top, which contain utterances `<u>`, and these, in turn, other subdivisions established "by prosodic or syntactic criteria" (Johansson 1995: 87).

In basic practice, such as is the case in the spoken files of the BNC, the `<u>` element thus essentially corresponds to what is generally referred to as a speaker *turn* in the literature on dialogue analysis (cf. Leech et al. 2000: 56). This is why the label *u* is probably best avoided, especially because the dialogue research community often refers to exactly those subdivisions referred to above as (segmental) *utterances*, and which may be assigned so-called *utterance tags* in order to reflect their function (ibid.: 57 ff). Hence, a better solution would be to consistently replace the misleading `<u>` element by `<turn>` instead, as was done in the DART annotation scheme.

This also leaves researchers with a choice to possibly use the utterance as a concept that reflects structural units at a level below the turns, which was the approach taken in the recommendations expressed in Leech et al. (2000), where the corresponding element name was `<UTT>`, written in all uppercase. The same approach was adopted for the SPAAC project (Leech & Weisser 2003), where the all lowercase tag `<utt>` was employed. However, in the course of the SPAAC project, it proved that the `<utt>` element was in fact redundant because separate, mainly syntactically motivated, lower level element names were more useful in establishing functional units. These had previously often been subsumed under one single *move* label, as in the following example from the "HCRC Dialogue Structure Coding Manual":

> F: Yeah, that's what I thought you were talking about. (Carletta et al. 1996: 5)

The move, as seen here, corresponds more to the general concept of utterance discussed above, but also with the additional communicative function, as defined by Taylor and Carletta (1996: 10):

> A move is a piece of continuous speech by one participant in pursuit of a particular communicative intention. […] Turns […] are also made up of moves, […].

This definition makes it clear that the move is considered a functional, purposive sub-unit of a turn. However, as we will discuss in more detail in Chapter 4 below, according to the classification adopted here, the above example of a move would in fact consist of two functional units, which are here marked off orthographically by a comma. Perhaps this tendency towards 'lumping together' various distinct units is also what has previously lead to problematic definitions of the utterance, as e.g. the one used on the Verbmobil project.

1. An utterance corresponds to a clause;
   it must contain a finite verb. (Alexandersson et al. 1997: 17)

As we shall see later on, this definition would effectively create a segmentation problem, since we could then not treat some of the short unit types, which might not contain a finite verb, but nevertheless constitute a whole turn, as valid independent units, something that Alexandersson et al., despite the original definition given above, also seem to have recognised, because they do list certain types of short units under their exception to the above rule (ibid.: 18).

The TEI also specifies a number of *global* attributes for all elements, namely `id`, `n`, `lang`, and `rend`. The `id` attribute is a *unique* identifier for an element, and thus it may only ever occur once in a single document, whereas the remaining three are non-unique, and can therefore be used as many times – and at as many different levels as necessary – within one and the same document. The `n` attribute is generally a number (although it can also be a name), the `lang` attribute specifies either a particular language or writing system, and the `rend` attribute is supposed to be used for declaring a specific type of rendering, not for general formatting purposes, but to enable a faithful rendering of an original written text element, and is thus not of direct relevance here.

As already shown in 2.2.2, out of these attributes, the `id` and the `lang` attribute appear in the `<dialogue>` element in our corpus data in order to specify the relevant features for a given dialogue document that also make it possible to load particular analysis resources dynamically. Some of these resources will be covered in slightly more detail in Section 3.3. The `id` attribute, however, here only makes sense in conjunction with the newly introduced `corpus` attribute where, taken together, the two represent a unique identifier for the individual dialogue. For pragmatics purposes, perhaps the most important addition to the list of potential attributes at a lower level, though, would be one that reflects the function of each unit, i.e. its speech act, which is referred to by the attribute name `sp-act`. Further useful attributes will be discussed in the respective sections below, in conjunction with the automated annotation system, and once the relevant features have been identified.

In contrast to the TEI recommendations for spoken language, the DART annotation scheme introduces the following hierarchical (non-empty) elements: at the top level, there is the `<dialogue>` element, within which are nested the individual speaker `<turn>s`; each `<turn>` can contain one or more sub-elements that reflect their 'syntactic' classification, namely – in alphabetical order – `<address>`, `<decl>`, `<dm>`, `<frag>`, `<imp>`, `<no>`, `<q-wh>`, `<q-yn>` and `<yes>`. The discussion of their exact specification will be deferred until Chapter 4.

2.3 Problems and specifics in dealing with spoken language transcription

In transcribing spoken language, it is sometimes not enough to purely represent the words that occur in a dialogue in the same way that we would represent them in an ordinary transcript, where it may only be relevant what was said, rather than the way in which something was said. This is why researchers who investigate spoken language have developed various means of representing these additional features in ways that extend the type of normal everyday orthography employed in e.g. keeping the minutes of a business meeting or producing a basic classroom transcript. A comprehensive overview of standard practices and recommendations for representing dialogue on different levels is given in Leech et al. (2000), on which many of the aspects discussed here are based. I will mainly consider those features here that are, or potentially could be, relevant to analysing and annotating pragmatic data. In the process, we will also consider the pros and cons of various approaches and their suitability for catering for different aspects of computational analysability and human readability.

2.3.1 Issues concerning orthographic representation

It is important to note here that, in general, almost all researchers, unless they do research for phonetic purposes, tend to stick to the orthographic representation of spoken language referred to above, despite the fact that this cannot usually reflect the whole wealth of detail present in any spoken utterance. Although it may not immediately be obvious, the motivation for this is that an accurate phonetic representation is in fact very difficult to understand when written down, partly because of the high degree of variability that exists, not only between different speakers, but also within one and the same speaker's productions. Apart from this general type of variability, spoken language also exhibits various features, such as *contraction, assimilation*, *elision*, and the occurrence of *weak forms*. These characteristics may change the appearance of each individual word from the expected *dictionary* or *canonical form* (cf. Leech et al. 2000: 17 & 19) to something that is highly context-dependent, potentially difficult to understand, or may easily be confused with a different word.

The highly frequent co-ordinating conjunction *and* may serve as a straightforward example here. In its unreduced dictionary form, this is normally represented as /ænd/. However, it actually rarely ever occurs in this form, but mostly in its *weak form*, where it is usually reduced to either [ənd], [ən], or simply [n̩]. Since the prepositions *in* or *on* may equally well be reduced to the latter two weak forms, it should not be very difficult to see that it actually makes sense to use the orthographic representation form *and* as a kind of *disambiguated* representation of the word, rather than opting for the full phonetic detail.

The orthographic representation of spoken texts, however, is not unproblematic. This begins with the issue of whether to capitalise specific words or not. Of course, in written language, – at least in those languages that have an alphabetic script based on the Latin or Greek character sets – we have conventions that dictate to us not only to capitalise proper nouns, but also words that belong to any other PoS category when they appear at the beginning of an orthographic sentence. This convention is pure *redundancy*, as generally the end of a prior sentence is marked by a punctuation mark, anyway[8], and in fact may lead to issues of ambiguity that affect those word forms that have the potential to both represent proper nouns and other PoS categories, such as *Smith* vs. *smith* (proper noun or ordinary noun) or *Tailor* vs. *tailor* (proper noun, ordinary noun, or verb). The approach taken here is therefore to avoid this redundancy and only capitalise proper nouns, such as the names of persons or places, in English. Obviously, for other languages one may want to process, similar or different rules might need to be employed to reflect the different properties of each individual language. For French, essentially the same rules apply as to English, while for German, one would certainly have to respect the convention that all nouns ought to be capitalised, which in fact makes their morpho-syntactic recognition easier, and can thus simplify the identification of syntactic structures.

There will be more to say about syntactic units and punctuation later, but for now, we will continue to concentrate on some of the other issues that affect lexical entries and their associated representational and functional properties. Amongst these entries are compounds, which – especially in English – may cause distinct problems with regard to their orthographic representation, as they can often be spelt in different ways, either joined completely without any intervening space or hyphen (*icecream)*, with a space in between the two components (*ice cream*) or joined by a hyphen (*ice-cream*). Depending on the approach to analysis chosen, this would either make it necessary to store each of the three forms in a lexicon

8. Spanish even marks the beginning of specific types of sentences, such as interrogative ones, by the same punctuation character that marks the end.

or to write special functions to identify compounds during the syntax-oriented processing phases. A third alternative that could simplify corpus use after the compilation phase would be to employ one single standardised form that all the individual different occurrences could be normalised to. If necessary, the normalisation, including the original form, could then be indicated via an empty element, e.g. something like `<correction orig="…" />`, which is a practice followed in preparing the data for this book when obvious spelling errors were identified. This would then also make it possible to store only one single canonicalised word form in the lexicon and use it in any potential searches conducted on the data. In the methodology adopted here, however, compounds do not present much of a problem, due to the simplified approach to syntax employed, as we will see in later chapters.

What does represent more of an issue in our data, though, are *quasi-lexical vocalisations* (cf. Leech et al. 2000: 21), which may either represent *interjections* (e.g. *oh, ah*), *backchannels, discourse markers*, or *response tokens* (e.g. *aha, uhu*), or features that are generally referred to simply as *fillers* or *filled pauses*. As there are often many different forms that only really constitute an approximate transliteration of the real phonetic values, anyway, these ought to be identified in each corpus before converting it to the analysis format, canonicalised if necessary, and then categorised according to their function, and often added to the appropriate lexica or analysis routines that need to be able to deal with them.

Although these vocalisations tend to have little semantic content, or even none whatsoever, they may play an important role in a dialogue context, depending on where exactly they occur. For instance, fillers at the beginning of a turn frequently indicate hesitation in responding to a question, while those inside the turn or syntactic unit signal other planning issues. Unfortunately, though, such items often tend to be represented in different ways in different corpora, with sometimes specific local preferences. Thus, for instance, in British data, one often finds the representation forms *em, er*, or *erm* for filled pauses, while American corpora exhibit a preference *um* or *uhm*. Whilst these forms are unproblematic, and either simply need to be standardised for comparability or treated as identical by annotation routines, apparently, at least according to some of the examples cited in Adolphs and Carter (2013: 56 & 58), the transcribers of the Limerick Corpus of Irish English (LCIE) used the rather more problematic variant *am*. This form, of course, can easily be confused with the 1st person singular form of BE, and may thus seriously skew both frequency lists and automated analysis routines that rely on accurate PoS identification.

The issue of identifiability can also become more problematic if many variant forms exist that may represent somewhat similar or rather different features at the same time. This may e.g. be the case for the minimal response that is variably represented as *aha, uh-huh, huh-huh, mhm*, or *mm-hmm* in the corpora I have worked with so far, and where at least those forms that include the grapheme sequence <huh> may represent genuine issues for any attempts to standardise or detect them automatically, as this sequence can of course also have the function of a querying tag that expresses surprise or doubtfulness when occurring on its own.

Spellings and numbers may also pose specific problems for representation in dialogue corpora (cf. ibid: 17–18). While numbers were not spelt out in much of the data used for here, as is sometimes done to remove ambiguity between semantically different combinations of them, such as e.g. dates or amounts, at least in the Trainline data, they necessitated special treatment. The reason for this was that the data also contained private information in numerical form, such as address or credit card details, which absolutely required anonymisation, and this procedure is much easier to carry out on digits than on a verbal representation of numbers, although it is not unproblematic then, either. For instance, some of the departure or arrival times contained in the data were automatically converted into times that did not make sense in the context, such as e.g. *half 9 in the afternoon*, which had to be manually corrected to something more sensible during the pre- or post-editing phases.

In general, the widespread practice of representing numbers as words does not really make sense because it actually interferes with treating numbers as such, and makes it more difficult to accord them the right kind of treatment (for instance by 'lemmatising' them) in any kind of n-gram or basic frequency analyses. One of the reasons for doing so in the first place has always been the claim that it is important to faithfully represent how something was said in corpus data, but this is questionable because the exact phonetic representation is (a) not relevant in many contexts and can (b) just as easily be indicated in comments added in the form of empty XML elements. For the disambiguation of years, for instance, it is not too difficult to specify their respective contexts as they tend to occur in fairly fixed patterns, such as deictic PPs beginning with *in*, etc., their length tends to be restricted to either 2 or 4 digits, etc. Rendering them in spelt-out form, on the other hand, presents serious issues of processability, due to various formats that transcribers may employ, sometimes representing compound numbers variably as e.g. *seventy-three, seventy three*, or even *seventythree*, where frequency counts tend to be affected more and more the longer the number becomes. In addition, some realisations, such as [əʊ] for 0 (*zero*) can easily be confused with spelt-out letters, especially if they do occur in combinations with them, for instance in British postcodes. Perhaps the only place where representing numbers in their numerical form may be a real issue is when filled pauses occur right in their middle, something that is not infrequent, especially in unplanned dialogues.

In order to protect other personal information, such as surnames, both as complete words or spelt out, further anonymisation steps were sometimes necessary to mask them, so that, in the first version of the SPAADIA (Trainline) data, they were turned into something like *{surname}%18%15%7%11%10%7*, with *{letter}%18 {letter}%15 {letter}%7 {letter}%11 {letter}%10 {letter}%7* as the spelt-out version, where each combination of a percent sign plus number constitutes a randomised numerical representation of the letter that remained constant throughout the dialogue. This was done so as to be able to e.g. identify *echoes*, repetitions of sequences of letters repeated for confirmation purposes (cf. 6.3.2). Anonymisation for the numbers, as discussed above, was done in almost the same way, again for the same purpose. Of course, all the relevant analysis modules later had to be made aware of these special markings and their functions. In the final version of the SPAADIA corpus, this anonymisation format was changed to an XML-based format using empty `<anonym />` elements to ensure comparability with other corpora.

Another issue is that of incomplete words, which often indicate hesitation phenomena, and either signal the complete breaking off of a textual unit, that is the choice of the speaker not to complete an utterance, or a potential correction, due to the speaker having changed their mind about using a specific expression in a given context. A common practice (cf. Leech et al. 2000: 19) is to mark the incomplete word by a special marking that immediately follows it. In this way, any program that processes the data can then choose to ignore this word. This marking often takes the form of an asterisk (*) or dash (-), which may not necessarily be the best choice, though, as a dash can be confused with a hyphen, as e.g. in the phrase *pre- and post-processing*, where the hyphen after the *pre* marks a deliberate indication of a partial word, whereas the asterisk could be confused with a *quantifier* or *wildcard* character that could potentially interfere with the computational processing. The choice made here is to use three dots (…) following the incomplete word instead, which is a familiar abbreviation for an ellipsis in written texts and will therefore probably intuitively make sense to anyone trying to interpret the data. This then also leaves the option for retaining the hyphen for indicating words deliberately left incomplete. In accordance with the two separate position-dependent functions of incomplete words described above, the DART analysis routines either skip over an incomplete word or can mark the complete textual unit with the speech act *abandoned*.

At this point, it may be necessary to stress again that, obviously, many of the issues described above do in fact have nothing to do directly with pragmatic aspects of the dialogues themselves, but rather only need to be dealt with if we want to handle and disseminate corpus data in a form that is globally usable, and also caters for issues of data protection properly.

2.3.2 Issues concerning prosody

The problems discussed above essentially concerned the representation of data on the segmental level. However, specific issues at the suprasegmental level are at least of equal importance, despite the fact that they are often completely neglected by spoken corpus compilers. The relationship between the words represented in standard orthography and their prosodic contexts, which determine their more or less 'exact' meanings in the dialogue, cannot simply be ignored. If we take prosody to also encompass the timing and chunking with which particular speech events are delivered, then some way is needed to somehow indicate what we could term the 'degree of prosodic cohesion' in such sequences.

In fact, attempts to do this already exist as conventions for using punctuation in written language, although so far these are less than perfect, especially with regard to the use of minor punctuation marks, such as the comma or semi-colon (cf. Chafe 1995: 57). Despite the obvious need for such conventions, there have been – and still seem to be – some researchers who assume that the use of punctuation marks or similar devices to delimit units in spoken discourse corresponds to an undue interpretation of the transcribed materials.

> […], one could still argue that the use of a transcription system that builds upon graphic punctuation symbols does not really capture the way words and expressions cluster together in spoken language ([…]). Even more critically, one could argue that the use of such devices forces us to think of such chunks as sentences, rather than providing an accurate representation of how speakers themselves produce language, e.g. as intonationally packaged foci of consciousness ([…]), as rhetorical amalgamations of clauses ([…]), in collaboration with interlocutors ([…]). To reflect such concerns, some discourse analysts exclude from their transcription systems those of punctuation (e.g. period, commas, capital letters) that are used in written language to indicate syntactic structure or closure, or to use such devices to capture aspects of speech production ([…]).
> (Schiffrin 1994: 25)

Such an attitude unfortunately also seems to have been prevalent in the compilation of most of the original data used for analysis here. This is why the analysis approach discussed here, which was originally developed for analysing the Trainline data and an equally unpunctuated much larger set of transactional dialogues from BT, initially did not include any mechanisms for proper handling of such types of prosodic information yet.

As far as the means for rendering such information in dialogue data is concerned, it would of course be quite possible to include a fairly accurate representation of the prosodic facts – at least at those points that actually delimit textual units from one another – in the form of either *Tonetic Stress Marks* or a *ToBI annotation* (cf. Leech et al. 2000: 39–48) inside empty elements in the running text, but as most researchers in pragmatics are probably not used to reading these, and they would also still need to be interpreted, Leech et al. (2000) advocated to resort to a clearly defined system that uses a limited number of punctuation marks to signal this for orthographic representation:

> As to the more general form of transcription, the use of a basic subset of the standard orthography is both normal and desirable. Sentence-initial capitals may be omitted, but, otherwise, normal capitalization and at least full stops tend to be used. This improves readability for the human user and improves processibility for taggers, parsers, and so on. Obviously, it is understood that such standard orthography is, to a considerable extent, interpretative when applied to speech, but its advantages outweigh its disadvantages. The use of punctuation characters other than full stops is an open question, but commas may sometimes have certain advantages as well. In English, for example, using a comma before a tag question is unambiguous and may actually help to identify the purpose of this particular phrase type as communicating a possible request for feedback: e.g. *Two o'clock, is it*. There is also a case for using question marks where the transcriber clearly perceives an utterance as a question. This can be useful especially where the structure of the utterance does not mark it as interrogative. There are many questions in which lack such marking (e.g. *Next week?*), and their import is not clear to a reader who does not have access to the prosodic level of annotation. (19–20)

In contrast to the above recommendations, however, I propose a simplified, 7-way distinction that employs empty 'phono-pragmatic' punctuation (`<punct />`) tags that include type information, where the attribute value

1. `comma` exclusively acts as a separator for list items,
2. `stop` signifies final 'declarative' intonation/completeness,
3. `query` marks different, relatively neutral, forms of 'interrogative',
4. `exclam` indicates an exclamatory nature,
5. `unsure` may mark certain types of 'incredulity' that are often expressed in form of a fall-rise contour,
6. `level` signals a non-final and non-interrogative prosody that indicates a 'trailing off' or 'please hold' pattern,
7. and `incomplete` an interrupted, and frequently also uninterpretable, unit.

Out of these seven values, 5 have been added to the corpora used for this book, with the exception of `comma` and `unsure`. Where and how these attributes can be employed to enrich the transcriptions and aid the analysis will be illustrated later.

Due the absence of punctuation marks to rely on in the identification of textual units in the original data, the DART approach uses two different mechanisms for handling the individual functional units within a turn separately. In those cases where it is syntactically more or less unambiguously possible to identify splitting points, the units will be split off and annotated automatically. Where this is not possible, the dialogue can either be *pre-processed* so that the relevant units appear on separate lines within the turn and `<punct />` tags added where applicable, or units that have not been correctly separated, and thus been marked up within a single syntactic element, can be split manually and hand-corrected in the *post-processing* phase. Further details concerning this will be discussed in Chapter 6.

Other prosodic issues that do not necessarily divide textual units, such as filled pauses, on the one hand may make it more difficult to analyse spoken language as a chain of coherent words or sentences, as they tend to disrupt the linear processing order one is so used to from written language. At the same time, their very occurrence – and potential importance for understanding spoken language – requires us to find a way to actually record them with an appropriate level of detail, as well as to try and make sense of their communicative function, either in separating chunks or phrases from one another, signalling opportunities for speaker change, or merely indicating non-responsiveness. Some unfilled pauses, though, will simply be there in the stream of spoken language due to pure physical necessity, since speakers need to stop for breath – although they will usually try to do so in places that allow them to control the aforementioned chunking.

As we have seen above, for *filled pauses* (e.g. *er, em, erm, um*), essentially the same principles apply as to the quasi-lexical vocalisations discussed above. And because pauses may be so important, the TEI guidelines for spoken language also contain a specific empty element `<pause />` for *unfilled* ones, which also envisages attributes for speaker identification `(who), type` – with possible values `long, short, medium` – and duration `(dur)`, if it has been timed (Johansson 1995: 83). The problem with assigning the `type` attribute, though, is that the communicative effect of the length of pauses may to some extent be situationally dependent, e.g. usually have to be seen as relative to the given rate of speech of a speaker, whether pauses may mark opportunities for a change in speaker turn, or just represent hesitation phenomena. Thus the arbitrary labels given above may really be more impressionistic than useful. Edwards (1993: 24) provides the following comment on this:

> Even if a pause is explicitly quantified, the classification of it as “short” or medium varies with research purpose. Researchers concerned with turn-taking smoothness may consider a “short” pause to be .5 seconds, while those interest [sic] in information packaging may consider it to be .2 seconds.

While the above comment makes it clear how important it is to record the length of pauses in general, a potential cause of confusion in many different annotation systems that are prior to, or not compliant with, the TEI is that rather different and – to my mind – fairly inconsistent conventions may also be employed in indicating the pauses themselves. Thus Sacks/Schegloff/Jefferson (1974) mark timed pauses in tenths of seconds in round brackets (e.g. *(.3)*) and untimed ones by what they refer to as a "long dash" (ibid.: 732), typographically an *m-dash*, but not inside round brackets. Chafe, on the other hand, uses two dots (..) "for a short pause", three dots (…) "for a long pause" and/or[9] "(optional) timing", presumably also indicated in tenths of seconds (1993: 43). Here, apart from the absence of any timing information for the short pauses, we can again observe an inconsistency in the representation, but one which is even more confusing in its application since three dots, as stated before, are conventionally used to signal an *ellipsis*.

To eliminate any confusion, and also make it possible to treat filled pauses as words, a dual approach is taken here, with unfilled pauses marked up as empty elements that may or may not contain a *length* attribute – depending on whether the approximate or exact length is known –, while filled pauses are generally canonicalised as much as possible and the different variants used are listed in the *generic lexicon* (cf. 3.3.1), although it would of course also be possible to incorporate a separate filler lexicon into the system. Apart from being more explicit, the empty element approach for unfilled pauses also theoretically makes it possible to include additional attributes that could render the impressionistic level, or any other 'comments' a researcher might want to add as to the significance of a particular pause, at a later stage of the analysis, and without disturbing the processing mechanism that can simply filter out any unwanted detail.

2.3.3 Issues concerning segmental and other features

In order to record other potentially relevant features of spoken language that do not form part of traditional orthography, various types of annotation formats have been devised, most notably and extensively perhaps within the CA tradition (cf. Atkinson & Heritage 1984). Many of these annotation types, though, deal with aspects of speech, such as speaker gaze, elongated vowels or syllables, emphasis, or subjective marking of intonation, which do not form part of the analysis strategies discussed here, although they are often well worth analysing in more sociologically oriented contexts. We shall return to some of these issues, however, in our brief discussion of multi-modality in Section 2.3.5.

While a substantial number of the types of annotation that do not use TEI conformant tags have the advantage of signalling what goes on in a conversation

9. Although he writes "or", he refers the reader to the following article in the same book, by DuBois, Schuetze-Coburn, Cumming & Paolino, where the timing information is appended after the three dots.

with a certain degree of immediacy because the features that do occur are marked more or less exactly where they do occur, they have the drawback of 'cluttering up' the transcription to an extent that sometimes makes it relatively unreadable. This is especially the case if one is not used to the conventions and may need to obtain the relevant documentation first. Furthermore, because there is no uniform marking of special features – such as enclosing all in round brackets –, as we have seen in the examples of pauses above, automatic processing of texts is made more difficult because all different conventions have to be specified in a computer program that may have to remove these annotations for specific processing tasks or to create a less 'cluttered' view of the data. Here is an example from Sacks, Schegloff & Jefferson (1974: 14) that illustrates this problem, at the same time highlighting other difficulties such a detailed kind of transcription introduces, including some of the issues discussed in earlier sections:

1 J: Oh I could drive if you want me to,
2 C: Well no I will drive (***I don' m//in'***)
3 J: ***hhh***
4 (1.0)
5 J: I mea***nt*** to offah.
6 (16.0)
7 J: Those shoes look nice when you keep on putting stuff on 'em.
8 C: Yeah I 'ave to get another can ***cuz cuz*** it ran out. I mean it's a//lmost ***(h)*** ou(h)*t=
9 J: Oh***:::***ah*e hh heh=
10 C: =yeah well it cleans 'em and keeps // 'em clean.
11 J: Yeah right=
12 C: =I should get a brush too and you should getta brush 'n // you should-* fix your hiking boo//ts
13 J: Yeah suh::
14 J: my hiking boots
15 C: which you were gonna do this weekend.
16 J: Pooh, did I have time this wk- well***::***
17 C: Ahh c'mon=
18 J: = wh'n we get- (uh::kay), I haven't even sat down to do any- y' know like
˙***hh*** today I'm gonna sit down 'n read while you're doing yur coat, (0.7)
do yur- hood.
(my emphasis; line numbers added)

The first thing to note in line 2 is that round brackets are used, according to the conventions (ibid: 733) in order to indicate that the transcriber was *unsure* about the exact wording. Yet, despite this insecurity about the exact wording, a place of *overlap* between speakers C and J is marked by a double forward slash before the vowel inside the word *mind*, which is already difficult enough to decipher because of the apostrophe at the end. The use of this apostrophe is not documented in the article, but most likely seems to mark elision of final consonants or initial *h*s, or to indicate *reduced syllables* in general. There is, however, also an alternative – and possibly easier to understand – way of marking overlap, which is to mark the start of an overlap by an opening square bracket ([), while the end is indicated by the corresponding closing bracket (]; ibid.: 732). In the DART approach, though, again the option of empty elements is chosen over any other way to mark overlap, as this not only allows to mark start and end via attributes, but also to potentially add further attributes later, which could, for instance, signal the overlap between different participants in multi-party dialogues more clearly.

In line 3, the triple *h* indicates *audible breathing* out, while the double *h* in line 20 – preceded by a raised dot, which is difficult to see – indicates breathing in. Presumably, the number of *h*s – rather arbitrarily – marks the length of this breathing period.

The underlining in line 5 represents "stressing" (ibid.: 733), but indicating *stress* on a consonant cluster without involving at least a vowel as a syllable nucleus is nonsensical, and it can therefore be assumed that in reality the whole monosyllabic word *meant* is stressed in this case.

In line 8, we find a non-canonical, impressionistic, rendering of *because* as "cuz". This is actually repeated as a *restart* or *dysfluent repetition* (cf. Leech et al. 2000: 35), which would lead one to assume that

a. there would be a pause between the two repetitions, and
b. that the explanatory unit is also preceded by a pause, as it seems to occur as some kind of an afterthought.

None of this is in fact indicated in an otherwise over-elaborate annotation system. The *h* enclosed in round brackets normally indicates strong *exhalation* or "explosive" *aspiration* (ibid.: 733). Yet, from a phonetic/phonological perspective is somewhat difficult to imagine occurring in the positions indicated here, especially in the word *out* and preceding the final plosive, with no indication whatsoever to be found in the article about the meaning of the asterisk. Atkinson and Jefferson, however, when discussing Jefferson's transcription system – which seems to provide the basis for the system discussed in Sacks/Schegloff/Jefferson – state: "Asterisks are used in a more ad-hoc fashion to indicate particular phenomena discussed in the text" (1984: 163).

Line 9 shows an example of a supposedly *elongated syllable*, where the number of colons represents the length of the elongation. Apart from the representation containing three colons, again appearing somewhat more arbitrary than realistic, the whole line is indecipherable and does not seem to bear any phonetic basis.

According to Sacks, Schegloff and Jefferson “a short dash indicates a ‘cut-off’ of the prior word or sound” (1974: 733). This seems to apply to their own example in line 18 in “any-” and possibly “wk-” in line 16, although the absence of any vowel letter in the latter is distinctly odd. However, it does not seem to have any basis in “should-” on line 12 or in “yur-” on line 18, where it rather seems to signal something similar to the apostrophe we discussed before. How this issue is handled here has already been described in 2.3.1 above.

As shown in the examples provided above, the coding system proposed by Sacks, Schegloff and Jefferson provides ‘solutions’ for dealing with a wide variety of specific problems in recording information pertaining to spoken language. However, it does so inconsistently and at the expense of *readability* and *computational tractability*, two of the three principles in annotation advocated by Edwards (1993: 21 ff.). The third one is *category design*, which I shall discuss later in conjunction with establishing suitable taxonomies for syntactic and pragmatic annotation. In lieu of a system that combines so much information into a ‘pseudo-orthographic’ representation, it would probably be better to choose some form of interlinear format where there are at least separate levels for orthography and phonetic detail, or to represent this information in separate files that can be displayed in parallel when necessary.

As illustrated earlier, the recommendations made by the TEI fulfil the requirement of computational tractability to a much larger extent than the CA conventions because they are more consistent, and the use of tags and attributes provides for a uniform system of annotation that is at the same time easily extensible if XML is used. Readability is also improved because words in a text are not normally interrupted by annotations, although the potential drawback here may be that it is not immediately visible where certain features, such as e.g. *prosodic* phenomena, occur inside the word. This, however, could to some extent be rectified by providing attributes inside start tags or empty elements that point towards the position inside the relevant word where the phenomenon is assumed to occur, provided that such an exact specification is possible at all and not more or less arbitrary, as in some of the examples encountered above.

The use of empty tags to signal special features that accompany spoken language, but are not actually considered verbal themselves, is already envisaged in the TEI proposals (Johannson 1995: 88 f.; TEI guidelines P4, REFTAG.pdf: 968).[10] Amongst those are:

1. `<vocal>s`: vocal noises made by one or more of the participants, such as laughter, coughing, etc.; associated attributes: `who`, for speaker identification, and `desc` for a description of the sound;
2. `<unclear>`: for indicating material that is not clearly identifiable; associated attributes: e.g. `reason`, stating why something cannot be transcribed, maybe due to background noises, low volume, etc., or `agent`, if the agent who caused the problem is known;
3. `<event>s`: other noises or occurrences not associated with any verbal or communicative action of one of the participants, e.g. ringing doorbells or telephones, etc.; associated attribute(s): usually only `desc`, unless the event can clearly be attributed to a participant or third party, in which case `who` may also be applicable;
4. `<kinesic>s`: non-vocal gestures made by one or more participants, such as nodding, winking, shaking one's head, etc.; same attributes as for vocals;
5. `<shift>`: used for paralinguistic features, e.g. marked changes in or deviations from vocal or delivery characteristics; associated attributes: `who`, `feature` and `new`; `<shift>` without any attributes marks the end of the special state

Examples of (1) in our data would include such attribute values as `laughter`, `breath`, etc. Instead of using the TEI attribute label `desc`, though, the label `content` is used, which also applies to the `<backchannel />` element, thereby signalling their common potential to contribute to the verbal interaction on the semantic or semantico-pragmatic level. If the source data used does not include any information regarding the person producing these features, no speaker attribution can be included, but the default assumption is probably in most cases that both types of content can usually be attributed to the party whose turn it currently is not, at least if there are only two interlocutors. In multi-party interactions, it may be necessary to include a `who` attribute to indicate which speaker has provided the contribution.

The `<unclear />` element is used to represent stretches of speech that cannot be identified as to their content, just as foreseen in the TEI guidelines. However, the main attribute used here is `length`, generally given in syllables,

10. Downloadable as part of a zip archive from <http://www.tei-c.org/Guidelines/P4/teip4.zip>

although a `content` attribute may also or alternatively be used to specify assumed content, which may then at least support the syntactic categorisation. The `length` attribute is also applicable to `<pause />`, if any information as to the concrete length exists, in which case the value is given in seconds.

Feature (3), `<event />`, occurs in the data and is qualified by a `type` attribute, and may e.g. refer to system-related information, such as `tape cuts off`. The `<overlap />` element, in contrast, has the `pos` (position) attribute, which only exhibits the binary values of `start` or `end`, and should also be identified by an `n` attribute that 'links' the overlapping passages for all concurrently overlapping interlocutors.

Some of these elements do not play a major role in the current analysis methodology, but are partly recorded for the sake of completeness and partly because they may be included in the methodology later on, once their potential relevance to different aspects of the analysis has been researched further. For the moment, most of the analysis routines simply 'skip over' them.

Another major problem in dealing with spoken data is that of *disfluencies*. Amongst those are *incomplete words*, *false starts* – where a speaker breaks off from what they have been saying and rephrases the utterance in a different way, correcting an error –, and *restarts* or *repetitions*. Leech et al. (2000: 22) suggest to mark these in a way similar to the following, using the TEI tag `<del>`:

<del type=truncation>s</del>see

<del type=repetition>you you</del>you know

<del type=falseStart>it's</del>he's crazy

However, while marking this may potentially be suitable for the finalised version of a corpus that is solely used for corpus analysis or machine learning purposes, a system such as the one introduced here needs to incorporate a certain kind of *robustness*, the ability to recognise such disfluencies, as well as other ungrammatical features of spoken language that lead to syntactic *fragments* (cf. 4.3.9 below) – such as the absence of auxiliaries, etc. –, which is why it may not necessarily be advisable, and therefore such hesitation phenomena are not indicated via any elements, but instead only recognised and handled by the appropriate analysis routines, although the disfluency is registered as an interactional feature in part of the annotation.

2.3.4 Issues concerning sequential integrity

We have already seen in earlier examples that the issue of speaker overlap is of potential importance in representing spoken interaction faithfully. So far, in many of the corpora I have worked with, it has been common practice to indicate such overlap through a variety of markers, such as the forward slashes (//) or square brackets ([or]) in the CA conventions, or through even more explicit devices, such as in the *International Corpus of English* (ICE) data, where the extent and number of multiple overlapped passages can be indicated in SGML tags (cf. Nelson 2002: 5). However, one thing that appears to have been ignored in the design of such markup schemes is the fact that not all overlaps have the same status and function, and that it may therefore be necessary to apply different forms of representing the different shapes they may come in.

Current schemes generally, for instance, represent overlapped content in separate turns for each speaker, thereby implicitly creating the impression of a sequence of all overlapped content uttered by a particular speaker, in the order that the content has been uttered. However, not only do some forms of overlap in fact represent *backchanneling* behaviour, i.e. are by no means intended to interrupt the interlocutor and take over the current turn, but such practice often also violates the representation of what we might term the '*sequential integrity*' of the interaction. What I mean by this will hopefully become clearer through the following example, taken from the (reformatted) Trains 93 data (see Section 3.1 for more details).

(1) <turn n="5" speaker="s_MF">
oh
i should tell you that you <overlap pos="start" /> can <overlap pos="end" />
only um <pause /> pull <pause /> 3 loaded boxcars at a time <punc
type="stop" />
</turn>
<turn n="6" speaker="u_ML">
<overlap pos="start" /> yes <overlap pos="end" />
oh
<pause /> okay <punc type="stop" />
<pause /> so <punc type="level" />
why don't we go from Avon <pause /> to Bath <pause /> get <pause /> um 2
boxcars <punc type="query" /> (d93-17.2)

In the above example, we can see that the overlap occurs relatively early on in turn 5, and speaker s_MF still continues her turn until she has completed a full declarative syntactic unit. Thus, both speakers do not overlap for a lengthy period of time, and the overlap does not appear to constitute any attempt on the part of speaker u_ML to take over the turn, either. In addition, the 'minimal response', *yes*, by speaker u_ML does not really constitute any logical response to the information she has just received, especially as it occurs before the major part of the propositional content has actually been conveyed. It should therefore rather be assumed to be a backchannel, especially because the surprise marker *oh* in fact does express a genuine response to the limitation expressed by s_MF, and which u_ML seems to not have been aware of before having received this information. Having 'digested' this information, he then goes on to propose an alternative solution that is perfectly coherent with these facts. The original representation format, however, separates the genuine response from the information it responds to, so that any software that processes turns sequentially would first have to be made aware of the fact that the first unit in turn 6 in fact needs to be ignored when investigating initiation–response sequences, which would be no trivial matter because it would have to keep a record of the true position (and potentially also length) of the overlapped sequence to determine whether the unit represents a backchannel or proper response, as it is perfectly normal for interlocutors to occasionally be talking at the same time, too, especially if this occurs towards the end of the 'initiating' speaker's turn. In some cases, such sequences may also signal attempts at the 'responding' speaker's to try and 'usurp' the turn, in which case the previous unit by the prior speaker may end up being incomplete. If the overlap happens to occur around the turn change, there may be no option but to represent it in the form we have just encountered, but if the overlap really does constitute a backchannel, as in Example (1), then a better alternative is to include it in the other interlocutors turn in the form of an empty `<backchannel />` element, as in the following, where Example (1) is appropriately rewritten.

(2) <turn n="5" speaker="s_MF">
oh
i should tell you that you can <backchannel content="yes" /> only um
<pause /> pull <pause /> 3 loaded boxcars at a time <punc type="stop" />
</turn>
<turn n="6" speaker="u_ML">
oh
<pause /> okay <punc type="stop" />
<pause /> so <punc type="level" />
why don't we go from Avon <pause /> to Bath <pause /> get <pause /> um 2 boxcars <punc type="query" />

In the above example, the actual response to the information in turn 5 is now clearly accessible to any program processing the data sequentially, and the coherence relations between the two turns also become immediately apparent to the corpus analyst.

2.3.5 Issues concerning multi-modality

In recent years, part of the focus in spoken corpus linguistics has begun to shift somewhat towards moving away from spoken corpora that purely take audio information into account. Some researchers are now claiming that, apart from having access to accurate transcriptions of spoken interaction, it is equally important to take aspects of non-verbal communication (NVC), such as supporting gestures or speaker gaze, into account:

> NVC comprises individual gestures or sequences of more discrete and structured gestural episodes which communicate messages between individuals involved in a conversation. (Knight et al. 2009: 2)

However, although increasingly research is being carried out on multi-modal data, as yet, it is not always clear to what extent exactly many of the features actually do contribute to conversations. As Knight quite rightly points out:

> […], while the addition of video and audio, as seen in current multimodal corpora, can arguably allow for a richer description of some of the extrinsic contextual features of interaction, it is difficult to fully quantify, qualify and analyse *all* such features in a meaningful way. (Knight 2011: 186)

Thus, while pragmatics-based research into textual and contextual features may already be fairly well defined, multi-modal approaches still remain more problematic. Perhaps this is partly also due to certain mis-interpretations of what multi-modality actually entails. Thus, Adolphs and Carter write that:

> […] spoken interaction is essentially multimodal in nature, featuring an interplay between textual, prosodic, gestural and environmental elements in the construction of meaning […]. (Adolphs & Carter 2013: 12)

Although part of the above statement is no doubt true, the problem here is that Adolphs and Carter appear to be guided rather too much by the fact that spoken corpora are generally rendered orthographically (cf. Section 2.2 above), which they refer to as "textual elements", but fail to understand that the textual level in the context of spoken language in fact already subsumes both verbal and prosodic elements. Thus, an orthographic representation of spoken data that excludes information about prosodic features is by definition incomplete, as the discussion in previous sections, and especially my critique of avoiding 'punctuation', should already have made clear.

In addition to such unfortunate misconceptions on the theoretical side, the development of multi-modal corpora still suffers from a variety of issues on the practical side, as discussed by Knight (2011: 183–184), most importantly perhaps that they tend to be highly domain-specific and that the conventions and means for representing them have not been suitably standardised, which currently makes it difficult to compare and draw generalisations from them. Furthermore, while having access to audio and video in addition to transcripts of corpus data may provide more flexibility for researchers to investigate different issues related to the same data, providing such access poses additional problems in that it generally ties researchers into using dedicated analysis software, which is often either very complex to handle and/or expensive, apart from presenting serious issues in terms of anonymisation (cf. Knight et al. 2009: 8).

In addition, while written or spoken corpora that do not contain audio materials already present ethical or privacy issues in terms of requiring the anonymisation of personal data such as names or addresses of informants, once audio and video are added to create multi-modal data, these issues tend to become compounded. Once audio is included, it may become possible to recognise speakers through their voices, and names or other personal details referred to might need to be masked acoustically to provide anonymity. And while it would be perfectly feasible to anonymise informants' faces using pixelation techniques, this would render multi-modal data less useful because potentially relevant facial expressions would then no longer be recognisable. For a more complete overview of these issues, see Knight (2011: 50–54).

The approach to pragmatic analysis presented here currently does not cater directly for, or take into account, any genuinely multi-modal information, such as head-nods or other non-verbal gestures. However, if such information were required, there are a number of possible steps that would allow researchers to make reference to such features by enriching the data in simple ways. First of all, each dialogue file could be 'linked' to the audio or video resources by adding an attribute for referencing the relevant multimedia file inside the XML container tag. This would enable the researcher to open an appropriate multimedia program and verify or reference the multimedia information. At the next level, each speaker turn could receive a time stamp attribute in order to be able to navigate better through the multimedia data and play back content in parallel to working with the text. Finally, to be able to concordance or calculate statistics on multi-modal events, it would be possible to add empty elements containing timestamps and information about these events, etc., to the relevant positions for the respective units. Future versions of the analysis tool DART may also provide ways for starting up such multimedia programs directly with the relevant files, although dealing with the time stamps directly would require a dedicated software solution or at the very least a scriptable multimedia program.

Having discussed the most important aspects concerning the representation and markup of dialogue data on the computer, we can now turn our attention to considering how these can most efficiently be applied to language data.

CHAPTER 3

Data, tools and resources

3.1 Corpus data used in the research

In the research for this book, primarily data from three corpora (SPAADIA Trainline, Trains 93, and the Switchboard Corpus) was used for different purposes, and to illustrate diverse phenomena relevant to the pragmatic analysis of corpora. In order to ensure relative comparability between the different sets of data, an equal number of 35 dialogues was randomly selected from the two larger corpora drawn from, as the smallest one only consists of this number of interactions. In the following, I shall give some brief descriptions of the corpora and specific issues that occurred in converting the data to the DART format, followed by a summary of descriptive statistics related to each corpus. Some of the issues raised here will hopefully also corroborate the points made above regarding the problems in representing spoken data accurately for this type of analysis. The conversion of the data that did not already exist in DART format was followed by extensive manual pre-processing in order to determine c-unit boundaries and add the phono-pragmatic 'punctuation' elements described in 2.3.2, but even the one corpus that already existed in DART format underwent extensive checking and some further modifications, as detailed below.

One slight caveat I need to point out again here, though, is that for none of the corpora I used, I actually had access to the original audio files. This is why it may not have been possible to fix all inconsistencies that existed in the original transcriptions, and I also sometimes had to resort to my own interpretation of the transcripts in order to make some of the decisions for splitting the data into the right units and adding the `<punc … />` elements.

3.1.1 The SPAADIA Trainline Corpus

The SPAADIA Trainline Corpus (see Leech & Weisser 2013) consists of 35 transactional dialogues between one British call-centre agent (Sandra) and unidentified callers, who communicate via the telephone in order to establish train timetable information and carry out bookings based on this information. This corpus constitutes a complete sub-corpus of the SPAAC (*A Speech-Act Annotated*

Corpus of Dialogues) project, carried out by Geoffrey Leech and myself between 2001–2002 at the University of Lancaster. The files had originated from another project carried out in our Department, and were only provided to us in transcribed form, but sadly without any of the original audio materials. As previously mentioned, it was during the course of this project that the first steps in developing the speech-act annotation methodology employed here were taken, and originally some 1,200 task-oriented dialogues comprising the Trainline data and a much larger number of dialogues provided by British Telecom were annotated and post-edited using the fore-runner of DART, SPAACy (Weisser 2004). However, due to copyright issues, we were unable to release the larger part of the SPAAC corpus, which later – at least for some time – became available as part of the Edinburgh OASIS Corpus.[1]

Version 1 of the SPAADIA Corpus was originally published in 2013, using the original SPAAC speech-act taxonomy (see Leech & Weisser 2003). Version 2, which features an enlarged taxonomy and the addition of `<punc />` elements, apart from eliminating a few inconsistencies, was released in 2015. The version of the corpus used here is version 3, which I annotated using the latest version of the speech-act taxonomy, and revised regarding a number of features. The main changes consisted in re-labelling speaker *A* to *Sandra* and *B* to caller_ + dialogue number – working on the assumption that all callers are unique – in order to be able to better distinguish between potential idiosyncratic features used by individual callers in the corpus, a change of the spelt-out number representations to digits, e.g. *2 o'clock* or *1400* – the latter with added pron(unciation) comment `"14 hundred"` –, capitalisation of all forms of address (e.g. *Sir*), the addition of inline `<quote>` … `</quote>` elements to be able to recognise reported speech better, etc.

3.1.2 The selection from Trains 93

The Trains 93 selection of 35 files was randomly chosen from the Trains 93 Dialogues[2] (Heeman & Allen 1995), which, in turn, form part of the larger TRAINS Dialogue Corpus.[3] This corpus was collected at Rochester University in the US over a number of years in order to investigate dialogue structure, and ultimately design dialogue systems that could perform collaborative problem solving. In the simulated dialogues of the corpus, two interlocutors, based on various prompts, need to collaborate in shipping commodities, such as bananas and

1. See <http://groups.inf.ed.ac.uk/oasis/> for more information.

2. Available from <http://www.cs.rochester.edu/research/cisd/resources/93dialogs/>

3. See <http://www.cs.rochester.edu/research/cisd/resources/trains.html> for details.

orange juice, via different routes to various locations, usually with time constraints imposed on them. In each dialogue, one user represents the *System*, and the other the human user.

The following changes and/or corrections were made to the data. As the speakers in the corpus were originally not labelled individually, I retrieved the relevant information from the documentation, and added it in modified form, prefixing the ID of the speaker mimicking the system by s_ and that of the user by u_. Furthermore, as the data also contains no punctuation, phono-pragmatic information had to be reconstructed as best possible. Many backchannels in the data were also formerly not integrated properly, and it appears that latching was frequently marked as overlap, thereby breaking the flow and 'masking' echoes, so that these features had to be corrected in order to ensure sequential integrity. As with many other spoken corpora, numbers are represented as words, and the 'proper names' of the engines used in to transport the boxcars or tankers that contain the commodities are represented as E + space + number, rather than marking them as names (e.g. E1, E2, etc.). The data also follow the general American conventions regarding the representations of filled pauses and minimal responses, as well as hyphenation, e.g. *uh-huh, mm-hm*, which I 'normalised' to *aha* and *mhm*, respectively, in order to ensure comparability with the British Trainline data. Some filled pauses that were originally represented as `<filled-pause>` were converted to *erm*, so that they could be identified as hesitation markers by DART. As with the Trainline data, I changed the numbers to digit representation. The transcriptions also contain a relatively high number of strong forms, in particular *i will*, which may indicate that the transcribers did not distinguish properly between contracted and non-contracted forms. However, as the presence or absence of (dys-)fluency phenomena is not a particular concern of the research for this book, as it would for instance be for the investigation of learner language, and as the audio data was not readily available, no attempts at correcting this feature were made. It may well be possible, though, that these strong forms in fact do not even constitute oversights of the transcription scheme or transcribers, but may actually be due to the nature of the interaction, where the 'human' interlocutors were trying to articulate more clearly as they were pretending to interact with a computer system.

3.1.3 The selection from the Switchboard Annotated Dialogue Corpus

The second selection was, again randomly, extracted from the Switchboard Dialog Act Corpus, a modified version of the original Switchboard Corpus (Godfrey & Holliman 1997) that already contained speech-act codes based on the SWBD-DAMSL tagset (see Jurafsky et al. 1997). These codes, however, were removed in the conversion process, so that only the units and general turn structure marked up in the original version were retained as far as possible. The turn structure itself, however, still required a fair amount of manual adjustment because clear backchannels were frequently not integrated properly, and instead treated as separate turns, and even pseudo-vocal noises like breathing or static noise, etc., regularly appear as turns attributed to the interlocutor (e.g. sw_0009_4329.utt) when they were either not attributable to any speaker or did not represent any form of verbal behaviour at all. Regarding units, markers of incompletion often do not mark genuine incompletion on units, but simply indicate list-like continuation, which is generally marked with <punc type="level" /> in the DART scheme.

As the Switchboard Corpus represents an example of relatively unconstrained dialogue on a variety of every-day life topics, the difference in nature between it and the other two corpora easily becomes apparent in a number of different ways. For one thing, it does contain a higher number of words and sometimes very long, expository, passages that are hard to break down into units, due to sub-ordinated structures which in turn contain co-ordination, so on average, the turns and units are longer than in the other two sets of data (see Table 3.1 below). Handling quoted speech also becomes more of an issue for this type of data, especially for longer passages, as these essentially need to be excluded from what the current speaker is saying as not constituting their own speech acts, but instead something that is merely reported. In addition, not all quoted materials are handled consistently in the data, as some are in fact represented in quotation marks, while others are not and instead simply marked by a quote-initiating *preface* or *tag*, where I use these terms in a slightly different way from Biber et al. (1999: 956), as they only apply them to clefted NPs, while I also use them for clause-like materials as in and ***she says*** *<quote>after working in there and and you know here she was a provider but you know there's only so much 1 person can do</quote> <punc type="stop" />* (sw_0015_4877, turn 30, unit 99), where the quote-initiating preface is marked in bold.

As with the other corpora, the speakers were originally labelled *A* and *B*, but I retrieved the exact corpus IDs for all speakers from the header info in the original transcripts, and inserted them into the data to be able to investigate idiosyncratic features if necessary, and replaced spelt-out numbers by digits. One other, minor issue was that some foreign words were oddly 'transliterated' in the original, e.g. "fianc3e" for *fiancé*, which needed to be fixed.

A summary of the corpus data primarily used in this book is shown in Table 3.1. For each set of data, the dialogue type, the number of c-units, turns, and words, as well as averages for units per turn, words per turn, and words per unit, are listed.

Table 3.1 Summary of corpus materials primarily used

Corpus	Type	c-units	turns	words[a]	units/turns	words/turns	words/units
SPAADIA Trainline (ver. 3)	task-oriented and transactional	6,060	3,136	25,921	1.93	8.27	4.28
Trains 93 (Selection)	task-oriented	5,035	1,997	19,264	2.52	9.65	3.83
SWBD Dialog Corpus (Selection)	unconstrained	6,287	1,894	36,335	3.32	19.18	5.78
total		**17,382**	**7,027**	**81,520**			

a. Including anonymised content.

3.1.4 Discarded data

I had initially intended to complement the Switchboard data with additional extracts from the BNC, and therefore created yet another selection of 35 files from file KDA, which contains 132 conversations from the demographically sampled part of the BNC. After successful conversion and during the manual pre-processing phase, however, it quickly became apparent that a sensible pragmatic analysis and annotation would be impossible, due to an extremely high number of `<unclear />` tags in the data, which even made a coherent interpretation of many passages by a human analyst impossible. Based on my earlier negative experience with the transcription of file D96, illustrated in Section 2.2 above, I thus became suspicious and first checked on the exact number of `<unclear />` elements in the multi-dialogue file KDA, and then calculated the percentage of these elements in relation to the overall number of words, including items marked as unclear as single word tokens, thereby of course potentially underestimating their number, as such elements may well represent more than one token. The result showed that 6% of all 'words' in this file are marked this way. Expanding this analysis to the whole sub-section KD to try and see whether other parts would perhaps be more suitable revealed that still altogether 3% percent of all 'tokens' were or could not be correctly identified by the transcribers, something that, along with the errors discussed earlier, sadly further undermined my confidence in the spoken part of the BNC as a source for analyses that go beyond simple concordancing, and made me come to the decision to disband my efforts of identifying suitable material for analysis there. Other issues I identified working through a limited number of dialogues were that *innit* was misrepresented as *in it* in a number of files (e.g. kda-003 & kda-009),[4]

4. Here, the numbers identify the dialogues extracted from the composite file.

backchannels were not integrated properly, some words had erroneous representations or capitalisation (e.g. *China town* for *Chinatown* or *derby* for *Derby*), and punctuation was used haphazardly by the original (linguistically untrained) transcribers, as questions are often marked by a full stop.

3.1.5 Supplementary data

For English, two further small 'collections' of data were also employed to illustrate specific points. These consist of two files containing excerpts from the film *A Fish Called Wanda* (Crichton 1988), altogether comprising 139 words, and one classroom dialogue from a corpus of Singaporean classroom dialogues (540 words). Both sets of data were originally given to me by colleagues who had asked me to test the original SPAACy annotation system on them, the first one to see whether SPAACy could handle non-standard 'everyday life' language including irony and swearing, and the second whether it would also be possible to employ the program to analyse classroom interactions. Whereas the former mini-corpus is clearly not task-oriented and its domain and activity type are difficult to define, the latter is again task-oriented, but perhaps *non-task-driven* (cf. Leech et al. 2000: 7) because it does not follow a relatively fixed plan or pattern, and the activity type can be defined as teaching.

3.2 The DART implementation and its use in handling dialogue data

The following discussion of the implementation of the DART research tool in this chapter is of central importance to this book, as it demonstrates the needs and options for providing an integrated solution to the creation and at least partial further analysis of pragmatically annotated dialogue corpora. It will also provide the basis for a better understanding of how the individual components necessary for a semi-automated identification and annotation of dialogue data can be brought together and 'controlled' efficiently in a kind of *corpus workbench* environment that provides a user-friendly, integrated mechanism for the cyclical process of annotation, evaluation of annotation results, and subsequent improvement of annotation routines. Furthermore, the discussion of the research tool will also highlight how certain preparatory steps for creating the necessary computational resources, such as domain-specific lexica, etc., can be performed within DART in order to enhance the flexibility of the tool. For a more detailed overview of the first version of DART, see Weisser (2016b). The tool itself is freely available in compiled form for Windows, and can be run through Wine on Linux or MacOS X.

3.2.1 The DART functionality

Many of the strategies and mechanisms for handling resources that are implemented in DART were originally developed in a more simplified form for the SPAAC project referred to in 3.1.1 above. For this project, the task was to produce a speech-act annotated corpus of more than 1,200 dialogues, altogether amounting to more than 182,300 words (cf. Leech & Weisser 2003). To annotate this quantity of data manually on all the different levels envisaged would have been an impossibility and thus an annotation tool called *SPAACy* (described in Weisser 2003 & 2004) was developed. This enabled the research team to open a single dialogue at a time, run some interactive pre-processing on it, then conduct an automatic analysis of the dialogue, and finally perform some post-editing of the annotated data. A screenshot of the final version of SPAACy, depicting the interface with a fully annotated dialogue, is shown below.

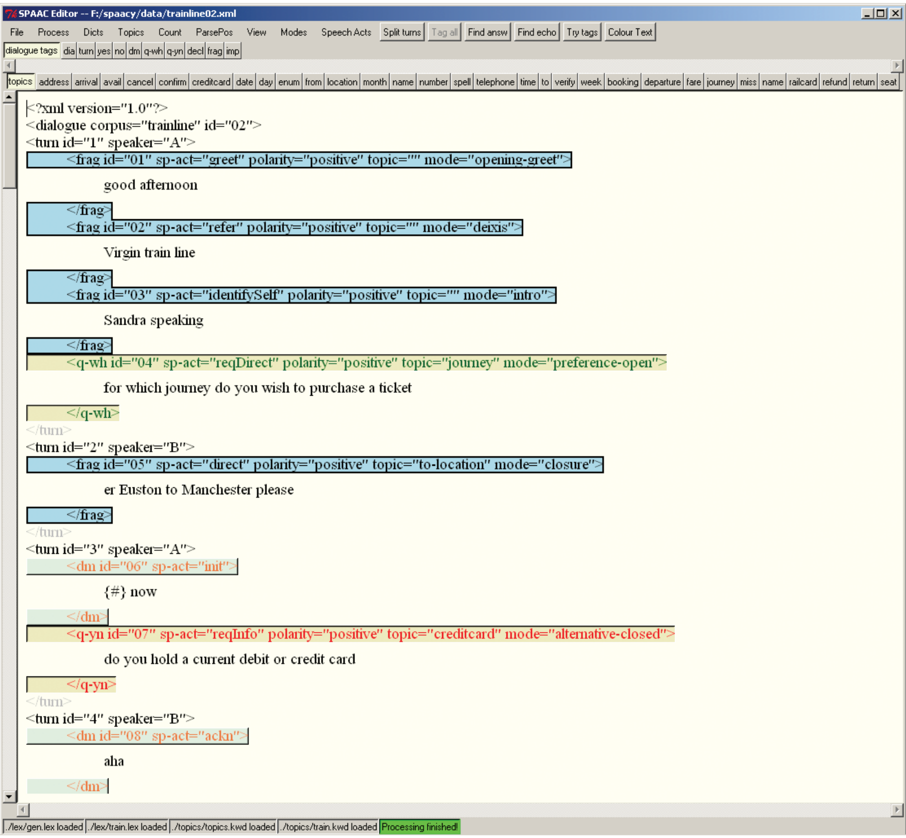

Figure 3.1 The SPAACy dialogue annotation tool

As can be seen when looking at the illustration above, the previous annotation format did not make use of empty elements yet, although the formatting conventions for non-textual data were already much more consistent and easier to understand than the ones used in CA that were discussed in 2.3 above. In terms of attributes, instead of the more TEI-conformant *n*, all elements were also given an *id* attribute at the time. This, however, has been replaced by *n* in the updated current methodology to bring it more in line with the TEI recommendations.

Amongst some of the further features that can be noted are the dynamically built toolbars containing elements and attributes that were supposed to provide the users of the tool with fixed options for annotating or post-editing structural and content elements, colour coding to reflect the 'semantics' of the annotated syntactic unit categories (cf. Section 2.2.1 above), as well as information pertaining to the equally dynamically loaded lexicon and keyword-spotting resources for particular domains, which can be seen in the status bar at the bottom of the illustration. All of these issues will be discussed and illustrated, in more or less detail, depending on their relevance the pragmatic analysis discussed here, in the following sections and/or chapters.

Although SPAACy performed fairly well, as demonstrated in Weisser 2004 by comparing an automated analysis of a dialogue with a *gold-standard* one produced by Geoffrey Leech, it still had a number of drawbacks. One of these was, as already pointed out above, that it was only possible to annotate a single dialogue at a time. This made it a very time-consuming effort to repeatedly annotate a sufficient number of dialogues whenever new features or improved algorithms were incorporated, which was especially inconvenient when it came to testing the correct behaviour or improvements resulting from these new additions.

Another major drawback of the original design was a certain lack of modularity with regard to the components for syntactic processing because all the analysis routines were tied into the routines that interacted with the *Graphical User Interface* (GUI), the editor window, which was essentially the reason for only being able to annotate a single dialogue at a time in the first place. In order to produce the more efficient multi-dialogue analysis methodology that is employed in this book, it was therefore necessary to separate the two processes from each other, as well as to improve the modularity of the overall analysis method in general.

The successor to SPAACy, the new DART tool (ver. 2 depicted in **Figure 3.2**), not only makes it easier than before to conduct the annotation process automatically, but also to test the outcome of any annotation process on a theoretically very large corpus, as well as analyse a variety of relevant features, within one and the same program.

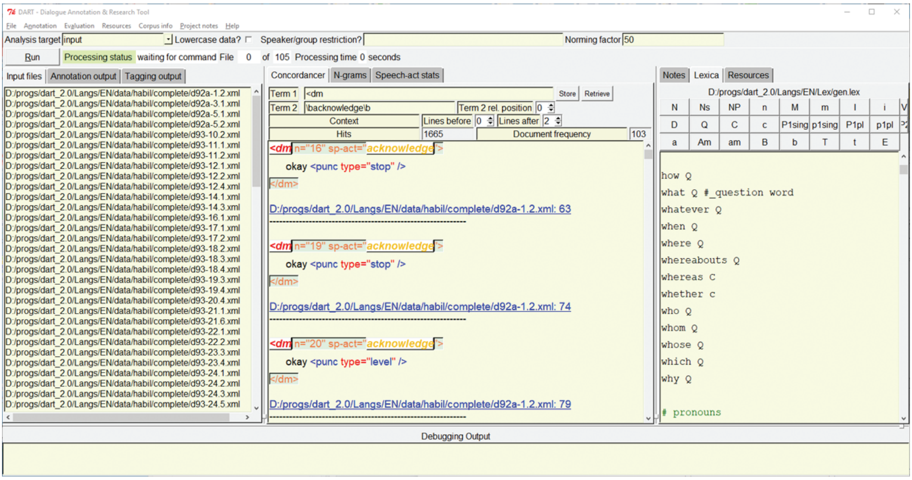

Figure 3.2 The Dialogue Annotation and Research Tool (DART ver. 2)

DART in fact constitutes an integrated research and testing environment, somewhat similar to the integrated development environments (IDEs) that are offered for a variety of programming languages, with the one major difference that the results of the linguistic annotation can immediately be verified and post-edited within the same environment, and the effects of any changes to different resources observed directly.

Inside the tool, researchers can load corpus data from individual files or whole corpora stored in a single directory into the 'Input files' workspace tab (left hand-side window), open the files for manual pre-processing in an annotation editor by double-clicking on them, and then either start an annotation or tagging process, with a list of the output files again accessible for investigation or post-editing from the 'Annotated files' tab. This already enables the computationally naïve user of the system to easily pre-process, consistently annotate, and post-edit large numbers of files without having to get used to a complicated interface or switching between different programs. Thus a cyclical processing and revision of the analysis routines becomes possible, which, in turn, makes it possible to devise and test new hypotheses about the linguistic nature of the data. The fact that the annotation process itself is carried out fully automatically clearly distinguishes DART from other linguistic annotation tools that only allow the user to add annotations manually.

DART's *Analysis* options also provide access to the functionality for creating the kind of frequency and n-gram lists for a given corpus, or reverse-sorted lists that may be used for identifying morphological features, many of us are used to from concordancers like AntConc (Anthony 2014). Within DART, such functionality also allows the user to prepare or improve different types of resources for new or existing domains, as well as identify morphological patterns that can be integrated into existing analysis routines, again all without the need to access external tools. In addition, though, DART allows the user to investigate speech-act statistics based on corpora previously annotated within it, and even includes options for filtering these according to individual speakers or groups. This feature helps the researcher to achieve an overview of the distribution of particular dialogue features, as well as identifying potential analysis problems on various levels, especially if the system has been unable to assign a relevant speech act.

The use of basic word frequency lists for e.g. the creation of domain-specific lexica will be discussed in more detail below, but the merits of being able to produce n-gram frequency lists in the context of pragmatic analysis probably already deserve some discussion here. As the design of the analysis methodology likewise caters for identifying and annotating both relatively domain-independent and domain-specific semantic information (c.f. Section 5.3), as well as largely domain-independent semantico-pragmatic constructions (c.f. Section 5.2), it is useful to be able to identify recurrent 'themes' or 'constructions', so that their patterns of occurrence can be induced and specified as regular expressions. These patterns can then be grouped under suitable category labels, and whenever such a pattern can be matched against a textual unit, the corresponding label can be recorded along with other information pertaining to this unit. This, in turn, may later potentially be used as part of the inferencing process in identifying the speech act(s) (c.f. Section 6.5).

Even if the information matched in this way is purely semantic and cannot necessarily be used as part of the inferencing process, it can still be recorded as a conversational topic and may later help to identify particular stages of a dialogue, or allow researchers to investigate how similar topics are talked about across different corpora.

The *Evaluation* menu contains options for creating a random selection of files from a given data set and copying them to a special evaluation folder, thereby eliminating the risk of the user making a biased selection. These files can then be investigated in various ways, either 'manually' or using the various analysis features offered by DART. In addition, an option for checking the well-formedness of files that have been converted to the DART format from other sources can be found here.

Access to a concordancing facility is provided via the *Concordance* tab in the analysis section in the centre of the interface. This concordancer can be used to investigate unprocessed, annotated or tagged data for verification purposes or in order to test the effect of different processing routines at different levels. It can also be employed for efficient post-editing, e.g. by listing all units that could not be assigned a speech act attribute, which will then be immediately accessible for editing via hyperlinks that open up an editor window with the cursor positioned on the relevant line. Such an option not only makes the researcher independent of any external concordancing utilities that would otherwise have to be used before or after any annotation or verification task, but the ability to open and edit the relevant files via hyperlinks is also something that improves the efficiency of the research task, and which would usually not be possible using general concordancing tools, as well as probably most other existing linguistic annotation tools, which generally also do not provide any options for automated analyses.

Once a given corpus has been finalised, the concordancing facility will also allow the user to create illustrative sample materials of particular phenomena, maybe for producing textbooks, other reference materials, or research articles, by saving the concordancing results to a file. Apart from being able to restrict the concordance to individual speakers or groups, it is also possible to look for specific combinations, such as all units that have been marked up with a particular syntactic tag plus a specific speech act, etc., as two *search terms* can be defined in parallel. The second one of these terms may also be specified as occurring relative to the first, which e.g. enables the researcher to look for a particular speech act only if the actual realisation of the act – which occurs on the next line – contains a certain expression In the same way, one can also search for tags or speech acts that occur at the beginning or end of a turn, etc.

To identify the relevance of a particular feature concordanced on, or identified as part of the frequency list options, within a particular corpus, all these analysis features also indicate the document frequency of the item, thereby providing a measure of dispersion.

Apart from the annotation and analysis options described above, which are most relevant for the research described in this book, DART also sports a variety of different facilities for creating or editing annotation-relevant resources, ranging from being able to set up configurations for new corpora, editing lexica and keyword/-phrase 'thesauri' for semantic and semantico-pragmatic analysis routines, to being able to edit the tag configurations in the built-in editor. For more details, please refer to Weisser 2016b or the DART manual.

3.2.2 The DART XML format

As already discussed to some extent in earlier chapters, the basic input and output format used by DART is a fairly simple form of XML that should be transparent even to a user who is not familiar with many of the features of XML. This basic structure is depicted in Figure 3.3.

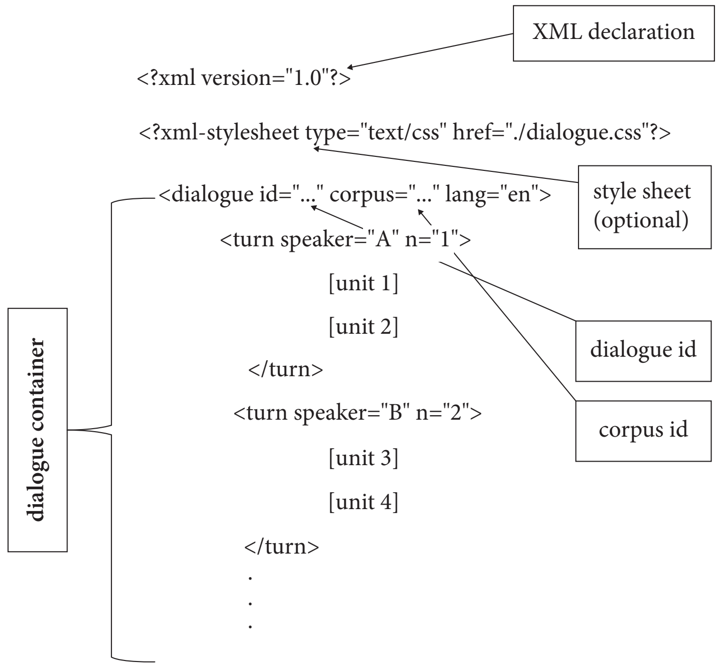

Figure 3.3 Basic DART XML dialogue structure

The raw, unprocessed files consist of the obligatory XML declaration and a container *dialogue* element whose attributes are *corpus, id*, and *lang*, where the first and the last of these allow DART to load the relevant resources for processing automatically. This may optionally be followed by a link to a style sheet. The only nested non-empty elements inside the container element are the individual speaker turns, which are characterised by two attributes, an incremental *n* containing the running number of the turn, and *speaker*, which generally consists of a single speaker identifier.

Potential empty elements inside the turns represent pauses and the 'phono-pragmatic' punctuation markers, as discussed in Section 2.3.2 above, but can also be augmented to include user-defined comments related to any other types of phenomena, such as the multi-modal information referred to earlier. Apart from empty elements appearing inside the textual data contained in the turns, all markup is otherwise separated from the dialogue text proper, and any turn may consist of one or more newline-separated lines, where each line generally represents one or more functional/pragmatic units.

The annotation process adds the syntactic elements `<address>`, `<decl>`, `<dm>`, `<frag>`, `<imp>`, `<no>`, `<q-wh>`, `<q-yn>` and `<yes>` with the attributes `n` (consecutive numerical identifier), `sp-act` (speech act), (surface) `polarity`, `topic` (semantic information), and mode (semantico-pragmatic information). The exact significance of all the attributes will be illustrated in the relevant sections below.

SPAACy originally contained some built-in routines for converting non-XML source data to the SPAAC/DART XML format, but since there can be major differences between different corpora that one might want to analyse in DART, no import routines were included in this tool. However, as the basic XML format required for processing in DART is very simple, it is easy enough to either do a manual conversion or write external conversion scripts that will produce the right format. Perhaps more importantly, all data to be processed should be UTF-8 encoded, as this is the encoding DART expects and produces. Of course, for English language data that does not contain any special accented or Umlaut characters, plain ASCII will also be interpreted correctly, as their code points correspond to the UTF-8 ones (see Weisser 2016a: 39ff).

Having outlined the basic framework the analysis methodology is embedded in, it is now possible for us to proceed to a discussion of the individual linguistic and computational steps involved in conducting the analysis and annotation, beginning 'bottom-up' with the levels of morphology and morpho-syntax.

3.3 Morpho-syntactic resources required for pragmatic analysis

In this section, I will describe some of the resources required in order to be able to carry out the computer-based pragmatic analysis and annotation in DART. Such a description is necessary to enable the reader to understand the issues and difficulties involved in creating the annotated materials to be analysed in later chapters. At the same time, the discussion will hopefully also shed some light on how an automated analysis of this type may become possible in the first place, as well as partly illustrating at least some of the connections between the different levels of processing involved in extracting pragmatics-relevant meaning from dialogues.

For any kind of linguistic processing that is not purely based on pattern matching, it is necessary to have access to some basic lexical information. In non-computer-based pragmatics, it is implicitly assumed that all the vocabulary contained in the materials under discussion is known and that the reader – or presumed hearer – is familiar with the words and their parts-of-speech, inasmuch as the latter is relevant for the interpretation process at all. In computer-based pragmatics, and especially when dealing with a variety of different domains or even unconstrained discourse, one can hardly assume that the computer will 'understand' any of the words that form part of a pragmatic unit, and therefore a suitable lexicon has to be included in any program that analyses language 'intelligently'.

Modern computer-readable lexica are created for various purposes, some for (language) learner use, others that serve purely as encyclopediæ, and yet others, such as the ones discussed here, for tasks in Natural Language Processing (NLP). Most researchers will certainly at least have encountered examples of the former two, but may have little or no experience with the latter, which is why this section will provide a short introduction as to their *content*, potential *structure*, and *coverage*. For a more in-depth overview of the topic, see van Eynde & Gibbon (2000) or Ooi (1998).

Essentially, any intelligent automatic analysis of language has to start at the level of words because this is what texts mainly consist of, apart from the kind of non-verbal information pointed out earlier. Many recent approaches to language analysis or the description of syntax – such as Lexical Functional Grammar (LFG; cf. Bresnan 2000), Head-Driven Phrase Structure Grammar (HPSG; cf. Pollard & Sag 1994), etc. – are heavily lexicalised. Unlike earlier (Chomskyan) approaches, which stress syntactic rules and only incorporate a few selectional restrictions/constraints, these approaches generally make use of large-scale lexica that contain many different types of information associated with each word inside the lexicon. These types of information frequently include word class – also known as part-of-speech (or PoS for short) –, phonetic, semantic and syntactic (case, complementation), gender, number, tense, etc., information, usually in form of (typed) feature structures (cf. Ooi 1998: 92 or Gibbon 2000). A relatively simple example from LFG will illustrate this.

(34)	a.	*a*	Det	(↑ DEF) = -	
	b.	*girl*	N	(↑ PRED) = 'girl')	
				(↑ PERS) = 3	
				(↑ NUM) = SG	
	c.	*me*	N	(↑ PRED) = 'PRO'	
				(↑ PERS) = 1	
				(↑ NUM) = SG	
				(↑ CASE) $=_c$ ACC	(Sells 1985: 159)

Figure 3.4 A short sample from an LFG lexicon

Here, one can clearly see – without going into lengthy explanation about the individual details of the theory – that each lexical entry has a PoS feature associated with it, plus a few other features in the form of *attribute-value pairs*, depending on the word class it belongs to, as well as functions it performs (morpho-)syntactically.

The drawback of any analysis that relies too heavily on the lexicon and this type of associated information, rather than also including an appropriate amount of (morpho-)syntactic analysis, is of course that *out-of-lexicon* words are difficult to handle and any analysis may already be doomed to fail at a rather early stage, unless appropriate fail-safe mechanisms are integrated into it. Furthermore, the creation of extensive lexica of the sort just described above is also rather costly in terms of time and person-power involved, apart from very often being tied into specific theoretical assumptions about the kind of grammar that is to be implemented.

In contrast, one of the explicit ideas behind the simplified lexicon model employed here is to remain as theory-neutral as possible. If one wants to be able to create corpora for a variety of different domains quickly and efficiently, it is also unrealistic to continuously adapt one single lexicon to ever-growing domain-specific needs, or to strive for high-coverage lexica, especially if the aim is to allow single users or small teams of researchers to work on and create flexible resources for varying domains and needs. This is why the current approach limits itself to a minimum of information contained inside the lexicon and includes only those items of information that are absolutely essential for recognising the relevant syntactic units that later serve as a basis for identifying the speech act. This is not to say, though, that it could not later be expanded to include further types of information, should it become necessary to do so to improve the analysis methodology.

It is at least partly possible, though, to generate lexica that contain minimal PoS information from pre-tagged corpora (cf. McEnery & Wilson 1996: 127). These can then be enriched with additional features as necessary. If one opts for such an approach, however, it is definitely necessary to ensure to pick the right kind of corpus (or corpora) as a basis for generating the lexicon, since e.g. a corpus of purely written or literary language will hardly be the right source for creating a lexicon to be used for analysing spoken interaction. This is at least partly so because written sources will not necessarily contain specific words that are typical only of spoken language or, if they do, may not reflect their typical use, and consequently even correct PoS. Thus, e.g. the word *well* in written language is primarily used as an adverb, whereas its main use in spoken interaction is as a *discourse marker* (cf. 4.3.3 below). Usually, general corpora, such as the BNC, where *well* occurs with a much higher relative frequency in spoken language (cf. Leech, Rayson & Wilson 2001: 242), do not indicate this distinction by assigning specific PoS categories to it.

Nevertheless, having PoS information in a lexicon is definitely the minimum requirement for any kind of language analysis, whether it be purely syntactic, semantic or pragmatic. However, one certainly does not always need to incorporate all the particular features discussed above in the lexicon, and does not even have to employ any particular typed feature structure, as I shall demonstrate in the next section.

3.3.1 The generic lexicon concept

Although most linguists generally use lexica extensively, there is one thing they might never really become aware of, which is that the language used in them to describe and explain entries is usually quite restricted and formulaic (cf. Barnbrook 1996: 154ff). The fact that many of the descriptions of lexical items follow these formulaic patterns is of less interest to us here than one of its correlates, that is that these formulae tend to go hand in hand with only a fairly limited set of words that is used in describing the content. The basic idea behind this is that, in order to describe complex phenomena, we should not necessarily make recourse to complex or difficult words that the reader may not understand, but that we can use a *common core vocabulary* that forms the basis for describing all other words.

A similar idea is the one behind the generic lexicon concept proposed here, only that, in our case, the basic notion is that there is a common vocabulary that accounts for most of the core words used in conducting basic verbal interactions, and thus applies to many situations that one may want to analyse pragmatically. In a sense, this is somewhat similar to the basic words in West's *General Service List* (West 1953), but geared more towards everyday interaction. This gives the DART approach a certain flexibility because the same analysis method can easily be applied to a number of different possible domains, and thereby achieve a high degree of morpho-syntactic and syntactic recognition for any analysis right from the start, but without necessarily having to modify the basic lexicon at all. A similar approach is, for instance, taken by Butt et al. (1999) in the context of developing LFG grammars for parsers, who refer to their (more restricted) 'generic lexicon' as "the core lexicon" that "includes closed class items such as auxiliaries, determiners, and prepositions", as well as "entries for verbs or nouns which behave exceptionally" (ibid.: 168). This lexicon is then augmented by a number of additional lexica, including a "lexicon file which deals with technical terms or specialized uses of words" (ibid.), in other words, a *domain-specific lexicon* (c.f. 3.3.3 below). Butt et al. define the advantages of such a modular approach 'negatively' as "that when the grammar is used for a different text or application, the exclusion of this specialized vocabulary simply requires not loading that particular file" (ibid.), while I would define the advantage in rather a more positive way as that augmenting the core lexicon makes it possible to improve the performance of the analysis by making adjustments to it that are more appropriate to a given domain only as and when required.

The composition of the generic DART lexicon, which only comprises between 700–800 words, is partly based on empirical evidence, through analysing word frequency data from different domains, but also draws on very basic linguistic notions, such as that the most frequent words in any language tend to be words which can be listed easily as a *closed class* of *function words*. Furthermore, the lexicon also incorporates some of the most frequently occurring general content words from the BNC, as listed in Leech, Rayson & Wilson (2001). In order to illustrate the bases for its composition, primarily frequency data from the Trainline and the Trains 93 Corpus will be drawn upon to help the reader understand what the most important generic lexical elements may be and what their significance in pragmatics is.

The first group is formed by determiners and pronouns, as the only type of 'genre' that could do without the former would be the telegraph style. And, as far as the latter are concerned, spoken language is known to abound with them, as also evidenced in the frequency list of word classes in the LLC provided by Altenberg (1990: 185), where pronouns figure with a relative frequency of 17.3%, only surpassed by verbs, which occur with a frequency of 20.1%. This especially applies to 1st and 2nd person pronouns, since dialogue participants frequently need to refer to one another, or themselves, when interacting.

Table 3.2 Frequencies for 1st and 2nd person pronouns in two illustrative corpora

pronoun	**Trainline (abs.)**	**Trainline (rel.)**	**Trains (abs.)**	**Trains (rel.)**
i[a]	505	2.24%	359	1.90%
me	76	0.34%	27	0.14%
my	19	0.08%	13	0.07%
our	1	0.00%	10	0.05%
us	4	0.02%	18	0.10%
we	38	0.17%	428	2.26%
you	800	3.54%	205	1.08%
your	257	1.14%	6	0.03%
yourself	7	0.03%	0	0.00%
yous	1	0.004%	0	0.00%
totals	**1,708**	7.56%	**1,066**	5.63%

a. In the DART representation format, all words that do not represent proper nouns are lowercased to facilitate the processing.

As can be seen from the percentages in the totals row, the combined frequency counts for the different 1st and 2nd person pronouns alone makes up a fairly large percentage of the overall words in both corpora. In total, the percentage of 1st and 2nd person pronouns in the Trainline corpus is slightly higher than in the Trains 93 selection, most likely due to the fact that, in the former, more personal information is elicited as part of the transactions.

Of course, the topic of *personal reference* or *person deixis* (cf. Levinson 1983, 68 ff. or Huang 2007: 136 ff.) is also one that has traditionally been covered in pragmatics to a great extent as part of a general discussion on *reference* and *definiteness* (cf. Levinson 1983: 83). According to Ariel (2008: 44 ff.), pronouns signal a high degree of *activation* of items or participants in discourse, in other words, largely those that are currently in focus, which is why they may also be indicative of specific *discourse functions* (ibid.: 53).

Auxiliaries, or auxiliary and full verb forms of *be* and *have*, also tend to occur in many constructions, ranging from basic existential expressions, such as *and **there's there*** is *no others* (trainline04) or *but **there is** those restrictions on the week day that i've explained to you* (trainline18), via passive constructions as in e.g. *i've just **been** quoted 19 pounds* (trainline03), modal expressions, such as in *if you **have to** cancel* (trainline01), to expressions that express basic tense and aspect, e.g. and *if you miss the service that **i have booked** you on* (trainline09). The first function mentioned above again relates to *definite descriptions* (cf. Levinson 1983: 60 or 170 ff.) as discussed in the traditional pragmatics literature, whereas the latter ones are connected either with certain types of *time deixis* (including the passive) or forms of modality that signal conditions pertaining to the current dialogue situation.

Another group of highly frequent words comprises determiners (*the, a, an, this, that*, etc.), numerical modifiers (cardinal and ordinal numbers) or quantifiers (*some, any, many, most*, etc.). Together, they form a class of words which has 'determiner or specifier character' and thus represents a part of almost all noun phrases that are not instantiated by pronouns. The first two groups again would be treated in traditional pragmatics under the heading of definite description, while the latter – although similar to numerical modifiers – is generally discussed in conjunction with issues of *scope*, but often more from the perspective of semantics where attempts are made to arrive at a representation of the logical form of an 'utterance' that either incorporates the existential quantifier $\exists$ and/or the all quantifier $\forall$. Attempts at reducing natural language to such logical representations frequently leads to irresolvable ambiguities or incorrect formal representations, though, as can be seen in the following example of a 'donkey sentence', taken from Carstensen et al. (2004: 302) in the original German.

(3.75)
Jeder Bauer, der einen Esel hat, schlägt ihn.
(Every farmer who has a donkey beats it. My translation)
$\forall x \forall y.\mathrm{bauer}(x) \wedge \mathrm{esel}(y) \rightarrow \mathrm{schlagen}(x,y)$

Here, the limitation of the expressiveness of the all and every quantifiers, where the original intention was to refer to a single referent that each one of the *x*s possesses respectively, does not produce a correct rendition of the natural language facts as the genuine translation of the logical representation from above should in fact be something like *All farmers who have all donkeys beat all of them*. This weakness of formal logic as a means of representing natural language facts has already been known for a long time – albeit apparently ignored by formal semanticists up to this day –, as can be seen in the following extended quote by Minsky, originally from 1974.

> 1. "Logical" reasoning is not flexible enough to serve as a basis for thinking: I prefer to think of it as a collection of heuristic methods, effective only when applied to starkly simplified schematic plans. The consistency that logic absolutely demands is not otherwise usually *available – and probably not even desirable!* – because consistent systems are likely to be too weak.
> 2. I doubt the feasibility of representing ordinary knowledge effectively in the form of many small, independently true propositions.
> 3. The strategy of complete separation of specific knowledge from general rules of inference is much too radical. We need more direct ways for linking fragments of knowledge to advice about *how* they are to be used.
> 4. It was long believed that it was crucial to make all knowledge accessible to deduction in the form of declarative statements; but this seems less urgent as we learn ways to manipulate structural and procedural descriptions.
>
> I do not mean to suggest that "thinking" can proceed very far without something like "reasoning". We certainly need (and use) something like syllogistic deduction; but I expect mechanisms for doing such things to emerge in any case from processes for "matching" and "instantiation" required for other functions. Traditional formal logic is a technical tool for discussing either *everything that can be deduced from some data* or *whether a certain consequence can be so deduced;* it cannot discuss at all what *ought* to be deduced under ordinary circumstances. Like the abstract theory of syntax, formal logic without a powerful procedural semantics cannot deal with meaningful situations. (Minsky 1997: 141)

Here, I am not concerned with issues of deep semantic analysis, but, as stated before, only wanted to point out the relevance of quantifiers for pragmatic analysis because quantification, on whatever level it may eventually need to be analysed, is something that occurs frequently in natural language, and therefore quantifiers form an important part of its vocabulary.

Next, there are conjunctions, prepositions, and deictica (other than the ones already mentioned). This may seem like a somewhat odd choice for a group, but essentially, they all perform a similar function in 'gluing' the individual parts of a text together, to build up connections, or help to establish a context or 'situatedness' in the widest sense. Thus, the deictica comprise temporal (*then, today, yesterday, tomorrow*) and local adverbs (*here, there*), as well as nouns, such as the names of months, times of the day (e.g. *morning, evening, noon, afternoon*), etc. Both groups have traditionally been discussed under the headings of *temporal* and *place deixis* in pragmatics.

The last group of words to be discussed here comprises items that are of particular relevance to dialogues or conversation in general. They either help to control the flow of the verbal interaction, like question words (*who, what, where, when, how*), different forms of *yes* and *no* (*yes, yep, yeah, aye, no, nope*), discourse markers (*well, now, so*), to establish basic politeness or serve as attention-getters, such as terms of address (*Sir, Madam, guys, girls*), or to help speakers with the planning of their utterances, such as fillers or backchannels (*em, er, mhm*). I will later show that some of these can essentially be regarded as constituting one-word speech acts in their own right. Most of the above, though, have only relatively recently become the focus of more discourse- or dialogue-oriented pragmatics, but rarely play any role whatsoever in traditional, logic-based pragmatics, probably due to the fact that they cannot be interpreted with reference to *truth-conditionality*.

To illustrate the concrete form of the generic lexicon, a sample from its beginning is shown below:

```
a d #indefinite_determiner
an d
the D #definite_determiner
that D
this D
# question words
how Q
what Q #_question word
whatever Q
when Q
where Q
whereabouts Q
whether c
who Q
which Q
why Q
# pronouns
i P1sing
```

Figure 3.5 A sample from the generic lexicon

This sample shows the very simple structure of the generic lexicon, with a word form entry, separated from a PoS tag (to be discussed in the next section), as well as a potential comment indicating further information concerning the tag.

Comments are always 'introduced' by a hash mark[5] and may also appear on separate lines, where they indicate the beginning of a new section, normally a listing of words with a new PoS. This information is provided to help the reader of the lexicon understand its entries better, but is disregarded in processing.

Testing the 'raw' coverage of the generic lexicon against all word *types* and tokens occurring in the Trainline and Trains 93 selection data, and then comparing these to the LOB and FLOB corpora for reference, the following results were obtained.

Table 3.3 Lexical coverage of the generic lexicon with regard to 4 different corpora

Corpus	Types	Tokens	Common types	Common tokens	Type coverage	Token coverage
Trainline	1,265	28,075	397	18,513	31.38%	65.94%
Trains 93	907	50,744	363	39,789	40.02%	78.41%
LOB	85,716	1,022,364	632	605,330	0.74%	59.21%
FLOB	91,122	1,026,426	638	585,397	0.70%	57.03%

Below is a graphical illustration to help document and compare the coverage even better:

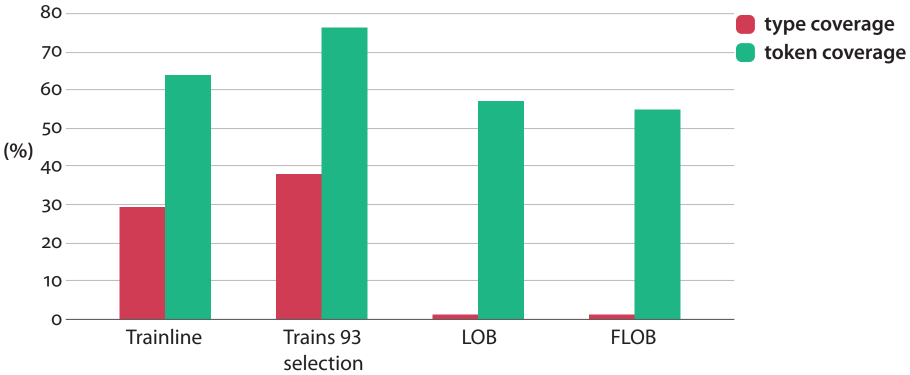

Figure 3.6 Comparison of type and token coverage of the uninflected generic lexicon for various corpora

As the above table and illustration clearly show, the items in the generic lexicon cover nearly around 30–40% of all word forms in the spoken corpora, whereas

5. … or pound symbol, for Americans.

they cater for less than .75% in the written reference corpora. This discrepancy is of course is due to the fact that the written corpora tend to be much more diverse – or *lexically dense* – in nature. However, what is extremely interesting is that, in terms of tokens, these lexicon items cover above 65% of the overall tokens in the spoken data, and even close to 60% of the ones in the written materials. This finding is well in accordance with Zipf's law (Manning & Schütze 1999: 23ff.), especially for spoken language – and also demonstrates that it should be possible to achieve a high degree of word recognition using only a small-coverage lexicon alone. Although this might seem to run counter to current tendencies in producing ever-growing high-coverage lexica for NLP and other purposes, it does not really do so, as one always needs to remember that the aim here is not a combined syntactic and semantic analysis via the lexicon, which would obviously necessitate a much higher coverage and level of detail of information stored inside the lexicon. Finally, taking into account that the lexicon actually (so far) only contains *uninflected* forms, it ought to be clear that this recognition rate can easily be improved for individual word forms in the two corpora by just adding a simple morphology, something that will be discussed in 3.3.3 below.

3.3.2 The DART tagset

As illustrated in the LFG lexicon sample entry earlier, there is a variety of different features one might potentially want to encode for a lexical entry. Lexica that are used purely for morpho-syntactic tagging, however, often do not contain all the information that may be relevant for a fully-fledged syntactic (or even semantic) analysis, but tend to be restricted to a number of features that are stored in a *compound tag*,[6] a tag that usually contains general PoS information, such as *N* for nouns, *V* for verbs, at the beginning, and is then followed by a more or less extensive list of other symbols that may represent more detailed information, such as whether the word is a proper or common noun, or morphological information, e.g. regarding *person, number, gender* or even *case* for inflected languages (cf. Cloeren 1999: 39). Perhaps the two most widely used tagsets (cf. Ide 2007: 293) employed in morpho-syntactic tagging are the *Penn* (Marcus et al. 1993; Santorini 1995) and *CLAWS* (Constituent Likelihood Automatic Word-tagging System) tagsets. The former is a very small tagset, comprising only 48 tags, due to the fact that its main purpose was to serve as a basis for parsing, for which a high level of granularity in the category distinctions is not

6. Please note that I am not using the term *compound tag* in the same way as e.g. Marcus et al. (1993: 2), who use this to refer to a combination of two or more PoS tags used to tag such constructs as contractions.

essential. The latter exists in different versions, both of which have been used to tag different versions of the BNC. The smaller one, C5, comprises only 62 tags, while C7 (see Garside, Leech & McEnery 1997: 257–260) has more than double that number, 152 tags. While C5 is only marginally smaller than Penn, and therefore is also more useful for parsing purposes, C7, with its substantially higher sub-categories for the major PoS tags, lends itself far more to in-depth grammatical analysis. Both the Penn and the CLAWS tagsets are essentially derived from the original Brown tagset, which makes them relatively comparable, though.

The DART tagset, shown in Table 3.4, is even smaller than the Penn one, with only 47 potential tags. The main differences between the DART tagset and both Penn and CLAWS are that the former is optimised for spoken language and also encodes the potential for what I call *grammatical polysemy*, the ability for a word form to belong to multiple word classes, apart from not actually being meant for 'serious' tagging in the first place, but instead only as a means to enable syntactic analysis.

Table 3.4 The DART tagset

Tag	Explanation	Examples
d	indefinite determiner	*a, an*
D	definite determiner	*the, this, that*
Q	question word, including relative pronouns	*how, what, whatever, when, where, whereabouts, which, who, why*
C	coordinating conjunction	*and, or*
c	subordinating conjunction	*as, because, cos, but, except, however, if, regardless, so, though, unless, whether*; *cause, otherwise*
M	modifier/quantifier	*able, afraid, all, any, available, bad, better, big, both, different, each, early, every, good*
m	numerical modifier or potential DM	*eight, eighty, eleven, fifty, five, four, forty, first, half; right, fine*
X	negation operator	*not, 't*
R	adverbial modifiers	*actually, again, anyway, apparently, here, hereby, indeed, just, maybe, perhaps, totally, unfortunately, usually, utterly*
r	DMs or similar modifiers	*aha, aye, no, nope, ok, okay, well, yeah, yep, yes; please*
T	temporal deictic	*then, today, tomorrow, yesterday*
t	DM or temporal deictic	*now, still*

(*Continued*)

Table 3.4 (*Continued*)

Tag	Explanation	Examples
E	existential *there*	
N	noun only	*April, August, December, May, Miss, Mr, Mrs, Ms, afternoon, card, choice, clock, customer, day, evening, hour*
n	predominantly noun, but also potentially verb, etc.	*account, address, book, date, end, head, house, man, number, part, quarter, type*
Ns	plural noun form or plural noun only	*men, people, women*
ns	plural form predominantly noun	*dates, times*
NP	proper name (not actually used, but included to be able to add proper names to the lexicon)	
P1sing	1st person singular pronoun	*I, me*
p1sing	1st person singular pronoun possessive	*mine, my*
P1pl	1st person plural pronoun	*we, us*
p1pl	1st person plural pronoun possessive	*our*
P2	2nd person pronoun	*you*
p2	2nd person pronoun possessive	*your*
P3sing	3rd person singular pronoun	*anyone, everybody, everyone, he, her, it, no-one, nobody, noone, nothing, she, somebody, someone, something*
p3sing	3rd person singular pronoun possessive	*its, his, hers*
P3pl	3rd person plural pronoun	*them, these, they, those*
p3pl	3rd person plural pronoun possessive	*their*
A	DO auxiliary/form	*do, don, does, doesn, did, didn*
a	HAVE auxiliary/form	*have, haven, has, had, 've*
Am	modal auxiliary	*could, 'd, 'll, may, might, must, need, shall, shan, should, shouldn, will, would, wouldn*
am	modal auxiliary form, potentially grammatically polysemous	*can, won*

Table 3.4 (*Continued*)

Tag	Explanation	Examples
B	BE form	*am, are, aren, be, been, is, isn, 'm, 're, was, wasn, were, weren*
b	BE or genitive marker	*'s*
V	verb only	*arrive, ask, assume, believe, come, exist, explain, find, get, give, go*
v	predominantly verb	*buy, call, change, check, contact, cost, cut, guess, hang, help, hold, like*
Ved	irregular or regular past tense/participle	*done, got, knew, made, said, seen*
ved	past tense/participle of predominantly verb form	*booked*
Ving	ing-form of verb	*arriving, departing*
ving	ing-form of predominantly verb form	*returning*
I	preposition/particle	*about, after, around, at, before, between, by, for, from, in, into, o, of, off, on, out, than, through, under, until, up, with*
i	*to*, both as preposition and infinitive marker	
F	filler	*ah, eh, em, er, erm, uh, um*
com	comment or unclear marker (empty element)	
ovl	overlap marker	
sil	pauses (silence)	only used in syntactic analysis and tagging
pun	punctuation	subsumes all `<punc … />` elements; only used in syntactic analysis and tagging

As Table 3.4 illustrates, its size and composition make the DART tagset more suitable for parsing than any detailed grammatical analysis. Although DART also provides an option for tagging dialogues, this feature is not really meant to be used for serious PoS tagging purposes, but instead primarily for debugging the syntactic analysis routines. The main purpose of the general tagging routines is rather to provide a basis for the syntactic analysis. Thus, unless the user specifically selects the tagging option, the tags never become visible in the output, but are discarded as soon as the syntactic analysis phase has been completed.

The DART tagset differs most strongly from Penn and CLAWS in one major respect, which is that the tag names are in fact in mixed case, rather than all uppercase, as in the other tagsets. This distinction is present for two reasons:

1. it makes it possible in many cases to keep the tag names shorter because some features – such as the possessive feature for pronouns or the coordinating vs. 'subordinating' feature in conjunctions – are encoded in the distinction between the upper- and lowercase variants of the (initial) letters
2. this distinction also provides the option to 'flag' the possibility of grammatical polysemy on specific word classes – mainly nouns and verbs –, something which would otherwise necessitate specifying two alternative tags in the lexicon.

This flexibility makes it possible to create very compact *character classes* or other patterns in *regular expressions*[7] when specifying tag options to be used for the syntactic analysis. One potential drawback of this approach, though, is that cases of multiple – rather than purely binary – word class associations, such as e.g. for the word *round*, cannot easily be distinguished in this way, in which case only the grammatically polysemous status of the word can be flagged and any disambiguation needs to be left to either the morphological analysis stage or the syntactic analysis routines. However, even if, based on DeRose's (1988) analysis of the grammatical polysemy in the Brown corpus, we can assume around 10% of word tokens to be polysemous, less than 1% actually have the potential for belonging to more than 2 word classes at the same time, so that this issue is probably negligible.

As has hopefully become apparent in the preceding discussion of the tagset, the DART approach explicitly takes ambiguity in form of grammatical polysemy into account and uses the general mechanism of the distinction between uppercase and lowercase to express a kind of probability. Unlike in stochastic approaches to language, though, where the probabilities are usually expressed in terms of frequencies automatically derived from tagged corpora, and disambiguation is often performed as a kind of post-processing step in tagging, the user of the system here needs to be aware of the potential for ambiguity and its repercussions at each step of the analysis. For some words, a certain margin of error is simply accepted, but again, this is not too dissimilar from stochastic methods, which, however, tend to apply their rules 'more blindly'.

7. For fairly detailed introductions to regular expressions and their syntax, see Weisser (2009) or Weisser (2016a).

3.3.3 Morphology and morpho-syntax

As the DART lexica only contain the most essential, non-variant, function words and the base forms for the most highly relevant content words, it would of course not be possible to identify all forms of the nominal, verbal, or adjectival paradigms that may need to be recognised yet. Since the aim in the syntactic analysis is to try and map words onto PoS tags, and then relate them to certain rules that specify syntactic patterns, purely relying on the lexica will therefore not work, despite the fact that the syntactic analysis used in DART is actually surprisingly simple compared to a fully-fledged parser. One of the main aims of a morphology in a system for pragmatic analysis, especially one that relies on small lexica of core items only, is therefore not only to try and recognise variants of the words listed in the lexica, but also what is commonly referred to as "out-of-vocabulary" words in speech recognition (Gibbon et al. 2000: 218) by attempting to recognise the word-class and/or other features via affix-recognition techniques.

Sproat (1992: xiii) provides two convincing arguments for relying on computational morphology in order to achieve such a task, as opposed to creating/using ever-growing lexica, which would certainly be feasible with today's computing power and memory resources:

(2) a. Simply expanding the dictionary to encompass every word one is ever likely to encounter is wrong: it fails to take advantage of regularities.
  b. No dictionary contains all the words one is likely to find in a real input.

Especially the latter point is highly relevant to this approach which is designed to be as generically applicable as possible, and hence should be able to work with dialogues from a variety of ever-changing domains for which producing new and exhaustive lexica each time one changes to a different domain would not only be extremely time-consuming, but also highly impractical.

Since I already stated earlier that regular expressions represent an excellent way of specifying linguistic patterns, it will probably come as no surprise that the morphology integrated into DART heavily relies on them. DART's morphological analyser is a simplified PoS tagger that provides a way to either match a word against its corresponding entry in the lexicon or apply some matching rules to 'hypothesise' about the possible PoS tag of the word it is currently trying to look up and map onto its tag. If the word is not found in the lexicon, the analysis routine goes through a number of 'rules' – generally expressed as matching operations mainly on suffixes, but sometimes also the beginning of the word –, occasionally also first reducing a word to its stem. This essentially represents a combination of *lookup* and *guesser* techniques (cf. Beesley & Karttunen 2003: 14 & 15). In addition, a small number of disambiguation patterns based on surrounding tags are also included in order to improve the accuracy, especially for identifying lexically polysemous words.

To be able to appreciate the significance of this approach, let us take a look at a few simple examples related to identifying modifiers here. Whereas the other two tagsets we referred to above for instance define separate tags for comparative and superlative modifiers, in DART, this is (currently) not required, and at least most superlative forms can be recognised as belonging to the class of modifiers via their {-est} ending. Of course, we cannot simply assume that every word ending in this suffix is really an adjective, so, in order not to misinterpret potential candidates, we need to include two simple constraints:

a. the word has to be longer than four characters, as otherwise it could hardly have a 3-letter suffix;
b. the word does not match the regular expression *^(?:arr|bedr|det|dig|gu|harv|inter|in[fg]|inqu|inv|manif|mol|pri|prot|sugg|temp)est$*, which covers many of the remaining exceptions by excluding specific word-initial sequences that definitely do not represent adjectival bases.

Identifying such constraints is relatively easy using standard corpus linguistics methods such as creating reverse-sorted frequency lists. For the comparative form, this is not quite as straightforward, due to the greater morphological productivity of the {-er} suffix, which is why a larger number of these forms may need to be listed in the lexicon. However, most ordinal numbers need not be included in the lexicon because they can again be identified via rules. Both cardinals and ordinals, when listed in the lexicon or identified via the morphological analysis routines, receive the lowercase *m* tag, which, for convenience, is also used for those adjectival or adverbial modifiers that may equally act as discourse markers, such as *right*. For an in-depth discussion of the latter, see 4.3.3 below.

Of course it is not only suffixes that may help to improve the analysis results. A number of known prefixes can also aid in potentially identifying the PoS of a word by either removing them and then looking up the remainder in the lexicon, or by assuming that a specific prefix can only ever occur with one or more specific word classes. Of course, the latter case may easily lead to over-generalisation and errors, e.g. if one were to erroneously assume that the initial character sequence <*un*> were (a) a negative prefix in all cases and (b) that e.g. it may only ever occur with adjectives.

3.3.4 'Synthesising' domain-specific lexica

In addition to combining the core generic lexicon and the morpho-syntactic analysis routines for improving lexical recognition, there is yet another option available in DART. Lexical coverage can easily be improved by adding a selection from the most important domain-specific vocabulary. The way the lexicon is implemented in DART facilitates loading any number of additional lexical resources along with the generic lexicon. This happens dynamically, based on configuration information that is retrieved from the *corpus* attribute stored in the wrapper element of the XML files and consulting additional configuration files that contain lists of lexica and other resources associated with each corpus, respectively. Furthermore, it is not only possible to add additional vocabulary, but also to override/-write the information stored for a given tag by adding the same word form again, only with different PoS information. A classic example for where this may be useful is the word *book*, which would usually be tagged as being predominantly a noun (*n*), but for domains connected to travelling, or making reservations in general, this tag should clearly be changed to predominantly verb (*v*). Such information can easily be gleaned from taking a few glances at domain-specific corpus materials, either by reading through a few dialogues or analysing the n-gram frequency lists that can be created in DART. A more 'statistical' approach, on the other hand, would generally not work here, as the data at hand would usually not be tagged, or – even if they were – probably not tagged appropriately, apart from the fact that small domain-specific corpora would most likely exhibit the problem of *sparseness* and thus make any stochastic approach unreliable.

One potential caveat of this simple augmentation approach, however, is that the latest 'definition' of a word always depends on the lexicon file loaded last, so that any additional file that 'redefines' a word may always also 'undo' an earlier redefinition. In other words, the order in which the lexicon files are loaded may be of extreme importance, and therefore the researcher working with a particular set of data always has to be aware of the contents of particular lexical (and other) resource files to a fair degree. This, however, is the same approach as taken in Butt et al. (1999: 170), the only difference there being that they provide more elaborate methods for adding or removing specific information to the lexicon (ibid: 170–71), something that may be included in future versions of DART.

Because the generic lexicon should already cover the most important words needed for any analysis, it is also feasible to use a kind of 'subtraction mechanism' in order to identify important domain-specific vocabulary, and thereby 'synthesise' a domain-specific lexicon. To identify whether a word is worth adding or not, it is of course important to look at the raw frequency information, as it is hardly worth adding low-frequency words, especially if they may be identifiable in some way via the morpho-syntactic analysis component. In addition, though, it might also be relevant to have an indication as to how many documents a word in fact occurs in – that is, its dispersion –, and to use this information as a factor.

In *information retrieval*, the relationship of occurrences in documents and overall frequency is often employed as a measure of determining the significance of a word for a *query*. Here, the individual word is usually given a negative *weighting* if it occurs as a high-frequency word in many different documents via a mechanism referred to as *tf x idf, term frequency by inverse document*[8] *frequency* (cf. Salton & Buckley 1988: 516). The basic assumption here is that these high-frequency words are not very good indicators of a specific domain, and hence less useful for discriminating between relevant documents. In our case, however, the inverse situation is true because there is a need to achieve a better lexical coverage for the whole corpus, and thus, while the most frequent generic vocabulary is already covered in the generic lexicon, high-frequency words that occur in more documents of the new domain should rather be added to the lexicon first. This way, a high document frequency can be employed as a positive factor to help decide which words to retain from a synthesised list.

DART provides such a feature, which allows the user to simply create a frequency list of any given corpus, 'take out' the words that already occur in the generic lexicon and their associated variants, identified again via the morphosyntactic analysis component, and output a weighted frequency list sorted in numerically descending order to an editor window. In this window, the file can then be edited and the final version of the new domain-specific lexicon saved. There is still a fair amount of editing to do after creating the frequency list, though, because such a frequency list obviously does not contain the relevant PoS information yet, but instead only the frequency information that helps to identify the most frequent words. Furthermore, despite filtering, the frequency list may still contain many inflected forms and – along with deciding whether to keep a word in the dictionary at all – it may also be necessary to change the word form from the frequency list to its base form and rely on the morphology for further analysis later.

Three additional features make 'synthesising' such lexica yet a little easier, (a) that the threshold for the minimal frequency of a word can be set prior to creating the word list, (b) that user-defined words or tags can be filtered out via a regular expression, and (c) that the interface has buttons for inserting the most commonly used tags, including tool tips that provide hints for the PoS-relevant definitions.

A sample output synthesised from the Trainline data is provided below. Read from the left, this contains the word form, the weighted frequency, and the raw frequency.

8. Or collection frequency.

credit	129.00	129
train	91.71	107
ticket	52.00	70
fare	51.37	62
tickets	47.14	66
travelling	44.80	56
journey	42.43	45
purchase	42.00	49
debit	40.80	42
seat	37.03	54
current	36.57	40
non	36.14	55
return	34.80	58
smoking	34.29	50
cheapest	30.86	40
advance	22.80	42
cancel	20.11	32
super	19.29	45
value	19.03	37
travel	18.46	34

Figure 3.7 Sample from a synthesised domain-specific lexicon

CHAPTER 4

The syntax of spoken language units

As illustrated above, spoken language presents the researcher with highly specific problems that need to be resolved. Nevertheless, when faced with the task of analysing it, many scholars apparently still find themselves tempted to intuitively resort to the same means employed as when handling written language. The reason for this is very simple: almost all of us have been brought up in 'literate' societies, societies where, despite the fact that we have learnt to speak before learning how to write, we have been 'drilled' by our education systems to see written language as the 'perfect' or 'ideal' means of expression. Thus, even in school, reciting poetry or producing a composition are valued more highly as skills than being able to produce a good oral presentation. Literature is (still) 'praised' – or at the very least presented – as the idealised form of language and anything as 'imperfect' as ordinary spoken language is treated as an inferior form of language because it happens to be full of little irregularities and may not be as neat and complex as a well-written report, etc.

The following passage from Blanche-Benveniste (1997: 10f) illustrates this problem more 'graphically'.

> Nous avons interrogé des enfants d'une dizaine d'années pour savoir s'ils estimaient bien parler le français. La réponse, comme nous nous y attendions, était «non», mais la justification nous a étonnés: ils disaient qu'ils ne parlaient pas bien parce qu'ils parlaient **avec des fautes d'orthographe**. Une telle réponse – qui ne semblait pas étonner l'institutrice – en dit long sur notre représentation du parlé, entièrement façonnée par l'écrit, dans la culture lettrée qui est la notre.
>
> (We interviewed children who were about ten years old in order to find out whether they thought they spoke French well. Their answer, as we had expected, was "no", but their justification surprised us: they said that they didn't speak well because they spoke **making orthographical mistakes**. Such a response – which didn't seem to surprise the teacher – tells us a lot about our notion of the spoken language, which is fashioned entirely after the written language in this literate language of ours.) [my translation]

However, one should not forget that spoken language, after all, is the primary means of communication, and it is therefore not surprising that more modern and immediate forms of verbal interaction used in ever-increasing and ever-popular forms of social and other Internet-based media, such as Facebook, chat rooms, web pages, email, etc., exhibit more of these 'unorderly' features of spoken language than those of its written, stylised, counterpart.

In order to be able to handle spoken language on the computer, it becomes necessary to be able to model the different, even irregular, forms of syntax and/or disfluencies that make it so 'messy'. This is why it is essential to begin this section with a discussion of the potential 'well-' and 'ill-formed' syntactic units one may encounter, starting with the general question of what the 'right' kind of unit in such an analysis may be in the first place. I shall here argue clearly against the traditional, more theoretically oriented approaches to language that see the sentence or the clause as the most important units in language, as well as some of their immediate corollaries in language processing, which manifest themselves in parsing techniques that are based on ideas derived from generative approaches to syntax, and therefore often embody an ill-conceived notion of *well-formedness* that is utterly inappropriate in dealing with real-life spoken language.

4.1 Sentence vs. syntactic types (C-Units)

Traditional grammar usually employs the notion of the *sentence* as an appropriate unit of syntactic description. However, the concept of 'sentence' is equally badly defined in linguistics as the notion of 'word'. I will soon demonstrate why exactly the concept of sentence is not a very useful one in the analysis of spoken discourse – and perhaps even its written counterpart –, but shall first provide an illustration of how some of the major grammars of the English language, the *Comprehensive Grammar of the English Language* (Quirk et al. 1985; henceforth CompGr), the *Longman Grammar of Spoken and Written English* (Biber et al. 1999; henceforth LongGr), and the *Cambridge Grammar of the English Language* (Huddleston & Pullum 2002; henceforth CamGr), handle the concept of the syntactic unit.

CompGr, being the oldest one of the three, is still firmly committed to treating the sentence as the highest-level syntactic unit, and distinguishes between four major different types thereof:

> DECLARATIVES are sentences in which the subject is present and generally precedes
> I. the verb […]
> II. INTERROGATIVES are sentences which are formally marked in one of two ways:
>
> i. *yes-no* interrogatives: the operator is placed in front of the subject […]
> ii. *wh*-interrogatives: the interrogative *wh*-element is positioned initially […]

> III. IMPERATIVES are sentences which normally have no overt grammatical subject, and whose verb has the base form […]
> IV. EXCLAMATIVES are sentences which have an initial phrase introduced by *what* or *how*, usually with subject-verb order (CompGr: 803)

Apart from these major types, there are some "minor types" that are generally treated under the heading of "Irregular sentences" (CompGr: 838ff), and which are mainly elliptical ones.

Both LongGr and CamGr, in contrast, no longer consider the sentence to be the highest-level syntactic unit, but instead attribute this role to the *clause*, which in the CompGr appears only in the second-highest position. LongGr still lists (in brackets) the sentence as part of its "hierarchy of units" (LongGr: 50), but also states – with regard to their corpus material used as a basis for the grammar – that:

> We argue […] that the 'sentence' is a notion that is not applicable to spoken language. It is true that the transcriptions contain orthographic sentences, defined as units beginning with a capital letter and ending with a period or other sentence-final punctuation mark. However, punctuation in the spoken texts must be treated with caution; the corpus transcribers marked punctuation to reflect spoken prosody, but there are no hard and fast rules to follow when punctuating natural conversation. (LongGr: 10)

This already hints at the problematic status of the sentence with regard to spoken language transcription, but further evidence of this will be introduced later, once the basic positions described in all three leading grammars have been outlined.

In addition, LongGr distinguishes between *dependent* and *independent* clauses, where "An embedded clause is a dependent clause. The superordinate clause, in which it is embedded, is termed the main clause." (LongGr: 192). These notions of embedding and dependence will again be of further interest later in the discussion of the status of such constructs as *if*-clauses in terms of their function and meaning in dialogues.

CamGr also still lists the sentence at the top of the syntactic hierarchy, stating that "We take sentences, […], to be units that occur sequentially in texts, but are not generally contained one within another." (CamGr: 44), but also makes it very clear that the sentence is conceptually quite different from other syntactic units concerning its status as an element of syntactic analysis or description:

> 'Sentence' is not a syntactic category term comparable to 'clause', 'noun phrase', 'verb phrase', etc., and does not figure in our constituent structure representations. (CamGr: 45)

Hence, despite still employing the term sentence, CamGr hereby implicitly more or less discards the (ill-defined) concept sentence – unlike contemporary generative theories would do – and denies its applicability to syntactic description. It refers to, and loosely defines, the clause as "[…] a syntactic construction consisting (in the central cases) of a subject and a predicate." (CamGr: 44). The additional qualification in round brackets here is important because, without it, there would be no way for the authors to handle subject- or verbless clauses, such as the ones mentioned in (their) Chapter 10: "We include within the category of clause various kinds of verbless constructions, such as open interrogative *What about the others?* or exclamative *What a disaster!*" (CamGr: 855). In general, though, CamGr bases its descriptions on "canonical clauses", simple, non-negative, active, non-embedded, declarative clauses. This may initially strike us as somewhat reminiscent of ideas from traditional philosophy concerning the status of declaratives as being the 'default type' of a sentence. However, in CamGr, this is only seen as a convenient descriptive mechanism:

> When we say, […], that *Is she still working?* involves subject-auxiliary inversion, we are not suggesting that a speaker actually starts with the declarative *She is still working* and then reverses the order of the first two elements. (CamGr: 48)

In other words, the canonical form is only used as a basis for comparing the other forms of clause type and their associated various grammatical properties to, rather than assuming that it represents some form of *deep* or *logical structure*, as is often more or less the case in traditional generative approaches.

However, since the term clause, despite the additions of the CamGr, can still be seen as somewhat conceptually biased, evoking notions of the traditional clause concept in many readers, my preference is to discard it altogether and instead consistently employ the concept of "C-unit" as defined in LongGr in this way:

> […] clausal and non-clausal units […] that […] cannot be syntactically integrated with the elements that precede or follow them. (Biber et al. 1999: 1070)

This definition not only provides a handle on 'ill-formed' types of syntactic categories, but also makes an important statement as to how one should be able to discern between the units of analysis by referring to their status as independent units. The independent status of individual units, however, if they cannot clearly be categorised in syntactic terms, can only be explained in terms of their semantico-pragmatic or pragmatic content and function, which is at least partly influenced or disambiguated by prosodic phonology.

Chafe approaches the topic of units in such a phonological way and says that:

> […] it is plausible to suppose that during the production of language a speaker will focus on the information he or she is verbalizing at that moment. Against this

> background, an intonation unit is plausibly viewed as the verbal representation of just the information that is in the speaker's focus of consciousness at the moment it is uttered. A speaker's intention in uttering an intonation unit must then be to introduce something resembling that particular focus of consciousness into the attentive listener's consciousness. If each intonation unit, indeed, corresponds to a focus of consciousness, intonation units can give us important insights into how much and what kinds of information can be active at one time in a speaker's mind […] (Chafe 1993: 39)

This would seem to provide yet another convincing argument for the existence of independent smaller syntactic units – such as discourse markers, which are discussed further below – but also include longer and more complex intonation units or chunks that may follow more 'sentence-like' patterns.

Edwards (1995: 20), in her discussion of units of analysis in discourse research, also refers to similar syntactic units identified via prosodic means.

> In written language the basic units are clearly set off by punctuation marks or indentation, and are usually defined syntactically: clause, sentence, paragraph. The units most natural to spoken language are of a different type and often shorter […]. In spoken discourse, many utterances lack explicit subjects or verbs and would be considered incomplete sentences in written language (e.g. "Bob's." in response to the question, "Whose Picasso is that?"). Discourse researchers often employ units based primarily on intonation (e.g. "tone units" or "intonation units") or bounded by pauses. ([…]) or some combination of these. […] An interesting syntactic unit is the "macrosyntagm" of Loman and Jørgensen ([…]), that is, a grammatically cohesive unit which is not part of any larger grammatical construction ([…]). Unlike a sentence unit in writing, a macrosyntagm may vary greatly in length, from a monosyllabic interjection to a multiword sentence expanded by a large number of subordinate clauses ([…]).

The type of categorisation procedure and syntactic makeup of the units Edwards discusses here rather seems like a kind of combination of the motivation for the C-unit given in LongGr and the ideas expressed by Chafe in the quote from above, which would further support my argument of preferring C-units over sentences or clauses, especially because, as pointed out by Moneglia (2011: 481), "[t]he main problem is that in spoken language a lot of configurations that are not clauses may turn out to be utterances in the speech flow. Almost 1/3 of speech events, according to the C-ORAL-ROM for the Romance languages and the Longman Grammar for English, do not have a verb and therefore do not show a clear syntactic structure.".

While the definition of the macrosyntagm given by Edwards earlier is certainly an interesting and useful one, it also indirectly raises one particular issue, that of the status of subordination, which she includes as an option for the overall composition of such a unit. However, certain syntactic constructs, such as *if clauses*, which are commonly assumed to belong to the group of subordinate clauses, do pose a problem in this respect. They appear to be more of a construct of written language which, with its 'neat' ordering, seems to impose a stronger hierarchy on complex syntactic constructs than their semantic or pragmatic content would warrant. To illustrate this point, one can observe these two alternative constructions:

a. if you miss the service i've reserved you on you are able to get the next available train
b. you are able to get the next available train if you miss the service i've reserved you on

Whereas (a) – which actually occurs twice in the Trainline data – may seem more natural and likely to occur – although one probably would not find it in a grammar or text book in this form –, surely (b) is equally possible. Actually, a similar example to (b) involving *because* can be found in the Trains data: *so i guess all the boxcars will have to go through oran… <pause /> through Corning because that's where the orange juice <vocal content="brth" /> orange factory is* (trains d92-1). In (b), the focus of attention in comparison to (a) is simply reversed, with the possibility expressed in the first unit being constrained in the second as a kind of afterthought or detail of minor importance, whereas in (a), the condition is brought to the foreground and therefore emphasised as being important. Thus, we only have two different perspectives on the same set of 'facts', or *truth conditions* in terms of formal semantics, rather than one being in any (other) way subordinate to the other than in terms of focus. This is not to say, though, that there is not a certain degree of interdependence present between these units, but this is more a matter of coherence than actual hierarchical dependence.

The preceding example already provides fairly strong evidence for the existence of two separate units in terms of syntax, semantics, and pragmatic force alone, but if one also pays attention to the likely prosodic rendering of the two examples, one will notice that there needs to be a distinct pause after the word *on* in (a) before the subject of the next unit, and one would also expect the pitch on this word to be a rising or level tone, including some *final lengthening*, which gives prominence to the otherwise unstressed, mono-syllabic function word. In (b), on the other hand, the pitch pattern would probably be continuous and there would be no prosodic break between the words *train* and *if*. As a matter of fact, the latter would probably even be linked to the former, the original *coda* /n/ of *train* becoming the *onset* of the monosyllabic conjunction *if*. In this way, the 'disjunctive force' of the conjunction effectively seems to be weakened in (b), whereas it is heightened in (a).

This indication of force clearly demonstrates that there is a difference between the two structures, and that one therefore ought to treat the two sub-parts as two pragmatic – or interactional – units that are not necessarily super- and subordinate to one another, but instead form a *coherence chain*. In other words, the so-called 'subordinating conjunction' in this case establishes a causal coherence-relation between the two units. The logical connection in terms of a coherence chain is also supported by the fact that the if-clause in (a) may actually 'initialise' a series of options that can follow, each in an independent clause. In other words, we could easily continue (a) by adding, *but you could also travel on the next day or you may even choose to travel next month*, etc. In this case, it would probably correspond to what is referred to as a "patterned construction" in (Cresti 2014: 374), one that is "performed across textual units, with each developing a different information function" (ibid.) in *Language into Act Theory* (Cresti 2000). In contrast to this, (b) would presumably be treated as a case of "linearized syntax" (ibid.) where the if-clause is genuinely subordinated to the initial clause.

Similar ideas regarding coherence chains and embedding also seem to have been expressed by Chafe (1984; cited in Miller & Weinert 1998) with regard to the conjunction *because*:

> It has been suggested that *because* clauses have a different discourse function in sentence-initial and sentence-final positions (Chafe 1984). In sentence-initial position, *because* clauses (indeed, adverbial clauses in general) function as a guide to information flow. As Chafe puts it, 'the adverbial clause … [signals] a path or orientation in terms of which the following information is to be understood'. The devised example given by Chafe is *Because it has such a big memory, I decided to buy it*, where the information that the computer in question has a large memory is the background against which the listener or reader is to assess the information that the speaker decided to buy it. In sentence-final position adverbial clauses are used by speakers simply to add something to the assertion conveyed by the main clause. (Miller & Weinert 1998: 57)

However, this does not hold true for all cases of conjunctions that carry the same label 'subordinating', but one can essentially create four sub-groups that exhibit different 'degrees of independence'. What is meant by the latter is their potential to occur either in initial position of a separate unit or whether or not they trigger a prosodic break before them if they occur at the beginning of the second unit that forms part of a complex coherence chain. The following taxonomy is still fairly tentative, somewhat graded and based on my own intuition, which however, has been substantiated through corpus analysis:

a. always independent: *but, however, so, whereas, while, whilst, for*
b. frequently independent: *if, unless, lest, except, because, as, though, when, after, before, once, since, provided (that), rather (than), whether, where*
c. hardly independent: *till, until, like, albeit*
d. never independent: *that*

Those items that appear in the *always independent* category seem to trigger an independent unit at all times, either because they tend to appear in initial position of two conjoined units, or because they force a prosodic break in between two units. The items in category (b) are a little less likely to appear initially or frequently do not cause the prosodic break, as could be seen in the *if* example above. For category (c), the likelihood of appearing initially or causing the prosodic break tends to decrease even further, and for *that* as a conjunction, it becomes impossible because it generally indicates an *embedded unit* (cf. also Verhagen 2005: 90ff. for a similar discussion, but one that excludes the prosodic side). In written representation, the ability of a conjunction to mark independent units is usually indicated by a comma or semicolon.

The taxonomy given above could help to identify what may or may not have to be classified as a separate syntactic unit when encountering complex syntactic units, essentially in declarative structures, as the other syntactic categories do not usually tend to exhibit such complexity, probably since this is specifically a feature of syntactic structures whose main function is to be 'informative' in some way. Issues of hypo- and parataxis, however, are not the only reasons for why we might need to postulate the existence of more independent units in spoken language than commonly posited for its written counterpart. I shall introduce further new syntactic categories in the later sections of this chapter, and also justify their independent status.

4.2 Units of analysis and frequency norming for pragmatic purposes

In general, it is common practice in corpus linguistics to report frequencies of occurrence normed by the relative number of words multiplied by an often arbitrary factor, such as 1 million (cf. Biber et al. 1998: 264). As functional utterances may consist of a variable number of words, though, and sometimes even a single word may constitute a meaning unit, this type of norming does not make much sense when investigating meaning. Apparently in an attempt to overcome this problem, Callies (2013) reports some of his findings normed by the number of turns, stating that "[i]n view of the manifold problems to operationalize the concept of sentence in transcribed spoken language and thus, to count the amount of sentences in the corpora, I chose to apply the number of speech turns as a basis of comparison" (2013: 18). However, even if the turn may already constitute a slightly better option for norming frequencies than the mere number of words, it essentially still suffers from an issue similar to the one indicated above, which is that a turn may again be made up of a variable number of meaningful units, starting with a single minimal response, but potentially encompassing a substantial number of c-units, especially if sequences of facts or actions are described. To illustrate this potential difference, we only need to look at the average number of units per turn in the Trainline, Trains and SWBD data, where the first has the smallest, 1.93, the second 2.52, while the last already amounts to 3.34. Given the fact that our three sets of data also come from three different *activity types* (see Leech et al. 2000: 8) and two of them are clearly *task-driven* (ibid.: 7), we can not only deduce that such domain-dependent factors are likely to have an influence on the number of units in a turn, with unconstrained dialogues apparently exhibiting the greatest potential for longer turns, but also that even within the same activity types, there may be considerable differences depending on the exact nature of the task. Thus, it e.g. appears that the collaborative problem solving in Trains requires slightly more elaborate turns than the interaction in the Trainline data, where the operator is perhaps more skilled in providing guidance (Weisser 2016c), and the interactional routines are more fixed. Furthermore, given the fact that the learner corpora that Callies contrasts with their native-speaker counterpart actually mix three different activity types in every dialogue (Gilquin et al. 2010), his decision seems to make even less sense in view of the discrepancies caused by different activity types we just saw.

Therefore c-units, despite potentially also being made up of a highly variable number of words, appear to be the most sensible way of counting units of meaning. This decision essentially conforms with Searle's idea discussed in 1.2 above that the speech act – and not the word – constitutes the minimal unit of meaning in a text. The fact that c-units represent meaning in either the form of a proposition, or embody some other type of textual function, makes them more meaningful as a unit of analysis than words or turns, which is why all frequencies in this study are normed according to them. And, as the lowest common denominator (see Weisser 2016a: 175) between the number of utterances is close to 5,000 units, this factor was chosen for norming to provide maximally representative frequencies.

I will mainly discuss how, or to what extent, the unit types introduced in this chapter can be recognised and annotated automatically in Chapter 6, but may give some 'hints' as and when necessary beforehand. The unit types are discussed in what would roughly be the order of processing in DART, though.

4.3 Unit types and basic pragmatic functions

As stated earlier in Section 2.3 above, it is still necessary to discuss the issue of *category design* raised by Edwards (1993), and this is the perhaps the most appropriate place to do it because the syntactic categories discussed here differ much more widely from established ones. Edwards provides the following principles for category design:

> In designing a category set for describing a dimension of interest, three properties are particularly important. First, the categories must be *systematically discriminable* in the sense that for every case in the data it is clear for every category whether or not it applies. […] Second, the categories needs [sic] to be *exhaustive*: For each particular case in the data there must be a category which fits (even if only "miscellaneous"). Third, and most importantly, the categories must be systematically *contrastive*. (1995: 5)

Each of these principles shall be adhered to in the classification procedure employed here as much as possible, although it may sometimes become necessary to strike a balance between exhaustiveness and over-generation of categories, and it may also not always be feasible to be systematically contrastive in all cases, especially because the criteria for categorisation might in effect belong to different linguistic levels and because there may be some similarity in the functions of some of the unit types.

Since I stated above that c-unit types ought to be mainly categorised in terms of their semantico-pragmatic or pragmatic content or function, this section will begin with a preliminary discussion of the different possible syntactic forms one can encounter in spoken language analysis. Furthermore, I shall also provide some basic indication as to their potentially diverse functions and how they may contribute towards establishing interactional meaning, which I shall then expand upon in more detail in later sections. That this is probably not an easy task can be seen in the following statement by Sperber and Wilson:

> What undeniably exists is not a well-defined range of syntactic sentence types but a variety of overt linguistic devices – e.g. indicative, imperative or subjunctive mood, rising or falling intonation, inverted or uninverted word order, the presence or absence of Wh-words, or of markers such as 'let's' or 'please' – which can guide the interpretation process in various ways. While it may be possible to build a theory of syntactic sentence types around these devices, as far as we know this work has not yet been done. (Sperber & Wilson 1995: 247)

This quote is, to some extent, reminiscent of Searle's idea of the speech act as the unit of linguistic communication and its potential realisations discussed in Section 1.2 above. What both of these comments on (mainly) syntactic constructs share is that they do refer to specific lexico-grammatical indicators that may, in conjunction with a particular syntactic category, help to identify the function of the unit. Fraser (1996), in his discussion of what he terms *pragmatic markers*, also refers to these as *basic markers*, and in particular distinguishes between *structural* – that is syntactic – and *lexical basic markers* – such as performatives or *pragmatic idioms* –, but also identifies *hybrid basic markers* that achieve their effect through a combination of structural and lexical markers. Essentially, we shall follow a similar line here, beginning with an overview of which syntactic unit types have which potential for being exploited in this way. The basic idea behind this is similar to the one expressed by Ariel (2008: 60):

> The function of most syntactic structures, especially of unmarked constructions, is to enable certain meanings/functions, rather than to strictly encode them. Some constructions facilitate (but by no means encode) other meanings […] Extragrammatical strategies, then, can make crucial use of the grammar without being grammatical themselves. Discourse profiles are not part of (the synchronic) grammar, but it does not mean that they have no role to play in processing utterances.

Of course, Ariel here refers to processing in the sense of human interlocutors handling and using specific grammatical markers in their processing strategies and in specific communicative settings, but one can try to exploit more or less the same strategies in computer-based processing. Therefore, having established some basic syntactic premisses or potential indicators, one can proceed to first identifying the syntactic category of a unit, look for specific lexico-grammatical or semantico-grammatical markers, and then relate the two to each other so as to identify the potential speech act expressed by the unit, whenever necessary also including other contextual information.

Still bearing in mind Edwards' criteria, we will begin by discussing the third one, that of contrastivity, and how it may apply to syntactic unit categorisation. The first thing to note here is that there can be no direct contrastivity between all units in the strictest sense because what one is dealing with here are essentially categories that are contrastive on two separate – and then further on subordinate – levels. These levels relate to Chafe's (1993: 37) distinction between *substantive* vs. *regulatory* units, in other words, between those units that are more semantics or content-oriented and those that are more pragmatics or discourse-oriented.

Amongst the former, there are the traditional 'sentence' categories of *declaratives, interrogatives, imperatives* and *exclamatives* already discussed above, as well as syntactic *fragments*, units that are incomplete for a variety of different reasons that might be related to disfluency or issues of cohesion, such as the interlocutors assuming some prior context or *common ground*. The latter comprise syntactic units that 'regulate' the interaction between the dialogue participants. Here, one encounters the categories of *yes* or *no* units/responses, *discourse markers* (henceforth DMs), and *addresses* (traditionally referred to as *vocatives*), which represent specific classes of syntactic 'objects'. They formally tend to consist of short – often single-word – constructions that are syntactically 'ill-formed' in the traditional sense, as well as often being fragmentary or elliptical with regard to their semantic content. Positionally, they mainly occur at the beginning of turns and fulfil either 'attention grabbing', responding, or topic initiating/initialising functions. Due to their largely prefatory nature, LongGr (LongGr 1073) also refers to them as "[u]tterance launchers", although this grammar also includes other non-clausal elements, such as fronted NPs, or "overtures" (LongGr 1075) – both here grouped with fragments – in this category. Each of these individual syntactic unit types will be discussed in detail in the next sections, outlining their potential for expressing specific pragmatic functions, their particular distributions in the different corpora analysed, and providing a suitable number of illustrative examples. The speech-act labels occurring in the sample illustrations, and partly introduced in the discussion, will be explained in more detail in Chapter 7, but an overview of all labels, together with brief explanations, is provided in Appendix A.

4.3.1 Yes-units

As illustrated above, traditional grammars, like CompGr, do not usually categorise *yes* units, their 'negative counterparts', the *no* units, and other short interactional units, as sentences. CompGr even explicitly classifies them as "REACTIONARY SIGNALS" under the sub-heading of "Formulae" in the section on "Nonsentences" (CompGr: 852). This would, however, seem to somehow relegate them to a lower status in the 'hierarchy' of syntactic units, purely because they do not conform to any of the 'well-formed' and established categories, and thereby also – perhaps unintentionally – de-emphasise their functional importance. That yes-units are not purely 'reactionary' and do have a very important role in keeping a conversation going, however, cannot be disputed. How exactly they do so will hopefully become clear from the following discussion.

In general, when thinking about yes-units that occur in dialogues, one usually assumes that they represent answers to requests for information, just as it is generally assumed that questions tend to express requests for information. However, looking at our data, it turns out that in two of the corpora, Trainline and Switchboard, *minimal response tokens*, i.e. basic acknowledgements of receipt of information, account for the majority of all yes-units. Only in Trains, answers are actually slightly more frequent, as can be seen clearly in Table 4.1, which provides a comparison of the major functions of yes-units, including their intra-category percentages and normed frequencies.

The minimal responses essentially come in two forms here, one where they occur directly as responses to information, and one where they represent 'echoing' responses to prior acknowledgements by the interlocutor, in which case they

Table 4.1 Main function types, intra-category percentages, and normed frequencies for yes-units

Function	Trl%[a]	Trl NFreq	Trs%	Trs NFreq	SWBD%	SWBD NFreq
minimal response	44.5%	156.88	34.0%	66.51	78.3%	238.27
'answering'	38.6%	135.96	38.1%	74.45	12.8%	39.09
accepting	3.2%	11.26	13.7%	26.80	0.7%	2.25
initiating	0.5%	1.61	1.5%	2.98	2.7%	8.27
emphatic	12.6%	44.25	7.6%	14.89	4.2%	12.78
phatic	0.7%	2.41	5.1%	9.93	1.2%	3.76

a. From now on, the corpus names in table headings will be abbreviated in order to save space. Trainline is shortened to Trl, Trains to Trs, and Switchboard to SWBD.

essentially appear to concur with the interlocutor's 'assessment'. This feature can be seen in unit 18 in Example (3) (from Switchboard), which, incidentally, also serves as an illustration of an acknowledging yes-answer (unit 17).

(3) <q-yn n="14" sp-act="reqInfo" polarity="positive" mode="benefit-closed-query">
is that pretty typical for your area <punc type="query" />
</q-yn>
</turn>
<turn n="6" speaker="1122">
<dm n="15" sp-act="muse">
well <punc type="level" />
</dm>
<frag n="16" sp-act="refer" polarity="positive" mode="partial-frag">
for the neighborhood i'm in <punc type="level" />
</frag>
<yes n="17" sp-act="answer-acknowledge">
yeah <punc type="stop" />
</yes>
</turn>
<turn n="7" speaker="1254">
<yes n="18" sp-act="echo-acknowledge">
yeah <punc type="stop" />
</yes> (sw_0087_2775)

Here, the relevant interaction sequence begins with a request for information by speaker 1254 in unit 14. Speaker 1122 is initially a little hesitant in responding (unit 15), then first prefaces the answer by a restrictive referential adverbial (unit 16), and finally provides the answer in the form of a simple *yeah* (unit 17). This *yeah* is then in turn echoed again by speaker 1254 in unit 18. We can thus represent the final part of the interaction sequence as *answer-acknowledge* → *echo-acknowledge*, a practice I shall adopt for indicating such sequences from now on.

What I have summarised under the label 'answering' above does not always only have to be a response to a request for information (*reqInfo* → *answer-acknowledge*), though, but may also constitute a response to a request for confirmation (*reqConfirm* → *confirm-acknowledge*), request for confirming an option (*reqOpt* → *stateOpt-acknowledge*; Example (4), an answering negation (*reqInfo* → *answer-negate*), a rejection (or refusal) of a request for confirming an option with negative polarity (*reqOpt* → *reject*; Example (5), a rejection of something the current speaker themselves has been deliberating and uttered in some form of a negative polarity statement (*stateX*[1] → *rejectSelf*; Example (6), or even some other form of self-correction (→ *correctSelf-acknowledge*; Example (7).

(4) <q-yn n="232" sp-act="reqOpt" polarity="positive" topic="booking" mode="request-closed-query">
<overlap pos="end" /> can i just repeat the reference please <punc type="query" />
</q-yn>
</turn>
<turn n="109" speaker="Sandra">
<yes n="233" sp-act="stateOpt-acknowledge">
yes <punc type="stop" />
</yes> (trainline04)

(5) <q-yn n="55" sp-act="reqOpt" polarity="negative" topic="booking-time-day" mode="constrain-closed-query">
can i don't have to book a time on the Sunday <punc type="query" />
</q-yn>
</turn>
<turn n="31" speaker="Sandra">
<yes n="56" sp-act="reject">
yes <punc type="stop" />
</yes>
<frag n="57" sp-act="elab-stateOpt" polarity="positive" mode="decl">
you do <punc type="stop" />
</frag> (trainline05)

(6) <decl n="58" sp-act="stateNonConstraint" polarity="negative" topic="transport_means" mode="nonconstrain-decl">
i don't need to pick up a tanker then <punc type="stop" />

1. Here X is a placeholder for one of the different types of 'statement' discussed in Chapter 7.

</decl>
<yes n="59" sp-act="rejectSelf">
<pause /> yes <punc type="stop" />
</yes>
<frag n="60" sp-act="state" polarity="positive" mode="decl">
i do <punc type="stop" />
</frag> (d93-24.3)

(7) <no n="30" sp-act="answer-negate">
no <punc type="stop" />
</no>
<dm n="31" sp-act="hesitate">
<pause /> uh
</dm>
<frag n="32" sp-act="reqConfirm" polarity="positive" mode="query">
the same way <punc type="query" />
</frag>
<yes n="33" sp-act="correctSelf-acknowledge">
<overlap pos="start" /> <pause /> yes <overlap pos="end" />
</yes> (d93-19.4)

Accepting uses of <yes> may be responses to a number of different request types, suggestions, or offers. Only explicitly marked ones, such as *yes please*, can be recognised automatically, while the remainder needs to be identified through the context, as we shall see below. Although one might assume that the surprisingly high number of accepting uses in Trains is due to the collaborative nature of the task, this is only partly true. Although the Trains data contains a mix of offers and suggestions that can potentially trigger this response, the formulaic (playbook) opening *(how) can i help you*, which usually follows a greeting at the very beginning of each dialogue, is responsible for exactly a quarter of these, whereas only the remaining three quarters are genuinely due to collaborative efforts. Despite the fact that the Trainline data is also collaborative – at least in some sense –, the number of accepting yes-units there is clearly lower. The reason for this partly lies in a different type of interaction style. Rather than making overt offers or suggestions, the agent, Sandra, often describes the options available to her callers, rather than eliciting acceptance, which the callers simply first acknowledge as potential options to accept, as can be seen in Example (8).

(8) <decl n="42" sp-act="stateOpt" polarity="positive" topic="time-enum-location_Anglo-from" mode="exists-decl">
there's a train at 14:30 from Manchester Picadilly <punc type="stop" />
</decl>
<decl n="43" sp-act="state" polarity="positive" topic="location_Anglo-arrival" mode="decl">

<pause /> you arrive in London Euston for 1700 <comment type="pron" content="17-hundred" /> <punc type="stop" />
</decl>
</turn>
<turn n="26" speaker="caller_02">
<dm n="44" sp-act="acknowledge">
ok <punc type="stop" />
</dm>
<decl n="45" sp-act="approve" polarity="positive" mode="decl">
that's great <punc type="stop" />
</decl> (trainline02)

In other words, instead of a direct *offer/suggest* → *accept* mode of interaction, the callers generally respond to the facts of the timetable and ticketing options being presented by first acknowledging them through the use of acknowledging DMs, and then apparently evaluate these before signalling their readiness to accept such options through different markers of approval or appreciation. In addition, the same triggers for accepts may of course also prompt rejects, which generally occur in no-units, so this may equally be a balancing factor, although, comparing the number of accepts and rejects, both numbers are still higher in Trains. Interestingly, though, there is even one instance of a rejection in the form of a yes-unit present in the Trainline data. As can be seen in Example (9), this occurs in response to a negative-polarity request for an option (reqOpt), and, as it is a response to a query, was grouped with the 'answering' yes-units.

(9) <q-yn n="55" sp-act="reqOpt" polarity="negative" topic="booking-time-day" mode="constrain-closed-query">
can i don't have to book a time on the Sunday <punc type="query" />
</q-yn>
</turn>
<turn n="31" speaker="Sandra">
<yes n="56" sp-act="reject">
yes <punc type="stop" />
</yes>
<frag n="57" sp-act="elab-stateOpt" polarity="positive" mode="decl">
you do <punc type="stop" />
</frag> (trainline05)

Yes-units with an initiating function are relatively rare in the all three corpora, although they do occur. If they do, they tend to follow on from some form of agreement that has been reached, or an acknowledged fact or option that provides a basis for continuation, as can be seen in Examples (10) and (11).

(10) <turn n="108" speaker="u_PH">
<q-yn n="219" sp-act="reqOpt" polarity="positive" mode="closed-query">
<overlap pos="start" /> can we go <overlap pos="end" /> back and do that
<punc type="query" />
</q-yn>
</turn>
<turn n="109" speaker="s_CB">
<dm n="220" sp-act="stateOpt-agree">
sure <punc type="stop" />
</dm>
<yes n="221" sp-act="elab-emphatic">
<pause /> <overlap pos="start" /> yep <overlap pos="end" /> <punc
type="stop" />
</yes>
</turn>
<turn n="110" speaker="u_PH">
<dm n="222" sp-act="acknowledge">
<overlap pos="start" /> okay <overlap pos="end" /> <punc type="stop" />
</dm>
</turn>
<turn n="111" speaker="s_CB">
<yes n="223" sp-act="init">
yeah <punc type="stop" />
</yes>
<dm n="224" sp-act="init">
<pause /> so <punc type="level" />
</dm>
<decl n="225" sp-act="expressPossibility" polarity="positive"
mode="poss1-decl">
we can yeah <pause /> <backchannel content="okay" /> change the plan
<punc type="stop" />
</decl>

(11) <q-yn n="117" sp-act="reqInfo-abandon" status="interrupted"
polarity="positive" mode="closed">
are you talking about like Spring Break and <punc type="incomplete" />
</q-yn>
</turn>
<turn n="37" speaker="1532">
<dm n="118" sp-act="answer-acknowledge">
right <punc type="stop" />
</dm>
</turn>
<turn n="38" speaker="1442">

<dm n="119" sp-act="echo-acknowledge">
right <punc type="level" />
</dm>
</turn>
<turn n="39" speaker="1532">
<yes n="120" sp-act="init">
yeah <punc type="level" />
</yes>
<frag n="121" sp-act="referProcess" polarity="positive" mode="frag">
getting a lot of breaks <backchannel content="right" /> between quarters
and that <punc type="level" />
</frag>
<frag n="122" sp-act="referProcess" polarity="positive" topic="leisure"
mode="decl">
and having a lot of vacation <punc type="stop" />
</frag> (sw_0059_4028)

In Example (10) from Trains, speaker u_PH first inquires after a potential option (*reqOpt*) in unit 219. Speaker s_CB then agrees with this being an option (*state-Opt-agree*) and uses an emphatic yes-unit to 'reinforce' their commitment to this fact (*elab-emphatic*; units 220 & 221). u_PH then acknowledges 'receipt of' this piece of common ground, whereupon s_CB initiates the new phase of the planning. Something very similar happens in Example (11) from Switchboard, where speaker 1442 tries to establish common ground via an interrupted and incomplete request for information (reqInfo-abandon) in unit 117. Before speaker 1442 can even finish this question, speaker 1532 already acknowledges this through the DM *right* in unit 118, which is in turn acknowledged by the interlocutor (unit 119), before speaker 1532 then launches into details related to the topic through an initiating yes-unit (120). As Table 4.1 shows, initiating yes-units occur with the highest frequency in the Switchboard data, perhaps because, there, the interlocutors tend to negotiate their (sub-)topics more frequently, due to a more flexible dialogue structure.

We have already seen an example of an emphatic use of a *yes* in unit 221 in Example (10) above. As pointed out in the discussion above, such uses tend to somehow reinforce or emphasise the commitment of the speaker to a given fact, which may have been expressed in a preceding answer, confirmation, approval, or agreement, but also potentially a general acknowledgement that does not occur in the shape of a yes-unit, but instead a DM. The high number of occurrences in Trainline is in fact related to the callers' signalling their acceptance of options by acknowledging, agreeing with, or approving of them first in some form, as referred to above, and then reinforcing the response emphatically, e.g. by saying *that's right, yeah*, etc.

The final form that yes-units can take is that of phatic signals that do not occur as a response to what the interlocutor has been saying, but mainly somewhere inside or at the end of a turn, but without the emphatic function just described. An instance of this from Trains, where these uses occur most frequently, can be seen in Example (12).

(12) <turn n="27" speaker="s_SB">
<dm n="64" sp-act="init">
so <punc type="level" />
</dm>
<yes n="65" sp-act="phatic">
<pause /> yeah <punc type="stop" />
</yes>
<dm n="66" sp-act="init">
so <punc type="level" />
</dm>
<decl n="67" sp-act="predict-abandon" status="abandon"
polarity="positive" mode="predict">
it'll take <punc type="incomplete" />
</decl>
<dm n="68" sp-act="hesitate">
um
</dm>
<dm n="69" sp-act="init">
so <punc type="level" />
</dm>
<decl n="70" sp-act="reqInfo" polarity="positive" mode="query">
you want to do what <punc type="query" />
</decl>

Unit 65 in Example (12) appears to have no direct functional meaning within the turn, but instead rather appears to act like a filler, allowing the speaker to deliberate about the potential course of action. This can be seen in speaker s_SB's repeated attempts to initiate a new stage in the planning (units 64, 66, & 69), evaluating and/or discarding particular options or conditions (as e.g. in unit 67), and then finally handing over the responsibility for deciding on an action to the interlocutor in unit 70.

Having investigated the different main uses of yes-units and their distributions in the three different corpora now, we can conclude that, although their main function undoubtedly remains to provide some form of response to something that has been said or asked, yes-units are in fact less likely to provide any real information, rather than supporting the ongoing conversation. This is perhaps also the reason why grammars tend to compare them to "backchannels" (LongGr: 1091), or refer to them as “reaction signals” (CompGr: 444), although this does not really do justice to their varied functions we have seen above. CamGr does not even accord yes-units and their ‘counterpart’, no-units, the status of separate units, but instead refers to them only as “lexically simple adverbs” (CamGr: 570) or “polarity adjunct[s]” (ibid.: 848) in responses to questions. It does, however, recognise that “*Yes* and *no* are used in response to statements” or “can be used to express intention to comply with a positive and negative directive” (ibid.), a function we did not observe in our data, but which is of course similar to the responses to offers or suggestions we discussed.

4.3.2 No-units

Even if one may intuitively presume that no-units simply form a logical counterpart to yes-units, a view perhaps enforced by more traditional grammars in stressing their function as negative responses to yes-no questions, this is just as unjustified as assuming that most yes-units constitute answers. What does become very clear soon, though, when looking at the summary of our data in Table 4.2, is that, in contrast to the same type of yes-units, no-units as minimal responses are far less common. Yet, this is perhaps not surprising, due to the inherently negative prosody of the word *no*, which hence does not lend itself readily to acting as a ‘go-on signal’ in the same way as a *yes* does. The only reason why we can still find it used in such a way in our data is that it occurs entirely in instances of *echo-negate*, where the current speaker simply acknowledges the interlocutor’s prior use of a no-unit, hence in *no* → *echo-negate* sequences.

In terms of automatic identification without taking prior context into account, the same difficulty as in distinguishing between yes-units also arises with no-units. Only explicit rejection, *no thanks/thank you*, can be unambiguously identified automatically in DART, whereas all other cases initially need to be treated as negating in the current implementation and then hand-corrected during a post-editing phase. However, future implementations of DART may also implement further contextual modelling, as e.g. already does exist in identifying responses to queries of various forms. Table 4.2 again summarises the main functions I identified after disambiguation, once more including their intra-category percentages and normed frequencies.

As far as ‘answering’ usages are concerned, we essentially find the counterparts to the corresponding yes-units, *answer-negate* (responding to requests for information) and *confirm-negate*, with the addition of *disConfirm* across all three corpora. The difference between the latter two, which both constitute responses to requests for confirmation (*reqConfirm*), is that *confirm-negate*s confirm

Table 4.2 Main function types, intra-category percentages and normed frequencies for no-units

Function	Trl%	Trl NFreq	Trs%	Trs NFreq	SWBD%	SWBD NFreq
minimal response	6.4%	5.63	2.1%	0.99	14.6%	4.51
'answering'	62%	54.71	38.3%	17.87	46.3%	14.28
rejecting	7.3%	6.44	10.6%	4.96	–	–
rejecting oneself	3.6%	3.22	23.4%	10.92	2.4%	0.75
emphatic	15.5%	13.68	17.0%	7.94	9.8%	3.01
initiating	1.8%	1.6	–	–	17.1%	5.26
phatic	0.9%	0.80	–	–	2.4%	0.75

negative-polarity requests, while *disConfirm* does the same for positive-polarity ones. Examples (13) and (14) illustrate this difference.

(13) <decl n="201" sp-act="**reqConfirm**" polarity="negative" mode="query">
he **didn't** go howl <punc type="query" />
</decl>
</turn>
<turn n="63" speaker="1285">
<no n="202" sp-act="confirm-negate">
no <punc type="incomplete" />
</no>
<decl n="203" sp-act="elab-state" polarity="negative" mode="decl">
he wouldn't go howl <punc type="stop" />
</decl> (sw_0088_3073)

(14) <decl n="19" sp-act="**reqConfirm**" polarity="positive" topic="enum" mode="query">
you want to drop off all 3 <punc type="query" />
</decl>
</turn>
<turn n="8" speaker="u_CK">
<no n="20" sp-act="disConfirm">
no <punc type="stop" />
</no>
<decl n="21" sp-act="stateIntent" polarity="positive" topic="enum" mode="preference1-intent-decl">
i just want to drop off 1 <punc type="stop" />
</decl> (d93-24.5)

However, a disconfirming no-unit can also follow a unit that does not contain a *reqConfirm* if that unit is a declarative that embodies an assumption by the interlocutor in the form of *i thought…*, in which case it is of course, strictly speaking, not answering. Other types of 'assumptions', such as predictions, etc., instead get contradicted by *no*, and this will generally initiate a follow-up correction, as in Example (15) from Trains, which is the only corpus that contains such contradictions.

(15) <frag n="29" sp-act="**predict**" polarity="positive" topic="duration-enum" mode="decl">
and the travel back would be <pause /> another 7 hours <punc type="stop" />
</frag>
</turn>
<turn n="7" speaker="s_AM">
<dm n="30" sp-act="hesitate">
um
</dm>
<no n="31" sp-act="contradict">
<pause /> no <punc type="stop" />
</no>
<frag n="32" sp-act="referDirection" polarity="positive" mode="partial-frag">
on the way back <punc type="level" />
</frag>
<decl n="33" sp-act="**correct-predict**" polarity="positive" topic="enum" mode="limitation-frag">
it would only be 6 <punc type="level" />
</decl>
<decl n="34" sp-act="stateReason" polarity="negative" mode="nonconstrain-reason-decl">
because you wouldn't need to load <punc type="stop" />
</decl> (d93-16.1)

In unit 29, speaker u_CK[2] first predicts a certain duration. Speaker s_Am then hesitates (unit 30) while evaluating this prediction, but then contradicts it in unit 31 and, after re-contextualising it (unit 32), corrects the prediction (unit 33), and even provides a reason for the correction in unit 34.

Returning to the category of 'answering' no-units, the Trainline data exhibits one additional category that cannot be found in the other data, *direct-negate*. This category occurs in response to a request for a directive, expressed either in the form of a yes-no or wh-question, and is very specific to the interaction style in Trainline, where the query is designed to elicit a directive from the caller that allows the agent to gather information and carry out part of the booking procedure, directly

2. Speaker ID not shown in the example.

based on this information. A request for a directive is therefore much like an offer in question form, as can be seen in Example (16).

(16) <turn n="9" speaker="Sandra">
<q-wh n="15" sp-act="reqDirect" polarity="positive" topic="time" mode="preference2-open-query">
and when is it you'd like to return <punc type="query" />
</q-wh>
</turn>
<turn n="10" speaker="caller_30">
<dm n="16" sp-act="hesitate">
er
</dm>
<no n="17" sp-act="direct-negate">
no <punc type="level" />
</no>
<no n="18" sp-act="emphatic">
no <punc type="level" />
</no>
<decl n="19" sp-act="stateIntent" polarity="negative" topic="return" mode="intent-decl">
i won't be returning <punc type="stop" /> (trainline30)

As can be seen in Example (16), the *direct-negate* in unit 17 is not a rejection, but simply corrects an implicit assumption expressed in Sandra's request for directive that caller_30 actually does want to return by train as well.

Rejecting no-units fall into one of three different categories: those occurring as a response to a request for information (*answer-reject*) or confirmation (*confirm-reject*; Example (17)), those responding to non-elicited options or assumptions that are part of the common ground or ongoing planning (*reject*; Example (18)) and introduced by the interlocutor, and finally those where a given speaker rejects assumptions or options raised by themselves (Example (19)). The first of these is only present in the Trainline data, while Switchboard only contains one example of the last one.

(17) <turn n="35" speaker="Sandra">
<decl n="61" sp-act="**reqConfirm**" polarity="negative" topic="booking" mode="query">
you don't want to book at all <punc type="query" />
</decl>
</turn>
<turn n="36" speaker="caller_02">
<no n="62" sp-act="confirm-reject">
no thank you <punc type="stop" />
</no> (trainline02)

(18) <q-wh n="96" sp-act="reqInfo" polarity="positive" topic="location-to-from" mode="open-query">
how long would it take to get from Dansville to Corning <punc type="query" />
</q-wh>
</turn>
<turn n="31" speaker="s_NB">
<frag n="97" sp-act="answer-referDuration" polarity="positive" topic="enum-time-duration" mode="decl">
1 hour <punc type="stop" />
</frag>
</turn>
<turn n="32" speaker="u_CK">
<dm n="98" sp-act="acknowledge">
okay <punc type="stop" />
</dm>
<dm n="99" sp-act="hesitate">
<pause /> um
</dm>
<no n="100" sp-act="reject">
<pause /> no <punc type="stop" />
</no>
<dm n="101" sp-act="hold">
<pause /> let's see <punc type="level" />
</dm>
<decl n="102" sp-act="stateConstraint-abandon" status="abandon" polarity="positive" mode="constrain">
we need to <punc type="incomplete" />
</decl>
<decl n="103" sp-act="stateConstraint" polarity="positive" topic="location-transport_means" mode="suggest-decl">
<pause /> we should have both <pause /> of the boxcars at Dansville by noon <punc type="stop" />
</decl> (d93-14.3)

(19) <q-wh n="276" sp-act="suggest" polarity="positive" topic="location-to-transport_means" mode="closed-query">
how about we go back to Elmira <pause /> and then <pause /> fill up with the orange <punc type="query" />
</q-wh>
<dm n="277" sp-act="expressSurprise">
<pause /> ooh
</dm>
<no n="278" sp-act="rejectSelf">

```
no <punc type="stop" />
</no>
<no n="279" sp-act="emphatic">
no <punc type="stop" />
</no>
<no n="280" sp-act="emphatic">
no <punc type="stop" />
</no>
<dm n="281" sp-act="exclaim">
<pause /> gracious <punc type="stop" />
</dm>
<dm n="282" sp-act="exclaim">
gracious <punc type="stop" />
</dm>
<decl n="283" sp-act="state" polarity="positive" mode="decl-disflu">
<pause /> bad bad move <punc type="stop" />
</decl>                                                    (d93-22.2)
```

In Example (17), Sandra requests a confirmation regarding the booking intentions of caller_02 (unit 61), and the caller then rejects this option, at the same time confirming Sandra's assumptions. Note that the caller here uses an explicit rejection, adding *thank you*, which allows DART to disambiguate between a simple negation and a rejection. Interestingly, although such *reqInfo/reqConfirm → confirm-reject* sequences should be relatively common in any type of negotiation between interlocutors, the small amount of examples in our data are restricted to the Trainline data in their occurrence.

Example (18) is somewhat more complex. Here, speaker u_CK first enquires about the duration of one of the legs of the transport journey (unit 06), to which speaker s_NB provides an answer (unit 97). After initially acknowledging receipt of this information, u_CK hesitates, apparently digesting the information, and then rejects this information as not representing a valid option (unit 100), followed by listing a number of constraints that serve as an explanation for the rejection, of which only the first two units are shown here.

At the beginning of Example (19), speaker u_JH starts by making a suggestion in unit 276, but almost immediately rejects it (unit 278) very emphatically, as evidenced by the two ensuing emphatic no-units, after first uttering an expression of surprise, apparently having realised her mistake. The emphatic nature of her self-rejection is made even more evident through the two exclamatives that precede the 'explanation' of the rejection in unit 283, where she classifies her previous suggestion as a *bad bad move*.

As previously in our data for yes-units, sometimes polarity may also come into play and provide some perhaps rather less-expected examples of responding no-units. For instance, Trainline contains two instances of accepting (*answer-accept*; 1.8%; Example (20)), and SWBD three instances of agreeing no-units (*agree*; accounting for 7.3%; Example (21)).

(20) <turn n="21" speaker="Sandra">
<decl n="53" sp-act="stateOpt" polarity="positive" topic="time-enum" mode="exists-decl">
there's a 7:15 in the morning <punc type="stop" />
</decl>
<q-yn n="54" sp-act="reqInfo" polarity="positive" topic="time" mode="alternative-closed-query">
or is that going to be **too early** <punc type="query" />
</q-yn>
</turn>
<turn n="22" speaker="caller_11">
<no n="55" sp-act="answer-accept">
no <punc type="stop" />
</no>
<decl n="56" sp-act="stateIntent" polarity="positive" mode="intent-decl">
i'll get on the 7:15 <vocal content="laughter" /> <punc type="stop" />
</decl> (trainline11)

(21) <decl n="140" sp-act="state" **polarity="negative"** mode="decl">
Guns and Roses **doesn't** have too much of a message <comment type="restart" /> too much to say <punc type="stop" />
</decl>
</turn>
<turn n="44" speaker="1496">
<frag n="141" sp-act="acknowledge" polarity="positive">
<comment content="laughter" />
</frag>
<no n="142" sp-act="agree">
no <punc type="level" />
</no>
<no n="143" sp-act="emphatic">
no <punc type="level" />
</no>
<decl n="144" sp-act="state" **polarity="negative"** mode="preference2-decl">
they **don't** have anything really important you want to hear <punc type="stop" />
</decl>

Interestingly, in Example (20), the polarity feature that projects a potential rejection by caller_11 does not originate in any kind of surface-polarity signal in the form of negation, but is instead due to the negative prosody of the relational adverb *too* in unit 54. Sandra initially states a potential option in terms of the availability of a particular connection (unit 53), but then literally 'questions' the suitability of that option, thereby giving the caller the option to reject it, and mitigating a potential threat to their *negative face* (cf. Brown & Levinson 1987: 62ff.). Against the background of the negative prosody, however, the acceptance in what would otherwise be a negative form now becomes perfectly congruent, so we could even refer to this phenomenon as *(pragmatic) prosodic harmony*. The same kind of harmony essentially exists in Example (21), with the main difference that the negative prosody is more explicit, due the use of *don't*, uttered by both interlocutors (in units 140 and 144), as it were 'framing' speaker 1496's agreement.

Basically, emphatic no-units are very similar to their yes-counterparts, only that, in all three corpora, they are actually far less frequent in terms of their normed frequencies. We can therefore cautiously assume that emphasising a no-unit, due to its stronger negative and insisting effect, is somewhat dispreferred. This appears to be especially the case in unconstrained dialogue as, in the SWBD data, the ratio of empathic yes- to no-units is about 4:1, whereas, in Trainline, it is around 3:1, and in Trains even close to 2:1. Thus, perhaps emphasising no-units do become more acceptable the more collaborative the interaction is, and the more making joint decisions becomes an important factor.

No-units with an initiating function can be *echo-reqConfirm* (1.8% in Trainline; Example (22)), or regular *init*s, similar to the yes-units described in the previous section (SWBD).

(22) <turn n="117" speaker="Sandra">
<dm n="236" sp-act="init">
now <punc type="level" />
</dm>
<decl n="237" sp-act="state" polarity="positive" topic="journey-railcard" mode="decl">
you say that the person's travelling has a disabled railcard <punc type="stop" />
</decl>
<q-yn n="238" sp-act="reqInfo" polarity="positive" mode="closed-query">
<pause /> do they require assistance at all <punc type="query" />
</q-yn>
</turn>
<turn n="118" speaker="caller_28">
<no n="239" sp-act="answer-negate">
no <punc type="level" />
</no>
<no n="240" sp-act="emphatic">
no <punc type="level" />

</no>
<no n="241" sp-act="emphatic">
no <punc type="stop" />
</no>
</turn>
<turn n="119" speaker="Sandra">
<no n="242" sp-act="echo-reqConfirm" mode="query">
no <punc type="query" />
</no>
</turn>
<turn n="120" speaker="caller_28">
<no n="243" sp-act="confirm-negate">
no <punc type="stop" />
</no>
</turn>
<turn n="121" speaker="Sandra">
<decl n="244" sp-act="approve" polarity="positive" mode="reassurance-decl">
that's fine <punc type="stop" />
</decl> (trainline28)

(23) <decl n="71" sp-act="refer" polarity="positive" mode="frag">
the monotonous <punc type="level" />
</decl>
<no n="72" sp-act="init">
no <punc type="stop" />
</no>
<decl n="73" sp-act="state-abandon" status="abandon"
polarity="negative">
that's **not** <punc type="incomplete" />
</decl>
<decl n="74" sp-act="expressDislike" **polarity="negative"** topic="time"
mode="dispreference1-decl">
i **don't** enjoy that at all <backchannel content="aha" /> when they're just
talking about <quote>well i'm better than you and blah blah blah</quote>
<punc type="stop" />
</decl> (sw_0093_3227)

Example (22) is a very good illustration of initiating no-units in situations where no-units occur as requests for confirmation because, in this case, the no-unit in fact initiates a continuation phase as part of a verification process. In such cases, a default assumption (implicit in units 237–238) on the part of one of the interlocutors, here the agent Sandra, is negated by the other interlocutor (units 239–241). The person whose default assumption has, probably to their surprise, been violated then initiates a verification phase (unit 242) that is completed by a response of the interlocutor (unit 243) finally being acknowledged by the initiator (unit 244).

In other cases, such as in Example (23), the exact reason for a speaker using a no-unit to initiate a sequence is frequently not as clear-cut as for the corresponding yes-units. In this example, the speaker uses a deictic reference in order to – presumably – raise something as an issue or at least establish a focus on it (unit 71). This is then followed by a sequence of units discussing that particular point that share a common prosodic negative harmony.

Phatic uses of no-units are relatively rare, but in both instances occurring in the data, one in Trainline and Trains each, invariably follow an acknowledging yes-unit, as if the speaker were somehow unsure about the acknowledgement.

One other minor use is that of a no-unit expressing surprise (*expressSurprise*), or rather perhaps incredulity in the sequence of *oh* followed by *no*, where there is a tight prosodic link between the two, as opposed to an expressSurprise followed by a *no* expressing a *rejectSelf*, as shown in Example (24).

(24) <dm n="288" sp-act="expressSurprise">
oh
</dm>
<no n="289" sp-act="rejectSelf">
no <punc type="stop" />
</no>
<decl n="290" sp-act="state" polarity="positive" topic="enum-transport_means" mode="limitation-decl">
i only <backchannel content="okay" /> need 1 <pause /> 1 boxcar
<punc type="stop" />
</decl> (d93-22.2)

As the preceding discussion has hopefully shown, the options for using no-units to achieve different communicative purposes are again rather more varied than standard grammars may lead us to believe.

4.3.3 Discourse markers

Discourse markers (DMs), sometimes also referred to as *discourse particles*, especially if they are mono-morphemic, have been extensively discussed in the literature (cf. Schiffrin 1987; Jucker 1993; Jucker & Ziv 1998; Fraser 1996, 1999 & 2006; Aijmer 2002; Fischer 2006), but there is still relatively little agreement which words or phrases exactly belong into this category and which do not (cf. Jucker 1993). LongGr subsumes them under *inserts*, but that particular linguistic category itself is rather ill defined as a “word class” (LongGr: 1043), when in fact the definitions LongGr offers for inserts frequently include compound expressions that contain multiple 'words', and do not distinguish sensibly between their various types and associated functions. Often, *discourse connectors*, such as *therefore, however*, etc., are included under this heading, as for example in the definition of DMs by Fraser (1996, 1999, or 2006), or even what may be called 'pseudo-DMs', such as *like*. Adolphs and Carter (2013: 71) also include expressions like *I guess/think* amongst them, which, however, is a rather unfortunate choice because DMs can generally be assumed to function as independent c-units that do not integrate with other units and may often even be left out without changing the meaning of what is being said too much, while said expressions actually do require syntactic complementation. This is the case even if they do not always occur at the beginning of syntactic units, and may thus appear more like an afterthought in some cases.

An exact definition or exhaustive listing of particular DMs, however, is not important for our purpose here because it is possible to provide a broad framework for the description of their function without this. Nevertheless, we can at least list a few distinguishing criteria that may help us to differentiate them from other types of c-units. In the first instance, as pointed out above, DMs often consist of only a single, and relatively invariant, word form, such as *well, now*, etc., and often act as *prefaces* (cf. LongGr: 1072) to other units. If they formally correspond to adverbs or conjunctions, then their core function as a DM will differ from the adverbial/linking one, often making little or no contribution to propositional semantics, with their main purpose being either to express some form of attitudinal (e.g. *frankly, honestly*) or epistemic (e.g. *definitely, obviously*) marking, or to create/enhance textual cohesion and/or coherence (e.g. *however, well, now*). In the first case, they belong to the ideational or interpersonal level, else they represent features on the interactional level. For instance, the difference between *now* as a temporal adverb and as a DM is that the former clearly refers to the 'current moment in time', while the main function of the latter is to signal the beginning of a new (sub-)topic, perhaps retaining some small part of the original temporal meaning. This functional difference can also be seen in the fact that *now* may combine with another originally temporal adverb, *then*, in the DM *now then*, where the semantics of the two adverbs would in fact clash with one another and thus ought to be deemed incompatible from the point of view of formal compositional semantics.

Prosodic features frequently also play an important role in identifying either whether an ambiguous form represents a DM, or in distinguishing between the different potential functions of such a DM. For instance, the difference between *so* as a logical connector and a DM (see below) may be indicated via the presence or absence of a prosodic break between *so* and the following word and/or its degree of phonological (non-)reduction. Thus, if *so* is followed by a prosodic break that is either indicated through a short pause or a pitch reset, as well as having a longer duration, and without exhibiting any phonological reduction, it is most likely a DM. In contrast, if *so* is prosodically cohesive with the following word in that there is no break in the pitch pattern, there is linking present if the following word starts in a vowel, and/or it is phonologically reduced, then we can assume that it functions as a logical connector. In other words, *so* as a logical connector is characterised by the typical features pointed out by Halliday and Hasan for the group of "CONTINUATIVE items" (1976: 268) that are normally de-accented. Apart from the presence or absence of continuative prosodic marking, the exact meaning may sometimes also be dependent on the different options for variability of the pitch contour itself.

In general, I draw a distinction here between two different types of discourse markers, those that keep a dialogue going by responding to what the other speaker has said – often somewhat erroneously subsumed under the rather vague category of *response tokens* in the literature (e.g. in Adolphs & Carter 2013: 56–57) – and those that have an initiating/initialising function with regard to a new discourse stage. Such a stage may variably be referred to as a *game, transaction*, or simply *sub-dialogue*.

Some basic discourse connectors, such as *and, but, because*, or *if*, are specifically excluded from my definition of DMs, although they certainly play an important role in determining the verbal interaction between interlocutors. The role they play, however, is more on the semantico-pragmatic or lexical, rather than a functional syntactic, level, which is why most of them will be referred to in more detail in the context of semantico-pragmatics in Section 5.2 below. While, in general, my chunking practices – and probably intuitions – for splitting turns into c-units are very similar to those of Sinclair & Mauranen (2006), this is one area where they diverge. In the same way, my definition of DMs also differs from that of Schiffrin (1987), who includes *and, but*, and *or* in her discussion, due to their functions in establishing coherence relations, but disregarding their phonological properties that disallow them to act as independent c-units. Similar functional and/or phonological considerations also lead me to discard certain items, such as *just* and *sort of*, that Aijmer (2002) includes in her discussion of discourse particles for my scheme, as they neither respond nor initiate, nor in fact are phonologically separated – or separable – from their context, and therefore, again, cannot act as independent units. And although some DMs, such as *however*, can certainly appear in positions where they appear to be embedded in other units, they nevertheless need to be considered independent units as they are always clearly marked as 'parentheticals' by prosodic means such as pauses and pitch resets. While such instances cannot always easily be identified and marked up in an automatic analysis, which is why DART currently only marks up DMs that precede other units, their semantico-pragmatic meaning potential can, provided that they can contextually be disambiguated, still be taken into consideration in other ways, for instance by treating them as IFIDs that can be used in DART's inferencing procedures.

I would argue, though, that if such DMs do not occur initially, they might not in fact be as relevant to an interaction in terms of their cohesive effect. In other words, they may still affect the overall coherence of the text, but not necessarily its cohesion, to the same extent, as they then tend to lose their 'marking' effect, at least to some degree. By thus relating the marking function of DMs to the notion of *markedness*, we can perhaps also explain the fact that textual coherence is often not affected if (certain) DMs are simply deleted; instead of being required to use DMs, it is at least partly because speakers may choose to use them to mark particular stages or parts of the interaction that they can become valuable rhetorical devices (cf. Aijmer 2002: 15–16). Using DMs efficiently to delimit parts of the interaction thus becomes a valuable skill that may need to be acquired, just like using connectors to enhance cohesion and coherence in writing. In addition, the fact that most DMs tend to be very brief makes them perfect devices to signal initial quick responses to queries or facts introduced into the common ground, as we have already seen for yes- and no-units, rather than having to respond immediately with a 'full-blown' proposition.

Table 4.3 again summarises the main function types of DMs, along with their intra-category percentages and normed frequencies for comparison.

Table 4.3 Main function types, intra-category percentages and normed frequencies for DMs

Function	Trl%	Trl NFreq	Trs%	Trs NFreq	SWBD%	SWBD NFreq
initiating	35.79%	433.70	23.84%	523.03	14.80%	233.01
'delaying'	23.37%	283.23	24.42%	536.92	54.65%	860.64
'acknowledging'	29.95%	362.89	35.53%	781.06	17.04%	268.34
accepting	–	–	1.20%	25.81	0.14%	2.25
'answering'	2.26%	27.36	4.65%	102.22	1.48%	23.30
'reacting'	5.31%	64.37	4.24%	93.29	7.16%	112.75
'attitudinal'	2.66%	32.19	0.23%	4.96	1.38%	21.80
textual	0.66%	8.05	5.96%	131.00	3.34%	52.62

Initiating DMs are partly similar to their yes- or no-unit counterparts in that the majority of them simply marks the beginning of a new stage in the interaction (*init*). In this capacity, they thus represent what Aijmer (2002: 57) calls "topic-changer[s]", and are normally represented by expressions such as *now, alright, so,* *well then, basically, essentially, by the way, anyway, in any case, you know what, here/there we go*, etc. They are also generally preceded by a pause if they occur inside the turn, but may also occasionally occur with a preceding coordinating conjunction. Interestingly, Fraser, in his discussion of pragmatic markers, explicitly lists these markers not as DMs, but as *conversational management markers* amongst the group of *parallel pragmatic markers*, which, according to him, "signal a message separate from the basis [sic] message" (2006: 190). Talking about a "message", however, to my mind, would accord most of these expressions too much of a 'semantic status', something which does not seem to be warranted by their actual semantic (or propositional) content, when their function, as Fraser rightly realises, is definitely one that helps to manage the interaction. Examples (25)–(27) provide some illustrations of such initiating DMs from all three corpora.

(25) <q-yn n="2" sp-act="offer" polarity="positive" mode="benefit2-closed-query">
<pause /> can i help you <punc type="query" />
</q-yn>
</turn>
<turn n="2" speaker="U_PH">
<yes n="3" sp-act="accept">
yes <punc type="stop" />
</yes>
<dm n="4" sp-act="init">
okay <vocal content="brth" /> <punc type="level" />
</dm>
<dm n="5" sp-act="hesitate">
<pause /> um
</dm>
<decl n="6" sp-act="suggest" polarity="positive" topic="departure-to" mode="suggest-decl">
both boxcars <pause /> and both engines should leave Elmira <pause /> and go to Corning <punc type="stop" />
</decl> (d93-10.2)

(26) <turn n="5" speaker="1610">
<decl n="11" sp-act="state" polarity="positive" topic="location" mode="decl-disflu">
it's <comment type="restart" /> uh it's really bad here <punc type="stop" />
</decl>
<dm n="12" sp-act="hesitate">
uh
</dm>
<dm n="13" sp-act="init">
for example <comment content="throatclearing" /> <punc type="level" />

</dm>
<decl n="14" sp-act="refer" polarity="positive" topic="education" mode="frag">
the uh local high school <punc type="level" />
</decl>
<dm n="15" sp-act="hesitate">
uh
</dm>
<decl n="16" sp-act="state" polarity="positive" topic="enum-education" mode="decl">
they've already found 2 students with sawed off shotguns <punc type="stop" />
</decl> (sw_0022_4320)

(27) <dm n="37" sp-act="init">
<pause /> so <punc type="level" />
</dm>
<dm n="38" sp-act="init">
basically <punc type="level" />
</dm>
<decl n="39" sp-act="stateCondition" polarity="positive" topic="journey-day" mode="condition-frag">
if it's the Sunday you were travelling back <punc type="level" />
</decl>
<decl n="40" sp-act="stateOpt" polarity="positive" topic="time" mode="decl">
you come back at any time <punc type="stop" />
</decl> (trainline18)

In Example (25), speaker u_PH has already accepted the interlocutor's offer of help, so the following DM in unit 4 can no longer be interpreted as acknowledging the offer and hence has to be seen as signalling the beginning of a new stage in the interaction. In this case, the new stage proper begins by stating an objective in the form of a suggestion.

Speaker 1610 in Example (26) first raises an issue (unit 11) in referring to the "bad" situation where they live, and then, after a brief hesitation signal (unit 12), initiates a stage of description (unit 13) exemplifying one particular aspect of this issue (units 14–16).

In Example (27), the initiation of an ensuing explanation actually occurs in two steps, first via an initiating use of *so* (unit 37), which is followed by another initiating DM, *basically* (unit 38), before Sandra finally provides a precondition (unit 39) for an available option (unit 40).

In addition to simply – and relatively neutrally – indicating a change of topic, initiating DMs may also fulfil a variety of other initiating functions that all more or less respond to some information present in the common ground, but are not triggered by any form of request. Thus, for instance, expressions like *sorry, pardon (me), i beg your pardon,* or even initial *huh* (*pardon*), or *ok, ok now/then* (*reqConfirm*), realised with a querying intonation, all invite the interlocutor to respond, in the case of the former by providing further clarifying information, or, for the latter, to acknowledge that all or part of the interaction has been completed. What is labelled *pardon* in the DART annotation is a polite – or, in the case of *huh*, not so polite, perhaps more colloquial – signal of non-understanding that needs to be distinguished from the non-interrogative use of *sorry*, labelled *expressRegret* in the DART scheme. Of course, *huh* can equally be used to signal a request for confirmation, but then it can no longer exist as an independent unit, but instead has to be added as a tag. Other forms of initiation that do not occur with querying intonation serve to foreshadow or ground upcoming propositional content. The forms identified in the three corpora are labelled using the following list of speech acts shown and illustrated in Table 4.4

Table 4.4 Proposition-signalling initiating DMs

Speech act	Expansion	Expressions
initCounterExp	initiate counter-expectation	*actually, in fact, (the) fact is, as a matter of fact, the fact of the matter is, (but) the thing is (that)*
initContrast	initiate contrasting information	*however, nevertheless, yet, instead (not followed by of), besides, in contrast, on the one/other hand, that (being) said, having said that, be that as it may, even then, for 1 /one/another thing*
initReason	initiate reason	*in (that/which) case, that being the case*
initOpinion	initiate opinion	*as i/we see it, as far as i/we 'm/'re/am/are concerned, (at least) in my/our opinion, (at least) to my/our mind*
initQ	initiate question	*my/our/the/their (most) 1st/2nd/3rd/first/second/third/ big/biggest/crucial/final/important/last/pressing/serious/ urgent question is*

As the patterns representing them make it easy enough to understand how these expressions function in setting up the various proposition-modifying options, I shall not provide examples from the corpora here. Instead, I want to discuss the individual functions a little further and contrast them, as and when applicable, with alternative options for modifying propositional content that I do not consider to be DMs, but that have been treated as such in the literature. What is hopefully easy to appreciate regarding all the above DMs is that they do fulfil my criterion of being prosodically separated from the following units, and, incidentally, in most cases the subjects of these units or the beginning of a that-clause. Regarding *initContrast, initReason*, and *initOpinion*, we also find equivalents that have been treated as DMs in the literature, but whose prosodic behaviour generally does not allow them to act as independent units, i.e. *but, cause/because*, and *i think*. I deliberately said "generally" here because evidently *i think* may also occur in final position as a kind of afterthought, in which case it actually is prosodically separated from the prior propositional content. *Init-CounterExp* is different in that there is no single-word equivalent that would serve as a discourse connector. These inits are often referred to as *evidentials* in the literature, and Aijmer (2002: 251) also calls *actually* an "expectation marker", which, to my mind, is a slight misnomer because her term does not include any reference to the fact that whatever is being signalled in fact runs *counter* to the (assumed) expectations of the interlocutor, something that Aijmer actually does state herself. The final DM listed above, *initQ*, also has no single-word equivalent, and is also distinguished from the other initiating DMs through the fact that it initiates a question. Its only true equivalent would really be the question itself, but using this question without the 'preface' would remove the option to mark its importance through the use of appropriate adjectives or to indicate that it may be only one out of a whole series of questions.

In contrast to the general initiating DMs, such as *so* and *now*, the proposition-modifying ones are relatively rare in our corpus data, but we can assume that they would be far more common in written language, where they would help to enhance the potential argumentative nature of a text. In the Trainline data, the basic initiating DMs account for the highest overall intra-category percentage of all DMs, clearly indicating that the interactions between Sandra and her callers occur in relatively fixed stages, while in Trains, they come second to acknowledging DMs, presumably because the structure, partly due to the varying complexity of the individual tasks, is less fixed and therefore more acknowledgements of suggestions are required. In Switchboard, however, initiating DMs appear to play a far less important role, again partly possibly because the interaction structure is even less fixed than in Trains, but predominantly due to the fact the clear majority of DMs in Switchboard is of the 'delaying' type discussed next.

My definition of 'delaying' DMs includes a number of slightly disparate features that, however, have one thing in common – they allow the speaker to gain time to either reflect upon what to say or actually do something physically. What may come as the biggest surprise here – and might well be seen as somewhat contentious – is that I include filled pauses under the label *hesitate*. However, they represent such a prevalent and important floor-holding signal in interaction that I consider this decision justified, at least again if they occur in initial position, in other words before the current speaker has finished planning what to say. In contrast, filled pauses within c-units are treated as minor dysfluencies and therefore not accorded any specific status.

As with yes- and no-units, we also again encounter *phatic* uses of DMs that, due to the fact that that they occur in positions where neither initiation nor any form of response is possible or necessary, need to be deemed redundant and can thus be interpreted as more or less strategically employed 'fillers'. Under phatic DMs, I also include some expressions like *(and/but) i mean, (and/but) you know, you see*, and *like*, which other researchers (e.g. Schiffrin 1987 or Jucker and Smith 1998) have previously ascribed more meaning to, apparently misled by their potential to also enter into more meaningful constructions. Thus, of course, *you know* may occur as a shortened form of *do you know?* or *i mean* may be part of an explication as in *i mean the one on the left*, etc., but claiming that they either appeal to common ground or shared knowledge, or always initiate explanatory sequences, is simply not true for most of their occurrences, and here again the criterion of prosodic cohesion may be invoked in disambiguating between their uses as DMs and their 'embedded' uses.

Another potentially contentious choice is to include the main usage of *well* in this category, using the label *muse*. This usage is, however, distinct from *well* in a purely initiating function, where the unambiguous form already listed above would in fact be *well then*. In this sense, my interpretation of this DM differs slightly from that of Jucker, who seeks to find one single, over-arching interpretation of its use as a DM from the point of view of relevance theory, claiming that "[i]t signals that the context created by an utterance may not be the most relevant one for the interpretation of the next utterance. The discourse marker *well* is used to indicate a shift in the relevant context." (1993: 450–451). This interpretation, however, does not seem to fit the use of *well* with an initiating function, as can be seen in Example (28).

(28) <turn n="33" speaker="s_MF">
<frag n="83" sp-act="referProcess" polarity="positive" topic="to-location" mode="frag-disflu">
<overlap pos="start" /> to go <pause /> <overlap pos="end" /> to go <pause /> to <pause /> A… Avon <punc type="level" />
</frag>
<frag n="84" sp-act="referDirection" polarity="positive" topic="location" mode="partial-frag">
Dansville Corning <punc type="level" />
</frag>
<decl n="85" sp-act="state" polarity="positive" topic="location-to" mode="decl">
that's a bit faster than going <backchannel content="okay" /> uh than going down to uh <pause /> to Bath <punc type="stop" />

</decl>
</turn>
<turn n="34" speaker="u_CK">
<frag n="86" sp-act="abandon" status="interrupted" polarity="positive">
<overlap pos="start" /> o… <overlap pos="end" />
</frag>
</turn>
<turn n="35" speaker="s_MF">
<decl n="87" sp-act="stateDuration" polarity="positive" topic="enum-time-duration" mode="decl">
<overlap pos="start" /> it's 1 <overlap pos="end" /> hour faster <punc type="stop" />
</decl>
</turn>
<turn n="36" speaker="u_CK">
<dm n="88" sp-act="acknowledge">
alright <punc type="stop" />
</dm>
<dm n="89" sp-act="init">
<pause /> well then <punc type="level" />
</dm>
<decl n="90" sp-act="approve" polarity="positive" mode="decl">
that sounds good <punc type="stop" />
</decl>
<decl n="91" sp-act="stateIntent" polarity="positive" mode="intent-decl">
we'll do that <punc type="stop" />
</decl> (d93-17.1)

In this example, the speakers are trying to establish the fastest option for going from one place to another, and initially speaker s_MF attempts to state such an option (units 83–85), but u_CK first appears to have some reservation about this, as she attempts to interrupt (unit 86) unsuccessfully. s_MF disregards this attempt at taking over the floor and continues by actually stating the shorter duration that justifies her suggesting this route (unit 87). Apparently convinced now, u_CK then acknowledges the option and initiates the next stage (unit 89) where she signals her approval (unit 90) and the intention to go along with s_MF's plan (unit 91). Thus, even if a new stage is entered here, the relevant context itself is still preserved and has by no means undergone a shift.

In contrast, in the *muse* usage, what Jucker interprets as a shift in context rather seems to indicate a hesitation on the part of the current speaker. This hesitation could either be due to a need to deliberate on what to say in general or, if the speaker assumes that the answer would be dispreferred, whether it might not be possible to say it in a way that 'softens the blow' and eliminates a potential face-threat, for either of the two parties concerned.

(29) <turn n="66" speaker="u_JA">
<q-yn n="159" sp-act="reqInfo" polarity="positive" mode="closed-query">
am i <pause /> finished <punc type="query" />
</q-yn>
</turn>
<turn n="67" speaker="s_RD">
<dm n="160" sp-act="hesitate">
um
</dm>
<dm n="161" sp-act="muse">
well <punc type="stop" />
</dm>
<dm n="162" sp-act="hold">
let's see <punc type="level" />
</dm>
<frag n="163" sp-act="expressOpinion" polarity="positive"
mode="opinion-decl">
i guess <punc type="stop" />
</frag> (d92a-1.2)

(30) <turn n="53" speaker="Sandra">
<q-wh n="83" sp-act="reqInfo" polarity="positive" topic="creditcard"
mode="open-query">
and what type of credit card is that do you hold <punc type="query" />
</q-wh>
</turn>
<turn n="54" speaker="caller_06">
<dm n="84" sp-act="hesitate">
er
</dm>
<dm n="85" sp-act="muse">
well <punc type="level" />
</dm>
<decl n="86" sp-act="state" polarity="positive" topic="creditcard"
mode="decl">
i'm looking at a Switch card <punc type="stop" />
</decl>
<q-yn n="87" sp-act="reqInfo" polarity="positive" topic="creditcard"
mode="constrain-alternative-closed-decl">
or do you need a credit card <punc type="stop" />
</q-yn>

</turn>
<turn n="55" speaker="Sandra">
<decl n="88" sp-act="answer-state" polarity="positive" mode="decl">
Switch is fine <punc type="stop" />
</decl> (trainline06)

In Example (29), s_RD clearly first has to think about whether u_JA has indeed fulfilled all the requirements for completing the task. This hesitation is not only indicated through *well* in unit 161, but also the preceding hesitation marker (*hesitate*; unit 160) and the *hold*, that is an explicit request to be given time to consider or physically do something – yet another delaying device. The next example illustrates what I stated above about a potentially dispreferred response, as caller_06 seems to be afraid that they may not have the right type of payment means, but 'only' a debit card, when Sandra has in fact referred to a "credit card". Caller-06 therefore delays the response using *well*, thereby also indicating that perhaps a dispreferred response may need to be expected, and which could, in turn, lead to a rejection of the available option on Sandra's part. This is further corroborated by the request for information in unit 87 regarding the potential alternative which appears to be almost a request for confirmation of the caller's negative assumption of this potential rejection. Based on these examples, and similar ones in the three corpora, my explanation seems to be more neutral than Jucker's, at the same time still distinguishing between two fairly distinct uses. What is perhaps also worth noting in this context is that *well* here seems to share a feature with *actually* in that both have the potential to 'raise' or signal counter-expectations.

Delaying DMs account for over 50% of all DMs in Switchboard, at least partly due to the fact that around half of these are represented by hesitation markers. This should probably not come as a surprise, though, if we bear in mind that the interaction between the interlocutors there is far less constrained than in the other two sets of data, so that the choices available for responding or introducing new topics may well require more deliberation.

The next category of DMs to be discussed here is that of 'acknowledging' DMs. I deliberately used scare quotes here to indicate that not all members of this category signal acknowledgement only, but that there is in fact a cline of instantiation ranging from simple minimal responses (*acknowledge*: e.g. *aha, mhm, (al) right, ok, i see*), which consistently show the highest intra-category frequencies, via agreements (*agree*: e.g. *sure, no problem*), to even stronger endorsements (*approve*: e.g. *excellent, lovely, fine*). These differences will be discussed in more detail in Chapter 7. For now, we only need to note that, just like the acknowledgements we discussed earlier in conjunction with yes- and no-units, the DMs in this category also respond to 'statements' of various types, rather than to questions. In considering their relative frequencies in the corpora, though, we also need to bear in mind that similar acknowledgments, and in perhaps also the same forms, are frequently also provided in backchannels, particularly if the overall length of turns is higher, which may explain why their relative frequency in Switchboard is so low. Unfortunately, though, speech-act analysis of backchannels has not been implemented in DART yet, so it is fairly difficult to verify this.

Accepting DMs account for a relatively small percentage of instance in Trains and Switchboard, and do not occur at all in the Trainline data. This is perhaps not surprising, as the prototypical form of acceptance is generally the yes-unit. As there is no particular form of realisation for accepting DMs, they formally correspond to 'acknowledging' ones, but constitute responses to offers or suggestions in the form of declaratives, interrogatives, and imperatives, rather than general 'statements'.

The sub-category of 'answering' DMs essentially corresponds to that of 'acknowledging' DMs formally, with the exception that there are no *approve* forms present in any of the corpora. These DMs generally respond to the same types of 'answer-eliciting' units we discussed in 4.3.1 above.

Under the category heading 'reacting' DMs, we find two separate sub-categories, those of exclamatives (*exclaim*) and expressions of surprise (*expressSurprise*). The former is commonly realised via expressions such as *oh my God, gosh, (oh) my goodness, oh boy, goodness gracious me, Jesus (H) Christ, geez, ouch, whoa, wow, oh dear, what, how* + Adj., *cool, shoot, hell, (oh) really, oh yeah*, etc. Please note that these expressions also include some that both O'Keeffe, Clancy and Adolphs (2011: 111), and Adolphs and Carter (2013: 49–50) refer to as "[r]eligious references and swear[ ]words". However, using such labels ignores the fact that most of these expressions, even if they may have originated as religious terms, swear words, or euphemistic variants thereof, have since become semantically bleached to such an extent that they can hardly be considered to retain such connotations. The latter sub-category comprises expressions like *oh, oh no, you're kidding, ooh, (wh)oops*, etc., that tend to signal more incredulity or surprise than 'pure' exclamatives, even though it is sometimes doubtful whether this distinction is always justifiable. The relative percentages of these forms in the three different corpora are fairly low and the differences can perhaps be explained rather easily. The highest frequency is present in Switchboard, probably again due to the unconstrained nature of the interaction and more 'colloquial' atmosphere, which makes both exclamatives and expressions of surprise slightly more likely. Comparing Trainline and Trains, the relatively higher instance in the former may be due to assumptions of the callers regarding their travelling options occasionally being disappointed, and that they tend to react using exclamatives, which is indeed the only one of the two sub-categories present in this corpus.

Under the label ‘attitudinal’, I have grouped a number of DMs that describe the personal attitude of a speaker in different ways, ranging from expressions of regret or empathy (*expressRegret*), via personal attitude (*expressStance*), to those of a more epistemic or evidential nature (*reaffirm, expressAwareness*). Some examples of these are presented below.

(31) <turn n="4" speaker="u_PH">
<frag n="8" sp-act="answer-referDirection" polarity="positive" topic="from" mode="partial-frag">
from Dansville <punc type="level" />
</frag>
</turn>
<turn n="5" speaker="s_CB">
<dm n="9" sp-act="acknowledge">
okay <punc type="stop" />
</dm>
</turn>
<turn n="6" speaker="u_PH">
<frag n="10" sp-act="correctSelf-referDirection" polarity="positive" topic="to" mode="partial-decl">
to Dansville <punc type="stop" />
</frag>
<dm n="11" sp-act="expressRegret">
sorry <punc type="stop" />
</dm> (d93-11.2)

(32) <dm n="144" sp-act="expressStance" polarity="positive" mode="frag">
<overlap pos="end" /> to be perfectly honest <punc type="level" />
</dm>
<decl n="145" sp-act="expressStance" polarity="negative" mode="stance-decl">
it doesn’t really matter <punc type="stop" />
</decl> (trainline26)

(33) **<dm n="5" sp-act="reaffirm" mode="constrain">**
needless to say <punc type="level" />
</dm>
<frag n="6" sp-act="referPlace" polarity="positive" topic="location-location_Anglo" mode="partial-frag">
here in Washington D C <comment content="laughter" /> <punc type="level" />
</frag>
<decl n="7" sp-act="state" polarity="positive" mode="decl">
this is the war zone <punc type="stop" />
</decl> (sw_0022_4320)

(34) **<dm n="12" sp-act="expressAwareness" mode="awareness">**
as far as i know <punc type="level" />
</dm>
<decl n="13" sp-act="stateConstraint" polarity="negative"
topic="education-location" mode="alternative-decl">
the school doesn't have any kind of programs or anything out here <punc
type="stop" />
</decl> (sw_0009_4329)

The exchange in Example (31) is concerned with establishing which direction to go next, and u_PH first provides an answer in unit 8 that is acknowledged by s_CB (unit 9), only to be retracted and corrected again (*correctSelf-referDirection*) by u_PH in unit 10. Speaker u_PH then follows up the self-correction by an expression of regret which essentially constitutes an apology for having made a mistake.

Example (32) actually contains two examples of stance expressions, one in the form of a DM (unit 144), and the other in the shape of a declarative. The *reaffirm* in Example (33) stresses the assumed common ground between the interlocutors (unit 5) before then going on to 'state the obvious', while the expression of awareness in 34 (unit 12) hedges the reported personal knowledge of the speaker that is conveyed to the interlocutor.

While the *expressRegret* act may occur either before or after a statement it 'qualifies', the other three forms function like the proposition-modifying initiating DMs we saw earlier. The relatively higher occurrences of 'attitudinal' DMs in the Trainline data is in fact due to a combination of features. While for Trains, the example we saw above has shown that, there, typically, *expressRegret* constitutes an apology for getting something wrong, the same feature applies to Trainline, too, but additionally Sandra often also frequently expresses her regret about certain options not being available to the callers using the adverb *unfortunately*.

The final category of DMs, which I have labelled *textual*, comprises a set of discourse-connecting expressions that are perhaps more typical of argumentative and/or sequential descriptions in written texts. The speech-acts that occur in our corpora comprise *initFollowUp, listSequence, initConclusion*, and *initSummary*. The first of these, represented by expressions such as *and/but/so then, furthermore, in addition, additionally*, etc., introduces a follow-up item that may be part of a list of actions, as in a set of instructions that occur in temporal sequence like a recipe (Example (35)), or an elaboration of an already established topic. The second is similar, but specifically refers to the numbers of items in such a listing, such as *first(ly), first of all, secondly, thirdly, fourth*, etc. Next, *initConclusion*, instantiated via structures like *hence, in other words, in that way, therefore, thus*, and *at last*, prefaces an assumption by the current speaker about the meaning of something the interlocutor has said, as exemplified by unit 187 in Example (36), or may also occur before a clarification by the same speaker. Finally, *initSummary*, as its name already suggests, occurs before a summary is presented by the current speaker and is expressed through words or phrases like *finally, in conclusion, to conclude, to sum up, to summarise*, or *summarising*.

(35) <turn n="39" speaker="1454">
<frag n="98" sp-act="referProcess" polarity="positive" mode="frag">
and pour cocktail sauce over the top of it <punc type="level" />
</frag>
<dm n="99" sp-act="initFollowUp">
and then <punc type="level" />
</dm>
<frag n="100" sp-act="referProcess" polarity="positive" topic="food_types" mode="frag">
get these little canned shrimp and uh wash them and cool them <punc type="level" />
</frag>
<dm n="101" sp-act="initFollowUp">
and then <punc type="level" />
</dm>
<frag n="102" sp-act="referProcess" polarity="positive" mode="frag">
dump them in the uh sauce <punc type="level" />
</frag>
<dm n="103" sp-act="initFollowUp">
and then
</dm>
<frag n="104" sp-act="referProcess" polarity="positive" mode="decl">
take crackers <punc type="stop" />
</frag> (sw_0057_3506)

(36) <q-yn n="186" sp-act="reqInfo" polarity="positive" topic="location" mode="closed-query">
were <overlap pos="start" /> there witnesses <overlap pos="end" /> <punc type="query" />
</q-yn>
</turn>
<turn n="65" speaker="1042">
<dm n="187" sp-act="answer-initConclusion">
<overlap pos="start" /> in other words <overlap pos="end" /> <punc type="level" />
</dm>
<decl n="188" sp-act="elab-reqConfirm" polarity="positive" mode="alternative-report-condition-constrain-query">

you're saying that w… <comment type="restart" /> you have to find out whether it's premeditated or <backchannel content="right" /> what the circumstances <overlap pos="start" /> were <overlap pos="end" /> <punc type="query" />
</decl> (sw_0083_4830)

The higher incidence of textual DMs in both Trains and Switchboard is due to descriptive sequences, such as the stages in transporting items in Trains, or the recipe description or anecdotes in Switchboard, being marked more explicitly in these two corpora. Although such descriptive sequences do occur in Trainline, too, in particular when the individual legs of a journey are summarised, Sandra does not use such explicit marking, but instead relies on a different mechanism, which is to describe them as a sequence of processes (*referProcess*), as shown in Example (37).

(37) <turn n="25" speaker="Sandra">
<dm n="40" sp-act="init">
<pause /> so <punc type="level" />
</dm>
<decl n="41" sp-act="state" polarity="positive" topic="from-location_Anglo-enum" mode="frag">
it's the 14:19 from Preston <punc type="level" />
</decl>
<frag n="42" sp-act="referProcess" polarity="positive" topic="location_Anglo-time-arrival" mode="partial-decl">
arriving Rugby at 16:05 <backchannel content="yeah" /> <punc type="stop" />
</frag>
<decl n="43" sp-act="state" polarity="positive" topic="month-date" mode="frag">
that's on the 9th of October <backchannel content="yeah" /> <punc type="level" />
</decl>
<frag n="44" sp-act="referProcess" polarity="positive" topic="from-month-return-location_Anglo-enum-date" mode="frag">
returning the 11th of October 18:45 from Rugby <punc type="level" />
</frag>
<frag n="45" sp-act="referProcess" polarity="positive" topic="location_Anglo-time-arrival" mode="decl">
arriving in Preston at 20:33 <punc type="stop" />
</frag> (trainline06)

As was the case for the yes-units discussed earlier, the occurrence of the same words in units with slightly different meanings makes it difficult to discern their exact functions easily, but at least some of the words or word combinations may prove to be strong indicators of roughly which of the two main groups – initiating or responding – the individual markers may belong to.

Although the majority of DMs tends to occur in initial position in a turn, there may also be some that are tagged on to the end of a unit. These, however, occur more rarely than the initial ones and have a similar function to question tags, which are discussed in 4.3.7 below.

4.3.4 Forms of address

Forms of address (labelled *address*) are another type of small unit, which, just like yes/no units and DMs, often occur at the beginning, or the end, of a complex unit. If they do so, they function as 'discourse' deictica fulfilling a special type of focussing role, which is to pick out one or more addressees in order to get their attention. Fraser (2006: 190) includes some forms of address among his parallel pragmatic markers, but the set he includes is of a very limited nature, as is apparent from his sub-category label of *deference markers*. As pointed out earlier, the notion of a parallel message being expressed through such markers is probably too strong, but, on the other hand, limiting the group to terms of address that express some form of deference, such as *Sir* or *your honor*, as Fraser does, seems to somehow underestimate the general focussing potential that terms of address have in dialogues or other forms of spoken interaction.

In all three of our corpora, there are very few instances of addresses. In terms of raw frequencies, we find 37 in Trainline, only one in Trains, and 6 in Switchboard. Nevertheless, even then, there are noticeable differences. In a call-centre interaction, we would perhaps expect a high number of deferential expressions used by the agent, but in fact Sandra only uses them about half of that time (18), and only in 11 of the calls. At other times, she addresses her callers by *Mr* four times, *Mrs* three times, and *Miss* twice, and sometimes not at all, while one of the callers uses Sandra's name, one addresses a third party, and caller_28 employs a variety of terms of endearment particularly characteristic of Northern British English use, *Love, Darling*, and even the rather unusual *Good Lady*. In Trains, the interlocutors tend not to address each other at all, apart from one time where the speaker acting as the user refers to the interlocutor as *System*, which suggests that she was taking the simulated task rather seriously.

4.3.5 Wh-questions

Wh-questions, those types of question that contain one of the so-called 'wh-words', *who, what, when, where, why*, etc., but also *how*, are relatively easy to identify, due to the occurrence of such a wh-word in a relatively initial position within the unit. However, purely having a wh-word in the beginning of the unit does not uniquely identify these units unambiguously all the time, as certain exclamatives or fragments that express suggestions also tend to exhibit this feature, albeit with the restriction of the wh-word being either *what* or *how* for the former and *why* and *how* for the latter. In addition, declarative structures that represent relative clauses may also start in a wh-word, and there may of course also be declarative units that contain a wh-word in the third slot, such as *this is what I did*, where the wh-word marks the beginning of an embedded relative clause that serves as a subject complement.

Wh-questions, unlike the yes/no- or alternative questions to be discussed below, do not limit the associated responses to a closed set of options, such as a *yes*, a *no*, or one item out of a set of alternatives. This is also why they are generally referred as *open-type questions* or *open interrogatives* (CamGr: 856).

If one consults a textbook on how wh-questions are formed, one is also generally likely to find statements to the effect that the first element in a wh-question always is/has to be the wh-word, but while this may largely – but by no means completely – be true for written language, in spoken language, the wh-element is often preceded by conjunctions or prepositions, as in the following examples.

(38) **and** how long would it <pause /> take <pause /> to get <pause /> to <pause /> the Ava… Avon (trains d92a-3.2)

(39) **but** who's it that's travelling tomorrow (trainline14)

(40) **for** which journey do you wish to purchase a ticket (trainline02)

Surprisingly, even a modern grammar, such as LongGr – which after all is also a grammar of spoken language – does not make this explicit in the opening explanation on wh-questions: "*Wh*-questions open with a *wh*-word which indicates an element to be specified by the addressee." (LongGr: 204). This feature should be considered a relatively serious shortcoming in such a recent and corpus-based grammar. And, of course, when designing a computer system that handles wh-questions in spoken language, strictly following this information would cause serious issues because a very high number of wh-questions does not follow this rule. To take but one example, in Trainline, 84 out of all 224 wh-questions (in 33 of all 35 dialogues) start with either an *and, but, or*, or *from*.

As far as the communicative function of wh-questions – or rather questions in general – is concerned, CompGr states that "QUESTIONS are primarily used to seek information on a specific point". This is somewhat misleading and certainly not applicable to all types of dialogue, as Table 4.5 clearly indicates.

Table 4.5 Functions, intra-category percentages and normed frequencies for wh-questions

Function	Trl%	Trl NFreq	Trs%	Trs NFreq	SWBD%	SWBD NFreq
reqInfo	61.62%	111.04	91.40%	179.63	80.41%	58.63
reqConfirm	2.23%	4.02	–	–	3.09%	2.25
reqDirect	31.26%	56.32	0.50%	0.99	–	–
suggest	4.91%	8.85	8.08%	15.88	15.46%	11.27
phatic	–	–	–	–	1.03%	0.75

While the majority of all wh-questions in all corpora, including those that have been abandoned, either due to interruptions or the current speaker deciding to rephrase, do indeed constitute requests for information, in Trainline nearly a third of all questions in fact represent requests for directives. Collins (2006: 186) refers to such requests for directives a *deliberative questions*, although it is not quite clear to me what type of deliberation should be involved when he states that they "differ from typical information-oriented questions in that answers to them have the force of directives rather than statements". A classic example of such *reqDirect*s would be the fairly stereotypical call-centre opening *how can/may I help you?*, which, despite Trains not being a call-centre corpus, occurs once there. Instead of using such a 'non-committal' and vague question, Sandra in Trainline frequently jumps straight *in medias res* and uses the far more direct and goal-oriented question *for which journey do you wish to purchase a ticket?* at the beginning of the interaction. What these examples have in common is that the wh-question is trying to elicit a preferred modus operandi from the other party. In other words, the respondent is given some degree of control over the next action taken. In the Trainline example, where the two interlocutors have unequal status, the customer qua role has the authority of being able to make a choice the operator is obliged to comply with – provided of course that circumstantial factors, such as availability of tickets, etc., are guaranteed –, whereas in the example from Trains, the participant asking the question is essentially pretending to fulfil the role of an operator in 'acting out' the part of the computer system.

Another important function of wh-questions is less apparent in Trainline (only close to 5%), but far more in Switchboard (above 15%), and, to lesser extent, in Trains (close to 9%). This function is one of the few that Adolphs (2008) actually establishes a functional profile for, that of suggestions (*suggest*). The forms of wh-question we encounter in our corpus data are *how/what about* X, *why don't we* X, *what if* X, or *why not* (X), thus covering a wide range of options that signal sometimes subtle differences we shall discuss in more detail in Chapter 7.

Another minor function that is fairly similar to the request for information is that of requests for confirmation (*reqConfirm*), which we have already seen in some of our earlier examples, only not in the shape of wh-questions, as they can in fact be expressed via a wide variety of syntactic categories. The interesting thing about wh-*reqConfirm*s, though, is that they tend to occur mainly as single word utterances, such as *what?, where?*, *when?*, or *how?*, with the one exception in the Trainline data being *i must what?* (trainline21). What all these forms have in common, too, is that, with an appropriate intonation, they may potentially also be used to signal incredulity, as in Example (41).

(41) <turn n="78" speaker="caller_35">
<dm n="149" sp-act="exclaim" mode="exclaim">
oh God <overlap pos="end" /> <punc type="stop" />
</dm>
<q-wh n="150" sp-act="**reqConfirm**" polarity="positive" mode="exclaim-partial-query">
<vocal content="laughter" /> **what** <punc type="query" />
</q-wh>
<frag n="151" sp-act="reqConfirm" polarity="positive" mode="query">
even to to any one of those stations <punc type="query" />
</frag>
</turn>
<turn n="79" speaker="Sandra">
<frag n="152" sp-act="confirm-refer" polarity="positive" mode="decl">
even to any one of those stations <punc type="stop" />
</frag> (trainline35)

In Example (41), the effect of incredulity is not only created through the use of a single-word wh-question in unit 150, which, in itself, is somewhat similar to an exclamative, but also through the occurrence of a preceding exclamative and the following request for confirmation that begins with the adverb *even*.

One rather unusual option for wh-questions is that they can also occur purely phatically, as in Example (42).

(42) <frag n="6" sp-act="referTime" polarity="positive" mode="frag">
way back <punc type="level" />
</frag>
<q-wh n="7" sp-act="phatic" polarity="positive" mode="partial-query">
what <punc type="query" />
</q-wh>
<frag n="8" sp-act="referTime" polarity="positive" topic="duration-enum" mode="alternative-frag">
12 years ago or something <punc type="level" />
</frag>

<decl n="9" sp-act="referTime" polarity="positive" topic="computers-time" mode="frag">
when i got my first computer <punc type="level" />
</decl>
<decl n="10" sp-act="state" polarity="positive" mode="open-frag">
i did what everybody else does <punc type="level" />
</decl> (sw_0024_4688)

In this example, the function of *what* is vaguely similar to qualifying unit 8 by prefacing it with either *perhaps* or *maybe*, in other words, the current speaker is deliberating about the exact time period. However, as unit 7 stands alone prosodically, it does not really contribute to the interaction in any way other than delaying unit 8 slightly to give the speaker time to reflect.

4.3.6 Yes/no- and alternative questions

As already pointed out above, yes/no-questions and alternative questions limit the set of responses that can occur in reaction to them. This is also why CamGr refers to yes/no-questions as *closed interrogatives* (CamGr: 856). However, surprisingly, it does not even refer to or discuss *alternative questions*. CompGr handles both of these categories separately, including samples like the following for alternative questions:

Would you like CHÓcolate, vaNÍLla, or STRÀwberry (ice cream)?
Which ice cream would you LÌKE? CHÓcolate, vaNÍLla, or STRÀwberry?
(CompGr: 823)

While the first example unquestionably represents a genuine alternative question, it is rather doubtful whether this label can in fact be accepted for the second one because what is here perceived as a single question really consists of two separate units, generally marked by a clear pause and *pitch reset* (cf. Wichmann 2000: 24). Thus, CompGr's justification for claiming that there are alternative wh-questions seems seriously flawed, as one should really see the example as a combination of a wh-question plus a sentence fragment containing a list of options. LongGr, on the other hand, treats similar cases correctly as two separate syntactic units, as can be seen in their example and associated description:

Which one should I use, the blue or the pink? (CONV)
Here, the *wh*-question is followed by an elliptic alternative question. The alternative question narrows down the range of possible answers offered by the *wh*-question. (LongGr: 208)

Intonation also plays a major role in distinguishing between what may be referred to as 'false or fake alternative questions', that is closed-type questions that contain a marking of an alternative in form of an *or*, but that really require a response in form of a *yes* or *no*, such as *Would you like some ↗ tea or coffee?*. As the intonation in this case indicates, the force of the question in the first instance is to establish whether the addressee would like a hot beverage or not, which actually constitutes the equivalent of an expanded yes/no-question like *Would you like some ↗ coffee?*. The corresponding genuine alternative question to this would be *Would you like ↗ tea | or ↘ coffee?*, which has a clear list structure plus a distinct prosodic break/pause between the alternatives presented (indicated by the vertical bar), something which is absent in the fake version, where the co-ordination is expressed by 'running' the two options together in one chunk. What is important to note in this context is that, in the absence of any prosodic information indicated in the data to be analysed, there is no way for an automatic analysis procedure to distinguish between genuine and fake alternatives, apart from maybe looking at the answer.

In the present scheme, both yes/no and alternative questions are subsumed under one category labelled *q-yn*. This was done because, syntactically, they both follow the same rule in English, containing *subject-verb inversion* in contrast to the declarative. Although such an inversion may also occur in wh-questions that contain an auxiliary as the second (non-adverb) element, the absence of the wh-word in an initial position makes it easy to distinguish between the two major question categories.

As far as the main functions of closed questions are concerned, we get an even more diverse picture for yes/no-questions than for wh- ones, as can be seen in Table 4.6.

Table 4.6 Functions, intra-category percentages and normed frequencies for alternative and yes/no-questions

Function	**Trl%**	**Trl NFreq**	**Trs%**	**Trs NFreq**	**SWBD%**	**SWBD NFreq**
reqInfo	64.00%	160.12	49.99%	64.51	87.20%	81.93
reqDirect	10.93%	27.36	–	–	1.60%	1.50
offer	–	–	24.61%	31.76	–	–
direct	9.00%	22.53	–	–	–	–
reqOpt	7.40%	18.51	11.54%	14.89	–	–
reqModal	0.32%	0.80	2.31%	2.98	2.40%	2.25
reqConfirm	4.82%	12.07	10.00%	12.90	8.80%	8.27
suggest	–	–	1.54%	1.98	–	–

Although requests for information (*reqInfo*) still constitute the majority of these questions, they only amount to approximately two thirds in Trainline and slightly less than half in Trains. Only in Switchboard do they actually approach 90%, which is due mainly to the virtual absence of other functions apart from *reqConfirm*s and a tiny number of modal requests (*reqModal*). The latter essentially reflect 3rd person epistemic possibility, as can be seen in Examples (43) and (44).

(43) <q-yn n="14" sp-act="reqModal" polarity="positive" topic="from" mode="alternative-closed-query">
<pause /> **would it** <pause /> be faster for an engine to come from Elmira <pause /> or Avon <punc type="query" />
</q-yn> (d92a-3.1)

(44) <q-yn n="55" sp-act="reqModal" polarity="positive" topic="time" mode="closed-query">
would it be appropriate to uh drink with the <comment type="restart" /> uh prior <comment type="restart" /> when you have the hors d'oeuvres <punc type="query" />
</q-yn> (sw_0057_3506)

The 1st person counterpart to *reqModal*, the request for options (*reqOpt*), may also reflect epistemic possibility (or ability), as well as possibly deontic modality, and is more relevant to the joint interaction between two participants. This can clearly be seen in the frequencies in Trainline and Trains, whereas it is completely absent from Switchboard, where evaluating options or making joint decisions is simply not relevant in the same way. While Examples (43) and (44) in a sense also reflect enquiries about options, they do so in a more 'neutral' or general way by referring to an entity, *it*, in the 3rd person, as well as using a hypothetical modal form, *would*. In contrast, a *reqOpt* is more personal and/or involved, as will hopefully be obvious from the next two examples.

(45) <q-yn n="219" sp-act="reqOpt" polarity="positive" mode="closed-query">
<overlap pos="start" /> can **we** go <overlap pos="end" /> back and do that <punc type="query" />
</q-yn> (d93-11.1)

(46) <turn n="40" speaker="caller_04">
<dm n="94" sp-act="acknowledge">
right <punc type="level" />
</dm>
<dm n="95" sp-act="hesitate">
em <punc type="level" />
</dm>
<q-yn n="96" sp-act="reqOpt" polarity="positive" topic="day" mode="closed-query">

can i be a pain and ask you about Thursday night <punc type="query" />
</q-yn>
<decl n="97" sp-act="reqInfo" polarity="positive" topic="day"
mode="condition-exists-query">
if there's anything Thursday night <punc type="query" />
</decl> (trainline04)

In Example (45), the speaker uses an inclusive *we* to find out about an option (or possibility) for a joint action, while in 46, the caller basically also uses their request for an option as a polite way to request permission for enquiring in the first place, indicating that this may in fact be an imposition on Sandra.

We have already seen that requests for directives play an important part of Sandra's strategies when occurring as wh-questions, so it should come as no surprise that their yes/no-counterparts also occur with a frequency of around 10% in Trainline. They can also be found to a small degree in Switchboard, but are completely absent in this form in Trains. A typical example of a yes/no request for directive would be *do you want me to book this ticket <punc type="query" />* (trainline03).

However, yes/no questions can also express the 'opposite' of such *reqDirect*s, polite directives themselves, such as *could you put me in smoking as well please <punc type="query" />* (trainline09). This feature only occurs in Trainline in our data, but accounts for a non-negligible 9% of all yes-no questions.

I have already referred to the fact that Trains contains a fairly high number of offers in 4.3.1 above in conjunction with accepting yes-units. The offers are invariably triggered by the opening *(how) can i help you*, so that they can occur in both wh- and yes/no-form. Suggestions in yes/no-form are rather limited, though, but do occur twice, once as *shall we do that <punc type="query" />* (d92a-3.1) and once in abandoned form as *and should i <punc type="incomplete" />* (d93-16.1).

As we have seen above, both wh- and yes/no questions may have more varied functions than simply requesting information, in particular if they involve the use of modal auxiliaries. However, it is perhaps due to the fact that most of the research into modality (e.g. Palmer, 1990 & 2001; Collins 2009) has predominantly focussed on modality in declaratives that these other functions have not found their way into the descriptions in traditional grammars.

4.3.7 Declaratives

Declaratives are considered the prototypical 'sentence' type in most grammars, and this is why they are usually not covered in much detail. In the introduction to the section on “Distinctive grammatical properties of the major clause types”, CamGr states to this effect:

> Declaratives are the default clause type: a clause is declarative if it lacks the special properties that define the other types. In this section, therefore, we outline the distinctive properties of the other four major types with respect to main clauses. (CamGr: 855)

Thus, the basic assumption is that the declarative SV(O) structure for English is the default canonical and unmarked form, a form which can be used more or less straightforwardly as a basis for comparison to other categories. However, even though this may be the case, we still need to bear in mind that this canonical form does not always go hand in hand with an equally canonical meaning of making a statement or providing information, as is often simplistically assumed.

In terms of illocutionary force, there are at least three further options discussed in the literature, which are tied to particular controlling factors outside the syntax. The first of these is rising final intonation, which may turn a formal declarative functionally into a question, usually referred to as a *declarative question*:

(47) <decl n="37" sp-act="reqInfo" polarity="negative" topic="time-day" mode="query">
you don’t have any at any time on Saturday</decl> (trainline05)

Although tag questions are similar in nature, they are usually not discussed under the heading of declaratives, but, due to their presumed function, as questions. This happens despite the fact that most of their occurrences formally consist of a declarative unit with a tagged-on subject-auxiliary-inversion, which, as already stated, is generally the trademark of yes/no-questions. However, a different type of tag usage will be illustrated in the discussion of imperatives further below. Generally, tags that occur with declarative units invite the interlocutor to confirm an assumption made through the proposition in the declarative part, something that they have in common with discourse markers appearing in the same final position (cf. LongGr: 210; CompGr: 814).

The second case is when the main verb of the unit is a performative one and the subject is in the first person, as in *I promise to do* X, *We apologise for the inconvenience*, etc. (cf. Austin 1962; CompGr: 804). This only works if the verb is in the present tense, non-3rd person because otherwise the declarative is purely stating an event that occurred in the past, which we can shown by converting one of the previous examples to its past tense equivalent: *I promised to do* X. A slightly more unusual option for this also allows for 2nd or 3rd person subjects, but in conjunction with a passive construction, and in (1) also the prototypical performative marker *hereby*, as in:

1. You are hereby authorized to pay....
2. Passengers are warned to cross the track by the bridge only. (Austin: 1962: 57)

The third option is that of indirect requests, where the relevant declarative unit, as Collins (2006: 184) puts it, "can be endowed with indirect directive force". The two examples he cites for this are "You will/must be here by five" and "I would like you to accompany me" (ibid.). Both of these examples also clearly demonstrate that the speaker needs to have some degree of authority over the hearer in order for the indirect request to work, as otherwise we would be dealing with a suggestion.

Looking at the summary of main functions in our three corpora shown in Table 4.7, not only can we see quite easily that the above descriptions again represent a rather gross oversimplification, but also that there are clear differences between the corpus categories/genres regarding some of the distributions.

Table 4.7 Main functions, intra-category percentages and normed frequencies for declaratives

Function	Trl%	Trl NFreq	Trs%	Trs NFreq	SWBD%	SWBD NFreq
stateX	65.89%	842.59	58.63%	703.65	63.83%	428.13
report	0.82%	10.46	0.69%	8.93	3.18%	68.40
expressX	5.60%	71.62	8.47%	75.43	23.02%	91.74
predict	7.17%	91.74	15.56%	200.48	2.10%	45.10
reqX	9.38%	119.91	9.86%	168.72	1.54%	119.91
direct	1.83%	23.34	–	–	–	–
suggest	0.19%	2.41	2.23%	28.78	0.73%	15.78
offer	0.06%	0.80	0.15%	1.98	0.03%	0.75
agree/approve	6.29%	80.48	1.23%	15.88	1.75%	37.58

Before discussing Table 4.7, though, it is important to explain some of the distinctions and conventions adopted there. Regarding conventions, as before, the *X* here acts as a placeholder for a number of different sub-categories, which will primarily be discussed in Chapter 7. The main distinction in terms of speech acts that requires explanation here is that between *stating* and *expressing* ones. Stating speech acts predominantly express more factual information, e.g. as in *state* itself, but also *stateCondition, stateConstraint, stateOpt*, etc., while the expressing ones tend to embody personal feelings, judgements, or evaluations, as in e.g. *expressOpinion, expressLiking*, or *expressPossibility*, to list but a few. The category *report* is a special case of stating that represents reported speech, essentially referring to a past event, whereas *predict* constitutes its 'counterpart' in the expressing category that refers to a potential future event.

Stating speech acts occur with high frequency across all corpora, but are not equally distributed, with the highest frequency occurring in Trainline, where only around a third (32.41%) are general statements, while the remainder relate to providing contextually important information, such as the presence or absence of constraints, conditions, durations, dates, processes, and intentions. Here, the latter constitutes an exception to the rule that stating acts normally do not reflect a personal nature, as intentions are normally presented more like facts to be accepted by the interlocutor, rather than indicating judgements or evaluations. The high number of special, circumstantial, stating acts is a strong indicator of the task-/goal-oriented nature of Trainline, which clearly necessitates the exchange of such information. Reporting is used here to convey information that the speaker has either obtained prior to the interaction, i.e. the caller has already made an attempt to find out some information prior to calling up Sandra, or, in Sandra's case, that she has just looked up on the computer.

In contrast, in Switchboard, which has the next highest number of occurrences of stating acts, more than half (52.19%) of these are general statements. Together with the highest incidence of reported speech (3.18%), this reflects the 'story-telling' nature of ordinary conversation, where information about precise circumstances is of lesser importance. Reporting here, in addition to talking about what other people have said, often consists of relaying to the interlocutor what the speakers themselves have said in conversation with others, and often involves the use of *like*, as in e.g. *i'm like <quote>okay i'll sit down here and call it</quote>* (sw_0009_4329).

The lowest frequency is exhibited in Trains, where the number of general statements is again considerably lower at only 22.42%, and the majority again relates to stating facts that are relevant towards achieving the mutual goal of the interlocutors, once more indicating the task-oriented nature. In both Trainline and Trains, reporting is also commonly used to indicate information pertaining to the instructions the speaker has received as part of the problem-solving activity.

With respect to the interactional features of stating acts, 8.4% of them constitute answers in Trainline, 4.85% in Trains, and only 3.03% in Switchboard. This should probably come as no surprise, considering the fact that it partially reflects the number of interrogatives used across the corpora, which is in turn related to the nature of the interaction. Here, the need to elicit factual information from the caller is clearly most important in Trainline, but least so in Switchboard, with Trains being in-between, as the interlocutors there already have some instructions to guide them in the problem-solving process, thereby reducing some of the need for elicitation.

Interestingly, the situation is reversed for expressing speech acts, presumably because the interaction in Switchboard is of a more personal nature. Therefore, the questions being asked also allow the respondents to evaluate what is being talked about in such a personal light in their responses, e.g. expressing opinions, wishes, likes or dislikes, as well as other attitudes or assumptions, more liberally, something that is also reflected in the overall much higher incidence of expressing acts. In Trainline, in contrast, such expressions of a personal nature of course tend to be limited to expressing preferences regarding the travel arrangements, while in Trains, evaluations tend to be of a more epistemic nature, expressing (non-)awareness of options or the (im)possibility/probability of taking particular actions. One feature that is particularly striking, though, is the far higher number of predictions occurring in Trains. This is predominantly due to the time constraints under which the logistics problems are to be solved, so that the interlocutors frequently make predictions about when different types of commodities can be assumed to arrive in different places, based on an envisaged scenario. The predictions in Trainline account for less than half the number of declarative structures in Trains, but are still considerably higher than in Switchboard, where they only make up a very small percentage. Predictions in Trainline are somewhat similar to the ones in Trains, as they also tend to pertain to arrival times, but this time those for the callers if they book a particular option.

The fairly high percentage of interrogatives (marked *reqX* in Table 4.7) in both Trainline and Trains again stands in stark contrast to that in Switchboard, but can once more easily be explained by the task-oriented nature of the interaction. The directives in declarative form occurring in Trainline are mainly attributable to Sandra's strategy of using requests for directives, which effectively turn the responses into directives, but also partly constitute statements of intent that simply have a *please* tagged onto them such as *i'll go for that one please* (trainline09). The fact that they do occur at all is due to the imbalanced role of the interlocutors who are not equal partners in the exchange, with Sandra as the 'service provider'. The higher incidence of suggestions and – to a lesser extent – offers in Trains clearly marks a feature of the collaborative nature of the problem-solving activity. The relatively high number of expressions of agreement or approval in Trainline is mainly caused by the responses of the callers who not only acknowledge the options Sandra conveys to them, usually in the form of DMs, but frequently also tend to register their 'approval' through more or less exuberant expressions like *that's fine/great/wonderful/lovely* or simple agreement, e.g. *that's right*.

Other, less common, options for speech acts occurring in declarative form in the three corpora are expressions of confirmation (*confirm*), indirect requests to wait for a response/hold the line (*hold*), listing numbers (*enumerate*), (self-)corrections (*correctSelf, correct*), phatic expressions like *if you see what i mean*, rejections (*reject*; e.g. *that's no good, that's too early*), admissions (*admit*; e.g. *i have to say…*), negations (*negate*; e.g. *there is not any Virgin Value fares the entire day*), signals of misunderstanding (pardon; e.g. *i didn't follow that*), disagreements (*disagree*; e.g. *i don't agree with that at all*), explanations (*explain*; *i was actually talking about…*, *this/which means…*), and nominations (*nominate*; e.g. *you're first*). In addition, we sometimes encounter independently occurring clefted or focussed structures (*referThing*; e.g. *what we could do is…*, *from what i know/experienced…*), or relative clauses expressing temporal or personal deixis that may either preface information (e.g. *when we went to pick up the bananas*) or serve as a kind of afterthought, sometimes complementing something the interlocutor has said, as in Example (48).

(48) <turn n="45" speaker="1599">
<decl n="105" sp-act="stateConstraint" polarity="positive" topic="problem-weather" mode="constrain-decl">
and you need people who are trained for that type <backchannel content="right" /> of problem <punc type="stop" />
</decl>
</turn>
<turn n="46" speaker="1340">
<decl n="106" sp-act="refer" polarity="positive" mode="decl">
who know what they're doing with that <punc type="stop" />
</decl> (sw_0006_4108)

4.3.8 Imperatives

Imperatives are characterised by having an initial verb base form as the first element in their positive variant and *don't*, followed by the base form, in the negative counterpart. Their main function is generally assumed to be that of uttering a directive, but, as can clearly be seen in Table 4.8, this need not always be the case, but may instead be domain-dependent to some extent. While in both Trainline and Switchboard, the general assumption certainly does hold true, in the collaborative problem-solving task in Trains, only around one fifth of all imperatives do in fact represent directives, while close to half constitute holding acts (*hold*).

Table 4.8 Functions, intra-category percentages and normed frequencies for imperatives

Function	Trl%	Trl NFreq	Trs%	Trs NFreq	SWBD%	SWBD NFreq
direct	60.78%	24.94	20.29%	13.90	76%	14.28
hold	25.49%	10.46	47.83%	32.76	8%	1.50
suggest	11.76%	4.83	28.99%	19.85	12%	2.25
apologise	1.96%	0.80	2.90%	1.99	4%	0.75

We have already seen examples of *hold* earlier when discussing DMs. The *hold* can be seen as a special form of directive used for interaction management where one of the participants needs to 'stall for time', either to record some information, deliberate about a decision or, as is often the case in task-oriented dialogues such as the Trains or Trainline data, to try and find some appropriate information on the computer or elsewhere. Such verbal action is perhaps especially frequent and necessary if the conversation is conducted over the telephone because it is usually followed by a period of silence and the party at the other end of the line may otherwise be tempted to hang up, due to the irresponsiveness of the interlocutor. Two examples of such holds are provided immediately below:

(49) <imp n="106" sp-act="**hold**" polarity="positive" topic="verify" mode="hold-decl">
let me just uh <pause /> check <punc type="stop" /> </imp> (d92a-1.2)

(50) <imp n="17" sp-act="**hold**" polarity="positive" mode="hold-decl">
hang on <punc type="stop" /> </imp> (trainline23)

The above examples are probably rather typical of telephone conversations, even if the second one could equally well occur in a face-to-face situation, although there it might also be used as a means to make someone stop and think if it is followed by a suggestion like *wouldn't it be better to…*. Yet another hold construction, *let me think…* is generally used to 'buy some time' for deliberating about potential options, but is more similar in nature to a suggestion, as will become apparent a little later. The fact that this speech act occurs with an extraordinarily high frequency in comparison to other imperatives can be seen as particularly relevant to the problem-solving nature of the interactions in Trains, where frequent deliberation is required as part of the planning process.

To distinguish these holds better from more typical directives, let us look at some examples of the latter.

(51) <imp n="152" sp-act="direct" polarity="positive" topic="booking" mode="decl" >
book that for <overlap pos="start" /> me <punc type="stop" /></imp>
(trainline04)

(52) <imp n="128" sp-act="direct" polarity="positive" topic=" address-confirm" mode="request-decl">
and just confirm the rest of the address please <punc type="stop" /> </imp>
(trainline08)

(53) <imp n="1" sp-act="direct" polarity="positive" mode="decl">
tell me about your home <punc type="stop" /> </imp> (sw_0087_2775)

The first Example (51) is what one might want to call a 'straightforward' imperative, a command without any kind of attempt at conveying even the least degree of politeness. The customer here clearly seems to know that he or she has the authority to utter such a directive. This is fairly rare in transactional interaction, where it is far more common to 'soften' the directive by adding a *please* at the end, as is the case in Example (52). This use of *please* is in fact often described as an act of *mitigation*, but the scare quotes are deliberately used around the verb *soften* above because this may be seen as more of a ritualistic act, rather than involving any genuine notion of politeness. However, despite this sort of negative interpretation, one can definitely say that the word *please* still serves as a fool-proof indicator for recognising a directive. Looking at Example (53), though, we can see that the 'bare' imperative does not always have to be associated with more directness and authority. As previously, the force clearly depends on the relationship between the interlocutors, which, in this case, is that of equal 'partners' in the conversation, so that effectively the force of the *direct* speech act is more like that of a suggestion, rather than a command.

One other feature may strike one as interesting, though, when looking at the Example (52). This is the fact that – as already observed for the question categories above – even though the general assumption is that directives do start with a verb, this need not actually be the case at all. Instead, there may in fact be one or two words, conjunctions and adverbs, preceding the verb, although the latter category seems to be restricted to the word *just* in imperatives. This restriction, though, does, for instance, not apply to declaratives.

As already shown in Table 4.8, another important function of imperative units is that of making suggestions. These suggestions are frequently marked by the occurrence of *let's* or *let me* in the beginning of the unit, or, to put in more lexico-grammatical terms, a combination of the verb base form *let*, followed by a 1st person pronoun. It is conceivable that some suggestions may also involve a 3rd person pronoun, such as in e.g. *let them do it*, if this is not used as an order. In this case, however, the power to fulfil on the 'promise' made in the suggestion does not lie with the person making the suggestion. Below is an example of a suggestion involving the contraction *let's*, which is generally used in spoken interaction, rather than the more 'prosaic' full form *let us*.

(54) <imp n="40" sp-act="suggest" polarity="positive" topic="time-enum" mode="decl">
let's say <pause /> half 3 in the afternoon <punc type="stop" /> </imp>
(trainline02)

A special type of suggestion is an *offer*, but this is often difficult to distinguish automatically from a 'normal' suggestion, unless the benefit to the other person is clearly expressed, as in e.g. *let me do this for you*. This, however, does not occur in our data. What does occur with limited frequency in all three corpora is the speech act *apologise* in conjunction with an imperative form. In this case, the verb is invariably *excuse* in combinations like *(or) excuse me* or *excuse a moment*. Overall, imperatives are fairly rare in all three corpora, with perhaps Trains with its relatively high number of *hold*s, *direct*s and *suggest*s being marginally different from the rest.

4.3.9 Fragments and exclamatives

Fragments and exclamatives are two types of units that, like DMs, are syntactically incomplete. They may arise under a number of different conditions:

a. when it is semantically possible to only express partial information in response to queries, as part of longer information gathering processes, or in conventionalised, formulaic expressions,
b. when certain bits of information are focussed or clefted, such as in leading (verbless) adverbials, or other types of what is frequently referred to as *prefaces* or *heads* if they occur before the main clause, or *tails* or *noun phrase tags* if they as ‘afterthoughts’ (cf. Rühlemann 2006: 390 for a more comprehensive summary),
c. when a speaker has produced a unit that is somehow not completely grammatical or when essential syntactic elements are unintelligible, or
d. when an utterance has been abandoned before having been syntactically completed.

In the case of (a), one usually finds elliptical structures generally in the form of single phrases, such as NPs or PPs, and where the missing constituents can be ‘recovered’ from the context in that they have been ‘primed’ by a question or the idiomatic character of formulaic/‘ritualistic’ expressions. In the other cases, it is usually impossible to identify the missing information, either because it is prospected, indicating that an expectation of what is to come has been created as in case (b), or simply because the information cannot easily be inferred.

Before discussing Table 4.9, it is again necessary to clarify some important distinctions related to the syntax and speech act labels used here. Although fragments can fulfil many functions similar to those of declaratives, the main criterion for distinguishing between the two categories is the presence of an S + V structure containing a finite verb in the latter. If either the verb or subject is missing, the resulting structure can only be a fragment, an imperative, or an exclamative. In spoken interaction, automatically differentiating between imperatives and fragments when the subject is missing can sometimes be difficult, though,

Table 4.9 Important functions, intra-category percentages and normed frequencies for fragments

Function	Trl%	Trl NFreq	Trs%	Trs NFreq	SWBD%	SWBD NFreq
stateX	3.13%	50.00	12.36%	108.50	24.07%	181.00
refer	21.01%	329.00	14.40%	126.00	40.68%	303.50
referProcess	5.91%	93.00	18.03%	158.00	1.61%	12.00
referTime	9.14%	143.00	9.07%	79.50	4.83%	36.00
referDuration	–	–	12.24%	107.00	1.01%	7.50
referPlace	3.60%	56.00	1.59%	14.00	1.01%	7.50
referDirection	1.64%	26.00	7.37%	64.50	–	–
enumerate	19.77%	310.00	2.95%	26.00	0.40%	3.00
spell	1.34%	21.00	–	–	–	–
reqX	14.33%	225.00	8.73%	76.50	4.33%	32.50
social ritual	12.02%	188.50	4.54%	39.50	1.51%	11.50

especially in lists of actions performed by the same ‘agent’, as can be seen in Example (55).

(55) <decl n="273" sp-act="state" polarity="positive" topic="enum" mode="frag">
<pause /> engine E3 <pause /> is going to pick up 2 boxcars <backchannel content="mhm" /> <punc type="level" />
</decl>
<frag n="274" sp-act="state" polarity="positive" topic="to" mode="frag">
go to Corning <backchannel content="mhm" /> <punc type="level" />
</frag>
<dm n="275" sp-act="hesitate">uh</dm>
<frag n="276" sp-act="state" polarity="positive" mode="frag">
<pause /> pick up a tanker <punc type="level" />
</frag>
<frag n="277" sp-act="state" polarity="positive" topic="enum" mode="decl">
<pause /> and load oranges into the 2 boxcars <punc type="stop" />
</frag> (d92a-5.2)

Here, units 274, 276, and 277 formally look like imperatives, when in fact they simply constitute fragments where the subject from unit 273, *engine E3*, has been omitted. This example is more complex than most, however, as not only the subject has been left implicit, but also part of the verb structure, where we need to ‘interpolate’ either *is going to* or *will* into all incomplete units to retain the future orientation. In other cases, it is even more difficult to distinguish between such fragmentary units and imperatives, as the imperatives may look like a set of instructions, similar to those in a recipe.

(56) <decl n="5" sp-act="state" polarity="positive" topic="problem" mode="frag">
this is my problem <punc type="level" />
</decl>
<imp n="6" sp-act="direct" polarity="positive" topic="location-transport_means-to-from" mode="decl">
<pause /> plan a round trip <pause /> from Avon <pause /> to Elmira <punc type="stop" />
</imp>
<frag n="7" sp-act="referDirection" polarity="positive" mode="partial-frag">
<pause /> on the way out <punc type="level" />
</frag>
<imp n="8" sp-act="direct" polarity="positive" topic="enum-transport_means-commodities" mode="decl">
<pause /> take <pause /> 3 boxcars of bananas <punc type="stop" />
</imp>
<imp n="9" sp-act="direct" polarity="positive" topic="location-enum-transport_means-to" mode="decl">
deliver 1 <pause /> each <pause /> to Bath <pause /> Corning <pause /> and Elmira <punc type="stop" />
</imp> (d93-18.3)

In Example (56), the imperative structures look extremely similar to a listing of action sequences like the one in (55), but the main difference here is that the speaker has actually, somewhat indirectly, introduced the fact that he has been given a set of instructions in unit 5, and now renders them verbatim. And although these imperatives are reported, they effectively constitute a set of instructions the two interlocutors need to follow in the problem-solving activity as if a third party were present and uttering them.

In terms of speech acts, the main function of fragments is essentially either to provide information in the same way declaratives tend to do, only that here, the majority of information given is not constative, but instead referential, as can be seen in the relatively low frequencies of acts marked *stateX* in Table 4.9, where the cumulative frequency of referential acts always clearly outweighs that of constative ones. Most referential expressions belong to either type (a) or (b) listed above and generally consist of single NPs, PPs, ADjPs, or even single adverbs. If a verb is present, or can be inferred in rare cases, the distinction between stating and referring essentially depends on whether the (assumed) verb is finite or not. Most referential non-finite clauses in DART are labelled as *referProcess*, as they describe ongoing actions, such as arrivals or departures in Trainline, or movements between places in Trains. This speech act is particularly prevalent in Trains (18.03%), but even if such movements are also important in the domain represented in Trainline, the percentage there is lower (5.91%), as it balances more against other referential acts, in particular those of *enumerate* (19.77%) and *spell* (1.34%), which mainly represent the enumeration of credit card numbers and spelling out of name or address details as part of the actual booking process, something that is of no importance in either Trains or Switchboard. Enumerations do occur in Trains and Switchboard as well, but with relatively negligible frequencies, 2.95% and 0.4%, respectively

The number of stating expressions in fragments is still highest in Switchboard (24.07%), again probably due to the 'narrative' nature of the interaction. However, more than two thirds of these are of type (d), due to being left incomplete by a speaker, while this is true for less than a quarter in Trains, and none in Trainline. The incidence of such abandoned stating units thus appears to be rising the less constrained the dialogue is, which we can probably attribute to planning issues related to a higher number of choices about what to say next. This is corroborated by the overall (normed) frequencies of all abandoned units across the three corpora, where Switchboard again has the highest (545), Trains is in the middle (309), and Trainline the lowest (106.5). Not all of these instances of abandoned units are due to planning issues, though, but we also need to take into account interruptions, where, interestingly, a rather similar pattern of 51.9:31.8:15.3, again in the same frequency order, arises. In other words, the number of potential interruptions appears to rise at a relatively similar rate to the number of speaker-abandoned units the more unconstrained the interaction becomes. This observation of course disregards potential dispersion arising due to the influence of individual speaker behaviour, something that may well be worth investigating in more depth in a study that deals specifically with such speaker behaviour.

General referring expressions (*refer*) account for substantial intra-category percentages in all three corpora, but clearly constitute the largest sub-category in Switchboard, where we encounter mainly focussing expressions of type (b) or abandoned deictic references of type (d). In Trainline, where the percentage is only about half as high, though, referential deictic expressions tend to refer to names, titles, circumstances, means of payment, and ticket types or options, and around a third constitute some form of response, mostly to interrogatives, but occasionally also corrections. Finally, in Trains, where they occur least frequently, they generally represent means of transport or commodities.

In addition to the aforementioned frequent occurrence of references to ongoing processes in Trains, references to time (*referTime*; 9.07%) and durations (*referDuration*; 12.24%), as well as directions (*referDirection*; 7.34%) and places (*referPlace*; 1.59%), do play important roles in Trains, clearly characterising the domain and task involved, together with the references to the processes. Surprisingly, durations are not referred to in Trainline at all, but these only become implicit in the references to arrival and departure times (9.14%). This absence of durations goes hand in hand with a lower number of references to directions (1.64%), whereas departure and arrival locations feature a little more prominently (3.60%). One thing we should not forget, though, is that references to directions do in fact 'subsume' locations, too. The fact that, cumulatively, references to times, processes, and directions do occur with a considerably lower frequency than in Trains, when both domains involve information about travel, may initially come as a surprise. However, considering that, effectively, Trainline contains two subdomains or tasks, that of travel planning and that of booking, while Trains only exhibits the former, it becomes easy to see that somewhere, some of these frequencies need to be balanced, as pointed out before. Other than references to time (4.68%), in Switchboard none of the other types of reference, apart from the general deictic ones, occur with any substantial frequencies, again pointing towards the more narrative nature of the dialogues.

Interrogatives as fragments (*reqX*) occur with the highest frequency in Trainline (14.33%), mainly due to elliptical queries by Sandra about personal facts, such as *and your postcode/credit/card/number/initial*, or queries about departure times, destinations, or dates, e.g. *departing at what time, travelling to*, or *Saturday*. In Trains, they mainly relate to eliciting circumstantial information related to directions or routes, or which commodities to load, but clearly at only 8.73% only play a lesser role, while they appear to be least important in Switchboard, where they only amount to 4.33%, and no clear categorisation is possible. The higher incidence in the task-oriented dialogues is evidently related to the necessity of identifying relevant facts in order to perform these tasks, and performing these efficiently by leaving out material that can already be assumed to be part of the common ground.

Formulaic social ritualistic expressions, such as greetings (*greet*) and goodbyes (*bye*), self-introductions (*identifySelf*), and expressions of gratitude (*thank*) make up 12.02% of all fragments in Trainline, while they only account for 4.54% in Trains and 1.51% in Switchboard. This relatively high discrepancy can not only be attributed to general differences in the genres/domains covered, but is at least partly due to differences in corpus compilation strategies, so they have to be interpreted a little cautiously. While the (randomly chosen) dialogues in Trainline and Trains are all essentially complete dialogues, the ones in Switchboard in most cases do not contain the beginning or end phases of the telephone conversations where greetings, introductions, and goodbyes may occur. In Trains, the completion of a task is normally marked by statements such as *i'm finished*, but while the participant who enacts the 'system' in all but one dialogue opens the dialogue with a greeting, there are no ritualistic goodbyes present to close the conversation in any of them. Even in Trainline, where all but one dialogue contain a ritualistic greeting, only 29 of them have the goodbye counterpart. What is particularly noteworthy in Trainline, though, is that Sandra introduces herself in the beginning of all interactions with *Sandra speaking* as part of her playbook routine, which already accounts for a large part of the differences in frequency. Removing these incidences and the greetings and goodbyes from the comparison, the most striking feature is the number (32) of thanking expressions, but where again most of them form part of the playbook interaction, as Sandra utters 29 of them, and only 3 are reciprocated by the callers. Switchboard, in contrast, contains no such expressions at all, and in Trains, the 3 occurrences occur in only 2 dialogues, each time forming part of a ritualistic closing sequence like the one used by Sandra.

One further type of fragmentary c-unit is that of exclamatives (exclam), which receive their own tag (*exclam*) in DART. They generally signal surprise, amazement, incredulity, or anger. There are, however, only 6 instances of exclamatives in 4 documents in the Switchboard data, and none in the other two corpora. These almost exclusively take the form *how* + Adj, e.g. *how funny, interesting, neat*. The remainder of expressions of surprise is currently subsumed under the syntactic category of DMs, and future versions of DART may implement a more unified solution where all instances of a similar nature are either subsumed under DMs or exclamatives.

CHAPTER 5

Semantics and semantico-pragmatics

Before turning to a discussion of how to identify the form of syntactic units, I will first introduce two other levels of content identified in DART, namely those of semantics and semantico-pragmatics. The final, pragmatic, level, will be discussed in Section 6.5 once it has been demonstrated how to determine the other levels in order to make use of them in an appropriate manner in the deduction of speech acts.

Within syntactic units, one can essentially distinguish between two different levels of non-syntactic content, similar to what Sinclair (1992: 87) refers to as "planes of discourse". He distinguishes between an *interactive* and an *autonomous plane*, where the former is part of the *discourse management*, whereas the latter

> is where the meaning of the discourse is managed; where each new move, once its interactive contribution has been taken account of, is related to the preceding meaning as the text has organized it. As Hazadiah (1991) says, the autonomous plane shows the product of discourse, the shared meaning; the interactive plane shows the process, the means whereby the meaning is made available for sharing. Every utterance has a value on both planes. (ibid)

What Sinclair here refers to as the autonomous plane clearly seems to be equitable with semantic content that either already belongs to the *common ground* (cf. Clark 1996: 92ff) or is in the process of being added to it. This semantic content equally clearly belongs to the subject matter under discussion, or what Halliday and other scholars in Systemic Functional Linguistics refer to as *field* (as opposed to *tenor* and *mode*) and is thus distinct from any kind of more pragmatic content that functions on the aforementioned interactive level, that of *discourse management*. Gregory and Carroll (1978: 7; cited in Benson & Greaves 1981: 45) define the three categories as follows:

> In a general sense they are all related to the role being played by the user in the language event. Field of discourse is the consequence of the user's purposive role, what his language is "about", what experience he is verbalizing, what is "going on" through language. *Mode* of discourse, on the other hand, has to do with the degree of writtenness or spokenness of a text, while *tenor* accounts for points along the scale of formality to informality, and for 'what the user is trying to *do* with language (in a sense that is different from the purposive-role/field-of-discourse factors) for, or to, his addressee(s) – whether he is teaching, persuading, amusing, controlling, etc.'

The key to interpreting field certainly seems to lie in the *aboutness* feature, although Benson and Greaves' reference to things "going on" through language rather seems to point into a more pragmatic direction. Determining the mode – in the sense the term is being used in the extract, but not in general in this book – is certainly not an issue in the present analysis because we are only dealing with spoken language. Tenor, on the other hand, may be again relevant because it partly seems to encompass aspects of illocution and partly of interaction on such levels as politeness, something that combines aspects of pragmatics – in the form of the speech act – and the use of interactive conventions or devices that may signal illocutionary force – in other words, something similar to Searle's IFIDS.

Despite the fact that the ideas and terminology provided above indicate that there may be multiple levels of meaning, the quotation is still fairly elusive and does not provide any concrete indication as to how to define and annotate these levels in order to make use of them in understanding or explaining verbal interaction. One attempt to make these various levels more explicit is the DAMSL annotation scheme already briefly referred to in Section 1.3, which will now be discussed in some more detail, outlining its strengths and weaknesses in order to be able to contrast these better with the choices made for the DART approach.

5.1 The DAMSL annotation scheme

The DAMSL annotation manual, an outcome of the results of the first two meetings of the DRI (see Section 1.3), comprises a larger set of different content level *dimensions* (Allen & Core 1997: 4; cf. also Leech et al.: 2000: 57ff), to be applied in the form of *utterance-tags*, namely the following:

1. Communicative status,
2. Information level and status,
3. Forward-looking communicative function,
4. Backward-looking communicative function.

Each unit may be marked up on all four levels, but need not necessarily contain tags at all of them if this is not appropriate (cf. Allen & Core: 1997: 4).

Communicative status marks whether a unit is complete and intelligible, but also includes information concerning features of contextual relevance, such as *self-talk*. Examples for this from the Trainline data, using the DART conventions, would be.

(57) <frag n="96" sp-act="refer-abandon" status="abandon" polarity="positive" mode="abandon" >the telepho… <punc type="incomplete" /></frag>
(trainline03)

(58) < n="13" sp-act="answer-enumerate" polarity="positive" topic="enum" mode="decl" > 4 <punc type="stop" /> </frag> (trainline13)

(59) <frag n="69" sp-act="reqInfo" polarity="positive"> <unclear length="9 syllables" /></frag> (trainline07)

(60) <q-yn n="83" sp-act="reqInfo" polarity="positive" mode="counter_expectation -closed-query">does Stephen actually smoke <pause /> <event content="background voice says no" /> <punc type="query" /></q-yn> (trainline16)

Example (57) shows an abandoned, and hence incomplete, unit, indicated by the ellipsis (…), (58), how a single word spoken out of context may be uninterpretable, unless it is possible to link it to something in the prior discourse, while (59) demonstrates how it may sometimes simply not be possible to interpret a unit due to the bad quality of a recording or because a speaker is mumbling, etc., unless one inspects the context and is able to identify the function of the unit based on that. Of course, sometimes abandoned units, such as in (57), may also be interpretable from the surrounding context, which in this case was not possible, but even then, their status may be such that they were abandoned by the speaker precisely because he or she deemed the content unnecessary to the dialogue or because they were unsure about what exactly to contribute.

Example (60) represents communication which is not necessarily directly relevant to the interaction between the two main speakers because what is relevant for the exchange is only the actual result of the speaker's enquiry to the third party which is relayed to the other main interlocutor.

As DAMSL was primarily designed for marking up transactional dialogues, where there is a specific task to be fulfilled, this is also reflected in the sub-categories that are designed to reflect information level and status. Two of these categories are generally of less importance to the discussion here because they make a very fine distinction between communication that is simply there to achieve the task at hand (*Task*) and ways of managing the interaction at a level that is purely concerned with the task (*Task Management*), such as keeping track of the stages of the task and particular steps they may involve, something that is probably nearly impossible to achieve automatically, unless one is dealing with fixed plans (c.f. Section 1.3). The third one, however, *Communication-management* is something easily dealt with because it essentially comprises some of the features already talked about above, such as the use of DMs, formulaic expressions such as greetings and farewells, holding directives, etc. The fourth level, called *Other-level*, simply seems to be more or less a garbage category that may, according to its description (Allen & Core 1997: 8), also contain features like small talk that are reminiscent of some of those already discussed under communication status above, but which, strangely enough, "may be relevant to the dialog" (ibid.). The issue of relevance is treated in an odd way by Allen and Core, anyway, because they seem to regard units that signal communication-management as almost unnecessary to the dialogue when they claim that:

> In coding this dimension, you should remember that every utterance has a Communication component in some sense, but that utterances should be marked at the Communication-management level only when they make no direct contribution to solving the task. In other words, Communication-management level utterances are concerned exclusively with maintaining the conversation and if they were removed, the conversation might be less fluent but would still have the same content relative to the task and how it was solved. For instance, the greeting "hi" could be considered at the Task level in the sense that it starts the process of performing the task. Removing the utterance, however, would have no significant effect on the task or the way it was performed, thus we know its function is mainly at the Communication-management level. (ibid: 8)

However, since, as pointed out in 4.3.3 above, DMs that acknowledge may also signal a weak form of acceptance, and for instance accepting a suggestion is highly relevant towards keeping a dialogue going, the assumption that the dialogue simply becomes "less fluent" when such a unit is removed seems to be plainly incorrect.

The two remaining categories, *forward-looking communicative function* and *backward-looking communicative function*, as their names already suggest, represent categories that may either have an effect on how the dialogue is going to progress, or relate to something that has occurred in a prior unit or sequence of units, respectively.

The four sub-categories of the forward-looking communicative function, including some typical speech-act categories associated with them, are:

1. Statement: e.g. Assert, Reassert, Other-statement, etc.,
2. Influencing-addressee-future-action: e.g. Info-request, Open-option, Action-directive, etc.,
3. Committing-speaker-future-action: e.g. Offer, Commit, etc.
4. Other-forward-(looking-)function: dummy category for fixed, relatively rare functions like performatives, exclamatives, conventional-opening, conventional-closing, thanking, etc.

These categories clearly demonstrate that DAMSL was primarily designed to label discourse units according to their purpose manually, rather than achieving this automatically, as can also be seen in the following example from the DAMSL manual:

(61) Action-directive
A: Let's buy the living room furniture first.
B: OK.
OO, Assert, Offer
I have a red sofa for $150 or a blue one for $200. (ibid.: 13)

Here, first of all, the label for the unit produced by A seems to be peculiar, indicating that A has some kind of authority over B, when in actual fact, A simply seems to be making a suggestion. Using this label may be partly in line with the definition of *Action-directive*, which is that it "obligates the listener to either perform the requested action or communicate a refusal or inability to perform the action" (ibid.: 12), but only if one actually knows that this suggestion is coming from someone in authority and it may be taken as a definite directive. The first part of the label, *Action*, is also perhaps an unfortunate choice because an interlocutor who is obligated to perform a directive that includes a genuine action would actually have to comply by performing the action, with an acknowledgement being of secondary importance only.

The counterpart to the *Action-directive*, the *Open-option*, is the non-obligating version of an incentive to perform an action, but in B's final unit above, it is impossible to detect any kind of direct incentive, let alone an *Offer*, in the traditional sense, which would only express a commitment on the part of the speaker. In DAMSL, however, the *Offer* tag is "conditional on the listener's agreement" (ibid.: 13), so, in other words, the implicit assumption in the coding of the above example seems to be that A will essentially agree to one of the options, something that could only be determined with hindsight.

Applying the third label out of the forward-looking category in the example, the *Assert*, should be contingent upon the fact that "the speaker is trying to change the beliefs of the hearer" and "make claims about the world" (ibid.: 10). Those formulations alone seem to be somewhat philosophical and intangible, and I would personally relegate the issue of belief to the domain of religion, rather than making any strong claims as to what someone's beliefs were, purely based on one or even a few utterances. Some of the participants at the third DRI workshop at Chiba in 1998 seem to have thought about this along similar lines, and this is why they decided to revise the decision tree for the *Statement* supercategory as follows:

Does speaker make a claim?
  Yes. Tag as Statement
    Does speaker think claim already made?
      Yes. Tag as Reassert
      No. No tag.
No. No tag. (Core et al. 1998: 12)

Labels like the above *Assert* may also, at least partly, have been the reason why many of the participants of the Chiba Workshop found the DAMSL categories unwieldy and why these categories also had to be adapted and reduced to 42 basic tags for use with the Switchboard Corpus on the *Switchboard Discourse Language Modeling Project* (cf. Jurafsky et al. 1997).

The main problem with the *Info-Request* category is that it is a purely function-based one, that is it subsumes everything that "introduce[s] an obligation to provide an answer" (Allen & Core 1997: 10). Because of this, it also includes directives, such as "Tell me the time" (ibid: 11), where the function may well be to inquire about the present time, but with a further implication that the speaker clearly has some degree of authority over the addressee, information that would necessitate an additional tag *Action-directive* to clarify it when it is actually already part of the syntax.

The label for category 4 above is in fact a convenience label used in Leech et al. 2000 to group together a number of additional sub-categories, where DAMSL has this as a label for one of these sub-groups, namely anything that is not covered by any of the other forward-looking categories. I will retain this convenience label here because it groups together a kind of rag-bag of functions that cannot really be referred to as properly forward-looking.

While for instance a greeting may potentially be interpreted as forward-looking because it opens up the dialogue – but without actually contributing to the content proper –, the same can definitely not be said for saying *goodbye*, not even in German, where at least something like *Auf Wiedersehen/-hören* expresses an inkling of potential future interaction.

To find the examples listed under the heading of *Explicit-performatives*, "Thank you" and "I apologize" (Allen & Core 1997: 14), listed in a section on forward-looking functions is also decidedly odd, as they clearly represent different types of reaction to something that has gone on in the previous sections of a dialogue, and therefore rather ought to be characterised as backward-looking, or features that form part of communications-management because they help to maintain a 'peaceful' interaction between the dialogue partners. The same essentially goes for the remaining sub-categories of *Exclamation* and Other-forward-function, where the latter contains a single example of "Opps" (presumably *oops*) as "signalling an error" (ibid.: 15).

The DAMSL *backward-looking communicative function* comprises the following categories:

1. Agreement: e.g. accept, maybe, reject, hold, etc.,
2. Understanding: e.g. backchanneling, signal-non-understanding, signal-understanding, etc.,

3. Answer: generally signals compliance with a request for information,
4. Information-relation : utterances expressing explicitly how an utterance relates to the previous one,
5. Antecedents: any utterance may be marked as relating to more than just the preceding one.

Already the first category here, *Agreement*, provides food for discussion, as it seems to comprise quite a few of the more interactive types of speech act which were originally deemed of somewhat lesser importance and grouped under the communication-management level. The same goes for category 2, so the question here is really whether it should not be possible to set up a single functional category for all communications-management devices and then sub-classify them as to whether they may be deemed to be forward-looking (initiating) or backward-looking (responding).

Incidentally, the label *hold* in category 1 is used in a different, and rather misleading way in DAMSL in two respects. First of all, it does not really signal agreement, as its categorisation in DAMSL suggests (Allen & Core 1997: 18). Instead, it is generally used for labelling units that do not contain a direct response, but rather include such things as clarification questions, and therefore may only signal potential agreement in an indirect way, although they do not even need to do so at all. Secondly, as shown in 4.3.8 above, the term itself would suggest a type of unit whose function is to signal an interruption in the current phase of the dialogue on the verbal level, something that is clearly not the case judging by the examples cited in the DAMSL Manual.

Category (3), *Answer* does not present much of a problem, as it is mainly straightforward, apart from maybe one special feature. Allen and Core (1997) state in this respect:

> The Answer aspect is simply a binary dimension where utterances can be marked as complying with an info-request action in the antecedent. […] Most questions are answered with one or more declarative sentences although it is possible to answer a question with an imperative […]. In fact, answers by definition will always be asserts. (23)

The special issue hinted at above is that imperatives are also seen as answers, when generally they should probably better be considered directives, even if they occur in response to a question. Furthermore, if all answers were also asserts, as Allen and Core claim, then, in conclusion, directives that 'function' as answers are also asserts, which seems rather contradictory.

Categories 4 and 5 really belong together because *Information-Relations* are essentially supposed to express the relationship between a unit and its antecedent(s). However, what exactly these relations are is never made clear and the DAMSL

Manual only states that "[t]his category is currently not elaborated and will be subject to further study" (Allen & Core: 1997: 24).

As indicated above, DAMSL represents an interesting attempt to capture the multiple dimensions of meaning, and sometimes structure, that are inherent in spoken interaction. However, as I have hopefully demonstrated convincingly, and is also corroborated by the comments on DAMSL by the participants of the DRI Chiba workshop, applying this scheme in its original form is too difficult, due to the sometimes needless complexity and inherent contradictions. As I have equally illustrated, some of this complexity may be due to the fact that it is a purely functional scheme that largely neglects the syntactic form of units and their inherent default (semantico-)pragmatic properties. Nevertheless, since it was originally based on a consensus model, arrived at through the participation of experts from various fields of linguistics and computer science, many of the features discussed there are in fact highly useful starting points for building up taxonomies on various descriptive levels. The main problem, however, seems to be that these levels are not appropriately connected to the levels of grammar and meaning established in linguistic theory. In the following pages, I shall therefore make an attempt to rectify this, pointing out how one only needs to observe and describe a few separate levels and their interaction in order to understand the meaning of individual discourse units.

Apart from the default meanings already illustrated for the different syntactic units, in the following, two distinct levels of meaning will be discussed. These levels are those of *semantics* and *semantico-pragmatics*, which will be referred to as those of *topics* and *modes* respectively. The term *mode* is obviously quite distinct from the Hallidayan use (c.f. the beginning of this chapter) and was coined in order to show that the features that belong to this category of meaning largely describe the *modus operandi* of a dialogue, in other words what goes on between two or more participants at the communicative and interpersonal interaction levels, as opposed to the specific semantic content (*topic*) of a unit. Comparing the difference between syntax, topics and modes in terms of cognitive approaches to linguistics and construction grammar, one could say that syntactic unit types are *schematic* in nature, whereas topics and modes are *substantive* (cf. Croft & Cruse 2004: 225 ff.) in that they give the relatively abstract meaning potential of the syntactic units a more specific meaning. The following discussion of non-syntactic content will begin with a look at the different types of modes and which properties of the dialogue exactly they reflect.

5.2 Modes

Some generic aspects of language were already identified in the discussion of the generic lexicon earlier, and to some extent also in the discussion of prototypical functions of syntactic units. As a matter of fact, the whole design concept behind the DART approach to pragmatic analysis revolves around the assumption that it is possible to model those features in language that are common to all types of language interaction, and, just as with the generic lexicon, to augment these models by appropriate additional resources for specific domains of discourse.

Modes represent another level of 'genericity', a type of content that is essentially semantico-pragmatic in nature, and can be expected to be found in any type of spoken and written interaction. This is because modes are extremely repetitive textual patterns that express high-level categories of 'aboutness', cognitive concepts that characterise particular intentional aspects of spoken interaction, essentially bridging the gap between syntax and the meaning or function of individual words in a unit. As such, modes provide a simple way around having to perform a deep semantic analysis of a unit in order to help determine its speech act. The mode system in DART, however, although it is already being used successfully in DART's inferencing process, is still under constant development, so at least some of the categories described below will probably see further enhancements in the levels of distinction they offer in the future.

It may be best to begin illustrating this concept by looking at what Searle discusses under the heading of "The speech act fallacy" (1969: 136ff.) with regard to words as individual markers of speech acts – or IFIDs, if you will. There he says:

> In the classical period of linguistic analysis, philosophers often said things like the following:
>
> The word "good" is used to commend (Hare).
> The word "true" is used to endorse or concede statements (Strawson).
> The word "know" is used to give guarantees (Austin).
>
> The word "probably" is used to qualify commitments (Toulmin).
>
> Each of these is of the pattern: "The word *W* is used to perform speech act *A*." Furthermore, it was generally the case that philosophers who said this sort of thing offered these statements as (at least partial) explications of the meanings of the words: they offered these statements of the form "*W* is used to perform act *A*" by way of philosophical explication of the concept *W*. Notice also that, in doing so, they drew – in most cases explicitly – an analogy between the words that they were discussing and the so-called performative verbs. Just as "promise" is used to make promises, and "bet" to make bets, so they argued "good" is used to commend, and "true" is used to endorse, etc. (ibid.: 137)

However, he then goes on to refute this notion, providing two separate ways of disproving it, thereby making clear that it is not the individual words per se that allow the speaker to perform a speech act, but that they do depend on a particular syntactic or semantic context to be able to acquire this function.

If one follows this argument further, it becomes clear that it is not always possible to simply employ a one-to-one mapping between potential single-word IFIDs and their pragmatic uses, although in some – limited – cases this may be an option. Rather, what is essentially required is a slightly more complex method of relating linguistic constructs, ranging from single- to multi-word units, to a particular semantico-pragmatic function. In other words, one needs to identify specific *collocations* or *colligations* that may potentially be relevant towards performing a given verbal action, and relate these to particular syntactic structures in order to be able to determine the illocutionary force of the unit. By doing so, one essentially arrives at what Fraser (1996) refers to as "hybrid basic markers", although his classification scheme is nowhere near as extensive as the one used here.

Being highly grammaticalised, the meaning or function of many modes is what, following Grice (1989), we could frequently categorise as belonging to the class of *conventional implicatures*, as can be concluded from his famous discussion of the word *therefore*, which is classed as an interactional mode in DART:

> In some cases the conventional meaning of the words used will determine what is implicated, besides helping to determine what is said. If I say (smugly), *He is an Englishman; he is, therefore, brave*, I have certainly committed myself, by virtue of the meaning of the words, to its being the case that his being brave is a consequence of (follows from) his being an Englishman. But while I have said that he is an Englishman, and said that he is brave, I do not want to say that I have *said* (in the favored sense) that it follows from his being an Englishman that he is brave, though I have certainly indicated, and so implicated, that this is so. […] So some implicatures are conventional […]. (Grice 1989: 25–26)

However, modes certainly do not need to have this 'air' of implying anything "smug", as this is more of a function of the semantics of the two units being linked together logically through the implication inherent in the word *therefore*, which, in effect, indicates a conclusion. Grice's example also indicates an important link between modes and the DMs that fulfil a textual function discussed in 4.3.3 above. Thus, the main difference between the use of *therefore* in the above example and the same word as a textual DM is that the former is embedded within a unit, while the latter would be treated as a separate unit, due to its focussed, and thus textually more prominent, position. However, the function of this word as an IFID is still clearly highlighted by a similarity to the DM in that that it is prosodically marked as a parenthetical. This duality of existence and function of particular expressions is something we do encounter frequently, so that DM patterns often co-exist as modes, although, due to being embedded, they may sometimes be realised in slightly different ways.

To put modes further into perspective, one can draw an analogy to the DAMSL model. There, many of the properties that the DART scheme captures as modes are features of the dialogue that would fall under the heading of *Communications-Management* (Allen & Core 1997: 7), with the main difference that the DAMSL categorisation only marks very specific units as belonging to this level, whereas the DART scheme assumes these properties to be present in almost all units.

Modes can roughly be grouped into six relatively distinct conceptual fields:

1. grammatical modes,
2. interactional modes,
3. point-of-view modes,
4. volition and personal stance,
5. social modes, and
6. syntax-indicating.

In order to demonstrate their nature, and discuss their relevance to the analysis, I shall list and discuss some (non-exhaustive) patterns from each of the six categories. To simplify the patterns, I shall use PP1 to refer to 1st person subject pronouns, PP2 to 2nd person, and PP3 to 3rd person, as well as Adv for intensifying adverbs and Adj for potential modifying adjectives. The actual application of modes to the analysis of English dialogue data and the recognition of speech act types through inferencing processes will be discussed in more detail in Section 6.5 below. The potential application of the concept to three other languages, German, French, and Korean, is demonstrated in Weisser (2010).

5.2.1 Grammatical modes

Grammatical modes mainly tend to be represented by conjunctions or adverbials. Here, unlike for the other mode categories, one also frequently encounters single-word constructions, due to the heavily conventionalised grammatical nature of these items, which frequently makes them relatively unambiguous in their use. Grammatical modes often tend to function as items that foster both cohesion and coherence, and to reflect conditions, constraints, and/or logical connections.

Table 5.1 and those in the other sections contain non-exhaustive lists of potential expressions or patterns that have been simplified considerably from the actual coding in order to make them easier to understand for readers who are not familiar with the (customised) regular expression syntax used to model them in DART. Semicola are used to separate the individual expressions, and optional parts or comments have been added in round brackets, with alternative words being separated by forward slashes.

Table 5.1 Grammatical modes

Mode label	Pattern
condition	if; whether; unless; as long as; while (not followed by at/in)
constrain	have (got) to; must; need (not followed by negation), unfortunately, subject to; the problem is…; that's too…
reason	cos; because; that's why; due to
contrast	(al)though; but; however; instead; whereas; on the one/other hand
alternative	(n)either; (n)or; otherwise
exists	there's/ is/are; PP1 've got/have (not followed by to)

Grammatical modes can be grouped into two sub-divisions. The first comprises the modes *condition, constrain*, and *reason*, which all describe (external) circumstances that need to obtain for an option or set of options, such as a potential course of action, to become true, or to explain the necessity for doing something. The second group, containing the modes *contrast, alternative* and *exists*, express limitations, options, the availability of relevant items, or alternatives that are important in achieving a task or goal.

The importance of grammatical modes for particular parts of a dialogue lies in the fact that they signal conditions or circumstances imposed on the participants in the dialogue, which may force the interlocutors to take particular actions or adopt certain strategies, thereby directly influencing the outcomes of particular stages of the dialogue. Although many of the markers for grammatical modes tend to be conjunctions, not all conjunctions in fact exhibit this feature of influencing the interaction in the way that grammatical modes do. For instance, purely cohesive conjunctions like *furthermore* do not necessarily help to constrain the dialogue or the information conveyed with the same degree of immediacy as an *if* or *because* do. On the other hand, though, cohesive markers may play a crucial role in establishing rhetorical relations that span across a larger number of units of the dialogue when used as textual DMs.

5.2.2 Interactional modes

The second category comprises interactional modes. These mainly represent patterns that essentially express reactions to dialogue content the other participant has provided, such as backchannels, or may also comprise initiating *moves* that signal the intent to do something, such as making an offer. As such, they often contain features that the DAMSL model discusses under the heading of *Task Management* (Allen & Core 1997: 7). In the DAMSL-sense, many of these interactional modes may also be seen as both forward- and backward-looking. However, one can still more or less group some of the sub-categories together according to their primary orientation, which will be done in the following tables, starting with the primarily backward-looking ones.

Table 5.2 Backward-looking interactional modes

Mode label	Pattern
admit	PP1 (honestly) have to say/admit
reassurance	that's ok/fine; will do fine/nicely
agree	no problem; fair enough; i (absolutely/completely) (do) agree
refuse	PP1 (have to) refuse/decline
reject	it/that's too; this is too; not at all; never mind
correct	that's what PP1 mean(t)

These six modes are clearly of the backward-looking, reacting type. Most of these modes, apart from *correct*, have a very similar function to that of the corresponding DM or yes-no units described earlier. The main difference to most of the DMs, though, is that the mode patterns often tend to be more explicit. The label *reassurance*, for instance, is here used for expressions that clearly agree with what the other party has stated or suggested, and therefore represents a kind of acceptance that makes it possible for the dialogue to continue in the same direction, rather than e.g. reversing tack or starting over again completely with a particular (sub-)dialogue. The *correct* mode, in contrast, signals that there has been some kind of potential misunderstanding which is thereby signalled as being corrected or clarified.

The next group encompasses modes that tend to be primarily forward-looking, so that they either elicit responses or to establish new conditions affecting the interaction or status quo.

The modes in Table 5.3 signal functions that indicate to the interlocutor that the speaker is committing themselves to a certain course of action. The first five of them, *promise, offer, suggest, bet*, and *perform* are fairly reminiscent of the original performative speech acts, as envisaged by Austin (1962), although *suggest*

Table 5.3 Forward-looking interactional modes

Mode label	Pattern
promise	PP1 promise
offer	PP1 (can) offer
suggest	PP1 to PP3 should/ought to; i/we ('d/would) (like to) suggest
bet	PP1 bet
perform	PP1 hereby

(*Continued*)

Table 5.3 (*Continued*)

Mode label	Pattern
intent	PP1'll/will just: PP1'm (not) going to; PP1'd like to
benefit1	for/to me/us; give/help me/us
benefit2	for/to you; give/help you
request	please (not followed by NP)
query	at what time/when/where; how big/far/large/long/many/much/old/tall/wide (at the end of the unit)

frequently may not exhibit the same performative structure, but instead a more 'indirect' approach using an auxiliary, e.g. *i would (like to) suggest…*, may be chosen by speakers. The *perform* pattern is something of a 'garbage category' and meant to capture any other performative patterns clearly marked by the 'performative marker' *hereby*.

Almost all modes in this sub-group are '1st-person-centred', which signals the commitment on the part of the speaker, unless of course the pattern should occur inside a syntactic unit that is marked as being of a querying nature such as *why should I promise…*, etc. One exception to this general rule are perhaps the *benefit* sub-categories, where *benefit1*, that is 1st person-oriented benefit, in effect signals a requestive, while *benefit2* is related to offers. *Intent* is also still 1st person-centred, but while the previous examples suggested at least a partial involvement of both interlocutors, this mode is supposed to capture the concrete plans and intentions of the speaker.

(62) <decl n="89" sp-act="stateIntent" polarity="positive" topic="seat-booking" mode="**intent**-decl" >
i'm going to book you a forward-facing seat again <pause length="2s" />
</decl> (trainline01)

The next two modes, *benefit1*, and *benefit2*, are designed to ascertain whether either party might benefit from to the intended actions of the other, or provide a benefit to them both. Modes may be 'syntax-sensitive', though, so while type 2 may occur in both declaratives and questions, type 1 only has the function of an IFID if it occurs inside a request, but not in any stating expression, as can be seen in Example (64).

(63) <q-yn n="2" sp-act="offer" polarity="positive" mode="**benefit2**-closed-query">
<pause /> can i **help you** <punc type="query" />
</q-yn> (d93–11.2)

(64) <decl n="163" sp-act="state" polarity="positive" topic="enum-transport_means-commodities" mode="**benefit1**-decl">

<pause /> and we have <pause /> 2 boxcars of oranges waiting **for us** <punc type="stop" />
</decl> (d93–12.4)

The final two patterns represent the 'lexical counterparts' to the syntactically marked categories of imperatives and questions when their distinctive syntactic patterns are absent, for instance in some types of 'declarative' questions.

(65) <decl n="106" sp-act="direct" polarity="positive" topic="address-confirm" mode="request-decl" >
<pause /> you confirm that address for me please <punc type="stop" />
</decl> (trainline03)

(66) <frag n="98" sp-act="reqInfo" polarity="positive" topic="time-arrival" mode="**query**-suggest ">
should arrive **at what time** <punc type="query" />
</frag> (d93–16.1)

Having said that these modes are primarily forward-looking, of course none of them occurs in a vacuum, but each action the speaker makes a commitment to in using such a mode frequently also constitutes a reaction to something that has gone on previously to some extent. Mode patterns that signal this to an even stronger degree, and can hence be assumed to be both backward- and forward-looking, are listed in the following table.

Table 5.4 'Bi-directional' interactional modes

Mode label	Pattern
abandon	…
hold	bear with me; hold the line; let me think
report	PP1 am/are/were/was told/quoted/informed; PP1've been told/quoted/informed; you say/said/are saying; apparently
predict	PP3/it/that/'ll/ will be/get; you'll be able to; PP1 am/are sure we'll/'re going to; will/won't be
tag	is/does/has/will/won't/would/should/shan't/shall PP3; are/do/have/haven't/will/won't/would/should/shan't/shall PP1–3; (al)right; do/don't/are(n't) you; did you say

At the top of the table, the *abandon* mode is listed as a sort of 'odd one out' because, unless there is a clear interruption, all that is known about units identified as being abandoned is that for some reason the speaker has chosen not to complete whatever unit they were going to produce. Nevertheless, it can still be assumed that an abandoned unit is backward-looking in the sense that the speaker evaluates what he or she has said before, and at the same time forward-looking, due to the decision not to complete the unit as it has probably been deemed irrelevant or inappropriate to the current flow of the dialogue.

The *hold* mode pattern constitutes part of the dialogue management strategy and, as shown earlier in the discussion of *hold* imperatives in 4.3.8, prepares the interlocutor for a potentially elongated period of waiting. *Report* simply indicates that the speaker is in possession of some – sometimes presumed – knowledge acquired earlier, which may be relevant to the progress of the dialogue. This mode is clearly both backward-looking in the sense that it may even refer to some exophoric (cf. Halliday & Hassan 1976: 33) information established prior to the current dialogue situation, and forward-looking because the speaker may essentially be trying to influence the hearer in either making a collaborative effort to achieve the dialogue (sub-)goal or to 'play a trump card' to his or her own advantage, as in the following exchange.

(67) <turn n="27" speaker="Sandra">
<dm n="46" sp-act="init">
now <punc type="level" />
</dm>
<decl n="47" sp-act="stateIntent-hold" polarity="positive" topic="fare-verify-availability" mode="intent-hold-decl">
i'm just going to check with you what's your cheapest fare available to you
<punc type="stop" />
</decl>
</turn>
<turn n="28" speaker="caller_02">
<dm n="48" sp-act="agree">
sure <punc type="stop" />
</dm>
<decl n="49" sp-act="report" polarity="positive" topic="enum-fare" mode="**report**-decl">
<pause length="9s" /> **i was quoted** 19 pounds <punc type="stop" />
</decl>
</turn>
<turn n="29" speaker="Sandra">
<decl n="50" sp-act="state" polarity="negative" topic="availability-fare-date-enum" mode="decl">
the 19 pounds is not available at the dates and times specified <punc type="stop" />
</decl> (trainline02)

In this example, caller_02 apparently wants to let Sandra know that he is perfectly aware of the cheapest option for a ticket and that this should form the basis of further 'fare negotiations'. Unfortunately, though, this type of strategy does not work out because Sandra introduces new constraining circumstances that invalidate caller_02's claim for a cheap ticket, thereby achieving a certain kind of 'authority' over the customer despite the fact that she, as the one who is offering the service to the customer, is generally in a weaker position.

The pattern expressed by *predict* evaluates some potential options that have been referred to more or less explicitly in the preceding context as to their outcome, as in Example (68).

(68) <decl n="67" sp-act="predict" polarity="positive" topic="health" mode="**predict**-decl">
it will be interesting to see <unclear /> <punc type="stop" />
</decl> (sw_0003_4103)

(69) <decl n="80" sp-act="answer-predict" polarity="positive" topic="enum-to-duration" mode="**predict**-decl" >
that'll take <pause /> 3 hours to get to Dansville
</decl> (trains d92a-1.2)

The assumption(s) articulated by the speaker generally affect future decisions of either the hearer or both interlocutors as to how to proceed with the dialogue or decision-making process. The final mode pattern in this category, *tag*, captures the kind of interactional meaning usually expressed by tag questions or DMs that occur in unit-final position, something that is generally indicative of requests for confirmation.

5.2.3 Point-of-view modes

The modes listed in the table below include patterns that are often discussed under the headings of *knowledge* or *belief* in the relevant literature on dialogue acts, as also demonstrated above in the definition of *assert* in DAMSL, or categorised as being *epistemic* in discussions of modality.

Table 5.5 Point-of-view modes

Mode label	Pattern
awareness	PP1 know/realise/understand; PP1'm/'re aware
nonawareness	PP1 don't (quite/really) know/realise/see/understand; PP1'm/are (not) aware/unaware
doubt	PP1 doubt/wonder if; PP1'm/'re just wondering if
opinion	PP1 think/suppose/believe; my/our belief/opinion
poss1	PP1 can/could; PP1 might/may/am/'m/are/'re/be able (none followed by or involving negation)

(*Continued*)

Table 5.5 (*Continued*)

Mode label	Pattern
poss2	PP2 can/could; PP2 might/may/ be able (none followed by or involving negation)
poss3	PP3/there can/could might/may/are/be able (none followed by or involving negation)
imPoss1	negative counterparts to the above; 1st, 2nd, and 3rd person impossibility
imPoss2	
imPoss3	
probability	probably/likely; probability/likelihood

Treating the categories shown above as modes accords them less of a 'deeply philosophical' status, but rather just serves as a means of flagging what the particular point-of-view of a dialogue participant may be, or which ideational stance they may be taking. In order to illustrate why this approach may be more sensible, here is another quote from Searle:

> Human communication has some extraordinary properties, not shared by most other kinds of human behaviour. One of the most extraordinary is this: If I am trying to tell someone something, then (assuming certain conditions are satisfied) as soon as he recognises that I am trying to tell him something and exactly what it is I am trying to tell him, I have succeeded in telling it to him. Furthermore, unless he recognises that I am trying to tell him and what I am trying to tell him, I do not fully succeed in telling it to him. In the case of illocutionary acts we succeed in what we are trying to do by getting our audience to recognize what we are trying to do. But the 'effect' on the hearer is not a belief or response, it consists simply in the hearer understanding the utterance of the speaker. (Searle 1969: 47)

In other words, what Searle states here is that one should not simply assume to be actually changing someone's state of mind by performing an illocutionary act, but rather just to be providing them with the potential means for understanding and adjusting their ideas accordingly. Thus, what these modes capture is the potential state of mind or knowledge of the speaker, as they are trying to present them to the interlocutor, rather than any presumed attitudes or beliefs.

The four types in this group of modes almost all have a different degree of force, with *awareness* (70 & 71) having the strongest, allowing the speaker to signal that he or she is clearly conscious of what is really going on or what the outcome e.g. of a decision may be. *Nonawareness* (72), as the negative counterpart to *awareness*, has a somewhat similar force, but semantically obviously expresses a lesser degree of 'conviction'. The sub-category *doubt* (73 & 74) occupies the 'middle ground' in terms of strength, along with perhaps a negative connotation as concerns the outcome or status of something the previous speaker has stated, or some general insecurity concerning the potential options available. And finally, *opinion* (75) is the weakest because it simply signals a potential, but not a definite state of mind, such as e.g. *awareness* does.

(70) <decl n="100" sp-act="expressAwareness" polarity="positive" topic="relationships" mode="**awareness**-decl">
and and **i know** a friend of mine talked to someone in the midwest <punc type="stop" />
</decl> (sw_0090_3133)

(71) <dm n="183" sp-act="acknowledge" mode="**awareness**">
i see <punc type="stop" />
</dm> (d93–11.1)

(72) <decl n="19" sp-act="expressNonAwareness" polarity="negative" mode="**nonawareness**-decl">
<pause /> **i don't know** <punc type="stop" />
</decl> (d93–17.2)

(73) <decl n="90" sp-act="expressUncertainty" polarity="negative" mode="contrast-opinion-condition-**doubt**-decl">
but **i wonder if** maybe in some civil cases it doesn't work as well <punc type="stop" />
</decl> (sw_0017_4036)

(74) <n="17" sp-act="reqInfo" polarity="positive" topic="fare-availability" mode="**doubt**-condition-exists-query">
erm **i'm wondering** if there's any erm seats available on the Super Saver <punc type="query" />
</decl> (trainline16)

(75) <decl n="48" sp-act="expressOpinion" polarity="positive" mode="**opinion**-frag">
i think people are <comment type="restart" /> are working at that from more of a <comment type="restart" /> uh a gut reaction than a <comment type="restart" /> a reason <punc type="level" />
</decl> (sw_0061_4151)

As can be seen in Examples (70) and (71) above, *awareness* can be signalled via declarative units, as well as DMs.

Finally, the last group encompasses features of modality, *possibility* and *probability*, which may determine the outcome of a course of action. In terms of possibility, it is also important to distinguish between 1st (poss1), 2nd (poss2), and 3rd (poss3) person, as well as their negative counterparts reflecting *impossibility*.

5.2.4 Volition and personal stance modes

Rather than evaluating or pertaining to factual information, volition and personal stance modes reflect the wishes/hopes and attitudes of speakers or third parties being talked about.

Table 5.6 Volition and personal stance modes

Mode label	English
hope	PP1 (Adv.) hope; if (only) PP1 can/could/may/might; hopefully; PP1 (just/only) wish; my/our (only) wish is (all not followed by negation)
preference1	PP1 (Adv) prefer; (not preceded by *if*) PP1 (Adv) want(ed/s); PP1 (Adv) wanna; PP1 (Adv) hope; PP1 ('d| would) rather; PP1'll go for; PP1 ('d/'ve/ would) (Adv) enjoy/like (to); preferably; appeals to me; PP1 (Adv) enjoy/ like/love(not followed by *you*); PP1 (Adv) look forward to
preference2	you(se) prefer; you want(ed); you wanna; you wish; you hope; you('d/ would) like; you 'd/ would rather; would you rather; you'll go for; you like to; appeal to you
preference3	PP3 prefer(s); PP3 wanted/wants; PP3 wanna; PP3 wish(es); PP3 hopes?; PP3('d/ would) like(s) to; PP3 ('d/ would) rather; PP3 'll go for; PP3 like(s) to; appeals to her/him)/it
dispreference1	PP1 (Adv) don't/ do not/never enjoy/(dis)like; PP1'd/ would (Adv) hate; PP1 (Adv) can't/can ?not abide/bear/deal with/stand
stance	PP1 am/are not bothered/worried; PP1 don't (really) argue against/with it/care/mind/have a problem with; PP1 don't give a (optional Adj.) fuck/ toss/shit about; never mind; it absolutely/genuinely/really does (n't/not) (really) matter; PP1 tolerate; it's (Adv.) not worth
disappointment1	a bummer; PP1'm/ are (Adv) disappointed

The first mode in Table 5.6 expresses the hopes of a speaker (Example (76)), or perhaps their wishes, as a distinction between these is not always easy to make. Unlike the *intent* mode we saw earlier, this mode is not forward-looking in the same sense, and therefore also not interactional, as the speaker is generally not in control of what is actually going to happen. Of course, expressions of (good) hope or (well-)wishing may also be directed at the interlocutor, but these are quite different from the above and are currently not (yet) modelled in DART. If they were, though, they would need to be classed under social modes, anyway.

(76) <decl n="78" sp-act="expressHope" polarity="positive" mode="intent-**hope**-decl">
hopefully eventually we'll move in a larger home <punc type="stop" />
</decl> (sw_0087_2775)

(77) <decl n="174" sp-act="expressDislike" polarity="negative" mode="**preference1**-decl">
i never liked the uh insanity defense <punc type="stop" />
</decl> (sw_0028_4133)

(78) <q-wh n="14" sp-act="reqInfo" polarity="positive" topic="to" mode="**preference2**-open-query">
how **would you like** <pause /> to send them to Bath <punc type="query" />
</q-wh> (d92a-5.1)

Expressions of *preference* in DART were originally introduced to model potential preferences regarding seat or ticket choices elicited by Sandra in the Trainline data, but have since come to also encompass expressions that signify likes or dislikes, so currently no real distinction between liking and preferring is captured at the mode level. This, however, will probably be introduced at a later stage of the development. Nevertheless, we can already distinguish quite easily here between the different types, where type 1 (Example (77)) refers to such preferences/likes directly expressed by the speaker themselves, type 2 (Example (78)) models preferences for enquiries made by the interlocutor, that is how they may be elicited, and type 3 captures how either speaker may refer to them when they are not their own, but instead features related to third parties, i.e. in a more descriptive context. Example (77) is particularly interesting, as it illustrates the interplay between the mode and negative polarity. All three of these types essentially do have negative counterparts, but here, type 1 *dispreference* tends to be most relevant to spoken interactions in our three corpora.

While the preference and dispreference modes signal either positive or negative attitudes of a speaker, *stance* (Example (79)) indicates a more 'neutral' attitude where the speaker does not really care about what the options are or does not want to make a commitment in either direction. The label is thus used in a fairly different way from the way it otherwise tends to be used in linguistics, where it often refers to commitments made by a speaker or writer towards facts.

(79) <decl n="91" sp-act="expressStance" polarity="negative" topic="problem" mode="**stance**-decl">
i don't have that big a problem with it <punc type="stop" />
</decl> (sw_0041_4048)

The final label listed above, *disappointment1*, has no counterpart in any of the above, but may signal a general attitude towards something being referred to in the dialogue.

(80) <decl n="96" sp-act="expressDisappointment" polarity="positive" topic="habit" mode="**disappointment1**-decl">
that's always **a bummer** too <punc type="stop" />
</decl> (sw_0088_3073)

Incidentally, Example (80) is the only example of such an expression of disappointment in all three corpora.

5.2.5 Social modes

Social modes are relatively self-explanatory, so I will not provide separate examples for them. They mainly comprise 'the usual' greetings or goodbyes that are customary for initiating or concluding any type of interaction, as well as 'interpersonal' expressions, such as those of sympathy/empathy, regret or appreciation, but also including the negative interaction side in the form of insults.

Table 5.7 Social modes

Mode label	Pattern
apology	PP1 apologise; my/our apology/apologies; my mistake
appreciate	no problem, that would be brilliant/correct/fine/great/lovely/wonderful
thank	thanks; thank you
greet	hi; hello; good afternoon/morning/evening
bye	good(bye)
intro	name + *speaking*
regret	PP1'm (very)sorry, PP1 regret
expletive	oh shit; damn/blast (not followed by *you*)
insult	you bastard/idiot; damn/blast/fuck/screw you

One of the above modes, *intro*, tends to be relatively specific to task-oriented or telephone-based interaction, at least in the form that they appear in some of the dialogue material used for this study. The English example given for *intro* here refers to an introduction by the operator who is identifying themselves, but which is unlike a formal introduction. In the Trainline data, this introduction always takes the form *Sandra speaking*, but obviously any proper name followed by *speaking* at the end of the unit would be possible.

Although many of the labels given for the modes discussed above may in fact look like speech act labels themselves, they should not be mistaken for such, but in reality be seen as pointers towards the identification of a particular speech act as expressed in a unit and which may occasionally trigger the assignment of a speech act by the same name.

5.2.6 Syntax-indicating modes

There are two different types of syntax-indicating modes in DART, lexico-grammatical patterns and `<punc … />` elements. The first type consist in patterns that signal question tags and other queries, all found at the end of a unit.

Table 5.8 Syntax-indicating lexicogrammarical modes

Mode label	Pattern
tag	(not preceded by *who*) is/did/does/has/will/would/should (negation operator) PP1/PP3/there/they/we; (not preceded by *to* or *can*) are/do/have/will/would/should (megation operator) PP1/PP3; alright/right; yeah/yes; do/are (n't) you; did you say; (not preceded by *be*) ok (now/then); innit; (h)ey/huh; (not preceded by *that's*) correct
query	at/around what time/when/where; how big/far/large/long/many/much/old/tall/wide; (at least one word preceding) what; how many/much (at least one word following; you mean

The other type simply makes use of any existing <punc … /> elements and maps their type attributes to their corresponding mode values.

Table 5.9 Syntax-indicating modes based on <punc … /> elements

Mode label	Pattern
decl	<punc type=["']stop["']
query	<punc type=["']query["']
exclaim	<punc type=["']exclaim["']
frag	<punc type=["']level["']
abandon	<punc type=["']incomplete["']

As can be seen from the above two tables, the *query* mode may therefore be triggered either by a lexico-grammatical pattern or 'punctuation'.

5.3 Topics

The concept of *topic* in discourse is certainly not an uncontroversial one and there exist a number of different ways of defining what exactly a topic may be. With regard to this, Brown and Yule (1983: 70) state:

> The notion of 'topic' is clearly an intuitively satisfactory way of describing the unifying principle which makes one stretch of discourse 'about' something and the next stretch 'about' something else, for it is appealed to very frequently in the discourse analysis literature.
>
> Yet, the basis for the identification of 'topic' is rarely made explicit. In fact, 'topic' could be described as the most frequently used, unexplained, term in the analysis of discourse.

They then go on to discuss the different notions of *sentence topic* (ibid.: 70–71) and *discourse topic* (ibid: 71 ff.), illustrating the problems in deciding on exact 'labels' for these, also pointing out that there is a certain degree of arbitrariness to this process and that there may be multiple factors that could influence the assignment of such labels. To tackle these issues, as well as to be able to incorporate information from different context sources into this decision-making process, they suggest to develop a *topic framework*:

> Those aspects of the context which are directly reflected in the text, and which need to be called upon to interpret the text, we shall refer to as *activated features of context* and suggest that they constitute the contextual framework within which the topic is constituted, that is, *the topic framework*. (ibid: 75)

What elements exactly according to Brown and Yule find their way into this framework is of no immediate concern to the current approach, but another concern voiced by them may possibly help us to understand how a suitable topic identification process or framework can be approached from a computational point of view, and what its limitations may be.

> Remembering that any discourse data to which the analyst has access will only be a fragment, it would be extremely difficult for the analyst to predetermine the complete set of discourse subjects which participants share prior to a particular discourse fragment. The most he could hope to provide is a partial set. The problem to be faced is that of limiting the choice of the contents of even a partial set, in some non-arbitrary way. (ibid.: 80)

Topics are essentially semantic in nature, that is, each topic describes specific properties of a particular (sub-)part of the overall discourse, in other words what it is actually about. Since what is being talked about in a dialogue is generally more or less specific to the particular domain under discussion, or the 'job at hand', and some semantic content may even mark particular stages of a dialogue, topics also tend to be relatively domain-specific. Thus it should be possible to empirically arrive at partial sets, such as the ones referred to by Brown and Yule, by observing recurrent information structures or chunks of information in particular domains, and investigating how patterns that instantiate particular discourse topics might be formed, as well as which kind of variation could be present on the paradigmatic axis. Once these patterns have roughly been identified, it is then possible to specify them as regular expressions, and group them under an associated topic label which can again be recorded each time a particular pattern is encountered by an automatic analysis routine. Such a procedure is loosely reminiscent of attempts to describe particular frames (Minsky 1974) or scripts (Schank & Abelson 1977) for certain activities or domains.

However, despite the fact that particular domains of discourse generally contain specific topic patterns, there are also some topics that are more likely to recur across different types of discourse situation, and therefore merit the label *generic topics*. Just as with the items in the generic lexicon and the modes discussed above, these can directly be re-used even for an analysis of previously unexplored domains, and therefore represent a part of the generic core of all discourse elements.

In DAMSL terms, topics roughly encompass the semantic content that is expressed by the *Task* category at the *Information-Level* (Allen & Core 1997: 6–7), although, unlike DAMSL, the DART approach again assumes that this type of content is present in almost all types of unit, apart from the mainly interactional ones, such as DMs and Yes/No-like units.

5.3.1 Generic topics

Generic topics primarily encompass deictic references to times, places, directions, as well as personal details, such as addresses, telephone numbers, etc. These are partly also covered in Nattinger and DeCarrico's chapter called "Necessary topics" (63–64), that is those that even language learners "will be asked [about], or ones they need to talk about frequently" (ibid.: 60). Even though, for instance, topics like dates, times and places may at first glance seem very specific bits of information that mainly characterise the domain of appointment scheduling, they actually represent an integral part of many (if not most) different types of discourse because they relate to tasks that we commonly need to perform in daily life. They are not even necessarily restricted to transactional exchanges, but may also occur in *ordinary* dialogue (cf. Leech & Weisser 2003: 146). For example, just because an arrangement between two friends to meet at a particular time is not usually referred to as an appointment, this does not mean that the steps that need to be taken, as well as the semantic content expressed, to fix a time and a place for such a meeting are not roughly the same as those required for making an official appointment between business partners or at an office. Similarly, one also frequently needs to provide various types of interlocutors with one's credentials in the form of addresses, telephone numbers, etc., in daily life, no matter whether this may be on more official 'business', such as opening a bank account or making a booking, or simply to provide a new acquaintance with the means of contacting one for social reasons.

The following tables of individual sub-groups of generic topics used in the DART scheme will hopefully exemplify their generic status.

Table 5.10 Measures, enumerations & spellings

Topic label	Patterns
spell	<anonym type="letter" />; <anonym type="alpha" />; any single letter
number	how many; (not preceded by *this/that*) one (not followed by *of*); two; three; four; five; six; seven; eight; nine; ten; eleven; number; (not preceded by *and*) once; multiple digits
enum	single or multiple numbers in a row; double; treble; triple; words ending in -uple

This group is perhaps the most basic of all, as it contains items that are necessary for the processes of quantifying or spelling out details, something that can occur in any kind of dialogue. Spellings (*spell*) may be needed in clarifying misheard or ambiguous words, or often proper names, which may be notoriously difficult to get right. *Number*s are required for obtaining and/or providing information about quantities or frequencies of events. And last, but not least, enumerations (*enum*) are used to convey different types of numerical details, such as house, telephone or credit card numbers, amounts, times and dates; in short, anywhere where there is a sequence of multiple numbers involved. Enumerations themselves represent an extremely vague category, and their exact significance is usually established only through other contextual topic information.

(81) <decl n="263" sp-act="state" polarity="positive" topic="name-**spell**" mode="decl">
it's Mr **<anonym type="letter"** /> <anonym type="name" /> <punc type="stop" />
</decl>
<frag n="264" sp-act="refer" polarity="positive" topic="name" mode="decl">
<pause /> Mr <anonym type="name" /> <punc type="stop" />
</frag>
<frag n="265" sp-act="spell" polarity="positive" topic="spell" mode="partial-decl-disflu">
<anonym type="letter" /> **<anonym type="letter"** /> **<anonym type="letter"** /> **<anonym type="letter"** /> **<anonym type="letter"** /> **<anonym type="letter"** /> <punc type="stop" />
</frag> (trainline28)

(82) <decl n="78" sp-act="reqInfo" polarity="positive" topic="**number**-duration" mode="query">
and that's <pause /> **how many** hours <punc type="query" />
</decl> (d92a-1.2)

(83) <decl n="122" sp-act="state" polarity="positive" topic="age-**enum**" mode="frag">
she's **85** years old <punc type="level" />
</decl> (sw_0006_4108)

Example (81) illustrates how spellings may be used in providing details for names that may otherwise be difficult to write. As the example also demonstrates, for reasons of respecting and protecting the privacy of informants whose personal details appear in a corpus, such details generally tend to be anonymised. One further part of the patterns for spellings in Table 5.10 may require further explanation, though, which is the anonymised content of type alpha. This is possibly a rarer feature in corpora, apart from Trainline, and certainly does not appear in either Trains or Switchboard, but reflects the option where a speaker attempts to reduce the chances of spelt letters being misunderstood, due to perhaps bad sound quality on the phone or background noises, even further by providing a word that starts with this letter alongside the letter itself, e.g. *B as in Bravo*. The topics *number* and *enum* currently show some overlap in the pattern definitions, as single numbers appear in both. However, the former is mainly designed to capture references being made to numbers or amounts, as in Example (82), while the latter is mainly intended to cover the counterpart to spellings, numbers that occur in a row. These rather crude 'catch-all' definitions may still be fine-tuned in future to avoid such overlap and reduce redundancy in the annotations. Both categories of course often occur with other topic categories, such as references to times, durations, information pertaining to age (e.g. in Example (83)), etc., as they form an integral part of them. The next group of topics covers patterns related to providing or requesting time-related information, in other words, temporal deixis that is not exclusively expressed in terms of enumerations (such as AM or PM).

Table 5.11 Times & dates

Topic label	Pattern
time	at (number in time range) (not followed by currency); (not preceded by *each*) time; (not preceded by *good*) afternoon/morning/evening/night; hour/month(s) (not followed by *away/old*); o'clock; minute; when; half (past); midnight; dawn; dusk; a/p.m.; early/earlier/earliest; late/later/latest (not followed by *thing*); before; after; (1–2 digits): (1–2 digits); right now; year(s) (not followed by *old*)
day	word with initial capital letter ending in -day; (not preceded by *all*) day/tomorrow/yesterday
week	last/next/this week
month	(name of month)
date	(not preceded by *expiry*) date; first/second/third/*n*th of (followed by initial capital letter of month)
duration	(for) (about/around) (not preceded by *once/twice*) (number/a/all/many/several/a whole/an entire) second(s)/minute(s)/hour(s)/day(s)/year(s)/decade(s)/century/centuries/month(s) (not followed by *ago/away/before/old*); for a/some (brief/certain/considerable/little/long/short) time/while

The patterns are designed to match direct references to time or specific times or periods of the day (*time*), even more specific reference to particular days (*day*; both in 84), weeks (*week*), or months (*month*), and combinations of the former and literal references to a particular date (*date*; both 85).

(84) <imp n="18" sp-act="direct" polarity="positive" topic="**time**-to-**day**-number-enum" mode="poss2-decl">
<pause /> determine the maximum number of boxcars <pause /> of oranges <pause /> that you could <pause /> get <pause /> to Bath by **7 a.m**.
<pause /> **tomorrow morning** <punc type="stop" />
</imp> (d92a-1.2)

(85) <decl n="42" sp-act="reqInfo" polarity="positive" topic="**month-date**" mode="query">
and that's for the **8th of October** <punc type="query" />
</decl>

Some recent additions to this group, not discussed here for reasons of space, are *past_time* and *future*, which were introduced to be able to distinguish better between references to future and past events.

The next category encompasses items pertaining to local deixis, specifically to place-names or other locations (*location*), as well as directions to and from these locations, but excluding temporal ranges, which, of course, could also occur between or with the relevant prepositions, such as *at, from*, or *to* in English, as well as their equivalents in the other languages. Locations can frequently be identified relatively easily because, at least in the Indo-European languages discussed here, they are proper names and hence start with a capital letter, as well as often exhibiting fairly distinct 'compound suffix' patterns.

Table 5.12 Locations & directions

Topic label	Topic
location	here (not followed by *you are/go*); there (not followed by you *are/go/fore/is*); <anonym type="street" />; in (followed by capital letter)
location_Anglo	(capital letter, followed by lowercase letters) + (burgh/by/caster/chester/don/ford/ham/pool/port/ridge/shire/to(w)n); Cottage; Hall; Church; Bridge; Avenue; Road; House
from (locative)	from (followed by capital letter)
to (locative)	(not preceded by *according*) to (followed by capital letter)

As can be seen especially in Example (87) below, information about origins and destinations generally includes locations, unless of course deictic adverbs are involved.

(86) <decl n="26" sp-act="predict" polarity="positive" topic="time-**location**-enum" mode="query">
it'd get **there** at <pause /> at um <pause /> at <pause /> 2 a.m.
<punc type="stop" />
</decl> (d92a-3.1)

(87) <decl n="82" sp-act="stateIntent" polarity="positive" topic="**location_Anglo-to-from**-booking" mode="intent-decl">
i'm booking from Euston to Birmingham New Street <punc type="stop" />
</decl> (trainline01)

Example (87) also illustrates a particular feature recently introduced into DART, the *location_Anglo* topic pattern. This pattern originally used to be a part of the *location* pattern, but while working on data from the ICE Corpus, I realised that this was more specific to Anglo-American cultures, where it models many place names or parts of addresses with a fair degree of accuracy, although it may of course also identify some (family) names, particularly those ending in -{ton}, whereas this is not the case for corpus data that contains instances of local deixis pertaining to international settings. Nevertheless, it is still sufficiently generic to warrant its inclusion in this section, as it may apply to data from many countries that have been subject to Anglo-American cultural influence, possibly including those where outer circle varieties of English are spoken.

The next group of topics encompasses personal details, such as names, addresses, telephone numbers, as well as potentially more specific information, e.g. about credit or debit cards. Of course, further items of information, like date or place of birth, etc., could also easily be added to the list, but have not occurred in the corpus data used for this study yet.

Table 5.13 Personal details

Topic label	**Pattern**
name	initial/name/title
address	address/postcode/Avenue
telephone	telephone (not followed by *sale*)
creditcard	credit/debit card, expiry date, Master card, Visa
verify	check/consult/look it/this/that up/verify/i will find out/have a brief/quick look

While most of the sub-categories in the table above are probably rather straightforward, *telephone* may still need to be adjusted for other domains that, for instance, deal with more specific information regarding telephones, such as that of telecommunications, although, of course the analysis scheme would always make it possible to override the generic patterns with domain-specific ones. The last sub-category, *verify*, however, still poses a problem that will need to be resolved satisfactorily in the future, as the information captured by it is fairly reminiscent of the *hold* mode category and, apart from providing semantic information about the task of verification, also implies a potential interruption of the flow of the dialogue. This clearly demonstrates that it may occasionally be very difficult to distinguish between the levels of (pure) semantics and semantico-pragmatics, and further research will be necessary to establish where exactly the boundary between the two can be drawn.

The last group of generic topics to be listed here encompasses those that are related to the 'mechanics' of arranging business or personal meetings, or travel plans. These also encompass elements related to fixing time frames for meetings, such as the availability of the participants for meetings, or references to possible arrival or departure times, along with options for negotiating meetings, such as confirming prior arrangements or – if necessary – cancelling them altogether.

Table 5.14 Meetings and appointments

Topic label	Pattern
availability	available/availability; booked up, is/are provided
cancel	cancel; rain check
confirm	confirm
arrival	arrive/arrival/arriving; be there; come at
departure	departing/departure; leave/leaving

As there are different levels of register possible, depending on whether the meeting is of a more formal nature or not, some of the patterns above may also contain some more domain-specific content that might not necessarily be suitable for less formal occasions.

5.3.2 Domain-specific topics

As we have already seen in the discussion of locations above, the boundary between generic and domain-specific topics can be somewhat fluent and difficult to draw whenever there is some overlap between expressions used for everyday activities and more specialised ones. However, if there are more specific details or expressions associated with a similar activity in a specialised domain, these can relatively easily be captured and assigned new labels to draw an appropriate distinction if this should prove necessary. This can for instance be done for those locations that do not follow Anglo-American naming conventions, but instead may either exhibit their own distinctive patterns, or simply a list of relevant place names. Yet, as the primary aim of the research discussed here is not to provide an accurate semantic representation of the individual textual units, but only to make recourse to topic information in case the modes – in conjunction with the syntax – do not provide enough evidence to identify a speech act, no exhaustive analysis or deeper semantic representation is really required here. In other words, the semantic annotation provided through the methodology could, in most cases, almost be considered a bonus or by-product of the annotation, even though the resulting annotations are certainly more flexible and readable than existing semantic annotation schemes, such as USAS <http://ucrel.lancs.ac.uk/usas/>, which assign codes, rather than telling labels, and also tend to be predominantly based on single words, rather than patterns.

Some examples of domain-specific topics for the domains of hotel booking and train timetable information are therefore provided in Table 5.15 without any further comment.

Table 5.15 Domain-specific topics

Topic label	**Pattern**
booking	(book(ed up)/debited/reference)
fare	advance/cheap purchase; (nor preceded by *to*) return/saver/open/single/fare; reduce
room	room, double/single/standard/twin room/bed, suite
rate	discount/rate/charge/tax/percent
service	service

Despite the somewhat lesser importance attributed to topics in contrast to modes within the analysis methodology, DART allows the researcher to investigate new domains with regard to potential topics easily using its built-in analysis methods.

CHAPTER 6

The annotation process

Now that all the preliminaries necessary for characterising and being able to process the other levels of meaning in dialogue data have been presented, the basic strategies that may be employed in the analysis and (mostly) automated annotation resulting in the speech-act annotation can be illustrated. We shall begin by looking at general issues, such as those related to identifying the relevant units, as well as other features concerning the special nature of spoken unplanned dialogue that necessitate special handling in DART, but without going into too much technical detail.

6.1 Issues concerning the general processing of spoken dialogues

6.1.1 Pre-processing – manual and automated unit determination

In Chapters 4 and 5, I already discussed the other types of syntactic units and non-pragmatic content that may occur in dialogue units, but did not illustrate how these units can physically be identified and annotated. That this may not always be an easy thing to achieve has already been pointed out in 4.1 to some extent, where I demonstrated that the classical categorisation into matrix and subordinate clause is not necessarily applicable when talking about units of information. The fact that spoken language is also full of hesitation and disfluency phenomena, such as pauses, interruptions, false starts, restarts, and short, incomplete fragments, presents further difficulties that may at times almost be insurmountable for any approach that relies on automatic pre-processing only. This is especially the case if no prosodic information is given in the form of punctuation or pitch movement indicators, as was originally the case for our corpus data. To illustrate this problem, I shall first present a few more examples from the original versions of the Trainline and Trains data after the basic conversion to DART format. Potential unit boundaries are indicated by a double pipe symbol (||) in analogy to representing major tone unit boundaries in phonetic analysis.

(88) <pause length="7s" /> now || <pause /> it's either the 22:45 train in the evening <pause /> <backchannel content="mm" /> available || i'm just checking early morning || <pause length="18s" /> sorry || it's 10:35 in the morning <pause /> for returning <pause /> || those are the only 2 available (trainline11)

(89) the first train that you're ab… a… able to use is on the 10:45 from Birmingham New Street || arriving in Euston at 12 32 (trainline11)

(90) so || <pause /> can i take an engine from Elmira <pause /> to Corning <pause /> || um || <pause /> pick u(p)… <pause /> || oh || okay || <pause /> alright || how far is it from Corning to <pause /> Dansville (trains d92s-4.1)

In Example (88), there is a DM, bounded by two pauses. This is followed by a declarative that seems to have been intended as specifying two alternatives initially, but which ends in an elliptical relative clause, and contains a pause and a short overlap sequence before the latter. Next, there is another declarative, indicating a holding phase, which is clearly delimited from the rest of the turn by a very long pause of 18 seconds. Following this, there is another DM, *sorry*, plus another declarative, which seems to contain the phrase *for returning* as a kind of afterthought, as this is preceded by another short pause. Finally, yet another declarative occurs, ending in an elliptical modifier relative clause. The turn therefore consists of six units altogether, and an initial assumption could be that it might be possible to use the pause information as a potential aid towards identifying boundaries. However, that pauses alone cannot provide a sufficient basis for splitting becomes clear when one thinks about the fact that some unit endings, such as the elliptical relative clause *available* and the adjunct *for returning* are also preceded by pauses, and one would certainly not want to treat those as individual units. However, every time one encounters a pause followed by a pronoun and an auxiliary in Example (88)), one would undoubtedly want to posit the existence of a new unit, something that, in more general terms, may point towards the potential of using pauses followed by a noun phrase and a verb or auxiliary as a suitable indicator that would at least make it possible to spot the beginning of a new declarative structure.

In the second Example (89), the new unit in fact begins with a non-finite verb form (*arriving*). A potential indicator for the end of the first unit here may be the occurrence of the PP *from Birmingham New Street*, but this would in fact require either implementing a reasonably good *chunk parser* (cf. Hinrichs et al. 2000), or even a full parser to identify the different phrases. The question here is whether such a parser could indeed be robust enough to handle the first unit correctly, due to the two restarts and the ungrammatical subject complement, which here takes the form of a PP. Perhaps a more promising approach would therefore be to assume that any present participle form that is not preceded by an auxiliary or noun (phrase), but followed by a preposition or the word *that* may represent the beginning of a new unit.

Example (90) begins with an initiating DM, followed by a yes/no-question, a hesitation marker, an abandoned fragment, followed by an expression of surprise, another two DMs that look like they may be acknowledging ones, although they actually occur in a non-turn-initial position – and hence cannot be –, and finally, a wh-question. The first unit, as will be shown a little further below, is relatively easy to identify and separate from the rest of the turn, even without looking at the syntax of the following unit. Identifying the beginning of the third unit – the fragment –, however, would again probably only be possible with the aid of a parser that can recognise the end of the yes/no-question. The fragment itself may also pose some problems for analysis later. Not only does it start with a filler (*um*), which needs to be split off first, but it also looks like an imperative. If one does manage to split it off the turn, though, one can again split off the two following discourse markers with relative ease because they follow specific patterns that allow it to define their syntax as a *closed class* of items, similar to the closed class of function words in morpho-syntax.

As has probably become apparent in the preceding discussion, there are some indicators that might help to at least automate part of the splitting, but of course the rough 'rules' listed above would have to be thoroughly tested and refined in order to achieve any reliability and avoid any splitting errors that may be just as problematic for the analysis as not splitting. For the present study, this improvement has not been possible to achieve yet, and even if the built-in editor in DART implements some functionality to assist the splitting of long turns, at least some of this should be considered rudimentary at best. One part that does already work well, however, is the ability to split off the shorter units, DMs, yes- or no responses, terms of address, and formulaic fragments, which may be carried out in three different ways. During the data pre-processing phase, it is possible to either split off all initial short units for the whole dialogue, also adding appropriate `<punc … />` elements at the same time, especially as most DMs can generally be assumed to b of type `level`, or, if a longer unit needs to be split manually first, but after splitting the remainder of the turn has initial short units again, this operation can also be triggered for each line containing multiple units again. As there is no formal way of distinguishing between ambiguous short units, such as *so* used either as a DM or a logical connector, some of the split units may need to be joined again manually. To be able to prevent them from being split off again during the main analysis phase, where this operation happens again, such instances can be masked by adding an underscore (_) to the final word of the unit, which will prevent the automatic splitting from being triggered. All of these underscores are automatically removed again once the annotation has been completed and the resulting annotated files are created.

The remainder of the pre-processing needs to be carried out manually, which may be time-consuming, but has the advantage that the researcher will already get some preliminary ideas as to the nature of the data.

6.1.2 Fillers, pauses, backchannels, overlap, etc

When listening to spoken discourse, human 'consumers' tend to automatically filter out items that are redundant, ungrammatical, or simply disturb the flow of information, in a way creating a 'purified' representation of what was said. Because the computer actually deals with words as entities that are made up of characters and bounded primarily by spaces or punctuation, it has no easy way to e.g. distinguish a filler/hesitation marker like *erm* from a DM like *now*, unless it is provided with some kind of information that tells it that the first may be less important – if it indeed is – and the second may play a more important role in signalling the beginning of a new phase of the interaction.

In terms of what has commonly become known as the *Shannon/Weaver model of communication* (Shannon & Weaver 1963), but was actually already developed by Shannon in 1948 (Shannon 1948: 2), one would probably simply see fillers as 'noise' in the system, and therefore make every effort to remove them from as many different parts of the analysis as possible, a strategy that is also employed in the DART system in most types of syntactic and topic or mode analysis whenever applicable. A similar thing relates to pauses, overlap, background noises, etc. One exception to this rule are turn- or unit-initial filled pauses, which, as we saw before, are classed as DMs that signal hesitations.

In order to deal with occurrences of potential 'distractors', one first needs to identify and either classify them according to their (non-standard) word class, and/or describe their forms sufficiently exhaustively in order to be able to create a pattern for filtering out such unwanted items from any units to be processed. It might be tempting to do this filtering once and for all prior to processing the data, by producing a cleaned-up representation, but in doing so, one would run the risk of possibly removing some information that may later become relevant for particular aspects of the research not considered initially. Furthermore, if distribution of the data is planned, one may also be rendering parts of the data useless for other researchers' purposes because this would – at least to some extent – constitute a falsification, or, at the very least, simplification of the information passed on. It is therefore always preferable to create a suitably modified copy of whichever unit is to be processed in memory, and then to perform the analysis on this. In this way, one can simulate a similar kind of 'purification' mechanism to the one described above, but, this time, for the computer.

The question as to how to deal with the different bits of information to be ignored depends on what kind of status should be allocated to them. Here, the main question is whether one may want to treat some parts of the data as 'words', such as one might want to do with fillers, or to simply mark up everything 'non-standard' in the same way by using a common format, such as an empty XML element. The latter approach would have the benefit of making it possible to only ever specify one single pattern for removing unwanted items during the processing phase, regardless of what the content is, but for any other type of processing, would require to 'extract' the information from the element, minimally by specifying an unambiguous starting character sequence of the element, e.g. something like <*pau* for pauses or <*over* for overlap. In the DART system, a mixed approach is taken, with pauses, overlap or extra-linguistic information, as e.g. noises, enclosed in empty elements, while fillers, which also have a relatively easily identifiable phonetic form, are being retained as 'words' that can also be listed in the lexicon.

It is, however, extremely important to treat instances of overlap that represent backchannels formally as backchannels, too, rather than assigning them to their own turn. I already raised this as an issue in Section 3.1.2 above, in relation to the cleanup operations that needed to be carried out on the Trains data. If this is not done, it may lead to artificially induced ungrammatical structures, as can be seen in the following example from this corpus – again based on a slightly older format prior to pre-processing –, where the backchannel *mhm* had been marked up by the transcriber of the original data as a separate turn (35) and was automatically marked up as a DM that interrupts turn 34, creating two disjoint units (68 & 70) that originally do belong together and would be grammatical if they had been left joined together.

(91) <decl n="68" polarity="positive">the engine with the tanker</decl>
</turn>
<turn n="35" speaker="s">
<dm n="69" sp-act="acknowledge" polarity="positive">mhm[1]</dm>
</turn>
<turn n="36" speaker="u">
<frag n="70" polarity="positive" topic="number">will take <pause /> 2
<pause /> additional <pause /> boxcars</frag> (trains d92a-1.3)

In this case, separating the two parts artificially has split the subject of the original unit, which has the form of a post-modified NP, from the VP, which has 'tricked' the analysis routine into only identifying unit 68 as declarative (decl) structure,

1. The original backchannel is represented as *mm-hm*, which has been adapted to our analysis format

based on its initial NP, but 70, due to the absence of an initial NP, as a fragment. This issue would never have arisen if the backchannel had been appropriately included within turn 34. This example again demonstrates quite clearly how important it is to be able to identify – and use – the right units of analysis, particularly as even human annotators or transcribers can be responsible for introducing the basis for analysis errors if they are not completely aware of the nature of the data, and not just the computer that produces a 'blind' analysis based on the patterns specified inside a more or less suitable or accurate grammar.

6.1.3 Handling initial connectors, prepositions and adverbs

The potential role of conjunctions in the context of identifying or classifying syntactic units has already been pointed out in 4.1 above, but there is yet another (related) feature which makes them relevant to the analysis scheme. Along with prepositions, adverbs, and vocatives, all these word types have one common feature, which is that they may occur unit-initially, and precede the syntactic pattern that is characteristic of the individual unit types. Below are a few examples that illustrate this:

(92) <q-wh n="83" sp-act="reqInfo" polarity="positive" mode="open-query">
and how do you spell that <punc type="query" /></q-wh> (trainline03)

(93) <decl n="27" sp-act="stateCondition" polarity="positive" topic="journey" mode="contrast-condition-frag" >
but but if you're travelling on a weekday <punc type="level" /> </decl>
(trainline18)

(94) <decl n <decl n="243" sp-act="expressPossibility" polarity="positive" mode="poss1-poss-decl-disflu">
maybe i can <comment type="restart" /> i can get this yard in shape <punc type="stop" />
</decl> (sw_0087_2775)

(95) <frag n="6" sp-act="direct-referDirection" polarity="positive" topic="location_Anglo-from-to" mode="request-partial-decl">
from London to Crewe please <punc type="stop" />
</frag> (trainline09)

(96) <imp n="101" sp-act="suggest" polarity="positive" mode="decl">
then do it <punc type="stop" />
</imp> (d93–16.1)

As can be seen here, these features may occur in or with basically any of the major syntactic categories. Moreover, they may also occur in combination with some of the other 'distractors' discussed in 6.1.2 above. Thus, unless these leading elements constitute DMs and need to be split off, anyway, what is necessary in order to identify the grammatical structure of the unit, along with its potential function, is to skip over any irrelevant leading elements and determine the 'true' beginning of the syntactic structure.

6.1.4 Dealing with disfluent starts

Fillers, connectors, and the other features discussed above do not represent the only problems one may encounter in trying to determine the start of a syntactic unit computationally. There may also be various types of repetitions, where speakers seem to be unable to get started properly on what they want to say and therefore repeat parts of the unit. Amongst those, there are different patterns with varying types of complexity that can present problems for the computer, and which need to be identified in order to make the analysis *robust*. A few examples from the Trainline data will help to illustrate these disfluencies:

(97) is is that alright (trainline18)

(98) i i i tell the guy you can do any a… (trainline28)

(99) i can i can manage that (trainline24)

Example (97) represents the simplest pattern, where the initial word is just repeated once, while in (98), there is a double repetition of the same word. Number (99) is the most complex example of all presented here because the pattern that is repeated in fact consists of two words that recur. Theoretically, one could also conceive of this type of pattern occurring three times in a row, but this seems to be so rare that it does in fact not happen in the data. Disfluencies like these rarely show up in conjunction with the other ones discussed above, but if they do, they are more likely after initial fillers, etc. One can therefore usually assume that it is just necessary to spot the patterns and adjust the beginning of the unit accordingly by adding the number of positions that one needs to shift the starting point.

However, there is one exception to simply being able to deal with repeated 'word' patterns. This is that it may be necessary to ignore duplicate numbers because these often occur in fragments where a telephone or credit card number, etc., are given, and such a repetition would therefore in most circumstances not count as a disfluent one.

6.1.5 Parsing and chunking for syntactic purposes

Many of the more traditional approaches to Natural Language Processing (NLP) seem to assume that it is necessary to perform a *deep* analysis on different levels, namely the syntactic and semantic one, in order to be able to determine the meaning of a unit. When discussing finite-state methods, though, Jurafsky and Martin (2000: 385) observe that:

> Some language-processing tasks don't require complete parses. For these tasks, a **partial parse** or **shallow parse** of the input sentence may be sufficient. For example, **information extraction** algorithms generally do not extract *all* the possible information in text; they simply extract enough to fill out some template of required data.

Chunk parsing (cf. Carroll 2003: 234), which refers to the basic identification of major constituents, apparently mainly tends to be employed when the degree of accuracy does not play a major role because the assumption is that the data needs to be post-processed exhaustively, anyway, as may be the case in the creation of treebanks (cf. Leech & Eyes 1997: 34ff.). Shallow parsing or processing is for example used on the Verbmobil project, a multi-lingual appointment scheduling system. Here, a shallow mode, based on identifying key words, is used if the system is simply following the interaction of the two participants trying to schedule an appointment in the common language English, but if one of the participants fails to understand the interlocutor and asks the system to help by providing a translation into their mother tongue, the system switches to a deep analysis mode (Jekat et al. 1995: 2).

In the analysis system used here, it is also possible to use a shallow approach because, essentially, it is not necessary to conduct a full parse of each unit, since generally all that is required for identifying the type of a unit – after the initial pre-processing steps discussed above – is to look at the first few words to identify the basic patterns. Once this has been done, along with some lexico-grammatical analysis, one can then make use of the *mode* – or potentially even the *topic* – information in order to try and derive the speech act, as I shall demonstrate later. Because only information about the first few words in the unit is required, a full parse would probably even be a disadvantage in terms of run-time and memory, as, depending on the type of parser, there might be a fair amount of backtracking or the need to build up relatively complex data structures, such as charts, involved. In contrast, the DART system only needs to evaluate a few relatively simple conditions, illustrated in the next section.

6.2 Identifying and annotating the individual unit types automatically

6.2.1 Splitting off and annotating shorter units

The first real analysis step in the original SPAAC/SPAACy analysis system consisted of tagging all *yes* or *no*-like responses, but this previous approach was much less efficient than the present one because each type of unit could only be processed and removed once before the next processing stage began. Thus, theoretically, if one had encountered a *yes*, followed by a DM, followed by another *yes*, e.g. *yes well yes*, the first two would have been spotted, but not the third one. DART handles this more efficiently, as it checks for each type of unit repeatedly until none of them can be found any more, and the remainder of a longer unit is then assumed to be one of the longer types of units.

To be able to do so, DART uses a special set of files that, again, contain label – pattern mappings that detail the relevant patterns for yes/no-units, DMs, and formulaic fragments. These files are freely editable by the user, and can therefore be adjusted to be able to create new sub-categories, although one needs to be careful in doing this, so as not to mess up the original patterns, and also ensure that the ordering of patterns is correct, i.e. that the most specific and longest patterns are handled first in the splitting process. For instance, the patterns for yes-units contain a simple distinction between those yes-units that can automatically be assumed to be acceptances (*accept*), as they are followed by *please*, while the remainder are all deemed to be only acknowledging (*acknowledge*) in the automated part of the annotation. If one were to list the label – pattern without the *please* first, only the shorter pattern for acknowledgements would be identified and split off, leaving the *please* stranded and later to be identified as fragment that functions as a directive. For no-units, as already stated in 4.3.2 above, currently similar assumptions hold in that *no thank you/thanks* is automatically marked as a rejection (*reject*) and all other no units neutrally as negations (*negate*). As we saw earlier for the topic and mode patterns, we here obviously also need to restrict the patterns to some extent in order to exclude expressions like *no wonder/doubt, no problem*, etc. from accidentally being annotated as no-units.

The file for DMs specifies a far more diverse number of categories, as already indicated in 4.3.3. The ones that occur at the beginning of the line or unit are relatively easy to spot and mostly unambiguous as to their word class. DMs at the end of a unit are somewhat harder to spot, but also occur in relatively clearly specifiable contexts. However, along with question tags, these DMs do not need to be split off, but rather only detected and interpreted correctly via suitable mode patterns, since they almost always indicate requests for confirmation in a similar way to tag questions. The problem of having to disambiguate as to their 'word class' – e.g. *right* having the potential for being either a discourse marker or belonging to a number of different word classes – seldom occurs because these words that have the potential to be DMs can be marked in the lexicon and also syntactically disambiguated according to their context in that that they should not be preceded by an article, possessive pronoun or auxiliary.

Terms of address can, at least to some extent, be handled in the same way as the other short units. Currently, only a single pattern is stored in the relevant resource pattern file, but of course this could easily be extended if necessary, e.g. in order to deal with different levels of politeness or deference. The pattern currently used is not unproblematic, though, and a better way of handling terms of address may need to be found in future in order to be able to handle vocatives more reliably. The reason why this pattern could cause problems is that it does not only contain 'pure' forms of address or deference markers (cf. Fraser 2006: 190), such as *Sir* or *Madam*, which are completely unambiguous in this context, but also singular and plural noun phrases that can be used as 'attention-getters' such as *guys*, but may equally well constitute the beginning, i.e. subject, of a declarative unit. Further work on this is needed in order to determine potential unambiguous environments in the absence of prosodic information, which could of course again help us to disambiguate these cases with relative ease because attention-getters, as opposed to subject noun phrases, are always be followed by a pause, as well as either a fall or calling contour (c.f. Knowles 1987: 214ff).

Another type of short unit are fragments that often represent formulaic constructions, such as *greetings* and *goodbyes*, but may also contain specific deictic references to e.g. times or durations.

6.2.2 Tagging wh-questions

After having processed all the short units, one can now begin to 'eliminate' the more traditional unit types step by step by first testing for the most obvious surface markers. Here, wh-questions are the ones that are easiest to find automatically because of the occurrence of the respective question word. If a unit does not contain any question word, it can immediately be discarded as a potential wh-question candidate, even without having to perform any morphological or syntactic analysis, which is why this analysis step should be performed straight after those processing steps that split off the short unit types.

Perhaps the main theoretical/definitional problem in identifying and annotating wh-questions is whether it may make sense to try and design a set of purely syntactic criteria. Because the position of the wh-word is relatively variable and the question may also just consist of a single wh-word, this category is syntactically less clearly defined than the other 'long' unit categories, and also incorporates structures that are essentially fragments in that they may not contain any subject or verb, or start with a non-finite verb form, such as in *departing at what time*. In such cases, it may very well be tempting to say that anything that is not well-formed should be considered a fragment, but then – in the worst case – we are faced with the problem of having to do a full parse to identify whether a unit may really be complete in consisting of all subjects, objects or complements a given verb in the unit may require. As will already have become clear earlier, this would not be practical for this type of analysis, so it is necessary to find a better solution. The one adopted here is to some extent based on identifying the length of the unit, as well as the position of the wh-word inside the unit.

Essentially, the routine in DART that detects wh-questions works on the assumption that one of the first three words – if there are of course so many at all – in such a unit has to be a wh-word. The only problem to be avoided here is that there are some declarative focussing or clefting constructions, such as in *This is what …*, which may have the wh-word in one of these positions, but surely do not qualify as questions. Below are two more examples.

(100) <decl n="205" sp-act="suggest" polarity="positive" topic="location-transport_means-commodities-from-to" mode="poss2-suggest-decl">
what you could do <pause /> is you could take the boxcars from Elmira load the oranges at Corning <pause /> and drop off the oranges at Dansville on your way to Avon <punc type="stop" />
</decl> (d93–14.3)

(101) <decl n="135" sp-act="stateConstraint" polarity="positive" topic="from-to-location_Anglo" mode="constrain-decl">
what you're going to have to do <pause /> is get the train from Euston to Wilmslow <punc type="stop" />
</decl> (trainline12)

Cases like in (100) and (101) can easily be excluded because the word following the wh-word is generally a pronoun.

The routine for identifying wh-questions begins by testing the shorter unit strings first, initially checking to see whether there is only a single wh-word on the line. If a single wh-word is found, the unit is marked up as a wh-question with speech act *reqInfo*, but a mode attribute *exclaim* is also added to indicate its potential nature as an expression of surprise. Clear exclamatives, such as e.g. *what the hell, how interesting*, etc., which generally consist of an initial wh-word, followed by either a determiner or modifier or -ing form, are filtered out at this point and marked up as <exclam … > with speech act *exclaim*. Another fragmentary wh-category with incomplete syntax is commonly used to enquire about directions, i.e. to elicit information about *local deixis*. This type of question has the form *where to* or *where from*, and may optionally be followed by temporal adverbs like *now* or *then*.

Once the above types have been checked for, suggestions in the shape of wh-questions are handled. These essentially either contain a combination of *what/how + about/if* or start in *why*, followed by either *not* of *don't*. This type of wh-question constitutes the main part of Adolph's (2008) discussion and analysis of functional profiles, but only represents a fairly minor, yet important, part of the DART methodology for handling pragmatics-relevant linguistic structures. These wh-units receive a tag *q-wh*, a speech-act attribute *suggest*, and a mode attribute *closed* because the set of options in a suggestion is essentially limited, as in yes/no-questions. Most other wh-questions receive a default *mode* attribute *open*, reflecting the fact that the answering party is relatively unconstrained in the choice of possible answers, although of course the domain may impose some limitations on the range of potential responses. Examples for this type of suggestion would be.

(102) <q-wh n="128" sp-act="suggest-abandon" status="abandon" polarity="positive" topic="from-enum" mode="condition-closed">
how about <pause /> if i take engine 2 <pause /> from Elmira <backchannel content="aha" /> with <punc type="incomplete" />
</q-wh> (d92a-5.2)

(103) <q-wh n="175" sp-act="suggest" polarity="positive" mode="closed-query">
what about the next <pause /> best option for 19:00 <punc type="query" />
</q-wh> (trainline12)

(104) <q-wh n="104" sp-act="suggest" polarity="positive" mode="closed-query">
<overlap pos="start" /> what <overlap pos="end" /> about emissions <punc type="query" />
</q-wh> (sw_0027_4096)

Example (102) illustrates nicely how even incomplete units can frequently be marked up with a speech act, as most of the functionally relevant indicators of 'intentionality' occur towards the beginning of a unit. Both this and the next example also show how suggestions in both task-oriented domains tend to relate to options concerning the task at hand, while, in Switchboard, they are mainly used to raise issues or switch (sub-)topics.

Unlike in the above (exceptional) examples, in most cases the speech act expressed by a wh-question tends to be a request for information (*reqInfo*), so that the *sp-act* attribute can be set accordingly for many of the remaining wh-units. However, as already discussed in 4.3.5 above, a fair number of wh-questions may also constitute a request for a directive (*reqDirect*), indicated by expressions like *what do you want/wish…*, e.g. *what time do you like to depart on <overlap pos="start" /> Tuesday* (trainline07). Nevertheless, as I shall demonstrate in 6.5 below, it is not truly necessary to capture this during the syntactic analysis phase, but it is possible to correct the default later, based on finding appropriate mode attributes.

Capturing the 'regular' wh-questions is relatively straightforward because they follow some fairly well-defined patterns. All that remains to be done then is to identify a few syntactically 'ill-formed' structures, which are hence tagged as fragments, but nevertheless endowed with a speech act of *reqInfo* and an *open* mode. These wh-fragments tend to either start with a non-finite verb, either followed by a wh-word immediately, as in e.g. *departing when*, or by a preposition and then the wh-word, e.g. *departing at what time*. Alternatively, they may also begin with either a preposition or (numerical) modifier, followed by *what*, as in e.g. *you(r) what* or *fifteen what*.

6.2.3 Tagging yes/no-questions

Once the wh-questions have been handled, one can move on to tagging yes/no questions. The main principle in dealing with these questions is to look for syntactic inversion involving an auxiliary + either a noun, pronoun, or existential there, including, of course, negated forms. If one of these has been identified, then it is simply a question determining which subcategory it belongs to.

One of the essential distinctions in yes/no-questions is according to the person of the subject. If the subject pronoun is in the 2nd person and occurs in the second position, that is immediately after the auxiliary, then it is possible to distinguish between two different options, both being indirect speech acts. In the first case, exemplified by the pattern auxiliary (excluding *will*) + *you*,[2] but not followed by the verb *like*, the question fulfils the function of a directive.

(105) <q-yn n="90" sp-act="direct" polarity="positive" topic="telephone-number" mode="request-query">could you give me your contact telephone number please <punc type="query" /></q-yn> (trainline21)

(106) <q-yn n="65" sp-act="direct" polarity="positive" topic="journey-day-verify" mode="frag">
would you mind try… er checking for me then erm to travel on the Thursday <punc type="query" /></q-yn> (trainline22)

This pattern does not cover all potential directives of this type yet, but one can generally catch the remainder because they tend to contain the word *please* (not followed by a pronoun), which is identified as a *request* mode and can later be used to assign the *direct* speech act to the unit. The other potential pattern has *will* in the first position, followed by a 2nd person pronoun subject, then *be*, and the following word is not a determiner, pronoun, noun or verb, which would catch such imperatives as *will you be quiet*, etc.

2. One hereby basically also includes archaic seeming forms such as *shall you*, but if the analysis method were to be extended to analyse older literature, this would probably make more sense.

The next group of 2nd person subject patterns expresses a request for a directive (*reqDirect*). It encompasses patterns such as *would you like* or *do you want/wish*.[3]

(107) <q-yn n="74" sp-act="reqDirect" polarity="positive" topic="seat" mode="preference2-alternative-closed-query">would you like smoking or non-smoking <punc type="query" /></q-yn> (trainline01)

(108) <q-yn n="40" sp-act="reqDirect" polarity="positive" topic="journey-booking-departure-fare" mode="preference2-alternative-closed-query"> do you want to book the return or leave this return journey open <punc type="query" /></q-yn> (trainline07)

Again, this does not capture all the options via the syntax routine, but is supplemented by the mode *preference*, which will later make it possible to set the correct speech act tag. All requests for directives also receive a mode attribute *closed* to indicate the restriction of choices for the response. Requests for directives, however, tend to mainly occur between 'unequal' partners, such as in service exchanges, so that often questions regarding volition in unconstrained dialogues, such as Switchboard, where both interlocutors have equal status, can be interpreted as proper requests for information.

Offers can be spotted on the lexico-syntactic level by identifying patterns that contain *can, could, may* or *might* as modals in the auxiliary position, followed by a 1st person non-possessive pronoun and words such as *help, offer*, or *give* that express a mode *benefit*, generally to the hearer, such as in the stereotypical opening in Trains discussed earlier. As before, a *closed* mode attribute is assigned to all units thus identified.

Any other structure that starts with one of the same modals and a 1st person non-possessive pronoun is automatically identified as a request for an option (*reqOpt*), which is basically a neutral label for enquiries that may either be about potential options available or actually represent requests for permission, depending on the status of the requester. If, as in most of the examples from the Trainline data, the speaker is a customer and therefore qua role possesses some form of authority over the hearer, then the request needs to be interpreted as one enquiring for options, in other words, the choices available, but if the speaker is less powerful than the hearer, then the force of the unit has to be seen as a request for permission, as e.g. *may i go toilet <punc type="query" />*, which was uttered by a student in the Singaporean classroom data referred to in 4.3.4 above. The

3. Theoretically one could test for any verb expressing volition, although again, some forms would probably be much less likely to appear in non-literary texts.

same classroom data, however, introduces yet another option for the meaning of this structure, which is essentially that of an indirect command, as the following example shows.

(109) <q-yn n="222" sp-act="reqOpt" polarity="positive" mode="closed">
can we do it afterwards <punc type="stop" /></q-yn>
<imp n="223" sp-act="direct" polarity="positive">
just <punc type="comma" /> just finish this first <punc type="stop" />
(classroom)

Here, the teacher seems to be enquiring about an option in the first instance, but asserts her position of authority in the ensuing directive.

Suggestions that can be spotted via the syntax routines either consist of a pattern *shouldn't* + 1st or 2nd person pronoun, *shall/should* + 1st person pronoun, or can/could/may/might + 1st person pronoun + *suggest/propose*, as in for instance:

(110) <q-yn n="56" sp-act="suggest" polarity="positive" mode="closed-decl">
shall we do that <punc type="query" /> </q-yn> (trains d92a-3.1)

If, on the other hand, the line contains a combination of *can* or *could*, followed by a pronoun that is not in the 2nd person, then the unit will be tagged as 'modal request' (*reqModal*), e.g. *can i be a pain and ask you about Thursday night* or *can we just confirm the times*, which is similar in nature to a weak form of suggestion. This pattern, however, still requires further clarification and clearer specification because it may also unfortunately often yield the wrong results. If none of the above conditions hold, but there is still an inversion present, the default category request for information (*reqInfo*) can be assigned in the final stage.

6.2.4 Tagging fragments, imperatives and declaratives

In the final stage of the lexico-grammatical annotation part declaratives, imperatives and fragments are disambiguated. Rather than testing for declarative structures first, the analysis again begins by identifying fixed formulaic and short fragments first. These often represent various types of referential deixis, (dis)confirming responses, such as *i do/don't*, or potentially longer fragments that start with non-finite verb structures, often referring to ongoing processes, such as the arrivals/departures we already encountered for the Trainline data.

This part is followed by an analysis of imperatives, which is relatively straightforward, as the clear criterion is finding a verb base form in first position. The only potential problem is that, in longer descriptive sequences of actions where the initial pronoun may be implicit, the resulting fragments may actually be misinterpreted as imperatives, too, as already pointed out in 4.3.9 above. With 'genuine' imperatives, it is possible to distinguish between two different types, the first one representing suggestions (*suggest*), and the second one, as one would probably expect by default, directives (*direct*). Suggestions that can be identified lexico-grammatically are clearly marked by the occurrence of an initial *let* + 1st person pronoun, which may also occur in the shape of a contraction.

(111) <imp n="118" sp-act="suggest" polarity="positive" mode="decl" >
let me get it the right way round <punc type="stop" /> </imp> (trainline26)

(112) <imp n="2" sp-act="suggest" polarity="positive" topic="place_state" mode="frag">
let's talk about the uh wonderful abuses in the State of Pennsylvania of personal property taxes <punc type="level" /></imp> (sw_0035_4372)

As illustrated in 4.3.8, a special case of imperatives are those that allow the speaker to interrupt the current conversation in order to perform some other – usually non-verbal – action, or possibly also to prevent the interlocutor from taking any hasty, unreflected action or decision. These, according to the DART scheme, receive a speech act *hold*, and cases where the syntax routine would assign a *suggest* speech act in the first instance can later be redefined in this way if one or more appropriate mode attributes are spotted in the unit.

(113) <imp n="59" sp-act="hold" polarity="positive" mode="hold-opinion-decl" >
let me think <punc type="stop" /> </imp> (trainline02)

(114) <imp n="210" sp-act="hold" polarity="positive" topic="time-duration" mode="hold-decl">
wait a minute <punc type="stop" /></imp> (d93–12.4)

Another special type of imperative that is usually not discussed in grammars is the formulaic form of goodbye (*bye*) starting with have, e.g. *have a good day*, where first two words always remain fixed, while the remaining positively connotated adjective and temporal noun express some limited variability.

All remaining imperatives can be marked up as directives. One just needs to ensure that two conditions hold, the first being that – for positive units – the first element is a verb and the second not a 2nd person pronoun, as in e.g. *see you (later)*, and that negative units either start with a contracted *don't* or an uncontracted *do not*.

Having handled the explicit imperatives, one can turn to indirect directives, which take the form of declaratives. Here, one encounters patterns like *I would/'d like you to* or *I want you to*, as well as the negative counterpart to the latter, *I don't want you to*. Note, though, that we cannot include the negative form of the first pattern, *I wouldn't like you to*, amongst these because it does not carry the same directive force, although it could of course carry that of a warning. A warning may have a somewhat similar influence on the behaviour of the interlocutor to giving a negative directive, i.e. forbidding something, but of course it does not need to do so. Apart from this, one cannot even automatically interpret the sequence *I wouldn't like you to* as a warning under all circumstances, anyway, as more often than not, it may simply represent a well-meant wish on the part of the speaker.

One may also be tempted to include structures that Collins (2006: 184) cites as potentially "bei[ing] endowed with indirect directive force", such as "You will/must be here by five" (ibid.), but of course their directive force does not purely rely on this particular syntactic form and the use of the modals, but really on whether the latter carry particular stress, as otherwise they may simply express a prediction. The fact that they are prosodically marked units and can thus deviate from the default function that the syntax would signal is demonstrated by the fact that it is not possible to reduce the modal *will*, along with the other deontic modal *should*, which could also occur in this context, to their contracted, unstressed forms *'ll* and *'d*, both forms that would be normal representation of these words in spoken language. Incidentally, in the light of this functional difference, an insistence on an 'orthographically more correct' uncontracted form in written language thus seems positively absurd because it prevents us from disambiguating the two functions in a simple manner, which would otherwise only be possible by special means, such as capitalising the modals, an option that is rarely used in written documents.

The next step in the analysis is to identify initial declarative patterns. This mainly takes the form of specifying patterns that represent initial noun phrases followed by verbs or auxiliaries, initial existential *there*, or the clefted relative clauses already discussed in the context of wh-questions, where annotating them by mistake had to be avoided. These declarative structures encompass a variety of different speech acts, such as retractions (*retract*), promises (*promise*), other explicit performatives marked through *hereby* (*perform*), suggestions (*suggest*), constraints (*stateConstraint*), expressions of approval (*approve*), expressions of liking/dislike (*expressLiking, expressDislike*), expressions of (im)probability/possibility (*expressProbability, expressImProbability, expressPossibility, expressImPossibility*), etc.

All remaining declaratives simply receive a *decl* tag and their potential pragmatic functions are later resolved as best possible through the use of mode attributes. By testing only for initial declarative-indicating structures, there is of course a small element of error because, without a full parse, one may not really be able to identify incomplete, and hence fragmentary, units. To some extent this problem is resolved if the data contains final ellipsis marking in the form of three dots that are attached to the final word(-partial), but one simply has to accept the margin of error for the rest. However, as pointed out before, even a full parser may have difficulties in identifying all the potential complements and adjuncts that are necessary for making a unit syntactically well-formed, and a potentially greater problem than determining this well-formedness may be that the units have not been split properly and one may therefore be 'lumping' units together.

Nonetheless, there is yet another problem, this time one on the functional level. In the absence of prosodic information, there will always be the problem of distinguishing between true declaratives and declarative questions, which tend to have the same functional properties as questions. Incidentally, the same goes for fragments as well and – if anything – tends to occur with them much frequently than with declaratives. Although it may sometimes be possible to identify these interrogatives if one takes a look at the larger context, the few indicators for declarative and fragmentary questions that one may be able to derive syntactically do not seem to be reliable enough to warrant making positive assumptions. The strongest one of these at the moment seems to be an initial co-ordinating conjunction, either *and* or *so*, as in the following examples.

(115) <decl n="75" sp-act="reqInfo" polarity="positive" topic="location_Anglo" mode="predict-query">
and that's going to get into Birmingham at <punc type="query" /></decl>
(trainline11)

(116) <decl n="175" sp-act="reqInfo" polarity="positive" mode="query">
so the <overlap pos="start" /> total <overlap pos="end" /> <pause /> is
<punc type="query" /></decl> (d92a-1.2)

This indicator, though, is much too weak to really be reliable, as many non-interrogative units may also start in the same way, for instance:

(117) <decl n="94" sp-act="state" polarity="positive" topic="refund-fare-cancel-enum" mode="condition-constrain-decl">
and it's a full refund less 5 pounds if you have to cancel <backchannel content="right" /> <punc type="stop" /></decl> (trainline01)

(118) <decl n="26" sp-act="stateConstraint" polarity="positive" mode="constrain-decl">
so we need to make some <punc type="stop" />
</decl> (d92a-5.2)

Sadly, it therefore needs to be concluded that, at least for the moment, it is not possible to automatically identify this type of interrogative unless some type of prosodic information is provided along with the orthographic transcription, as can be done by adding appropriate <punc … /> elements.

Having processed all declaratives, one can treat the remaining units as fragments, and simply needs to try and determine their speech acts. If the line contains the words *thanks, thank you* or *cheers*, then one can automatically assign a speech act attribute *thank* to it. To assume this is indeed permissible at this stage because neither indirect speech in the form of quotes, such as *he/she never said "thank you"*, or e.g. a directive *say thank you* given to a child should still occur in our unit because all the syntactic structures that may contain them – declaratives and imperatives – have already been handled previously. Thanking expressions already constitute valid and complete units in their own right, so it is also not necessary to indicate that there is only partial information present.

If the unit either starts with a preposition or noun, then it is generally deictic, as in *<overlap pos="start" /> October, Saturday* or *London*. In this case, one can automatically assign a *refer*, or possibly even *referX*, speech act attribute along with the *frag* tag. A mode attribute *partial* can also be included because it is known that some of the information is missing. If the initial word is a preposition, then there is usually directional or temporal deixis, whereas one or more initial nouns may indicate locations or proper names. For all remaining cases, one can just return the tag information *frag*, but without any additional attributes because it is difficult to claim anything definite about the function of these units. There is one exception to this, however, which is if the unit is a fragmentary directive, as indicated by a *please* near the end of the unit. In this case, though, the function of the unit can be established through mode spotting, which will later help to supply a missing speech act tag during the inferencing phase.

6.3 Levels above the c-unit

While it is possible for most initiating or proactive units to determine their default speech act more or less directly through the kind of lexico-grammatical analysis discussed in the preceding sections, this is not really feasible for some of the responding or reacting ones. Before proceeding to a discussion of the final stage in determining or correcting the default speech acts in DART, I will first demonstrate how these 're-active' units may be classified. Essentially, one can here distinguish between two different types of responding categories, as will be shown in the next sections.

6.3.1 Answers and other responses

As I have already illustrated in the sections on yes/no-responses and discourse markers, some reacting units are relatively easy to recognise by their very shape, although they may not always be easy to categorise exactly in terms of speech acts. Their main function is to show some more or less strong form of consent or dissent, but from a semantic point of view, their information content is severely restricted. Informing answers and other forms of responses, on the other hand, usually embody some semantic content that provides information, as in e.g. a response to a request for information, or a directive in response to a request for directive.

In order to identify such responses, DART first identifies all units that have previously been tagged as a form of request in the earlier processing stages, as well as which type of response is to be expected, e.g. *direct, answer, stateOpt*, or *confirm*. It then goes on to check the next available turn for valid responses, those that cannot constitute a potential clarification question or hesitation marker. If the speech act attribute of the first appropriate unit found is not empty and is not already identical to the expected response type, for instance because an imperative has already been marked up as a directive, then the speech act attribute is changed to the value of the expected response and any existing attribute appended to it, as in Example (119).

(119) <q-yn n="68" sp-act="reqDirect" polarity="positive" topic="journey-booking" mode="preference2-closed-query">do you want me to book this ticket <punc type="query" /></q-yn>
</turn>
<turn n="46" speaker="caller_03">
<yes n="69" sp-act="direct-accept">yeah <punc type="stop" /></yes>
(trainline03)

If there is no speech act attribute present for the responding unit yet, then one can simply assign the expected response as a speech act. It is important to note, though, that, in the absence of any further speech act information, what is marked here is not really a 'unit-local' speech act function, but rather a function that applies to the larger context of the interaction. In a sense, this is a higher-level function than the local one because it does not reflect the semantico-pragmatic function of the unit in the same way a 'local' speech act attribute would.

In annotating levels above the local unit, it is indeed possible to do more than just mark up the immediate responses. Often, a response is in fact followed by some form of elaboration, too, which may provide further information concerning the reasons for or about the contents of the response. Example (120) illustratesthis.

(120) <turn n="3" speaker="Sandra">
<dm n="7" sp-act="init">now <punc type="level" /></dm>
<q-yn n="8" sp-act="reqInfo" polarity="positive" topic="creditcard" mode="alternative-closed-query">
do you hold a current debit or credit card <punc type="query" />
</q-yn>
</turn>
<turn n="4" speaker="caller_07">

<yes n="9" sp-act="answer-acknowledge">yeah <punc type="level" /></
yes>
<frag n="10" sp-act="elab-emphatic" polarity="positive" mode="decl">
i do <punc type="stop" /></frag> (trainline07)

As can be seen in Example (120), the same principle that applied to the annotation of responses also applies when assigning the elaboration speech act attribute (*elab*). If there is already some speech-act information present, this is preserved and appended to the *elab* attribute in order to try and mark the specific 'local' nature of the elaboration.

The term *elaboration*, incidentally, is loosely based in the concept of rhetorical relations as expressed in *Rhetorical Structure Theory* (RST; cf. Mann & Thompson 1987 or Matthiesen & Thompson 1988), where an elaboration provides further explanatory evidence for a *nucleus* in form of one or more *satellites*, where both nucleus and satellite(s) are clauses, similar to the units used in the present approach.

6.3.2 Echoes

Echoes are different from the responses discussed immediately above. Responses, thinking back to the DAMSL categorisation scheme, are essentially characterised as having a backward-looking function. Echoes, on the other hand, which are utterances that repeat all or at least part of the preceding unit, comprise elements of both backward- and forward-looking functionality. They are backward-looking in the sense that they 'react' to the preceding information by repeating it, thus partly acknowledging that the information has been received by the hearer, but also forward-looking in that the information is generally repeated in order to confirm that the hearer has also received it properly, and the dialogue can continue as planned.

The mechanism employed to identify these echoes works in a relatively similar way to the one that recognises responses, only that it does not apply to potentially any unit within the turn as was the case for the requests that triggered response spotting, but, by definition, only to the very last unit within a turn. Before this can happen, though, DART not only needs to check and see whether it is currently handling the last unit in the turn, but also whether the current unit does not contain some speech acts that do not 'allow' echoes in the strictest sense, namely *hold, init, pardon*, and *thank*. The reason why initiating discourse markers are excluded from echo spotting is that their formal equivalents, e.g. *well, now*, or *so*, are frequently grammatically polysemous and, especially if they occur at the end of the unit, their meaning is often very distinct from that of the formally identical discourse marker that may follow them.

6.4 Identifying topics and modes

When referring to dialogue content, the dialogue systems literature often refers to topic or *keyword spotting* in applying techniques for identifying such content. In e.g. Verbmobil, a German appointment scheduling system, even units below the word are used (Niemann et al. 1997), but interestingly none of the approaches documented explicitly seem to be using units above the word in order to detect what is usually referred to as 'topics' there, and which is potentially quite different from what we have just discussed earlier. The DART approach, however, is firmly anchored in the belief that, in most cases, keywords are simply not enough to identify such type of content, and therefore relies on the concept of 'key phrases' instead.

Looking at the comments in the tables containing various modes and topics in Section 5.3 on how certain words or constructions should not precede or follow the patterns to be identified, the advantage of a key phrase approach should become more or less obvious. Individual keywords not only need to be available in sufficient number in order to characterise a specific domain properly, but may simply be misleading if pure frequency lists, such as those for example employed in internet search engines, are used, rather than relying on some loose measure of collocation or colligation. Multi-word units or phrases for particular domains stored in *regular expressions* corresponding to the patterns shown in highly simplified examples earlier here provide the advantage of being able to identify particular relevant semantic (or other) content much more precisely, even if constraining these patterns may not always be easy, either. For example, when dealing with service dialogues in the telecommunications domain where one of the services is to provide wake-up calls, specifying a regular expression like *(alarm|early morning|reminder|wake up) call* that covers the variety of options available for referring to this type of service is certainly less ambiguous than trying to identify the particular topic of the utterance by looking at frequency/word lists that may only contain one or more of the individual words *alarm, early, morning, reminder, wake, up* and *call*.

In DART, both topics and modes for the individual languages and domains are stored in 'key phrase thesauri' that map the patterns to their mode or topic labels during the analysis. Information as to which key phrase files are to be loaded for which domain of analysis is stored in one of the DART configuration files. This information can be retrieved from there once the meta-information for a particular dialogue file has been read, and the name of the corpus, which serves as the key to the domain configuration, has been identified.

For modes, one file containing the lexico-grammatical patterns and one containing the 'punctuation' patterns are always loaded. For topics, minimally the file containing the generic topics is read in, with potentially any number of additional domain-specific ones being added in order to supply additional information. The same principle demonstrated for the lexicon files earlier also applies to the topic files. Whichever file is read in later may overwrite the patterns associated with a particular label or simply add a new label–regular expression combination to the analysis options.

In the course of the mode or topic spotting, the text of each unit – cleaned up of all empty elements and fillers – is matched against all available topics and modes, keeping a record of how often each relevant pattern is found. This record is then sorted to bring the most frequent topic or mode to the 'top', presuming that this is most indicative of what goes on in the unit on the semantico-pragmatic or pragmatic level. The exact frequency information is discarded when the annotation is written to a file, though, and also no distinction is made between the topics that occur with the same frequency. It is only the ordering of the labels that provides information about the relative importance of a particular pattern to the user of the annotated data. However, if it were necessary for further in-depth semantic analysis of the dialogue data, it would be simple enough to devise a way to include the frequency information alongside the labels or incorporate it into a kind of weighting scheme.

6.5 Inferencing and determining or correcting speech acts

Having determined, or at least restricted, the default functionality for all syntactic units as much as possible, a set of inferencing rules that help to either correct the original assumptions about previously assigned speech act attributes, or to assign one where it has not been possible to do so due to a lack of lexico-grammatical criteria to be used as evidence, are employed. It is at this stage that an attempt is made inside DART to bring together syntactic and mode – or, to a more limited extent, topic – information in order to determine the final speech act (combination) that is to be recorded for each unit and written to an annotated dialogue file. As shown earlier in the discussion of the lexical unit categories and the rules needed in order to identify them and their default functionality automatically, one often needs to use relatively fine-grained distinctions that are not only based on the PoS categories of the words, but possibly also the semantico-pragmatic functions inherent in particular words. Furthermore, because one may occasionally encounter intervening words, such as e.g. adverbs, in some structures, this lexico-grammatical problem could potentially lead to a proliferation of rules that makes it difficult to set up and maintain a suitable syntax-oriented and pragmatic grammar. If the aim was purely to identify the syntax, excluding the functionality, there would be no such problems because ‘all one would have to do’ is to create an accurate mapping from words to morpho-syntactic tags and specify a regular expression grammar that can handle all the different potential patterns, including optional adverbs, complements or adjuncts. This is what grammars written for parsers designed for analysing written, well-formed, language are generally good for, but unfortunately they are usually not robust enough to analyse the kind of ‘messy’ data that needs to be handled by a pragmatic analysis system.

In such a system, it is more sensible to follow a divide-and-conquer approach, and try to first capture the general functions that are easily identifiable via the syntax, and then combine or augment the information thus gained with that provided by lexical patterns that may express various types of interactional or intentional semantico-pragmatic meaning in the form of modes. In this way, it becomes possible to avoid the complexity of a complete syntactic parse, and of the generally ensuing deep semantic analysis (cf. Jurafsky & Martin: 2000: 546ff.), which would also necessitate a much higher degree of complexity for the lexica. Modes and topics also provide options for specifying ‘collocations’ – or possibly *constructions* in terms of cognitive linguistics (c.f. Goldberg 2007) – that might even be discontinuous and include a number of intervening words or phrases, including optional elements, as in *i’ll (just), i’m (not) going to, i(‘d/ would) like to*, which all signal intent as illustrated in Section 5.2.2 above.

Essentially, the inferencing mechanism consists of a set of ‘templates’ for redefining speech-acts, based on the type of unit and one or more mode attributes that make it possible to reinterpret the speech act in a certain way. The flexibility of the approach can perhaps best be exemplified through the handling of reported speech. In the original Trainline material, such reporting predominantly occurred in an indirect form, as in e.g. *i was quoted 19 pounds* (trainline02), so that the original rule devised in order to capture this included patterns containing references to past tense or passive forms of verbs of saying like *told*, *informed*, and *quoted* in combination with 1st person subjects. This was later revised to include past tense forms for verbs of saying related to 1st, 2nd and 3rd person subjects like *i/you/ they said*, already increasing the options for identifying both indirect and direct reported speech. Integrating these patterns into the syntactic analysis routines for both declaratives and fragments would have caused a proliferation of different patterns there, reflecting a mix of PoS tags and lexical information, which would clearly have been more difficult to write, and also more error-prone. Instead, in the current approach, the syntax identification routines can remain more general, but spotting the mode `report` can be used by the inferencing process to disambiguate between simple statements and those that contain some form of reporting. With the introduction of the inline element <quote> … </quote> to mark all direct quotations and reported thoughts, the identification of all direct reported speech then also became possible, even if no clear other indicator of reporting in the form of a reporting verb may have been present, as in e.g. *and the son was you know <quote>why</quote>* (sw_0088_3073). Such forms are especially prevalent in the Switchboard data, and would otherwise have been nearly impossible to identify.

Another example of the use of these inferencing templates is the identification of directives via the mode request, which simply identifies the 'directive marker' *please* somewhere within the unit, but obviously not when it is used as a verb. Together with the appropriate instruction inside the inferencing routine, this makes it possible to easily identify directives that have the shape of a declarative or fragment.

(121) <decl n="6" sp-act="direct-referDirection" polarity="positive" topic="location_Anglo-from-to" mode="request-decl" >em it's from Birmingham to em London Euston please <punc type="stop" /></decl> (trainline01)

(122) <frag n="61" sp-act="direct-refer" polarity="positive" topic="name" mode="request-decl" >
your surname please <punc type="stop" /></frag> (trainline06)

However, in order to avoid categorising the special case of a directive, the *hold*, as a straightforward *direct*, the inferencing process can be restricted not to apply when there is a mode attribute hold present.

(123) <decl n="40" sp-act="hold" polarity="positive" mode="request-hold-condition-poss2-decl" >
if you can <backchannel content="yes" /> just bear with me please
<punc type="stop" />
</decl> (trainline01)

As has hopefully become apparent in the preceding discussion, the inferencing strategy based on default syntactic and mode information is a very powerful tool that greatly simplifies the identification or correction of speech acts. However, one obviously needs to be very careful when generating mode definitions, and always be aware of their potential implications, as otherwise they harbour the distinct danger of leading to over-generalisations.

Having discussed all the mechanisms needed in order to create a pragmatic annotation, along with the syntactic, polarity, semantic and semantico-pragmatic features, it is now possible to turn to the actual core of the pragmatic annotation, the definition and distribution of the individual speech acts.

CHAPTER 7

Speech acts

Types, functions, and distributions across the corpora

So far, the implicit assumption in the discussion of our methodological approach has been that the names of the particular speech acts that occurred in connection with the default functions for the individual syntactic units were relatively obvious and largely self-explanatory. Now, however, it is time to investigate the different categories of speech acts in more detail, and not only identify their general functions on the level of the unit – the *micro-level* – but also to discuss what their influence on the larger dialogue context may be. In order to do so, I will make an attempt to group all speech acts that occur in the DART scheme into functional super-categories, and to discuss them with regard to their local and global effects or force, as well as describe their distributions across the three corpora, whose names will be abbreviated to *Trl* for *Trainline, Trs* for *Trains*, and *SWBD* for Switchboard in the tables listing the respective descriptive statistics discussed in the following chapters. To make the distributional characteristics more explicit, the tables in this chapter always list the normed frequencies and rank, along with the document frequency, i.e. the dispersion, within a particular corpus. Due to the frequency norming, there may be an odd effect, though, in that sometimes the number indicating the normed frequency might actually be lower than the dispersion reported, especially for rare speech acts.

The distinction between 'local' and 'contextual' effects is something that did not exist in the same way in the original SPAAC scheme, and indeed also not in the original approaches to speech acts devised by scholars like Austin and Searle, where essentially all speech act assignments were more or less implicitly presumed to pertain to the level of the individual unit itself. Thus, generally each unit in the SPAAC approach was only assigned a single speech act label, with a few exceptions where a dual local function, such as *informIntent-hold*, was deemed to be expressed within the same unit. In other linguistic approaches like CA or DA, though, interactional functions have, quite rightly, always played a larger role, and some scholars working in DA, such as Francis and Hunston (1992), have even already employed remarkably detailed taxonomies of (speech) acts that by far exceed Austin's or Searle's.

To overcome this initial weakness of the SPAAC approach, the DART scheme seeks to make the information on the two levels more explicit, with the result that many units are assigned multiple speech act labels that reflect both their local functions and roles in the interaction sequence more clearly. How exactly this is achieved will hopefully become clearer through the discussions of speech acts and their potential combinations that are provided in the individual sections below. As examples of many of the speech acts were already provided in earlier chapters, I shall only provide examples if it is necessary to do so in order to make it absolutely clear how a particular speech-act label is employed, in particular if the DART use of the label may diverge from the way it is conventially employed. As already pointed out earlier, a complete listing of all speech acts currently defined in DART is provided in Appendix A for quick reference and comparison of the individual labels.

7.1 Information-seeking speech acts

One of the main features of dialogic spoken interaction is that this form of verbal exchange is generally more or less strongly governed by attempts of the interlocutors to engage in setting up topics, or to keep them going, by eliciting information. This is true even for interview situations where one of the speakers may be assigned a more 'dominant' role in providing the main content by means of producing longer and more detailed turns, but even more so in situations where the participant roles require a more balanced type of interaction. We will therefore begin our discussion of speech acts by looking at those that enquire about information in the widest sense. As illustrated before, the general default assumption expressed in traditional grammars regarding information-seeking acts is usually that they are expressed through questions that constitute requests for information (*reqInfo*). However, as I have already demonstrated, often this is not the case, as many questions may essentially also constitute requests for directives, directives, suggestions, etc. In this case, they are generally referred to as *indirect questions*, a term which may be deemed rather inappropriate for many of these constructs because the majority of the structures/patterns employed to achieve their illocutionary effect are anything but indirect – especially if they contain features such as the directive marker *please* –, but are instead highly conventionalised. Table 7.1 lists the different sub-categories of genuine information-seeking speech acts defined in DART.

As can be seen in Table 7.1, in all our three corpora, the 'prototypical' type of information-seeking speech act is still the request for information. This speech act not only occurs in all dialogues in all corpora, but also with considerably high frequencies. However, looking at the normed frequencies, we perceive rather marked

Table 7.1 Information-seeking speech acts

Speech act	Trl NFreq	Trl Rank	Trl Docs	Trs NFreq	Trs Rank	Trs Docs	SWBD NFreq	SWBD Rank	SWBD Docs.
reqInfo	400	2	35	253.5	6	35	149.5	9	35
reqConfirm	129.5	11	35	148.5	10	31	53.0	18	24
echo	180.0	8	33	46.5	26	25	17.5	33	17
reqOpt	15.5	43	14	13	42	12	2	69	2
reqModal	0.5	72	1	2.5	58	3	2.0	69	2

differences between the corpora. In Trainline, requests for information are clearly the most important information-seeking speech act category, and, as a matter of fact, the second-most important amongst all of the speech acts in general. This feature clearly signals the interactive information-seeking nature of the Trainline domain, which is reflected in both sub-sections of that particular domain. Thus, in the timetable enquiry section of the typical interaction, both Sandra and the callers normally try to elicit some of the relevant information, and in the booking procedure, Sandra tries to ascertain the relevant personal information pertaining to the customers, including titles, addresses, and credit card details. In Trains, *reqInfo* is still represented amongst the top 10 most frequent speech acts, but only occurs in position number 6, while in Switchboard, it only ranks at position 9. For both of these domains, this clearly indicates that information-seeking is not an absolute priority.

In terms of syntactic realisation, it is interesting to note that, in the Trainline data and Switchboard, the majority of the requests for information is expressed in the form of a yes/no-question, thus offering closed sets of choices for response, whereas, in the Trains data, this is achieved through wh-questions that effectively represent open-ended options, which again indicates certain domain-dependent differences. Trainline also exhibits a high number of fragment realisations, nearly equally high to the yes/no option, while the other two domains do not exploit this 'efficiency feature' in any noteworthy manner. For all three corpora, though, the number of declarative questions is fairly low.

The next-most important information-seeking category is that of requests for confirmation (*reqConfirm*). In contrast to the *reqInfo*s, which tend to be more 'pro-active', initiating, in nature, this speech act combines features of both response and initiation, as it constitutes an explicit request to the interlocutor to clarify something assumed to be within the common ground, but at the same time also primes a response, usually either in the form of a confirmation (*confirm*) or negative response (*disConfirm*), although other options, such as e.g. counter-questions, are also possible. This category, despite being important enough in Trainline to occur in nearly all dialogues, only ranks 11th there, whereas it features at rank 10 in Trains, and merely 24 in Switchboard. In Trains, this ranking probably points towards the problem-solving nature of the domain, where it is important to avoid or resolve misunderstandings, and it is perhaps surprising that, in contrast, a domain like that of Trainline should not exhibit such clarifying devices, when, after all, it is at least equally important there to get all the information right. The key here lies in the fact that, in Trainline, Sandra often only implicitly requests a potential confirmation for certain units through the use of repetition of all or part of them. This act is labelled *echo* in the DART scheme, where it ranks in position 8 in the Trainline data, with a normed frequency of 180, 3 places in front of *reqConfirm*. It thus occurs with more than triple the frequency than in Trains (46.5), and it is only through the combination of *reqConfirm* and *echo* speech acts that the importance of ensuring the exchange of correct and complete information really reveals itself here. This use of *echo*es in Trainline is, on the one hand, a feature of efficiency on Sandra's part in that she frequently avoids resorting to a complete, and thereby lengthy, *reqConfirm*, but, on the other hand, at least partly made possible by the nature of the data elicited and requiring potential validation. While, in general, requests for confirmation tend to be designed to confirm whole propositions, echoes are geared towards confirming less substantial, usually elliptical, pieces of information.

(124) <**decl** n="154" sp-act="**reqConfirm**" polarity="positive" topic="enum" mode="constrain-tag-query">
<pause /> we need 3 boxcars all together **right** <punc type="query" /></decl>
</turn>
<turn n="76" speaker="u_PH">
<dm n="155" sp-act="confirm-acknowledge">mhm <punc type="stop" />
</dm> (d93–11.1)

(125) <frag n="21" sp-act="answer-referTime" polarity="positive" topic="duration" mode="decl">
the same day <punc type="stop" /></frag>
<frag n="22" sp-act="elab-referTime" polarity="positive" topic="time" mode="partial-frag">
<pause /> at quarter to 7 <punc type="level" /></frag>
<frag n="23" sp-act="referTime" polarity="positive" topic="time" mode="decl">
6:45 <punc type="stop" /></frag>
</turn>
<turn n="15" speaker="Sandra">
<**frag** n="24" sp-act="**echo**-referTime" polarity="positive" topic="time" mode="decl">
<pause /> **6:45** <punc type="stop" /></frag>

</turn>
<turn n="16" speaker="caller_33">
<dm n="25" sp-act="acknowledge">right <punc type="stop" /></dm>
(trainline33)

In Example (124), s_CB requests an explicit confirmation of a complete proposition through a (declarative) tag question in unit 154, and u_PH confirms the correctness of this proposition with an acknowledging DM, while in Example (125), the caller first clarifies his reference to a particular departure time using a fragment, which is then repeated in its (brief) entirety by Sandra for implicit confirmation. Doing so concurrently allows her to signal her understanding of the information, which is portrayed as being uncontentious, which is why the ensuing acknowledging response is only marked by the single speech act *acknowledge*. The main difference between *reqConfirm* and *echo* therefore lies in the presence or absence of an explicit marking of a need for clarification, rather than just receipt of information. In addition, the former predominantly tends to be realised in the shape of a declarative question, in other words, a well-formed and complete syntactic unit, while the latter primarily occurs in the shape of fragments or other short units.

Although the primary use of echoes is the one described above, there is, however, also another, albeit rarer one, one that is generally confined to the opening or closing stages of dialogues.

(126) <turn n="111" speaker="Sandra">
<frag n="210" sp-act="thank" mode="thank">
thanks <punc type="stop" />
</frag>
<frag n="211" sp-act="bye" polarity="positive" mode="closing-farewell-decl">
<overlap pos="start" /> bye <punc type="stop" />
</frag>
</turn>
<turn n="112" speaker="caller_16">
<frag n="212" sp-act="**echo**-bye" polarity="positive" mode="closing-farewell-decl">
bye <overlap pos="end" /> <punc type="stop" />
</frag>
</turn>
</dialogue> (trainline16)

Example (126) illustrates that echoes, when occurring with formulaic speech acts that generally require reciprocity, do not fulfil any kind of verification function, but are instead only used for politeness' sake to 'mirror' the interlocutor's social act.

The next sub-category to be discussed here is that of the speech act *reqOpt*. This category was already mentioned briefly in Section 6.2.2 above, pointing out that the true meaning of the speech act may range from simply asking about information concerning one or more potential options available to the speaker (127), to a (polite) request for permission (128), or even a 'forceful suggestion' (129).

(127) <q-yn n="38" sp-act="**reqOpt**" polarity="positive" topic="to-from" mode="closed-query">
can i <pause /> dispatch an engine and a boxcar from Elmira <pause /> simultaneously to Corning <punc type="query" /></q-yn>
</turn>
<turn n="19" speaker="s_LP">
<dm n="39" sp-act="hesitate">uh</dm>
<yes n="40" sp-act="stateOpt-**acknowledge**">yeah <punc type="stop" />
</yes>
<yes n="41" sp-act="elab-emphatic"><pause /> yeah <punc type="stop" />
</yes>
<decl n="42" sp-act="expressPossibility" polarity="positive" topic="to" mode="poss1-decl">
we can <pause /> uh <pause /> connect an <pause /> engine to the boxcar <pause /> and then take <pause /> have the engine take the boxcar to Corning <punc type="stop" /></decl> (d92a-3.1)

(128) <q-yn n="76" sp-act="**reqOpt**" polarity="positive" mode="closed-query">
can i change something <punc type="query" /></q-yn>
</turn>
<turn n="33" speaker="s_PL">
<dm n="77" sp-act="stateOpt-**agree**">sure <punc type="stop" /></dm>
(d93–23.3)

(129) <q-yn n="85" sp-act="reqOpt" polarity="positive" topic="booking-number" mode="request-closed-query">can we book on that one please <punc type="query" /></q-yn> (trainline29)

The query about the basic availability of a potential combination of options (unit 38) in (127) is, after an initial hesitation by s_LP, first acknowledged positively (unit 40), and then confirmed emphatically once more before finally being raised as a distinct possibility (unit 42). As this example represents a genuine request for available options, only an acknowledgement would have been required in this case, though. Example (128), on the other hand, is different here, as the status as a request for permission is indicated through an agreeing response instead. This is similar to other forms of polite (often indirect) request, such as *do you mind if I* X, which are also often conventionally responded to in the same manner, despite in fact requiring a negative response from a logical point of view. The agreement here at least partly seems to be indicated through the original semantics of the word *sure*, almost as if such requests were a certain right of the requester, although such an entitlement may of course depend at least to some degree on the role and status of the speaker. The force of the request in (129) clearly stems from the directive marker *please* co-occurring with the interrogative structure, again pointing towards a certain amount of authority of the speaker. The fact that the status of the speaker in the last two examples may have an influence on the exact illocutionary meaning of the unit at all also suggests that it could be useful to adapt the analysis system in the future to include some information as to the respective roles of the participants.

The final speech act category in this section is that of a modal request (*reqModal*), which is essentially a kind of 'garbage category' for yes/no-questions that cannot clearly be identified as to their exact interrogative nature, but that do contain a modal auxiliary.

(130) <q-yn n="14" sp-act="reqModal" polarity="positive" topic="from" mode="alternative-closed-query">
<pause /> would it <pause /> be faster for an engine to come from Elmira <pause /> or Avon <punc type="query" /></q-yn> (d92a-3.1)

(131) <q-yn n="55" sp-act="reqModal" polarity="positive" topic="time" mode="closed-query">
would it be appropriate to uh drink with the <comment type="restart" /> uh prior <comment type="restart" /> when you have the hors d'oeuvres <punc type="query" />
</q-yn> (sw_0057_3506)

At first glance, this category may not be any different from a basic *reqInfo*, but the fact that *reqModal*s frequently contain a hypothetical element, as in Examples (130) and (131), intuitively makes them different from their non-hypothetical counterparts. Further research may still show ways of identifying more unambiguously what the function of such units might be and how to distinguish them more clearly from other information-seeking requests. However, the frequencies and distribution across the corpora are negligible, so that most of the meaning potential seems to already be well covered by the other speech-act categories in this section.

Apart from perhaps *echo*es, all information-seeking speech acts tend to raise some explicit expectations in the dialogue context. In some cases, the expectation may nevertheless not be fulfilled immediately, as has often been noted in the CA and discourse analysis literature (cf. Levinson 1983: 303ff.), when for instance a so-called *insertion sequence* is started, perhaps due to the fact that the interlocutor needs to verify some other fact first before being able to provide a definitive response. The most likely start of such an insertion sequence would be a *hold* request or a *counter-question*, as in the following example.

(132) <turn n="29" speaker="Sandra">
<q-wh n="53" sp-act="**reqInfo**" polarity="positive" topic="time-return" mode="open-query">
<pause length="5s" /> and when is it time you're returning <punc type="query" /></q-wh>
</turn>
<turn n="30" speaker="caller_05">
<dm n="54" sp-act="hesitate">erm</dm>
<q-yn n="55" sp-act="**reqOpt**" polarity="negative" topic="booking-time-day" mode="constrain-closed-query">can i don't have to book a time on the Sunday <punc type="query" /></q-yn>
</turn>
<turn n="31" speaker="Sandra">
<yes n="56" sp-act="**reject**">yes <punc type="stop" /></yes>
<frag n="57" sp-act="elab-**stateOpt**" polarity="positive" mode="decl">
you do <punc type="stop" /></frag>
</turn>
<turn n="32" speaker="caller_05">
<dm n="58" sp-act="hesitate">erm</dm>
<q-yn n="59" sp-act="direct" polarity="positive" topic="duration" mode="query">
can you hold a second again <punc type="query" />
</q-yn> (trainline05)

Here, Sandra wants to elicit some information regarding the intended return journey of the caller (unit 53), who first counters by inquiring about whether there is the option not to have to book the return journey yet (unit 55). When the rejection – which interestingly starts with a positive acknowledgement – is provided by Sandra (units 56 & 57), the caller initiates a hold phase through another yes/no-question in order to reconsider.

7.2 (Non-)Cohesive speech acts

After discussing the group of information-seeking speech-acts, the next logical step is to turn to those acts that may in some form either respond to the former, or play some other role in keeping the dialogue going. While all of the acts in this group can be considered cohesive or non-cohesive in this sense, we can still distinguish between three different subtypes, depending on whether they (1) engage with the interlocutor in either an initiating or responding form,

(2) fulfil a dialogue-managing function, or (3) provide a sense of textual cohesion or coherence. We shall begin our discussion with engaging speech acts.

Table 7.2 Engaging speech acts

Speech act	Trl NFreq	Trl Rank	Trl Docs	Trs NFreq	Trs Rank	Trs Docs	SWBD NFreq	SWBD Rank	SWBD Docs.
acknowledge	525	1	35	872.5	1	35	527.5	2	35
answer	327	4	35	178.5	8	35	88.0	11	34
confirm	101	16	35	135.5	12	30	57.5	16	30
disConfirm	14.5	44	13	2.5	58	3	0.5	85	1
init	313.5	6	35	433	2	34	169	8	35
nominate	–	–	–	–	–	–	0.5	85	1

Looking at Table 7.2, we can see that engaging speech acts are extremely relevant to all three corpora, as, apart from *nominate*, they occur in nearly all dialogues across the corpora, and also tend to be ranked near the top, with the notable exception of *disConfirm*. The most important among these is *acknowledge*, which indeed does occur in all dialogues and is consistently ranked in either 1st or 2nd place. The reason for this is that acknowledgments are highly multi-functional in that they can both respond to interrogatives and serve as indicators of receipt of information for non-interrogatives. And both of these functions certainly represent a kind of engagement with the prior context the interlocutor has created, indicating that the speaker using acknowledgements is being co-operative in the Gricean sense, and wishes to keep the dialogue going. In addition to this, *acknowledge* may also be used for self-acknowledgments, that is to express certainty about one's own assumptions being correct, as in Example (133).

(133) <turn n="70" speaker="u_JA">
<decl n="175" sp-act="reqInfo" polarity="positive" mode="query">
so the <overlap pos="start" /> total <overlap pos="end" /> <pause /> is
<punc type="query" />
</decl>
</turn>
<turn n="71" speaker="s_RD">
<frag n="176" sp-act="answer-enumerate" polarity="positive" topic="enum"
mode="decl">
<overlap pos="start" /> 5 <overlap pos="end" /> <punc type="stop" /></frag>
<decl n="177" sp-act="elab-confirm" polarity="positive"
mode="reassurance-decl">
that's right <overlap pos="start" /> <punc type="stop" />

```
</decl>
<dm n="178" sp-act="acknowledge"><pause /> okay <overlap pos="end" />
<punc type="stop" />
</dm>                                                          (d92a-1.2)
```

In Example (133), u_JA first enquires about the complete duration of a (sub-)task, which s_RD – perhaps tentatively – answers in unit 176, but she then elaborates on this answer by confirming it once more, both to herself and her interlocutor, and, to reinforce this confirmation even further, also acknowledges it.

In the DART scheme, *answer* is defined as the response to a *reqInfo*, while other forms of interrogatives trigger different types of responses. Theoretically, of course the number of answers could be equal to that of requests for information, but in practice, partly due to interlocutors sometimes being able to simply acknowledge, ask counter-questions, or simply not wanting or being able to provide answers to them, this is not the case in any of our corpora. The corpus that comes closest to constituting a 'question–answer game' in our case is Trainline, where answers rank 3rd, following *reqInfo*s and acknowledgements. This is probably not surprising, since, as pointed out earlier, in Switchboard, information-seeking speech acts in general tend to be fewer and generally used to start off particular topics, and in Trains, the decision-making process that reflects the problem-solving activity is revealed more through the acknowledgement of facts or options, which is partly why the speech act *acknowledge* ranks 1st there, as well as through the occurrence of different types of statements related to the task, as we shall see in 7.3 below.

Both *confirm* and *disConfirm*, taken together, constitute the most likely, or preferred, positive and negative forms of response to either a *reqConfirm* or *echo*, as we have already seen above. The fact that the two 'triggering' information-seeking speech acts are still rarer than requests for information throughout the corpora, and also that not every *echo* requires an explicit confirmation, or every *reqConfirm* actually gets honoured with a response, explains why these speech acts figure less prominently in the data. One thing that is obvious in all three corpora, though, is that the assumptions behind either requests for information or echoes generally appear to be correct, as the negative response type is much rarer than the positive one.

While the first four speech acts in Table 7.2 engage (with) the interlocutor in a responding manner, the final two instead initiate the engagement. As the ranks and distribution show, *init* is one of the most important speech acts in all three corpora, as it helps speakers to mark the beginnings of particular topics or sub-topics within a dialogue. As we have already seen in some prior examples, this marking is largely achieved via DMs like *so, now*, or *well then, okay, (al)right (then)*, or yes-units containing *(oh) yeah*, as well as potentially their counterpart containing *no*, all of which may of course also fulfil other functions in the dialogue to a greater or lesser extent.

It is interesting to note, though, that in yes-/no-units, it is only the more canonical forms that appear to be able to mark initiation, at least in our data, although it should in theory be perfectly possible to use less standard form like *yep, aye, nope*, and *nay*, too. There may, however, be at least some phonological influence here in that forms ending in vowels appear to be preferred, perhaps because they can more readily be lengthened to indicate continuation prosodically.

The final form of engagement-indicating speech act, *nominate*, occurs once in all the data, but is a good example of a relatively rare speech act that nonetheless requires modelling in order to arrive at as complete a picture of human interaction as possible. It is used to indicate in which order speakers, or agents in other activities, are meant to proceed, and usually expressed through structures like *you're first/second/next* or *you go first/second/next*.

Whereas engaging speech acts actively foster the verbal interaction between interlocutors in a dialogue, the next sub-group, that of dialogue-managing acts, is generally used to overcome potential planning or other issues that present themselves during the course of the interaction.

Table 7.3 Dialogue-managing speech acts

Speech act	Trl NFreq	Trl Rank	Trl Docs	Trs NFreq	Trs Rank	Trs Docs	SWBD NFreq	SWBD Rank	SWBD Docs
abandon	86.5	17	32	265.5	5	35	468.5	3	35
hesitate	174.5	9	35	344.5	3	35	405	4	35
phatic	11.5	46	11	19.5	36	13	226.5	7	35
hold	70.5	19	29	75.5	18	25	9	45	9
muse	44	29	25	63.5	20	24	118	10	34
pardon	23.5	38	16	2.5	58	3	8.5	47	8

In abandoning (*abandon*) a unit, the current speaker chooses to discard dialogue content that is presumably deemed unsuitable or irrelevant in the present context, or has been uttered without proper prior planning. Abandoned units are clearly less frequent in Trainline than in the other two corpora, perhaps because, there the choices speakers can make are more clearly delineated, due to the relatively narrow domain that leaves little leeway for truly negotiating and pondering different options that one may be able to get wrong. In other words, the playbook question–answer game makes it easier to make decisions and say exactly what one means to say, rather than having to change tack too frequently. Changing one's opinions half-way through, however, is far more common in decision-making processes, such as the one in Trains, where different options constantly need to be evaluated, and the cognitive load, along with the pressure to complete the task, is clearly higher, thereby affecting speech planning. In contrast, in Switchboard, the options for topics, and how they are discussed and presented, are much wider than in the other two corpora. At the same time, there is less pressure on the interlocutors to actually get their facts straight, so they are perfectly at liberty to change tack whenever they realise 'mid-way' that the information they want to provide may be better presented in a different way.

There are, however, also situations in which units are abandoned involuntarily, due to the interlocutor barging in and attempting to take over the turn or, in more co-operative cases, to collaboratively complete a unit. Such interruptions do occur in all corpora, 19 times in 14 documents in Trainline, 32 in 18 in Trains, and 69 in 30 in Switchboard. Out of these, 11 in 7 documents in Trainline, 10 in 6 in Trains, and 8 in 8 documents in Switchboard constitute collaborative completions, all expressed as raw frequencies. Thus, while the readiness to interrupt appears to definitely increase when the conversational partners are of more equal status, the inverse seems to be true when it comes to using these interruptions in a collaborative way.

Rather unsurprisingly, hesitations (*hesitate*) occur in all three corpora. In the DART scheme, this speech act captures initial fillers that indicate a short-term planning process before a definite response is provided, or a potential option stated.

(134) <turn n="36" speaker="u_JA">
<decl n="78" sp-act="reqInfo" polarity="positive" topic="number-duration" mode="query">
and that's <pause /> how many hours <punc type="query" /></decl>
</turn>
<turn n="37" speaker="s_RD">
<**dm** n="79" sp-act="**hesitate**">uh</dm>
<decl n="80" sp-act="answer-predict" polarity="positive" topic="enum-to-duration" mode="predict-decl">
that'll take <pause /> 3 hours to get to Dansville <punc type="stop" />
</decl> (d92a-1.2)

As we can see in (134), the speech act *hesitate* can therefore be seen as strategic planning device that enables the speaker to reflect for a short period of time, and consider a sensible response. This assumption is very different from theories where initial *filled pauses*, as such *um*s and *er*s are often referred to, are simply seen as features of disfluency that mark the inability of a speaker to respond in a 'fluent' and coherent way. Interestingly, when comparing across corpora, we can again observe a rise in the use of such hesitators from Trainline, via Trains, to Switchboard, in terms of frequencies. As with abandoned units, this difference can probably again be explained with differences in task complexity, cognitive load, and absence of performance-related pressure, which allows speakers to gain more planning time at no penalty.

The speech-act label *phatic* is applied to units that, like hesitations, are rather devoid of meaning, but serve either as a way of gaining planning time, or to maintain social relations between speakers. This label is commonly used to identify expressions like *you know* or *i mean*, when they 'preface' other units, but do not precede a direct object or that-clause. The latter distinction here is crucial because, as these expressions have previously – most notably by Schiffrin (1987: 267–311) – been treated as appealing to or maintaining common ground between interlocutors, the former by referring more specifically to joint knowledge, and the latter by prefacing potential elaborations of an explanatory nature. The trouble with this approach is that it that these explanations rely too much on the original, now bleached, meaning of the expressions, and also do not take into account that such structures, when used as DMs, generally precede the subject of the next unit, so that there is no coherence relation between the subject of the DM and the following unit. Yet, looking at the rank and frequency distributions across our three corpora, it becomes rather apparent that the use of these expressions as DMs is more a question of register than any form of semantics, as both rank and frequency are far higher in the informal conversations in Switchboard than in the task-oriented domains of Trainline and Trains. If the assumption that they generally appeal to common ground were viable, though, then we would certainly not expect to find such differences, especially as maintaining common ground is far more essential in task-oriented domains than in informal settings.

In contrast to the two speech acts above, which only indicate an implicit planning function, *hold* represents an explicit plea to gain time for deliberation or physically do something. The name is inspired by the expression *(please) hold the line*, which is common in task-oriented telephone conversations between an operator and a client, when the operator needs to request a 'time-out' in order to look up some information on the computer. In task-oriented domains, other expressions like *bear with me* are also common, while in informal conversation structures such as *wait a second/minute* or *let's/let me see* occur more frequently. Normed frequencies, ranking, and distributions all clearly indicate that this strategic device is more important for task-oriented domains, where it is more common for participants to have to check on information, or to deliberate what to do next. The use of this label in DART is very different from that in DAMSL, where "Hold" (Allen & Core 1997: 19) is essentially used to mark any response that delays a recommended course of action, rather than addressing it immediately.

The label *muse* essentially covers two simple patterns, the DM *well*, usually spoken with a fall-rise or level intonation and suitable lengthening marking non-finality, and the minimal response *hm*. It generally occurs at or near the beginning of a turn, and the speech act serves to indicate some potential reservation or insecurity on how to respond to a query or assumption by the interlocutor. The use of *well* in this context differs from its, rarer, use as an initiating DM, which is more likely to be spoken without lengthening and a prosodic contour indicating finality. Although the DM *well* has been discussed extensively in the literature on DMs (e.g. Schiffrin 1987; Jucker 1993; Aijmer 2013; Heritage 2015) and a variety of, sometimes subtly, different functions have been identified there, I feel that the distinction between its initiating and musing functions is sufficient.

The dispersion and ranking suggest that this is still a highly important speech act in Switchboard, as it occurs at rank 10, as well as in all but one of the dialogues. In contrast, it is far less important in the two task-oriented domains. This difference is probably due to the more targeted nature of the task-oriented domains, where responses to binary choices are more likely, while dialogues of a more personal nature require more 'weighing' of options, especially when the questions responded to require one to express some form of evaluation or opinion, which, as we shall see later in 7.6, is corroborated by the high frequency, rank, and dispersion of expressions of opinion in Switchboard, a number of which occur directly after, or in the vicinity of, a *muse* speech act.

The speech act *pardon*, which is again associated with, or mainly represented by, a special DM, comes in two different flavours, as exemplified below.

(135) <turn n="3" speaker="Sandra">
<q-yn n="7" sp-act="reqInfo" polarity="positive" topic="creditcard" mode="closed-query">
<pause length="3s" /> and <pause /> is it yourself as is a credit and debit card holder <punc type="query" /></q-yn>
</turn>
<turn n="4" speaker="caller_08">
<dm n="8" sp-act="**pardon**" mode="regret">**pardon** <punc type="query" /></dm>
</turn>
<turn n="5" speaker="Sandra">
<q-yn n="9" sp-act="reqInfo" polarity="positive" topic="creditcard" mode="alternative-closed-query">
do you hold a current debit or credit card <punc type="query" /></q-yn>
(trainline08)

(136) <decl n="31" sp-act="stateOpt" polarity="positive" topic="from-time-enum" mode="exists-decl">
there's a train at 15:05 from Crewe <punc type="stop" /></decl>
<decl n="32" sp-act="state-abandon" status="abandon" polarity="positive" topic="arrival" mode="abandon">and you'd arrive <punc type="incomplete" /></decl>

<dm n="33" sp-act="**pardon**" mode="regret"><pause /> **sorry** <punc type="level" /></dm>
<frag n="34" sp-act="referDirection" polarity="positive" topic="from-location_Anglo" mode="partial-decl">from Euston <backchannel content="yes" /> <punc type="stop" />
</frag> (trainline09)

The first type (Example (135)) signals a misunderstanding on the part of the hearer, and its function is to get the speaker to repeat the previous question or piece of information. The second type functions as an indicator of an ensuing self-correction or, conversely, the fact that some previously supplied information may be incorrect, which can be seen in Example (136). This speech act plays no major role in any of the corpora, so we will not discuss it in any further detail here. If the focus of investigation in analysing the corpora were on identifying potential issues in misunderstandings or miscommunication, then it would of course need to be investigated further, and properly distinguishing between the two functions might possibly even necessitate creating an extra speech-act label for one of them.

The final sub-group of (non-)cohesive acts is different from the other two groups in that the function of its members on the interaction level is not to respond, but instead to mark cohesion within, or even across, speaker turns. We can therefore refer to it as *textual*.

Table 7.4 Textual speech acts

Speech act	**Trl NFreq**	**Trl Rank**	**Trl Docs**	**Trs NFreq**	**Trs Rank**	**Trs Docs**	**SWBD NFreq**	**SWBD Rank**	**SWBD Docs**
complete	6.5	53	7	7.5	52	5	5	54	8
attribute	–	–	–	–	–	–	–	–	–
listSequence	–	–	–	2.5	58	1	–	–	–
initFollowUp	18	41	14	110.5	14	34	44.5	21	27
initConclusion	–	–	–	–	–	–	0.5	85	1
initSummary	–	–	–	–	–	–	0.5	85	1
initGeneralisation	–	–	–	–	–	–	–	–	–
initQ	–	–	–	1	71	1	–	–	–
initCounterExp	3	63	3	12	45	10	12	39	13
initConstraint	–	–	–	1.5	63	1	–	–	–
initCondition	–	–	–	1.5	63	1	–	–	–
initContrast	–	–	–	1.5	63	2	8	48	6
initReason	–	–	–	–	–	–	3.5	59	4
initOpinion	–	–	–	–	–	–	3.5	59	4

The first speech act in this sub-category is *complete*, which is the only one of these acts that establishes cohesion across turns in that one speaker collaboratively completes the unit of another, as illustrated in Example (137).

(137) <dm n="29" sp-act="initFollowUp">and then</dm>
<frag n="30" sp-act="referDirection-abandon" status="interrupted" polarity="positive" topic="to">
back <pause /> to Bath <pause /> by <overlap pos="start" /> mid… <pause /> <overlap pos="end" /> <punc type="incomplete" /></frag>
</turn>
<turn n="11" speaker="s_EP">
<frag n="31" sp-act="**complete**-refer" polarity="positive" mode="partial-decl">
<overlap pos="start" /> by noon <overlap pos="end" /> <punc type="stop" /></frag>
</turn>
<turn n="12" speaker="u_DT">
<frag n="32" sp-act="echo-refer" polarity="positive" mode="partial-decl">
by noon <punc type="stop" />
</frag> (d92a-5.2)

Here, u_DT in unit 30 first starts referring to a location in the form of a direction, and then goes on to add a time for the arrival at that location. Before he can complete this unit, though, s_EP completes this reference to an arrival time in unit 31, albeit in a slightly different way, which, however, is accepted by u_DT as being the correct completion in unit 32 through his *echo*. By comparing the highly similar ranks of such completions, we can see that such collaborative behaviour seems to be independent of the domain.

The next speech act in this category is cohesive in the sense that it introduces the information following in the next unit as belonging to another speaker, who may or may not be the interlocutor. This speech act, which takes the form *according to* X …, does not actually occur in any of our three corpora, but was defined in DART as it occurred in some courtroom interactions in ICE-HK.

The speech act *listSequence* is used in DART to mark acts that indicate that the following unit is part of a hierarchical or temporal ordering, and may take a limited variety of forms, such as *at first, first of all, first(ly)/second(ly)/…*, etc. It can occur as part of any decision-making process that requires a kind of temporal or hierarchical ordering. In our corpora, it only occurs in one document in Trains, but altogether three times.

(138) <decl n="7" sp-act="stateConstraint" polarity="positive" topic="location" mode="constrain-decl">
all these commodities must be in Bath by noon <punc type="stop" /></decl>
<decl n="8" sp-act="stateTime" polarity="positive" topic="time" mode="decl">
<pause /> and it’s already midnight <punc type="stop" /></decl>

<dm n="9" sp-act="acknowledge"><vocal content="brth" /> alright <punc type="level" /></dm>
<dm n="10" sp-act="init">so <punc type="level" /></dm>
<dm n="11" sp-act="**listSequence**">**first** <punc type="level" /></dm>
<imp n="12" sp-act="suggest" polarity="positive" topic="commodities-transport_means" mode="decl-disflu">
let's <pause /> let's deal with the boxcar full of bananas <punc type="stop" />
</imp> (d93–22.2)

Example (138) clearly demonstrates that this speech act may have properties that both make it applicable in temporal sequencing and in prioritising certain actions, which may render it highly relevant to such domains as business meetings or any other meetings that may be based on some form of agenda.

Another act, *initFollowUp*, is also used in order to indicate a temporal or structural order, but does not (necessarily) mark any hierarchical function. It is typically realised through such structures as *also, (and/but/so) then, in addition*, etc., and marks additional, rather than first, units in a sequential description. In our three corpora, indicating a sequencing of actions is clearly most important in Trains, where this speech act occurs with the highest frequency and rank, as well as in almost all documents. This reflects the decision-making process where a series of actions need to be performed in order to achieve the task, and can be seen in Example (139). In contrast, in Trainline, the speech act appears to be far less important, probably because the journeys the callers enquire about require relatively few stages. In Switchboard, the speech act is again a little more important, but instead of indicating items in a sequence of transactional stages, as Example (140) shows, the follow-up information tends to refer to stages in which events unfold.

(139) <dm n="90" sp-act="**initFollowUp**">**and then**</dm>
<frag n="91" sp-act="referDuration" polarity="positive" topic="duration-enum" mode="frag">
4 hours <punc type="level" /></frag>
<frag n="92" sp-act="referDuration" polarity="positive" topic="number-to-location-duration" mode="decl">
<pause /> 4 hours to get to Bath <punc type="stop" /></frag>
<frag n="93" sp-act="enumerate" polarity="positive" topic="enum" mode="decl">
<pause /> so 11 <punc type="stop" /></frag>
<dm n="94" sp-act="**initFollowUp**">**and then** <punc type="incomplete" />
</dm>
<dm n="95" sp-act="init">now <punc type="level" /></dm>
<decl n="96" sp-act="stateDuration" polarity="positive" topic="duration-enum-time" mode="decl">
we're at 13 hours <punc type="stop" /></decl> (d93–18.3)

(140) <decl n="118" sp-act="state" polarity="positive" topic="location-enum-duration" mode="decl">
she worked there as a volunteer for about 10 years <punc type="stop" />
</decl>
<dm n="119" sp-act="**initFollowUp**">**and then**</dm>
<decl n="120" sp-act="state-abandon" status="abandon" polarity="positive">
she decided that <punc type="incomplete" /></decl>
<dm n="121" sp-act="hesitate"><comment type="restart" /> uh</dm>
<decl n="122" sp-act="state" polarity="positive" topic="age-enum" mode="frag">
she's 85 years old <punc type="level" /></decl>
<decl n="123" sp-act="state" polarity="negative" topic="family-cars" mode="decl">
that eventually you know she she doesn't want her family taking care of her and being a burden <punc type="stop" /></decl> (sw_0006_4108)

As the next three speech acts in Table 7.4 are only defined in DART, but do not actually occur in any of the three corpora, I shall only introduce them here briefly. In the same way that the speech act discussed previously prefaced the introduction of additional information about individual further steps in a sequence, *initConclusion* indicates that the next unit will provide a statement containing a logical conclusion about something, *initSummary* that one or more summarising statements are to follow, and *initGeneralisation* that some form of generalisation is to follow. The former is realised through expressions like *hence, in other words, in that way*, etc, the second through e.g. *in conclusion* or *summarising*, and the last through *as a rule (of thumb), in general, on the whole*, etc.

initQ only occurs once in Trains, and not at all in the remaining corpora. However, it may well be important in dialogues where issues are raised frequently, and is generally expressed through stuctures that contain *the X question is*, and where X may either be empty or represented by an ordinal number, or some other form of pre-modifying AdjP like crucial or most important.

The speech act *initCounterExp* again occurs in all three corpora, but nowhere with any remarkable frequency. Its function is to introduce information that may be regarded as being contrary to expectation, and it is generally realised via *actually, in/as a matter of fact, the thing is*, etc. I have already (Section 4.3.3) discussed how one of these, *actually*, has been treated in the relevant literature in the past.

The remaining speech acts in the textual sub-category all preface different types of statements in the widest sense. The first two, *initConstraint* and *initCondition*, preface a statement containing a constraint or condition affecting the ongoing dialogue, the next one, *initContrast*, which is realised through such expressions as e.g. *however, yet*, or *on the other hand*, introduces information that highlights a contrast to something the speaker has stated in the prior context. *initReason* prefaces a statement that provides some kind of justification, while the final one, *initOpinion*, indicates that an expression of opinion is to follow. None of these occurs with very high rank or frequency in the data, and these speech acts are even completely absent from Trainline. However, they complement the speech acts contained in direct expressions pertaining to the individual dialogue 'facts' they introduce, and which occur with much higher frequency, as we shall see further below. In theory, it would also be possible to add rules to DART's inferencing process that automatically mark the units following e.g. an *initConstraint* as a *stateConstraint* or *referConstraint*, respectively, depending on whether that unit was a declarative or fragment. However, this feature has not been implemented in the current version of DART yet, and it still also remains to be tested how many of the following units may in fact be in the scope of such an initiation, as it is technically well possible that multiple follow-up units are affected.

As has probably become apparent from the patterns listed above, the majority of textual (non-)cohesive speech acts generally takes the form of DMs, although they may not have treated as such by other scholars in the past.

7.3 Information-providing and referring speech acts

Information-providing speech acts form the counterpart to the information-seeking speech acts outlined at the very beginning of this chapter. Unlike the (non-)cohesive speech acts we discussed immediately above, which predominantly provide response signals or hints as to what may be expected next, information-providing acts actually furnish propositional content to fill the 'information void' created by a query that cannot simply be acknowledged. They comprise a number of sub-categories, namely *informing* and *referring, elaborating, explaining, awareness-indicating, hypothesising*, and *volitional* ones. I shall list all of these sub-categories in separate tables below, and illustrate their individual functions and/or importance relative to the different domains.

The most neutral informing speech act is *state*, which is assigned in DART if there is a finite verb present in a declarative or fragmentary structure and no other, more specific, label indicating an information-providing act can be assigned. The counterpart to *state*, which is assigned to fragmentary units without a finite verb, is *refer*. As the label suggests, such units tend to be deictic in nature, i.e. predominantly contain referring NPs or PPs, although other phrase types may also occur. Such referring speech acts either take the shape of focussing adverbial prefaces to other units, elliptical responses, or sometimes incomplete units. This dual patterning of informing and referring speech acts is also frequently reflected in the other

Table 7.5 Informing or referring speech acts

Speech act	Trl NFreqs	Trl Rank	Trl Docs	Trs NFreqs	Trs Rank	Trs Docs	SWBD NFreqs	SWBD Rank	SWBD Docs
state	357.5	3	35	329.0	4	33	1,119.0	1	35
refer	322.0	5	35	115.5	13	30	270.0	5	35
negate	48.0	27	32	17.0	38	16	19.0	31	18
report	8.5	51	11	8.5	50	6	63.0	14	23
enumerate	253.0	7	30	24.0	34	18	2.5	64	2
stateCondition	59.5	23	28	33.5	29	16	57.0	17	24
referCondition	0.5	72	1	1.0	71	1	2.0	69	3
stateConstraint	109.5	14	33	157.0	9	32	48.0	19	28
stateNonConstraint	1.5	68	2	7.0	53	6	0.5	85	1
referConstraint	6.0	57	9	1.0	71	1	0.5	85	1
stateOpt	30.0	34	24	16.5	39	14	1.5	76	2
stateNonOpt	2.0	66	3	–	–	–	1.5	76	1
referOpt	–	–	–	1.0	71	1	–	–	–
stateProcess	28.5	35	19	16.5	39	10	8.0	48	11
referProcess	74.5	18	32	136.5	11	25	10.5	40	8
stateHabit	1.5	68	2	–	–	–	9.0	45	11
referDirection	24.0	37	26	55.5	22	24	–	–	–
stateDistance	–	–	–	–	–	–	0.5	85	1
stateDuration	2.0	66	2	50.5	24	23	4.5	55	5
referDuration	–	–	–	92.5	15	22	6.5	51	8
stateTime	11.5	47	9	37.0	28	19	3.5	59	5
referTime	117.5	13	34	76.5	17	28	64.0	13	28
stateDate	7.0	53	10	–	–	–	–	–	–
referDate	5.0	59	4	–	–	–	–	–	–
referPlace	46.0	28	25	12.0	45	11	10.5	40	10
referPerson	–	–	–	–	–	–	1.5	76	1
referThing	–	–	–	1.5	63	2	3.5	59	5
referAct	–	–	–	–	–	–	1.5	76	1
referHow	–	–	–	–	–	–	–	–	–

categories we shall discuss below, although there is not always a stating equivalent to a referring act. General statements and referring expressions consistently rank among the top speech acts in all corpora. There is only one exception, Trains, where general referring expressions are not in the top 10, but 'only' rank in 13th place. This, however, is probably due to the fact that these expressions here 'stand in competition with' other, more specific, referring expressions, such as references to ongoing processes or actions (rank 11), durations (rank 15), and times (rank 17), which also occur amongst the top 20 items. Switchboard also stands out here in that *state* is the highest-ranked speech act of all, while this position is occupied by *acknowledge* in the other two corpora, but which only features in second position in Switchboard. This fact illustrates a major difference between the two task-oriented corpora, where responding to information-seeking acts is more important, and Switchboard, where keeping the narrative going is far more important, something that is corroborated by the ratio of general statements to acknowledgements being more than 2:1.

The speech act *negate* is essentially the negative, and equally neutral, counterpart to *state*. It primarily marks negative-polarity responses to requests for information or confirmation, as well as some other statements that cannot be responded to by either a *refuse, reject*, or *disConfirm*. The ranks for *negate* are relatively unremarkable for all three corpora, although Trains again stands out a little here with this speech act being least important. This possibly suggests that the assumptions behind the different types of information-seeking acts there are more likely to be correct, perhaps because the options in this domain are also more highly constrained and therefore it is easier to frame questions that elicit a positive response.

The label *report* indicates instances of reported speech, both produced by other parties or the speakers themselves. While the ranks and document frequencies for this speech act are relatively unremarkable for the task-oriented corpora, the importance of what someone else of the speaker themselves have said clearly appears to be a more important part of the narrative in Switchboard.

Enumerations (*enumerate*) in the DART scheme represent a highly restrictive form of providing information that consists only of a single number or a sequence of numbers. Supplying such information is clearly least important in Switchboard, while in Trains, it already happens somewhat more frequently in responses to how many engines or boxcars are required for transporting goods, or statements 'summing up' how many hours will already have passed. Nevertheless, both rank and dispersion do not single this act out as being of very high importance. In contrast, such enumerations rank in 7th place in Trainline, indicating that numerical information plays a very important role in this domain, despite the fact that the speech act does not even occur in all dialogues. However, we need to remember in this context that not all interactions between Sandra and her callers lead to an actual booking stage, which does explain the latter, as enumerations in Trainline generally reflect the callers' providing credit card numbers, etc., within the booking stage.

The remainder of the categories in Table 7.5 all constitute special cases of stating or referring that have the potential to influence the ongoing interaction in various ways. The speech acts *stateCondition* and its adverbial counterpart, *referCondition*, introduce conditions into the common ground. They are present in all three corpora, and one might initially assume that they ought to be more likely to occur in task-oriented domains where concrete conditions or options that affect the tasks are more relevant, as is evident in Example (140). Yet, this is not really the case, as the higher rank in Switchboard indicates. However, the conditions being talked about there tend to relate more to everyday-life concerns or abstract moral issues that are being debated, as in Example (142).

(141) <frag n="91" sp-act="refer" polarity="positive" topic="fare" mode="partial-frag">
with the standard open single <punc type="level" /></frag>
<decl n="92" sp-act="**stateCondition**" polarity="positive" topic="miss" mode="condition-frag">
if you miss the service i've reserved you on <punc type="level" /></decl>
<decl n="93" sp-act="expressPossibility" polarity="positive" topic="availability" mode="poss2-decl">
you are able to get the next available train <backchannel content="right" />
<punc type="stop" />
</decl> (trainline01)

(142) <decl n="86" sp-act="**stateCondition**" polarity="positive" mode="condition-frag">
if somebody gets the death penalty <punc type="level" /></decl>
<decl n="87" sp-act="state" polarity="positive" mode="frag-disflu">
they're <comment type="restart" /> they're judged guilty <punc type="level" /></decl>
<decl n="88" sp-act="state" polarity="positive" mode="frag">
they got the death penalty <punc type="level" /></decl>
<decl n="89" sp-act="suggest" polarity="positive" topic="enum-duration" mode="suggest-decl">
they should have 1 year and 1 appeal <punc type="stop" /></decl>
(sw_0013_4617)

Yet Example (141) also illustrates that it may not always be straightforward to decide what should count as a *stateCondition* if no overt marking for this is present, since units 87 and 88 may also have been intended as being part of an elliptical listing (or reiteration) of conditions, only with the *if* and overt linking conjunction missing.

The next pair of speech acts, *stateConstraint* and *referConstraint*, in contrast, are somewhat less important in Switchboard than in the two task-oriented domains, and clearly most important in Trains. These acts mark expressions that indicate requirements that may limit the choices of action of the interlocutors, and, in contrast to the conditions we just discussed, are usually not indicated via conjunctions like *if, whether*, or *unless*, but through expressions of deontic modality, most often reflecting necessity, as can be seen in Example (143).

(143) <decl n="35" sp-act="expressImPossibility" polarity="negative" mode="imPoss1-decl">
we can't <pause /> load <pause /> oranges into an engine
<punc type="stop" /></decl>
<decl n="36" sp-act="**stateConstraint**" polarity="positive" mode="constrain-decl">
we need a boxcar <backchannel content="mhm" /> to load them into
<punc type="stop" />
</decl> (d92a-3.1)

In (143), s_LP first raises an issue by expressing an impossibility, and then, in a sense, provides an explanation for this fact by stating a constraint imposed by the scenario. The 'negative' counterpart to *stateConstraint, stateNonConstraint*, is quite rare in all three corpora. It indicates the absence of a constraint or necessity to do something, commonly in response to the assumption of a constraint being present in the first place, and expressed by the interlocutor.

(144) <dm n="38" sp-act="acknowledge">right <punc type="stop" /></dm>
<decl n="39" sp-act="**stateNonConstraint**" polarity="positive" topic="time" mode="decl">
so you've got lots of time <punc type="stop" /></decl> (d93–9.2)

(145) <no n="177" sp-act="answer-negate">no <punc type="stop" /></no>
<decl n="178" sp-act="elab-**stateNonConstraint**" polarity="negative" mode="decl">
what i'm saying is you don't need to take the card <punc type="stop" /></decl>
<decl n="179" sp-act="stateConstraint" polarity="positive" topic="miss" mode="contrast-constrain-decl">
but you need a letter of authorisation <backchannel content="right ok" />
giving you permission to use the card <punc type="stop" /></decl>
<decl n="180" sp-act="stateConstraint" polarity="positive" mode="constrain-decl">
and it must be on the company headed paper <punc type="stop" /></decl>
(trainline14.xml)

Example (145) illustrates how units expressing a *stateContraint* and *stateNonConstraint* may also be linked together. Here, in response to the caller's assumption that they need to bring a company credit card with them (not shown in the example), Sandra first negates this, then goes on to state that it is not necessary to actually bring this credit card along, but instead lists the concrete requirements for being able to use a company credit card in such cases.

The next category to be discussed here, *stateOpt*, has already occurred previously in a number of examples, in the context of responses to a *reqOpt*. As before, there is also a referential counterpart *referOpt*, which, however, only occurs once in all the corpora, as well as a negative one, *stateNonOpt*. The latter also only occurs with very limited frequency in Trainline and Switchboard, and essentially captures references to non-availability or non-existence of certain options.

(146) <decl n="83" sp-act="**stateNonOpt**" polarity="**negative**" topic="**availability**" mode="alternative-condition-partial-decl-disflu">
that <comment type="restart" /> that option sometimes is not available if they do not have the <comment type="restart" /> uh <comment content="noise" /> either Medicare or insurance to cover it <punc type="stop" /></decl> (sw_0006_4108)

(147) <decl n="31" sp-act="**stateNonOpt**" polarity="**negative**" topic="journey-fare-enum-time" mode="**exists**-decl">
<pause length="4s" /> there's not any advance purchase tickets on the 7 a.m. train <punc type="stop" />
</decl>
<decl n="32" sp-act="stateOpt" polarity="positive" topic="enum-time" mode="alternative-decl">
<pause /> it's either 6 o'clock or 9 o'clock <punc type="stop" /></decl>
(trainline16)

As Example (147) shows, at least in transactional dialogues, such references to options not being available are frequently also balanced by units that provide existing alternatives.

Whereas, so far, for many other pairs of stating and referring speech acts, the stating variant has been the predominant one, for *stateProcess* and *referProcess*, this situation is clearly reversed, especially for the task-oriented domains. The reason for this is that processes are essentially indicated by non-finite forms, such as progressives or infinitives, that tend to occur in fragments which either precede some form of stating declarative, or form part of sequences of actions or steps required in order to achieve a task.

(148) <frag n="8" sp-act="**referProcess**" polarity="positive" mode="frag">
<pause /> to make OJ <punc type="level" /></frag>
<decl n="9" sp-act="stateConstraint" polarity="positive" topic="transport_means-location-commodities-to" mode="constrain-decl">
you have to get <pause /> a boxcar of oranges to Elmira where the OJ <pause /> factory is <punc type="stop" /></decl>

<decl n="10" sp-act="stateConstraint" polarity="positive" topic="transport_means-number" mode="decl">
and you'll need a tanker there to carry the OJ once it's finished <punc type="stop" /></decl> (d93–16.1)

(149) <frag n="24" sp-act="**referProcess**" polarity="positive" topic="education" mode="frag">
working for a university <punc type="level" /></frag>
<dm n="25" sp-act="hesitate">uh</dm>
<decl n="26" sp-act="state" polarity="positive" mode="exists-frag">
there is room for advancement <punc type="level" /></decl>
(sw_0042_4060)

(150) <turn n="20" speaker="s_CK">
<dm n="62" sp-act="init">okay <punc type="level" /></dm>
<dm n="63" sp-act="init"><pause /> so <punc type="level" /></dm>
<frag n="64" sp-act="**referProcess**" polarity="positive" topic="transport_means-to-enum-location" mode="decl">
follow the procedure of sending <pause /> an engine <pause /> engine 1 to <pause /> Dansville <punc type="stop" /></frag>
<frag n="65" sp-act="**referProcess**" polarity="positive" topic="transport_means" mode="decl">
pick up a boxcar <punc type="stop" /></frag>
<frag n="66" sp-act="**referProcess**" polarity="positive" topic="location-to" mode="decl">
<pause /> go to Corning <pause /> <punc type="stop" /></frag>
<frag n="67" sp-act="**referProcess**" polarity="positive" topic="commodities">
pick up oranges</frag>
<frag n="68" sp-act="**referProcess**" polarity="positive" topic="to-location-transport_means" mode="decl">go to Elmira <punc type="stop" /></frag>
(d93–16.1)

(151) <dm n="69" sp-act="init">now</dm>
<frag n="70" sp-act="referTime" polarity="positive" topic="month" mode="partial-frag">
on the 8th October <punc type="level" /></frag>
<decl n="71" sp-act="**stateProcess**" polarity="positive" topic="from-location_Anglo-journey-time" mode="frag">you're travelling from Birmingham New Street at 7:33 <punc type="level" /></decl>
<frag n="72" sp-act="**referProcess**" polarity="positive" topic="arrival-time-location_Anglo" mode="decl"><pause /> arriving in Euston at 9:05 <punc type="stop" /></frag> (trainline01)

In Examples (148) and (149), the references to processes essentially introduce and focus a topic. In (148), the fragment containing the to-infinitive + object brings a relatively clearly defined and concrete process as a sub-topic into the common ground, and the speaker then goes on to state a number of constraints affecting this process. The progressive in unit 24 in (149) essentially does something very similar, only that the process in this case is more abstract and less clearly defined, as is the statement that follows it in unit 26, which, incidentally, could possibly also be labelled as a *stateOpt*. The main difference between the non-finite structures thus appears to be that the to-infinitive expresses a stronger goal-orientation than the progressive.

Examples (150) and (151) instead illustrate how references to processes can form part of the description of ongoing, sometimes longer, sequences of processes, steps, or, as in (151), the legs of a journey. In (150), s_CK uses bare infinitives that may initially create the impression that we are dealing with a series of instructions, when in fact she summarises and reflects on a sequence of steps necessary to achieve the task. Example (151) also illustrates that the main structural difference between *stateProcess* and *referProcess* is the presence (unit 71) vs. absence (unit 72) of a subject, where the absence is responsible for placing the focus on the object. And while the focussing function illustrated in (148) and (149) is definitely relevant in all dialogue contexts, the function of this speech act in sequence listings is less important in unconstrained and non-goal-oriented dialogues, which explains the differences in frequencies and rank between Switchboard and the two task-oriented domains.

The next speech act, *stateHabit*, is again not of major importance in any of the corpora, and in fact completely absent from Trains. This act is currently identified in DART if the combination NP (+ frequency-indicating Adv) + *used to* is encountered. The reason why this speech act is completely absent from Trains is probably due to the impersonal nature of the interaction in the problem-solving domain, where the focus is on the activity itself, and not the interlocutors. In the other two domains, personal habits may be more relevant, though, for instance in Trainline when callers refer to the custom of having taken a particular train in the past, etc., or in Switchboard, when speakers recount past events that involve their or other people's habits.

The speech act *referDirection* is used to either preface other information-providing units (Example (152)), moving them into the focus of attention, or to provide an elliptical response to an information-seeking act (Example (153)), in which case it tends to combine with *answer* or *elab*, although it also occurs once in combination with a *confirm* in Trains. It has no stating counterpart, and only occurs in the task-oriented dialogues in our data. This fact is not surprising, as directions are important features of both travel- and logistics-oriented domains, although of course references to direction in either form could just as well occur in unconstrained dialogue, for instance if speakers talk about their holidays or about how to get to and from specific locations.

(152) <decl n="5" sp-act="state" polarity="positive" topic="problem" mode="frag">
this is my problem <punc type="level" /></decl>
<imp n="6" sp-act="direct" polarity="positive" topic="location-transport_means-to-from" mode="decl">
<pause /> plan a round trip <pause /> from Avon <pause /> to Elmira <punc type="stop" /></imp>
<frag n="7" sp-act="**referDirection**" polarity="positive" mode="partial-frag">
<pause /> on the way out <punc type="level" /></frag>
<imp n="8" sp-act="direct" polarity="positive" topic="enum-transport_means-commodities" mode="decl">
<pause /> take <pause /> 3 boxcars of bananas <punc type="stop" /></imp>
<imp n="9" sp-act="direct" polarity="positive" topic="location-enum-transport_means-to" mode="decl">
deliver 1 <pause /> each <pause /> to Bath <pause /> Corning <pause /> and Elmira <punc type="stop" />
</imp> (d93–18.3)

(153) <decl n="5" sp-act="stateIntent" polarity="positive" topic="journey-day-time-location_Anglo-to" mode="preference1-intent-decl">
i wish to travel to London tomorrow morning <punc type="stop" /></decl>
</turn>
<turn n="3" speaker="Sandra">
<frag n="6" sp-act="reqInfo" polarity="positive" mode="query">from <punc type="query" /></frag>
</turn>
<turn n="4" speaker="caller_32">
<frag n="7" sp-act="**answer-referDirection**" polarity="positive" topic="location_Anglo-from" mode="partial-decl">from Manchester Picadilly <punc type="stop" /></frag> (trainline32)

Most of the other paired speech acts in Table 7.5 are probably more or less self-explanatory, and work in a similar way to the one described above for processes in that, generally, the referring expressions either have a focussing or elliptical responding function, while the stating ones make full, explicit statements about dates, times or durations, such as *and that's on Wednesday the 7th, it is now midnight*, or *it's about an hour and 3 quarters*. Out of these three, only references to time occur with any noteworthy frequency, rank, and dispersion. And although these occur with the same rank in Trainline and Switchboard, the higher normalised frequency and occurrence in all but one of the dialogues in

Trainline indicates the comparatively more important status there. In contrast, references and statements of duration are obviously more important in Trains, which makes sense, as each individual task there does need to be completed with certain time limits. On the other hand, the almost complete absence of duration-related information also implies that the Trainline callers appear to be less concerned with how long their journeys might take than the modalities of transport.

It may at first appear odd that references to places (*referPlace*) have a relatively low occurrence in Trains, as the task is to move goods around between places. However, this low incidence is misleading, as references to directions, which are more common in Trains, actually 'subsume' references to places, often even multiple ones, as in *<pause /> from Dansville to Bath <pause /> <overlap pos="start" /> by <overlap pos="end" /> <overlap pos="start" /> Avon <overlap pos="end" /> <punc type="stop" />* (d92a-1.2). In contrast, references to places frequently occur in answers detailing destinations or addresses, responses to requests for directives providing destinations, as well as confirmations. They can thus, again, occur in different stages of the dialogue, which makes them more important, and hence also a slightly better indicator of the domain.

The absence of date-related references or statements from Switchboard simply indicates that no date-related topics are discussed in our selection from that corpus, while all dialogues in Trains always happen in the 'here-and-now', so references to future or past dates are not necessary in order to fulfil the task.

Direct references to (other) people (*referPerson*) are generally not important in the context of Trainline or Trains. In Trainline, it is more important to establish the number of travellers, rather than who they might be, which can be established via general references or enumerations, and this explains the absence of this category there, whereas in Trains the whole interaction only revolves around the task and the interlocutors, and other people who could be referred to are in fact not involved at all. In Switchboard, though, there are occasional references to other people, as in e.g. Example (154).

(154) <decl n="147" sp-act="referPerson" polarity="positive" topic="family-relationships" mode="reason-decl">
<comment type="restart" /> one of my friend's parents who went in because she had largely lost her abilities <punc type="stop" />
</decl> (sw_0014_4619)

The final three referring speech acts *referThing, referAct*, and *referHow*, are probably less intuitive to understand, and still highly tentative, so that I will not discuss them in depth here. They all represent clefting constructions that affect the topic focus. The first one of these, *referThing*, again has a focussing function, often directing the focus towards something that represents an ability or option, as in Example (155).

(155) <decl n="60" sp-act="answer-stateCondition" polarity="positive" topic="health" mode="condition-decl">
and it depends on how <comment type="restart" /> how sick the person is too <punc type="stop" /></decl>
<decl n="61" sp-act="elab-referThing" polarity="positive" mode="decl">
what you're capable of <punc type="stop" /></decl> (sw_0006_4108)

referAct, in essence, is very similar to *referProcess* in containing a progressive form, the main difference being that the progressive form in this speech act is always preceded by the word *by*, thereby emphasising the action expressed through the progressive and its potential consequences, which can be seen in (156).

(156) <frag n="40" sp-act="referAct" polarity="positive" mode="partial-frag">
by putting him in prison for life <punc type="level" /></frag>
<decl n="41" sp-act="expressPossibility" polarity="positive" topic="health" mode="exists-predict-decl">
there is still a possibility that he will get out again <punc type="stop" />
</decl> (sw_0028_4133)

The final one, *referHow*, does not even occur in our data, but is defined in the DART scheme as occurring in a declarative structure that starts with *how* or *however*, e.g. as in *how (exactly) one might want to do this is unclear*. Unlike *referAct*, which focusses on the action, *referHow* shifts the focus onto the manner in which something is achieved.

The next sub-category to be discussed, that of elaborating speech acts, only has two members, *elab* and *spell*.

Table 7.6 Elaborating speech acts

Speech act	Trl NFreqs	Trl Rank	Trl Docs	Trs NFreqs	Trs Rank	Trs Docs	SWBD NFreqs	SWBD Rank	SWBD Docs
elab	122.0	12	33	53.0	23	25	45.0	20	29
spell	17.0	43	8	–	–	–	–	–	–

Elaborations – marked *elab* – are 'add-ons' providing additional information to a response to any potential form of request, apart from requests for directives. The name is loosely based on the category of elaborations in Mann & Thompson (1987), but is limited to single units in DART.

(157) <q-yn n="76" sp-act="reqInfo" polarity="positive" topic="transport_means-enum" mode="contrast-closed-query">
<pause /> but can you have <pause /> 2 <pause /> things on <pause /> the same train <punc type="query" />
</q-yn>
</turn>
<turn n="23" speaker="s_AM">
<yes n="77" sp-act="answer-acknowledge">yes <punc type="stop" /></yes>
<decl n="78" sp-act="**elab**-expressPossibility" polarity="positive" topic="enum" mode="poss2-decl">
you can have up to 3 <pause /> loaded <punc type="stop" /></decl>
(d93–16.1)

The speech act label *elab* thus 'flags' the fact that there is some additional information present that is very likely to be of further relevance to the current topic under discussion, especially if the response only consists in an acknowledgement, as in Example (157). However, *elab* itself does not provide any exact detail as to the local functional nature of the unit. Such detail is then usually provided in the form an additional, more 'information-oriented', speech act attribute, in our case that of *expressPossibility*.

Spelling out names or other details to ensure that they are understood and taken down correctly is indicated vial the *spell* speech act label. This act is only relevant in taking down personal information for the bookings in Trainline, and does not occur in the other two corpora, although of course it would definitely be possible for someone to spell their own or someone else's name in unconstrained dialogues, too. In Trains, all the information about potential destinations is already known, so this type of clarification of information is not necessary. This speech act is classed as a form of elaboration here because it essentially repeats information already provided, only in a different form.

(158) <frag n="98" sp-act="reqInfo" polarity="positive" topic="name" mode="query">
and your initial <punc type="query" /></frag>
</turn>
<turn n="52" speaker="caller_23">
<frag n="99" sp-act="answer-**spell**" polarity="positive" topic="spell" mode="decl-disflu">
<anonym type="letter" /> <anonym type="letter" /> <punc type="stop" />
</frag> (trainline23)

When spellings co-occur with enumerations, they are classed as *refer*s, though, e.g. in postcodes.

The next sub-group is that of explaining speech acts. As the name implies, these generally provide some form of explanation regarding the content of the previous unit(s), or for the necessity of taking certain actions.

Table 7.7 Explaining speech acts

Speech act	**Trl NFreqs**	**Trl Rank**	**Trl Docs**	**Trs NFreqs**	**Trs Rank**	**Trs Docs**	**SWBD NFreqs**	**SWBD Rank**	**SWBD Docs**
stateReason	18.0	41	19	23.0	35	16	78.0	12	30
referReason	–	–	–	1.0	71	1	2.0	69	3
explain	–	–	–	–	–	–	2.0	69	3
conclude	–	–	–	–	–	–	–	–	–

Essentially, *stateReason* and *referReason* again represent the same speech act content-wise, but the latter only occurs in fragments, often in non-finite or abandoned verbless units.

(159) <decl n="17" sp-act="predict" polarity="positive" topic="transport_means-enum-location-to-duration" mode="decl">
to get the engine and the boxcars to Corning will take you 4 hours <punc type="stop" /></decl>
<decl n="18" sp-act="**stateReason**" polarity="positive" topic="location-enum-from-to-time-duration" mode="reason-decl">
<pause /> cause it's 1 hour from Dansville to Corning <punc type="stop" />
</decl> (d93–16.1)

(160) <decl n="28" sp-act="expressOpinion-abandon" status="abandon" polarity="positive" mode="opinion">
i think the biggest frustration is <punc type="incomplete" /></decl>
<frag n="29" sp-act="**referReason**" polarity="positive" mode="reason-partial-frag">
because of the cost of insurance going up every year <punc type="level" />
</frag>
<decl n="30" sp-act="stateConstraint" polarity="positive" mode="alternative-decl">
our district has had to alter <comment type="restart" /> uh you know change companies or <punc type="stop" /></decl> (sw_0059_4028)

Example (159) represents an illustration of a straightforward statement of reason that simply provides an explanation as to why the overall duration is supposed to be four hours, as the prior leg of the journey has already been established as being three hours long in the preceding context. Example (160), on the other hand, begins with an expression of opinion whose completion the speaker interrupts to insert an explanation they apparently assume necessary for the complete understanding of the circumstances, and finally completes the sequence by stating a constraint. In other words, the use of *referReason* here represents a strategic device to establish a clearer context.

The speech act *explain* is different from the above two acts, as it does not state the reason as directly through using conjunctions like *because, since*, or *due to*, but rather with 'hindsight', as the following examples from Switchboard, where this act exclusively occurs, will hopefully illustrate.

(161) <decl n="69" sp-act="**explain**" polarity="positive" mode="counter_expectation-explain-decl">
i was actually talking about businesses purchasing something mail order and then having <comment type="restart" /> having to pay sales tax on it <punc type="stop" /></decl> (sw_0035_4372)

(162) <decl n="118" sp-act="state" polarity="positive" mode="frag">
it's where it makes the least noise <punc type="level" /></decl>
<decl n="119" sp-act="**explain**" polarity="positive" mode="decl">
which means it's <backchannel content="oh" /> often the lowest support type roles <punc type="stop" />
</decl> (sw_0042_4060)

In other words, *explain* is not simply used to provide a reason, but in order to clarify the exact meaning of something that already represents part of the common ground. The final speech act in Table 7.7, *conclude*, does not in fact occur in any of the three corpora, but represents the counterpart to *initConclusion*, described in 7.2 above. While *initConclusion* represents a DM that prefaces an information-providing unit that is not clearly marked as a logical conclusion, *conclude* marks an information-providing unit that has embedded markers of conclusion, such as *hence, in that way*, etc.

All members of the next sub-category express different degrees of awareness or knowledge of circumstances surrounding the interaction. In contrast to the more general, 'fact-oriented', *stateX* labels, we here encounter the *expressX* labels, which reflect the fact that the information provided through these speech acts usually tends to be based more strongly on the personal judgements or attitudes of the speaker. Thus, some *expressX* acts may be vaguely similar in nature to Searle's *expressives* (Searle 1979: 356) or Austin's *behabitives* (Austin 1962: 81), although these scholars mainly apply their labels to formulaic acts expressing some form of speaker attitude.

For *expressAwareness*, most of the expressions this label is applied to contain the expression *i know/understand/am aware*, or *as far as i/we know*, etc. Its negative counterpart, *expressNonAwareness*, is mainly expressed through *i/we don't know,*

Table 7.8 Awareness-indicating speech acts

Speech act	Trl NFreqs	Trl Rank	Trl Docs	Trs NFreqs	Trs Rank	Trs Docs	SWBD NFreqs	SWBD Rank	SWBD Docs
express Awareness	0.5	72	1	–	–	–	18.5	32	14
expressNon-Awareness	5.0	59	8	7.0	53	6	40.0	25	27
expressUn-certainty	0.5	72	1	1.5	63	1	10.5	40	9

i don't remember, i'm not aware of/that, etc. The speech act *expressUncertainty* typically represents a state somewhere in-between the other two, and is realised through constructs like *i'm/i was (just) wondering if/whether, i'm not sure*, etc.

None of these three speech acts exhibits any remarkably high frequencies in any of the corpora, but it is interesting to note that there are no incidences of *express-Awareness* at all in Trains, and almost none in Trainline, but that speakers refer to their knowledge or awareness of people or issues somewhat more frequently in Switchboard. In contrast, indicating one's lack of awareness or uncertainty about something does occur rather more frequently, especially again in Switchboard, although this is partly also used as a feature that indicates a certain kind of indecisiveness, as in Examples (163) and (164).

(163) <frag n="18" sp-act="expressOpinion-abandon" status="abandon" polarity="positive" mode="opinion">
i guess <punc type="incomplete" /></frag>
<decl n="19" sp-act="**expressNonAwareness**" polarity="negative" mode="nonawareness-decl">
<pause /> i don't know <punc type="stop" /></decl>
<decl n="20" sp-act="expressPossibility" polarity="positive" topic="location" mode="poss1-frag-disflu">
i <pause /> we can start at Bath <punc type="level" /></decl> (d93–17.2)

(164) <decl n="136" sp-act="referTime" polarity="positive" topic="time" mode="frag">
when you have kids <punc type="level" /></decl>
<q-yn n="137" sp-act="reqInfo" polarity="positive" topic="health" mode="closed-query">
will you work <punc type="query" /></q-yn>
<q-yn n="138" sp-act="reqInfo-abandon" status="interrupted" polarity="positive" mode="closed">
do you <punc type="incomplete" /></q-yn>
</turn>

<turn n="58" speaker="1519">
<decl n="139" sp-act="answer-**expressNonAwareness**" polarity="negative" mode="nonawareness-decl">
i don't know <punc type="stop" /></decl>
</turn>
<turn n="59" speaker="1632">
<frag n="140" sp-act="reqInfo" polarity="positive" mode="query">
work now <punc type="query" /></frag>
</turn>
<turn n="60" speaker="1519">
<decl n="141" sp-act="answer-state" polarity="positive" mode="decl">
that's something i've considered <punc type="stop" /></decl>
(sw_0001_4325)

Somewhat related to indicating awareness is hypothesising about things, which is what the speech acts in the next sub-category reflect to varying degrees.

Table 7.9 Hypothesising speech acts

Speech act	Trl NFreqs	Trl Rank	Trl Docs	Trs NFreqs	Trs Rank	Trs Docs	SWBD NFreqs	SWBD Rank	SWBD Docs
predict	59.0	24	30	182.5	7	33	44.0	22	24
expressProbability	–	–	–	7.0	53	7	10.0	43	8
referProbability	–	–	–	–	–	–	1.5	76	2
expressImProbability	–	–	–	1.0	71	1	3.5	59	6
expressPossibility	35.0	31	23	38.5	27	23	38.0	26	27
referPossibility	–	–	–	–	–	–	2.5	64	3
expressImPossibility	6.5	54	5	13.0	42	9	21.0	30	14

The strongest form of hypothesising is *predict*, where speakers clearly indicate their expectations as to whether an event will or will not occur in the future. As rank, dispersion, and frequency all indicate, this speech act is very important in Trains, as the planning requires the interlocutors to make predictions about arrival times and durations of the stages involved in the planning process in order to ascertain whether the task can be completed within the time limit. A good example of this can be seen in (165), where the speaker first predicts a certain duration and then concludes that it will be impossible to achieve the (sub-)task based on this.

(165) <decl n="173" sp-act="**predict**" polarity="positive" topic="enum-to-**duration**" mode="predict-frag">
it'll take 2 hours to go back to Corning <punc type="level" /></decl>

<decl n="174" sp-act="**predict**" polarity="negative" topic="time"
mode="predict-decl">
so there won't be time <punc type="stop" /></decl> (d92a-1.2)

(166) <decl n="35" sp-act="**predict**" polarity="positive" topic="fare-enum-**availability**" mode="predict-decl">
the cheapest fare that's going to be available is the Saver Return at 33
pounds 50 now <punc type="stop" />
</decl> (trainline05)

(167) <decl n="53" sp-act="**predict**" polarity="positive" mode="decl">
that would make a bigger market <punc type="stop" /></decl>
<decl n="54" sp-act="**predict**" polarity="positive" topic="leisure"
mode="decl">
we'd have uh more recreation <punc type="stop" /></decl>
<decl n="55" sp-act="**predict**" polarity="positive" topic="time" mode="decl">
people would have time to do it <punc type="stop" /></decl>
<decl n="56" sp-act="**predict**" polarity="positive" mode="predict-decl">
there would be less of a stress level <punc type="stop" /></decl>
<decl n="57" sp-act="**predict**" polarity="positive" topic="crime" mode="decl">
we'd have <comment type="restart" /> we'd have less crime <punc
type="stop" />
</decl> (sw_0020_4109)

For both Trainline and Switchboard, making such predictions is clearly also not unimportant, yet not quite as important as for Trains, as the rankings are not even in the top 20. And although one would probably assume that in Trainline the predictions would mainly revolve around expected arrival times and durations, too, this is actually not the case, as Example (166) illustrates. Instead, most of the predictions either revolve around the assumed availability of certain ticketing options, what will happen once the booking is completed, and/or when the caller can expect to have the tickets sent out or delivered to them. As is to be expected, in Switchboard the predictions relate to less clearly defined objectives or options, but instead focus more on the effects of hypothetical actions to be taken on everyday life, as Example (167) demonstrates, where the potential effects of establishing a 32-hour working week are predicted.

The probable, or improbable effects of events happening are evidently of no concern in Trainline, as the statistics for *expressProbability, referProbability*, and *expressImProbability* show. Here, *referProbability* is simply again the fragmentary or elliptical form of *expressProbability*, while *expressImProbability* constitutes its negative counterpart. This 'grey zone' on the cline between hard-and-fast predictions and expressions of (im)possibility also appears to be of relatively little importance in Trains, and only slightly more so in Switchboard.

The label *expressPossibility*, together with its elliptical and negative counterparts, *referPossibility* and *expressImPossibility*, in fact captures what would normally be discussed in treatments of modality as both *possibility* and *ability*. In contrast to *stateOpt*, which mainly covered aspects of options not within the control of the subject of the unit, such as availability, this speech act represents an epistemic judgement regarding feasibility or ability, but frequently also with a built-in element of choice or control on the part of the subject, apart from in units expressing ability.

(168) <q-wh n="7" sp-act="reqInfo" polarity="positive" topic="transport_means-number" mode="closed-query"><pause /> how many um <pause /> boxcars can 1 engine take <punc type="query" /></q-wh>
</turn>
<turn n="3" speaker="s_NB">
<dm n="8" sp-act="hesitate">um</dm>
<decl n="9" sp-act="answer-stateCondition" polarity="positive" topic="transport_means" mode="condition-frag"><pause /> unl… if the boxcars are unloaded <punc type="level" /></decl>
<decl n="10" sp-act="elab-**expressPossibility**" polarity="positive" mode="poss3-decl">
as many as it <pause /> as it <pause /> can <punc type="stop" /></decl>
<decl n="11" sp-act="stateCondition" polarity="positive" mode="condition-frag">
<pause /> and if they are loaded <punc type="level" /></decl>
<decl n="12" sp-act="**expressPossibility**" polarity="positive" topic="transport_means-enum" mode="poss3-decl">it can carry 3 <punc type="stop" />
</decl> (d93–14.3)

(169) <decl n="154" sp-act="**expressPossibility**" polarity="positive" topic="journey-fare-departure" mode="poss2-frag">you can leave the return journey open <punc type="level" /></decl>
<decl n="155" sp-act="**expressPossibility**" polarity="positive" topic="time" mode="poss2-decl">
so you can return back within in a calendar month <punc type="stop" /></decl> (trainline34)

(170) <decl n="53" sp-act="expressOpinion-**expressPossibility**" polarity="positive" topic="time-week-cars" mode="condition-poss-opinion-decl">
i guess maybe this week someone may start getting them if the truck that picks them up is outfitted in time <punc type="stop" /></decl>
(sw_0055_3156)

Example (168) illustrates what is essentially two instances of (cap)ability that are being ex-pressed (units 10 and 12) in response to a question pertaining to this (unit 7). However, despite the fact that one would normally consider ability a relatively fixed property, it is quite obvious here that it may well vary depending on the conditions pertaining to a situation and thus may also require a certain element of judgement on the part of the speaker.

The next Example (169) shows the element of control on the part of the subject of the unit referred to above, while Example (170) indicates that expressions of possibility can sometimes go hand in hand with an even stronger element of judgement that is clearly marked as such in the form of an expression of opinion (*expressOpinion*). Please note that, unlike in Example (168), here the conditional clause is not marked as a separate unit as it is clearly not focussed and can therefore be considered subordinate to the main function of the unit. This issue was already discussed in more detail in Section 4.1.

Regarding the statistics related to expressions of (im)possibility in our three sets of data, there appear to be no major differences in the numbers of statements of or references to possibility, as well as their rank and dispersion. For *expressImPossibility*, however, there seems to be a notable decrease of such expressions from Switchboard, via Trains, to Trainline. The difference between Trains and Trainline can be explained through the fact that the interlocutors there tend to refer more frequently to both restrictions imposed by the capability of the means of transport, and how these means may be deployed, along with impossibilities arising from errors in their own planning in interpreting the former, while in Trainline, most of the expressions of impossibility result from a relatively low number of ticketing or booking restrictions. The higher incidence of *expressImpossibility* in Switchboard in comparison to the other two corpora is probably related to the wider range of options for making reference to something that is or was impossible. Not only are there more abstract moral judgments, as we already saw earlier, but the narrative nature of the dialogues also allows speakers to express facts related to the past, whereas the task-oriented dialogues only deal with matters that are impossible at the present moment.

The final sub-category of information-providing speech acts is that of volitional ones. I do not here use the term *volitional* in the more traditional sense as indicating intentionality (cf. Bussmann 1996: 1273), but instead to refer to (hypothetical) desires expressed by speakers.

Table 7.10 Volitional speech acts

Speech act	Trl NFreqs	Trl Rank	Trl Docs	Trs NFreqs	Trs Rank	Trs Docs	SWBD NFreqs	SWBD Rank	SWBD Docs
expressHope	4.0	62	5	–	–	–	4.0	56	6
expressWish	0.5	72	1	–	–	–	2.0	69	3

The two speech acts are triggered by slightly different modes that affect the inferencing process in DART, partly based on the occurrence of expressions of hope vs. wishing. In practice, however, it is very difficult to distinguish the purpose of the resulting annotations in terms of the strength or purpose of the desire expressed there. Until further research using more examples can yield better rules that allow for a clearer distinction between the two, it is currently probably best to treat them as functionally equivalent. One example for *expressHope* is (171).

(171) <decl n="111" sp-act="phatic" polarity="negative" mode="frag">
<pause length="2s" /> you never know <punc type="level" /></decl>
<decl n="112" sp-act="state" polarity="positive" mode="decl">
the new millenium's coming in soon <punc type="stop" /></decl>
<decl n="113" sp-act="**expressHope**" polarity="positive" mode="hope-decl">
so <backchannel content="laughter" /> hopefully we will get a new system
<punc type="stop" />
</decl> (trainline09)

One thing that is interesting to note, though, is that such expressions are completely absent from the Trains data. Although it is of course quite conceivable that an interlocutor in a problem-solving task might express e.g. a desire for all or part of a plan to work out, this domain, with its focus on results, and also its impersonal nature, makes this rather unlikely. In contrast, much of the discussion in Trainline centres around the personal preferences and wishes of the callers, making such expressions more likely to occur there, while unconstrained dialogue tends to be of a more personal nature, anyway, so that wishes and desires of the interlocutors being voiced should not appear uncommon. Interestingly, though, the example from Trainline above comes from a turn uttered by Sandra, which demonstrates that the use of volitional expressions need not be limited to people who, by their very roles, have more right to use them.

7.4 Negotiative speech acts

The next major category to discuss is that of units that are somewhat akin to the engaging (non-)cohesive ones we discussed earlier in that they are similarly interactive, and also frequently short, but, unlike the former, which were essentially fact-oriented or purely signalled different stages, negotiative speech acts also embody a certain degree of commitment to the goals or aims of the ongoing interaction. They also predominantly occur in short units that tend to be DMs, yeses or nos, and commonly also occur in pairings or triplets where one has positive and the other one or two have negative force. Table 7.11 shows the different options that currently exist in the DART scheme.

Table 7.11 Negotiative speech acts

Speech act	Trl NFreqs	Trl Rank	Trl Docs	Trs NFreqs	Trs Rank	Trs Docs	SWBD NFreqs	SWBD Rank	SWBD Docs
accept	10.5	50	12	47.0	25	34	6.0	52	7
refuse	–	–	–	–	–	–	–	–	–
reject	8.0	52	9	7.0	53	5	–	–	–
approve	62.5	21	26	25.0	33	17	26.0	28	17
disapprove	–	–	–	–	–	–	0.5	85	1
agree	32.5	32	20	9.5	49	8	25.0	29	21
disagree	–	–	–	–	–	–	2.0	69	2
contradict	–	–	–	1.5	63	2	–	–	–
correct	11.5	47	12	13.0	42	12	–	–	–
retract	–	–	–	–	–	–	–	–	–
admit	–	–	–	–	–	–	0.5	85	1
rejectSelf	2.5	65	3	10.5	48	9	0.5	85	1
correctSelf	6.0	57	9	5.0	57	6	–	–	–

Essentially, the first three positive speech acts form a kind of hierarchy, ranging from explicit acceptance (*accept*), via registering one's approval (*approve*), to expressing agreement (*agree*). I will discuss each of these in turn now, trying to point out their characteristics by providing suitable examples, and showing in which environments they predominantly occur.

The speech act *accept* responds in an active positive way to an offer, some form of proposal, or suggestion (172), or even a request for a directive (173).

(172) <q-yn n="56" sp-act="suggest" polarity="positive" mode="closed-decl">
shall we do that <punc type="query" /></q-yn>
</turn>
<turn n="28" speaker="u_DT">
<yes n="57" sp-act="**accept**">yes <punc type="stop" /></yes> (d92a-3.1)

(173) <q-yn n="68" sp-act="reqDirect" polarity="positive" topic="journey-booking" mode="preference2-closed-query">do you want me to book this ticket <punc type="query" /></q-yn>
<yes n="69" sp-act="direct-**accept**">yeah <punc type="stop" /></yes>
(trainline03)

As Example (172) shows, acceptance may also go hand-in-hand with another speech act. If an offer is responded to negatively, we have an instance of *refuse*, which, however, does not occur in our data. In contrast, a negative response to a proposal or suggestion constitutes a *reject* speech act.

(174) <decl n="61" sp-act="reqConfirm" polarity="negative" topic="booking" mode="query">
you don't want to book at all <punc type="query" /></decl>
</turn>
<turn n="36" speaker="caller_02">
<no n="62" sp-act="confirm-**reject**">no thank you <punc type="stop" />
</no> (trainline02)

Example (174) also illustrates that a proposal or suggestion may well be expressed indirectly, as in this case through a request for confirmation.

As stated previously, refusals do not occur in the corpus data, but this is perhaps not surprising as the overwhelming majority of offers occurs in Trains, where, however, this exclusively occurs in the form of the playbook opening *can i help you*, and it would be extremely surprising if this offer were ever refused by the speaker mimicking the system user. Rejections are also completely absent from Switchboard, but, as the number of offers there is also extremely low, and the interaction tends to be more narrative than collaborative, this is perhaps not surprising. The relatively low number of rejections in Trainline and Trains is probably due to the collaborative behaviour, where the interlocutors in fact make an effort to collaborate and therefore tend to provide more constructive suggestions that are also less likely to be rejected. This collaborative conduct is most clearly observable in Trains, where the number of acceptances is not only far higher than in the other two corpora, but also the rank of this speech act.

The relatively low incidence of acceptances in Trainline may initially be somewhat surprising, as Sandra is certainly making every effort to provide good recommendations to her callers, but this is at least to some extent balanced by the higher number of approvals, as the British callers in this corpus frequently tend to come across as more enthusiastic about the choices Sandra provides them with. Thus, while the American speakers in Trains and Switchboard tend to restrict their approvals to expressions like *that's okay/good/great*, the British callers frequently express their approval or acceptance via more effusive epithets, such as *brilliant, excellent, lovely*, or *wonderful*. Thus, although both *accept* and *approve* can both be used to signal the acceptance of offers, suggestions, or proposals, the main difference between the two speech acts lies in the presence or absence of an expression of the degree of positive rating on the part of the speaker.

Explicit disapproval (*disapprove*) is extremely rare, and only occurs once in Switchboard (see Example (175)).

(175) <decl n="67" sp-act="state-abandon" status="abandon" polarity="positive">
that does <punc type="incomplete" /></decl>
<decl n="68" sp-act="state-abandon" status="abandon" polarity="negative">

<comment type="restart" /> that's something you don't <punc type="incomplete" /></decl>
<frag n="69" sp-act="disapprove" polarity="negative" mode="decl">
<comment type="restart" /> doesn't even sound good loud really <punc type="stop" /></frag> (sw_0063_4334)

Agreement (*agree*) may be signalled in a variety of different ways, ranging from simple adverbial DMs, such as *sure, exactly, definitely*, to slightly longer declaratives like *that's it/correct/true, i('ll) agree (with you/that), i'm with you, i do/me too, it is/was* (Example (176)), or even negative-polarity expressions, such as *no problem*, or even a single *no(pe)* (Example (177)) if the proposition of the preceding unit also had negative polarity.

(176) <dm n="271" sp-act="exclaim" mode="exclaim">
<overlap pos="start" /> gosh <punc type="level" /></dm>
<exclam n="272" sp-act="exclaim">
was that wonderful <overlap pos="end" /> <punc type="stop" /></exclam>
</turn>
<turn n="68" speaker="1313">
<frag n="272" sp-act="**agree**" polarity="positive" mode="decl">
it was <punc type="stop" /></frag> (sw_0069_3144)

(177) <turn n="54" speaker="u_ML">
<decl n="144" sp-act="referDuration" polarity="**negative**" topic="time" mode="decl">
that takes no time <punc type="stop" /></decl>
</turn>
<turn n="55" speaker="s_TG">
<no n="145" sp-act="**agree**">nope <punc type="stop" /></no> (d93–18.3)

As Example (176) shows, it is even possible to agree with the content or proposition of exclamatory units. Frequencies, ranks and dispersions indicate that expressing agreement is not amongst the most important speech acts in any of the corpora, while expressing disagreement (*disagree*) only occurs in two of the documents in Switchboard.

Overall, the prevalence of positive-polarity elements of acceptance, approvals, and agreements, as opposed to their negative counterparts, would seem to indicate that all three corpora do not exhibit strongly argumentative characteristics. The one small exception to this can be seen in the two files in Switchboard that feature disagreements, which both involve discussions about legal issues, the use of trial by jury and the necessity of the death penalty.

The fact that disagreements are completely absent from Trains may perhaps appear a little odd at first, as one would assume that there cannot usually be complete agreement about all the tasks involved in planning processes. However, the pure absence of explicit disagreements does not mean that the interlocutors always have to, and do indeed, accept everything the other party says as true. As disagreements are more about positions or assent, but generally less about facts, there are of course other options to voice some form of non-assent. In Trains, these occur in two forms, contradictions (*contradict*) and corrections (*correct*), while in Trainline only the latter occurs. Yet, the contradictions that do occur in Trains are quite rare, and when they do occur, they are always used to overtly, and explicitly, mark an upcoming correction, as in Example (178).

(178) <frag n="169" sp-act="referTime" polarity="positive" topic="time-enum" mode="partial-frag">
at 3 a.m. <punc type="level" /></frag>
<decl n="170" sp-act="predict" polarity="positive" topic="to" mode="predict-decl">
it would get to Dansville <punc type="stop" /></decl>
<decl n="171" sp-act="predict" polarity="positive" topic="number-to" mode="decl">
it would pick up the 3 boxcars and then go back to <overlap pos="start" /> Avon <overlap pos="end" /> <punc type="stop" /></decl>
</turn>
<turn n="72" speaker="u_PH">
<no n="172" sp-act="**contradict**">
<overlap pos="start" /> n… <pause /> <overlap pos="end" /> no <punc type="stop" /></no>
<frag n="173" sp-act="predict-abandon" status="abandon" polarity="positive" mode="partial">
it would <punc type="incomplete" /></frag>
<dm n="174" sp-act="hesitate">uh</dm>
<decl n="175" sp-act="**correct**-predict" polarity="positive" topic="enum-to" mode="predict-decl">
<pause /> it will be carrying 3 boxes <correction orig="box" /> of bananas to Dansville <punc type="stop" /></decl> (d93–11.2)

In Example (178), s_CB first summarises her assumptions about one of the stages of transport (units 169–171). Speaker u_PH, however, recognising that she has made a mistake, then contradicts her in unit 172, produces a false start in 173, and finally provides his correction in unit 175. Yet, as Table 7.11 clearly shows, not all corrections in Trains are explicitly marked like this, and none of the ones in Trainline are.

(179) <decl n="60" sp-act="stateTime" polarity="positive" topic="location-time-enum" mode="decl">
it's 3 a.m. <overlap pos="start" /> now in Corning <overlap pos="end" /> <punc type="stop" />

</decl>
</turn>
<turn n="25" speaker="s_SB">
<frag n="61" sp-act="state-abandon" status="abandon" polarity="positive" mode="abandon">
<overlap pos="start" /> it's w… <overlap pos="end" />
<punc type="incomplete" />
</frag>
<decl n="62" sp-act="**correct**-stateTime" polarity="positive" topic="enum-time-location" mode="**counter_expectation**-decl">
it's **actually** 2 a.m. <overlap pos="start" /> in um <overlap pos="end" /> Corning <punc type="stop" />
</decl> (d93–12.1)

(180) <turn n="99" speaker="Sandra">
<dm n="215" sp-act="init">now <punc type="level" /></dm>
<decl n="216" sp-act="stateProcess" polarity="positive" topic="journey-day-from-location_Anglo" mode="decl">you're travelling tomorrow from Stockport <punc type="stop" /></decl>
</turn>
<turn n="100" speaker="caller_28">
<frag n="217" sp-act="reqConfirm" polarity="positive" mode="query">
10:30 <punc type="query" /></frag>
</turn>
<turn n="101" speaker="Sandra">
<frag n="218" sp-act="**correct**-referTime" polarity="positive" mode="decl">
10:38 <punc type="stop" /></frag> (trainline28)

Example (179) demonstrates that using a contradiction is not the only way of marking a correction explicitly, though. Here, s_SB uses the marker of counter-expectation *actually* to indicate that what she is saying is somehow at odds with u_PH's assumptions. Unlike the use of this marker in the 'prefacing' DM *initCounterExp* we encountered earlier, this inlined use is captured through the mode attribute counter_expectation. Example (180), on the other hand, illustrates that it is quite possible to provide a correction without any explicit marking at all, and, in fact, in absolutely minimal form, as in the reference to time in unit 218. As will hopefully have become apparent from Examples (178)–(180), the speech act *correct*, which indicates a pragmatic relevance of the unit across turns, needs to be accompanied by another speech-act label that again reflects the precise nature of the type of correction in the form of a 'local' speech act.

Apart from not assenting to what the interlocutor has said, a speaker can also signal non-agreement using suggestions or facts they themselves have introduced into the common ground. Doing so can essentially take four forms, retracting (*retract*) something that has been stated, admitting that something is/was wrong (*admit*), rejecting one's own ideas or suggestions (*rejectSelf*), and correcting one's own utterance (*correctSelf*). Unlike the forms that mainly provided responses to something the interlocutor had said in one or more prior turns, these forms of self-dissent often occur within the same turn. The first of these forms, *retract*, does not occur in our data, and *admit* only once in Switchboard. However, as before, we would probably rather expect such speech act to occur in argumentative interactions. Self-rejections, on the other hand, are generally a feature inherent to interactions that involve (logistical) planning, which is why it is not surprising that they have the highest incidence in Trains, but also do occur in Trainline. Correcting oneself, on the other hand, is more related to utterance planning, and may therefore theoretically occur in any kind of verbal interaction, even to some extent in the performance of skilled speakers, although we would perhaps expect the highest incidence in learner speech or that of verbally less skilled native speakers.

7.5 Suggesting or commitment-indicating speech acts

Obviously, in dialogues, speakers not only request or provide information, or negotiate meaning, but there are also situations where an even stronger element of co-operative behaviour or voluntary personal involvement is required of the participants. In other words, what is needed of them is that they make some kind of commitment, verbally indicating that they are prepared to undertake a certain course of action, or accept a specific responsibility. The options for speech acts that express this type of commitment, as they are defined in the DART system, are provided in the following table.

Table 7.12 Suggesting or commitment-indicating speech acts

Speech act	Trl NFreqs	Trl Rank	Trl Docs	Trs NFreqs	Trs Rank	Trs Docs	SWBD NFreqs	SWBD Rank	SWBD Docs
suggest	13.5	46	14	61.0	21	22	29.5	27	19
offer	0.5	72	1	29.0	31	34	0.5	85	1
promise	–	–	–	–	–	–	–	–	–
perform	–	–	–	–	–	–	–	–	–
direct	131.0	10	35	14.5	41	10	13.0	37	11
reqDirect	69.0	20	35	1.0	71	1	1.5	76	2
stateIntent	108.0	15	34	89.0	16	25	13.0	37	9
state-Attempt	–	–	–	–	–	–	10.0	43	8

The discussion of the above listed speech acts will start with the speech act *suggest*, as this may also involve a potential commitment on the part of both or all interlocutors. As can be seen in the examples listed below, *suggest* probably represents the most 'syntactically versatile' of all speech acts, as it can be expressed through a large variety of syntactic categories. In terms of traditional approaches to speech-act theory, most of these expressions used to perform this speech act would probably need to be considered indirect because they do not conform with the default expectations for the syntactic categories used to perform them.

(181) <**q-wh** n="43" sp-act="**suggest**" polarity="positive" mode="closed-query" >
what about <pause /> Avon <punc type="query" /></q-wh>
(trains d92a-1.2)

(182) <**q-yn** n="56" sp-act="**suggest**" polarity="positive" mode="closed-decl" >
shall we do that <punc type="query" /></q-yn> (trains d92a-3.1)

(183) <**imp** n="40" sp-act="**suggest**" polarity="positive" topic="enum-time" mode="decl">
let's say <pause /> half 3 in the afternoon <punc type="stop" /></imp>
(trainline02)

(184) <**decl** n="38" sp-act="suggest" polarity="positive" topic="time" mode="suggest-decl">
<pause /> and <pause /> we should leave <pause /> at midnight
<punc type="stop" />
</decl> (d93–14.1)

(185) <**frag** n="77" sp-act="refer" polarity="positive" mode="decl">
dessert <punc type="stop" /></frag>
<frag n="78" sp-act="**suggest**" polarity="positive" mode="query">
something sweet <punc type="query" /></frag> (sw_0057_3506)

Examples (181) and (182) take a question form and would therefore by default be assumed to be requests for information, or maybe, as we shall see soon, potentially also requests for directives, while imperatives, as in (183), are generally associated with uttering directives. Example (184) probably comes closest to what we might expect a prototypical suggestion to look, while (185) clearly demonstrates that, given that the topic is in focus (unit 77), even fragmentary units consisting of single NPs may function as suggestions if uttered with the appropriate intonation.

As the examples above indicate, suggestions are frequently designed to encourage two-sided commitment, especially if they contain a subject or object in the plural. If the subject or object is in the singular, though, we may either have a one-sided commitment expressed, or simply an option available to a 2nd or 3rd person subject.

(186) <turn n="92" speaker="caller_12">
<q-yn n="196" sp-act="reqInfo" polarity="positive" mode="tag-closed-query">
are you going to let me do it <punc type="query" /></q-yn>
</turn>
<turn n="93" speaker="Sandra">
<decl n="197" sp-act="answer-suggest" polarity="positive" topic="booking" mode="suggest-poss1-decl">
<pause /> i could book it just now <punc type="stop" /></decl> (trainline12)

(187) <decl n="170" sp-act="suggest" polarity="positive" mode="suggest-poss2-decl">
you could go back <punc type="stop" />
</decl> (d92a-1.2)

Of course, a one-sided 1st person commitment, such as in (187), may also be interpreted as an *offer*, but it is only safe to mark this automatically on a unit if a mode benefit2, possibly also in conjunction with a possibility (poss1), is clearly signalled for the other party (Example (188)), or an explicit performative is used, as in Example (189), which involves a 1st person subject doing the offering.

(188) <decl n="38" sp-act="answer-**offer**" polarity="positive" topic="verify" mode="benefit-poss1-decl">
i can check back for you <pause /> <punc type="stop" />
</decl> (trainline05)

(189) <decl n="30" polarity="positive" mode="***offer***">***i offer*** a complete and utter retraction <punct type="stop" /></decl> (wanda2)

Examples like (189) do not occur in any of the three corpora, but this example is from an extract of the comedy *A Fish called Wanda*.

Suggestions and offers do not play a major role in Trainline and Switchboard, although suggestions occur more frequently in the latter. In Trains, in contrast, suggestions potentially represent an important part of the planning process, even if they only rank at position 21, so just outside the top 20 speech acts. The high incidence of offers there, however, as we saw previously, does not signal a genuine commitment, but in fact represents the playbook opening of the 'system'. Although these do not occur in the corpus data, the current implementation also contains mode definitions and inferencing rules for identifying promises (*promise*), betting (*bet*), and other explicit performatives (*perform*) marked by *hereby*, and which would usually mark an 'official act', such as marrying, baptising, sentencing, etc. (cf. Austin 1962), in order to be 'forward compatible' with future data from less restricted domains.

Directives, as defined by Searle, "are attempts […] by the speaker to get the hearer to do something" (1979: 13). According to him, "Questions are [also] a subclass of directives, since they are attempts by *S* to get *H* to answer" (ibid. 14). However, as we have at least to some extent seen in 7.1 above, defining questions in this way is too one-sided, speaker-centric, since assuming that they merely adequately reflect the speaker's desire for the hearer to do something does not acknowledge the collaborative role that many questions play in spoken interaction. The definition of directives (*direct*) used in the DART scheme thus excludes questions and focusses exclusively on cases where the speaker – sometimes qua role – has the authority to request or order someone to follow some kind of instruction.

This type of speech act is most frequent, and significant, in Trainline, where it actually ranks 10, apart from occurring in all dialogues. Its fairly high frequency can be explained both by the fact that the callers need to provide instructions for Sandra regarding their bookings, but also partly because Sandra actively encourages them in nearly half the cases by first eliciting this through a request for a directive (*reqDirect*). These two speech acts thus potentially form an adjacency pair, where the *reqDirect* indicates the commitment by the speaker to do what they will be told by the interlocutor, while the *direct* tells the (then) hearer what to do, and is grouped under suggesting speech acts because the speaker still has a choice as to whether to comply or not.

In the other two corpora, directives occur with a far lower frequency, in fewer documents, and at lower ranks. They are also hardly ever triggered by requests for directives. The main reason why directives occur in Trains at all, where otherwise more co-operative speech acts prevail, is that occasionally the 'user' of the 'system' lists the sub-tasks that form part of the problem-solving activity in the form of instructions they have received. In Switchboard, on the other hand, we either get 'collaborative adhortatives' like *see* or *imagine the guilt that person would have*, directives 'reported' as direct speech, or even 'categorical imperatives' as in (190).

(190) <imp n="93" sp-act="**direct**" polarity="negative" topic="location-enum-duration" mode="benefit-decl-disflu">
don't <comment type="restart" /> don't let them sit up there on death row for you know <comment content="laughter" /> 15 years
<punc type="stop" /></imp> (sw_0013_4617)

The final two speech acts in this category are *stateIntent* and *stateAttempt*. The first of these signals a commitment on the part of the speaker towards doing something, generally in the non-too-distant future (191), while the latter indicates a commitment by stating that one has been trying to do something and will probably continue to do so (192).

(191) <q-wh n="4" sp-act="reqDirect" polarity="positive" topic="journey" mode="preference2-open-query">
for which journey do you wish to purchase a ticket <punc type="query" /></q-wh>
<decl n="6" sp-act="direct-**stateIntent**" polarity="positive" topic="location_Anglo-from-to" mode="intent-preference1-decl">i'd like to go from Preston to London <punc type="stop" /></decl> (trainline10)

(192) <decl n="24" sp-act="state" polarity="positive" topic="verify-enum" mode="decl">
we just have the 1 check book <punc type="stop" /></decl>
<decl n="25" sp-act="**stateAttempt**" polarity="positive" topic="date" mode="attempt-decl">
and we <comment content="inhaling" /> try to keep it up to date as much as possible <punc type="stop" />
</decl> (sw_0003_4103)

Example (191) also shows that a statement of intention may occur concurrently with a directive if the triggering unit is a request for a directive. As statements of intent are an integral part of providing information in order to obtain appropriate timetable information, it is not surprising that they occur with the highest frequency, dispersion, and rank in Trainline. In Trains, they are almost equally important in indicating how the interlocutors intend to carry out parts of the planning process. There is, however, one main difference that demonstrates the more collaborative behaviour of the interlocutors in Trains, which is that more than half the number of units containing statements of intention has a plural 1st person subject, signalling that a joint commitment is intended, while only about 5% of the ones in Trainline do, indicating that the remainder represent one-sided commitments. In Switchboard, statements of intent are much rarer, and 1st person singular statements are double as frequent as plural ones. All of these basically concern everyday-life planning or decisions.

Statements of attempt only occur in Switchboard, but also only with a relatively limited frequency, so they are clearly not a major concern in the interaction, but mainly contribute circumstantial information. The fact that they do not occur in Trains is not very surprising, as the interaction is essentially goal-driven and future-oriented, without any need to refer to past attempts, while in Trainline, callers may well have made earlier attempts at making bookings or obtaining information, but simply do not indicate them as being attempts, but instead may report prior interaction with another agent.

7.6 Evaluating or attitudinal speech acts

So far, most of speech acts we have discussed were either predominantly fact-oriented or designed to help the conversation move along in some form, but generally there has been little or no emotional involvement on the part of the speakers concerned, even in the *expressX* acts that may have embodied a sense of epistemic judgement. In the group of speech acts we want to discuss now, this is very different, since essentially all of them embody some form of personal evaluation, or signal the attitude of the speaker towards something being discussed – hence the labels *evaluating* and *attitudinal*.

Table 7.13 Evaluating speech acts

Speech act	Trl NFreqs	Trl Rank	Trl Docs	Trs NFreqs	Trs Rank	Trs Docs	SWBD NFreqs	SWBD Rank	SWBD Docs
expressOpinion	21.0	40	14	26.5	32	18	234.5	6	35
expressConviction	–	–	–	–	–	–	6.0	52	8
expressDoubt	–	–	–	–	–	–	2.5	64	3

The first of the evaluative speech acts, *expressOpinion*, reflects a personal evaluation of the facts surrounding a given situation or topic on the part of the speaker.

(193) <decl n="61" sp-act="**expressOpinion**" polarity="positive" mode="opinion-decl">
i think we're done <punc type="stop" /></decl> (d92a-3.1)

(194) <decl n="23" sp-act="**expressOpinion**" polarity="positive" mode="opinion">
it seemed like it did <punc type="stop" /></decl>
<decl n="24" sp-act="**expressOpinion**" polarity="negative" mode="contrast-decl">
but <comment content="laughter" /> it might not **i guess**
<punc type="stop" /></decl> (sw_0001_4325)

(195) <decl n="146" sp-act="**expressOpinion**" polarity="positive" topic="booking" mode="constrain-frag">
<pause /> that train **must be** particularly booked <punc type="level" />
</decl>
<frag n="147" sp-act="**expressOpinion**" polarity="positive" mode="constrain-decl">
<pause /> **must be** really busy <punc type="stop" /></frag> (trainline18)

As Examples (193) and (194) indicate, expressions of opinion are often marked by a 1st person subject in combination with a verb of mental action, such as *think*, *guess, assume*, or a 3rd person subject in conjunction with a verb of perception, e.g. *seem, appear, look*, possibly also involving a comparison with *like*. Another form such expressions can take is that of epistemic necessity, as in Example (194). As this example shows, unlike for some of the other categories we saw above, the DART scheme makes no distinction between expressions of opinion that occur in declaratives and fragments, so there is no *referOpinion* counterpart to *expressOpinion*.

Expressions of opinion are a common feature of all forms of dialogue, so that it is perhaps not surprising that they occur in all our corpora, but what is striking are the statistics for Switchboard in comparison to the other two sets of data. Not only do *expressOpinion*s occur in all dialogues in the former, but this speech act also ranks amid the top 10, clearly marking the Switchboard dialogues as being of a much more personal and evaluative nature, while the latter are obviously more fact-oriented.

While expressions of opinion are, even if they do reflect personal stance, generally relatively neutral, unless of course they contain an explicit positive or negative evaluation, the DART scheme also recognises two stronger forms of indicating epistemic stance, *expressConviction* and *expressDoubt*. Both of these only occur in Switchboard, again corroborating the assumption that more personal evaluation is involved in the interaction there. The first of these two acts is expressed through markers of certainty, such as *i'm sure/certain, no doubt, of course (not), evident(ly), obvious(ly)*, etc., while the latter is often signalled via structures that contain *i wonder if* that do not signal indirect requests, or, obviously also *i/we doubt (that)* X.

(196) <decl n="92" sp-act="expressConviction" polarity="positive" mode="conviction-decl">
it was fairly obvious that it was a piece of propaganda
<punc type="stop" /></decl> (sw_0034_4345)

(197) <decl n="7" sp-act="answer-expressDoubt" polarity="positive" mode="condition-doubt-decl">
i wonder if that worked <punc type="stop" /></decl> (sw_0001_4325)

In contrast to the more knowledge- or belief-oriented evaluating acts, attitudinal speech acts are less concerned with facts, but rather different degrees of preference or dispreference that the speakers exhibit towards an item or topic under discussion.

Although all attitudinal speech acts have the potential to occur in any kind of dialogue, they are of course more likely to occur in verbal interactions of a more personal, involved, nature. This fact is clearly borne out by the frequencies and distributions in Table 7.14, where many of the speech acts listed either exclusively occur in Switchboard, or at the very least rank much higher there and also show greater dispersion. As the labels for *expressLiking/referLiking, expressDislike*, and

Table 7.14 Attitudinal speech acts

Speech act	Trl NFreqs	Trl Rank	Trl Docs	Trs NFreqs	Trs Rank	Trs Docs	SWBD NFreqs	SWBD Rank	SWBD Docs
expressLiking	–	–	–	–	–	–	40.5	24	15
referLiking	–	–	–	–	–	–	1.5	76	1
expressDislike	–	–	–	–	–	–	17.5	33	10
expressDisgust	–	–	–	–	–	–	–	–	–
expressPreference	1.5	68	2	–	–	–	2.5	64	4
expressDis-appointment	–	–	–	–	–	–	0.5	85	1
expressStance	0.5	73	1	–	–	–	0.5	86	1
expressSurprise	32.5	32	21	68.5	19	25	60.0	15	26
exclaim	20.5	41	15	12.0	45	6	43.0	23	22

expressDisappointment are probably intuitive enough to understand, I shall here only list a few examples highlighting the respective triggers, but not provide any detailed discussion.

(198) <decl n="34" sp-act="answer-**expressLiking**" polarity="positive" topic="cars" mode="preference1-decl-disflu">i li… <comment type="restart" /> **i like** cars that are designed with <comment type="restart" /> with human beings in mind <comment content="thumping" who="1569" /> <punc type="stop" />
</decl> (sw_0010_4356)

(199) <decl n="109" sp-act="**expressLiking**" polarity="positive" mode="frag-disflu">
it's a very <comment type="restart" /> really **enjoyable** for me <punc type="level" />
</decl> (sw_0020_4109)

(200) <decl n="111" sp-act="refer-**expressDislike**" polarity="positive" topic="enum" mode="frag">
and <backchannel content="aha" /> and 1 of the things that i <comment type="restart" /> **i really hate** about jobs <punc type="level" /></decl>
<decl n="112" sp-act="**expressDislike**" polarity="negative" topic="time" mode="dispreference1-decl">
i don't <comment type="restart" /> **i don't like** bosses that <comment type="restart" /> that you know want to <comment content="inhaling" /> yell at you and you know are down on your back and all this and that <punc type="stop" /></decl> (sw_0020_4109)

(201) <decl n="99" sp-act="**expressDislike**" polarity="negative" mode="dispreference1-alternative-decl">
i can't <backchannel content="laughter" /> **stand** uh uh Debussy or Debussy <comment content="pronounced different ways" /> <comment content="laughter" /> <punc type="stop" />
</decl> (sw_0045_4312)

(202) <decl n="96" sp-act="**expressDisappointment**" polarity="positive" topic="habit" mode="disappointment1-decl">
that's always **a bummer** too <punc type="stop" /></decl> (sw_0088_3073)

As Examples (198)–(202) demonstrate, there is also a gradience in the degree of liking or disliking, but the most extreme form of dislike would be indicated through an expression of disgust (*expressDisgust*). This speech act, however, which currently DART would only spot in the form of a DM containing exclamatives such as *urgh* or *yuck*, does not occur in our data.

The speech act *expressPreference* tends to be expressed through structures containing *rather* or *prefer, preferably*, etc., generally also containing a 1st person subject, although this does not always need to be present, as in Example (203).

(203) <frag n="67" sp-act="**expressPreference**" polarity="positive" topic="seat" mode="preference1-decl">
forward-facing **preferably** <punc type="stop" /></frag> (trainline21)

The next speech act, *expressStance*, probably requires some more explanation. The term *stance* frequently tends to be associated with a variety of different expressions that cover aspects of modality, notions of hedging, and other forms of personal judgment (cf. e.g. Hyland 2005: 176 or Gray and Biber 2015: 219), many of which are already covered by more specific speech-act labels in the DART scheme. Here, though, it acts as a 'garbage category' used to cover a number of remaining expressions reflecting personal attitude that I have so far been unable to categorise specifically. It thus subsumes constructs that signal a more neutral attitude or the choice of a speaker in not wanting to position or commit themselves towards a topic in any specific way, such as *never mind, i don't (really) care, i don't mind* X, *it doesn't matter*, etc., as well as 'markers of sincerity' like *frankly (speaking), to be quite/perfectly honest/frank*, etc. Such expressions occur in all three sets of data, but as the label only needs to cover features not yet covered by the other labels, the frequencies and ranks are fairly low. Further research is still required to categorise the features covered by this label better.

The final two speech acts in this category, *expressSurprise* and *exclaim*, are different from those attitudinal markers that embody a more personal evaluation in that, in general, they both signal an 'attitudinal' reaction indicating an element of surprise about, rather than concrete evaluation of, the situation. In *expressSurprise*, the reaction consist purely in signalling surprise, while *exclaim* often additionally reflects a sense of 'awe' or exasperation. The former tends to be expressed via single-word DMs containing *oh* or *ah*, occasionally also in combination with *yeah*, and frequently also precedes DMs signalling acknowledgement as in Example (204).

(204) <dm n="26" sp-act="phatic">you know <punc type="level" /></dm>
<decl n="27" sp-act="state" polarity="negative" mode="exists-decl">
there's no <backchannel content="yeah" /> no no prospects right away for anything to happen <punc type="stop" /></decl>
</turn>
<turn n="8" speaker="1191">
<dm n="29" sp-act="acknowledge">u… <comment type="restart" /> aha <punc type="stop" /></dm>
<dm n="30" sp-act="**expressSurprise**">oh</dm>
<dm n="31" sp-act="**acknowledge**" mode="**awareness**">i see <punc type="stop" /></dm> (sw_0015_4877)

In Example (204), speaker 1697 first informs speaker 1191 about the prospects concerning the topic under discussion. 1191 then initially acknowledges this (unit 29), but appears to not have understood the implications fully at first, but then, surprised at their own realisation, acknowledges the fact once more in unit 31, indicating a change in awareness at the same time, which is captured by the mode attribute.

If *oh* occurs in combination with *no*, however, in the majority of cases it counts as an *exclaim*, unless the two words are clearly separated prosodically and *no* can occur as a reluctant answer. *Oh* may also occur in combination with other words indicating exclamatives, such as in expressions like *oh my goodness/God, oh gosh, oh boy, oh dear*, etc., but exclamatives may of course also be realised in the form of other well-known expressions such as *geez/jeez/Jesus (H Christ), wow, w(h)oa, ouch, hell*.

(205) <decl n="30" sp-act="reqInfo" polarity="positive" mode="constrain-query">
<pause /> and we have to get to where <punc type="query" /></decl>
<frag n="31" sp-act="reqConfirm" polarity="positive" topic="location-transport_means" mode="query">
Elmira <punc type="query" /></frag>
<dm n="32" sp-act="exclaim"><pause /> oh boy <punc type="stop" /></dm>
(d93–19.3)

Another typical form exclamations may take can easily be confused with questions, as the first element is in fact a wh-word, or there is syntactic inversion present, as in Example (206) and (207).

(206) <exclam n="79" sp-act="**exclaim**" polarity="positive" mode="decl">
how lucky <comment content="laughter" /> <punc type="stop" /></exclam>
(sw_0001_4325)

(207) <dm n="271" sp-act="**exclaim**" mode="exclaim">
<overlap pos="start" /> gosh <punc type="level" /></dm>
<exclam n="272" sp-act="**exclaim**">
was that wonderful <overlap pos="end" /> <punc type="stop" /></exclam>
(sw_0069_3144)

Example (207) also illustrates that exclamations may occur in sequence, enhancing their effect.

Frequencies, ranks, and dispersion indicate that both expressions of surprise and exclamations play a more important role in Switchboard, most probably due to the stronger interpersonal orientation, and, as far as exclamatives are concerned, also somewhat higher degree of informality. This, however, does not mean that both speech acts cannot also occur under more formal, transactional conditions, as especially the relatively high rank for expressions of surprise in Trains indicates. Here, in contrast to Trainline, it seems again to some extent the slightly less predictable nature of the planning process that is causing the surprises, as the callers in Trainline generally appear to be quite well informed about their options. One thing we need to bear in mind here, though, is that the exact distinction between exclamatives and expressions of surprise remains difficult to make in the absence of full prosodic information and access to the audio materials.

7.7 Reinforcing speech acts

The category of reinforcing speech acts is rather small, as it only comprises two acts, *emphatic* and *reaffirm*. The first of these involves either the repetition of the same item or addition of an item that fulfils a function already fulfilled by another unit in the interaction sequence, usually a <yes>, <no>, or DM.

(208) <q-yn n="101" sp-act="reqOpt" polarity="positive" topic="to-transport_means-time-location" mode="intent-closed-query">
<pause /> can i <pause /> do the <pause /> transporting of <pause /> the OJ at the same time <pause /> while i'm doing <pause /> while i'm sending the engine to Dansville and then back <punc type="query" />
</q-yn>
</turn>
<turn n="45" speaker="s_TG">

<**yes** n="102" sp-act="stateOpt-acknowledge">yes <punc type="stop" />
</yes>
<**yes** n="103" sp-act="elab-**emphatic**"><pause /> yes <punc type="stop" />
</yes> (d93–18.2)

In Example (208), the item that is repeated is in fact of the same type, even using the same word form, but emphatic items need not be the same at all, as we can see in Example (209).

(209) <decl n="138" sp-act="reqConfirm" polarity="positive" topic="enum-transport_means" mode="limitation-tag-query"><pause /> we only need 1 boxcar of OJ right <punc type="query" /></decl>
</turn>
<turn n="62" speaker="u_ML">
<**dm** n="139" sp-act="confirm-**acknowledge**">**mm** <punc type="stop" />
</dm>
<**yes** n="140" sp-act="elab-**emphatic**">**yes** <punc type="stop" /></yes>
(d93–18.4)

As in (208), the yes-unit in Example (209) is completely redundant, since the confirmation for unit 138 has already been provided in unit 139 in the form of an acknowledgement, so the only reason for using unit 140 in this context is further emphasise the confirmation.

While *emphasise* in essence repeats the function of a unit within the prior local scope of the turn, *reaffirm* is not locally bound, and may even be exophoric or cataphoric. Its function is to indicate that something is, or is assumed to be, part of the common ground.

(210) <dm n="150" sp-act="muse">well <punc type="level" /></dm>
<frag n="151" sp-act="**reaffirm**">as you say <punc type="level" /></frag>
<decl n="152" sp-act="stateDuration" polarity="positive" mode="decl">
it takes a while to build it up <punc type="stop" /></decl> (sw_0055_3156)

(211) <turn n="13" speaker="Sandra">
<q-wh n="19" sp-act="reqInfo" polarity="positive" topic="date-journey" mode="open">
and what date is it you're travelling</q-wh>
</turn>
<turn n="14" speaker="caller_13">
<dm n="20" sp-act="muse">well <punc type="level" /></dm>
<decl n="21" sp-act="stateIntent-abandon" status="abandon" polarity="positive" mode="preference1-intent-abandon">i want to know erm <punc type="incomplete" /></decl>
<dm n="22" sp-act="**reaffirm**" mode="conviction-reason">
<pause /> because **obviously** <punc type="level" /></dm>

```
<frag n="23" sp-act="stateIntent-abandon" status="abandon"
polarity="positive" mode="intent-preference1-abandon">i want <punc
type="incomplete" /></frag>
<decl n="24" sp-act="state" polarity="positive" topic="duration"
mode="decl">
it's just for the day <punc type="stop" /></decl>
<decl n="25" sp-act="stateIntent" polarity="positive" topic="duration"
mode="limitation-decl">
i only want to go for the day <punc type="stop" /></decl>
</turn>
<turn n="15" speaker="Sandra">
<q-wh n="26" sp-act="reqInfo" polarity="positive" topic="date-journey"
mode="open-query">
what date is it you're travelling <punc type="query" /></q-wh>   (trainline13)
```

In Example (210), the speaker appeals to the information that is part of the common ground by referring back to something the other speaker has said before, thereby also 'cataphorically reinstating' the topic to some extent. In contrast, in (211), caller 13 uses the adverb *obviously* when it is by no means obvious that the enquiry is meant to concern a day trip, as this has never been mentioned before in the dialogue. Hence, Sandra simply ignores the fact that the information has been presented as presumably being part of the common ground, and simply repeats her question from unit 19 in unit 26.

These two examples also illustrate to some extent the different forms *reaffirm* can take, one where an appeal is made to something that has been said before, in other words, similar to reporting, and the other through adverbs or adverbial constructions that 'appeal to the obvious', as e.g. *obviously, of course, naturally, evidently, needless to say*, etc. Table 7.15 lists the statistics for both speech acts in all three corpora.

Table 7.15 Reinforcing speech acts

Speech act	Trl NFreqs	Trl Rank	Trl Docs	Trs NFreqs	Trs Rank	Trs Docs	SWBD NFreqs	SWBD Rank	SWBD Docs
emphatic	52.0	25	27	19.5	36	13	14.0	36	14
reaffirm	3.5	63	5	1.0	70	1	18.0	33	15

With regard to the use of the *emphatic* speech act, it is interesting to note that, whereas the statistics for Trains and Switchboard are very similar, the frequency, rank, and dispersion are much higher in Trainline. This discrepancy may well be due to cultural differences, in that the British speakers in Trainline may exhibit more of a tendency to 'overemphasise' in comparison to the American speakers in the other two corpora. Whether this assumptions is indeed justified, though, would need to be verified on much larger and more diverse corpora.

The difference observable between the task-oriented corpora and Switchboard for the speech act *reaffirm*, in contrast, appears to be more one that is genre-based, as apparently the need to maintain or reaffirm common ground is much higher in unconstrained dialogue, perhaps also because speakers tend to voice their assumptions about the facts being discussed more frequently there.

7.8 Social, conventionalised speech acts

The items in the next category can be referred to as *social* or *conventionalised* speech acts. This type of speech act essentially has very little to do with any type of transaction, but more with the 'social niceties' – as well as their potential counterparts – that have to accompany any kind of trans- or interaction between dialogue participants.

Table 7.16 Social, conventionalised speech acts

Speech act	Trl NFreqs	Trl Rank	Trl Docs	Trs NFreqs	Trs Rank	Trs Docs	SWBD NFreqs	SWBD Rank	SWBD Docs
greet	30.5	33	33	31.5	29	34	4.0	56	4
identifySelf	24.5	36	34	–	–	–	0.5	86	1
bye	36.5	29	28	–	–	–	4.0	56	3
thank	60.5	21	31	2.5	58	2	2.0	68	2
acknowledge-Thanks	–	–	–	–	–	–	–	–	–
expressRegret	24.0	37	18	8.5	50	8	3.0	63	4
apologise	1.5	69	2	1.5	63	2	0.5	86	1
swear	–	–	–	–	–	–	–	–	–
insult	–	–	–	–	–	–	–	–	–

The first three speech acts essentially 'set or close the scene', and, apart from *identifySelf*, form integral parts of almost any kind of complete interaction, other than perhaps the most rude encounters, such as maybe a robbery or other type of attack on a person. But then again, one probably would not want to call this type of encounter an interaction in the strict sense, anyway. Because *greet* and *bye* are so conventionalised, there is hardly any need to provide any examples here, other than saying that, depending on the level of formality, most greetings would probably take the form of *hello, hi, good morning/evening/afternoon, nice to see you* *again, fancy meeting you here*, etc., where obviously the first three are often followed by the latter examples, if those occur at all, while goodbyes may be realised either as *goodbye, bye, have a good/nice day, see you (later)*, etc.

Greets that occur at the beginning of an 'ordinary' interaction sequence – such as an encounter between friends or acquaintances – are generally characterised by a reciprocal response by the party who has been greeted first. However, in transactional dialogues – as illustrated by the Trainline and Trains data – this reciprocity is not always present, presumably because the operator, or the 'System' in Trains, is, as repeatedly observed, in an 'inferior' position as the agent supposed to provide information, and the latter therefore does not have to employ the same level of politeness, but actually has the option of 'barging out' with their request immediately. As it happens, out of the 34 cases in the Trainline data where the operator opens up the dialogue with a greeting, this is only reciprocated 8 times, and only once in Trains.

One thing that may not be so obvious, though, is that some 'greetings' may also occur after a *hold* phase in transactions that are conducted over the telephone in order to re-start the conversation. Here, they usually also have the additional function of ascertaining whether the other party is still 'online' after the period of interruption that was necessary to check on some detail or verify some information. In this case, obviously no greeting in return is required, but it is still generally necessary for the hearer to acknowledge that he or she is still listening, as can be seen in Example (212).

(212) <q-yn n="46" sp-act="direct" polarity="positive" topic="duration" mode="hold-query">
can you hold the line a second <punc type="query" /></q-yn>
</turn>
<turn n="27" speaker="Sandra">
<yes n="47" sp-act="acknowledge">yeah <punc type="stop" /></yes>
<dm n="48" sp-act="agree">sure <punc type="stop" /></dm>
<frag n="49" sp-act="**greet**" mode="greet">hello <punc type="stop" />
</frag>
</turn>
<turn n="28" speaker="caller_05">
<yes n="50" sp-act="acknowledge">yeah <punc type="stop" /></yes>
<q-yn n="51" sp-act="reqOpt" polarity="positive" topic="number-journey-booking" mode="closed-query">can i book a ticket on that one <punc type="query" /></q-yn> (trainline05)

Here, the customer requests or signals the hold (unit 46) and Sandra initially acknowledges and agrees with the hold request (units 47 & 48), but after a while initiates the resumption of the dialogue (unit 49). The customer then acknowledges that he/she is still there in unit 50 and continues with an enquiry about a potential option in unit 51.

Greetings are frequently followed by a form of self-identification (*identifySelf*), especially in dialogues that are conducted over the telephone, or when the conversational partners do not know each other. Most often, this type of speech act occurs in fragments, but of course may also be found in declaratives, and can be captured by means of a mode attribute called *intro*, which matches constructions like X *speaking* or *my name is* X. It is, however, not possible to identify all potential names or name combinations that may represent a self-identification easily because, for example, a unit that contains the word sequence *this is* + (title +) first name (+ surname), would still be ambiguous, unless it was followed by the words *speaking* or *calling*, as there could equally well be a third party being introduced by one of the speakers, or, depending on the position inside the dialogue, it may represent an answer.

In the Trainline data, the introduction usually takes the form *Sandra speaking*. The speech act does not occur in any of the other two sets of data, although we would probably expect it to occur in Switchboard. However, as already indicated earlier, the Switchboard conversations do not always contain the beginnings and ends of the conversations, so that these may have simply been deleted from our data. In Trains, such introductions, as well as goodbyes, are obviously not necessary, as the assumption is that the 'human interlocutors' are supposed to be communicating with a machine.

Thanking (*thank*) is conceptually quite different in its social and dialogue function from greetings, self-identifications, and farewells. While greetings and introductions constitute preliminaries to the dialogue proper, and potentially also establish the roles and 'hierarchies of authority' between the participants, farewells 'round it off', marking the end. These former categories can almost exclusively occur at the beginning or end of the dialogue, while thanking, and its 'negative counterpart', swearing (*swear*), can occur in different places throughout a verbal exchange, and essentially represent conventionalised forms of response, signalling the (dis-)satisfaction with an action or proposal/commitment on the part of the interlocutor. The same goes for the positive and polite response to thanking, labelled *acknowledgeThanks* in the DART scheme, which may be realised as e.g. *don't mention it*.

The speech act *expressRegret* may at first appear similar in its function to the *pardon* DM, and there is indeed a certain amount of overlap, but unlike the *pardon*, usually the *expressRegret* does not signal misunderstanding, but rather functions as a politeness device, informing the hearer that something is or was impossible to do in a transactional context, even though it may also include 'more genuine' expressions of regret – such as *i'm sorry about your loss* – or signal an implicit apology, etc., in other contexts.

(213) <decl n="124" sp-act="stateOpt-***expressRegret***" polarity="negative" topic="journey-day" mode="***regret***">***i'm afraid*** that's not possible with that type of ticket for travelling tomorrow</decl> (trainline12)

(214) <decl n="20" sp-act="stateIntent" polarity="positive" mode="intent-constrain">
i'm going to have to phone you back</decl>
<decl n="21" sp-act="expressPossibility" polarity="positive" topic="number-location" mode="poss3">there might be two of us <pause /></decl>
<decl n="22" sp-act="***expressRegret***" polarity="positive" mode="***regret***">
i'm sorry about that</decl> (trainline31)

Apologising (*apologise*) is a more specific and explicit form of expressing regret, where this regret is communicated not for something that has simply happened to the interlocutor and/or other parties, but instead something the speaker has actually caused, or possibly an interruption they are about to cause. This speech act can only be identified unambiguously by DART if it occurs in the form of an explicit performative that contains a 1st person subject and the word *apologise*, or the imperative *excuse me*.

In Trainline, most of the expressions of regret are uttered by Sandra, essentially indicating that particular options the caller is interested in are not available, while one apology each is produced by her and another by a caller. In Trains, both explicit apologies, perhaps coincidentally, come from the speaker representing the system, and expressions of regret are considerably rarer than in Trainline. Essentially, most of the latter functionally, but not formally, constitute apologies for either having said something wrong or not having mentioned it at all. In Switchboard, the few expressions of regret do constitute implicit apologies, but all reflect regret about circumstances surrounding the topic that is being discussed, while the one apology is related to getting a fact wrong.

Swearing or insulting do not in fact occur in the main data used for this study, but the speech acts *swear* and *insult* were included and modelled in the DART scheme because they did occur in one of two excerpts from the British Comedy *A Fish Called Wanda* (Crichton 1988), which I was once asked to analyse using the original SPAACy program. To illustrate these two features, an excerpt from this file is presented below in Example (215). The speaker identifiers/attributes refer to the actors Kevin Cline (KC), John Cleese (JC) and Jamie Lee Curtis (JLC), respectively.

(215) <turn n="5" speaker="KC">
<dm n="5" sp-act="init" polarity="positive">now <punct type="stop" /></dm>
<imp n="6" sp-act="direct" polarity="positive" mode="apology">apologise
<punct type="stop" /></imp>
</turn>

[…]
<turn n="7" speaker="JC">
<q-wh n="8" sp-act="reqInfo" polarity="positive" mode="exclaim-partial">
what <punct type="question" /></q-wh>
</turn>
<turn n="8" speaker="KC">
<imp n="9" sp-act="answer-direct" polarity="positive" mode="apology">
apologise <punct type="stop" /></imp>
</turn>
<turn n="9" speaker="JLC">
<frag n="10" sp-act="**swear**" polarity="positive" mode="expletive">oh shit
<punct type="exclam" /></frag>
</turn>
<turn n="10" speaker="JC">
<q-yn n="11" sp-act="reqInfo" polarity="positive" mode="closed">
are you totally deranged <punct type="stop" /></q-yn>
</turn>
<turn n="11" speaker="KC">
<frag n="12" sp-act="answer-**insult**" polarity="positive" mode="expletive">
you pompous <punct type="comma" /> stuck up <punct type="comma"
/> snot-nosed <punct type="comma" /> English <punct type="comma" />
giant <punct type="comma" /> twerp <punct type="comma" /> scumbag
<punct type="comma" /> fuck-faced <punct type="comma" /> dickhead
<punct type="comma" /> asshole <punct type="stop" /></frag>
</turn>
<turn n="12" speaker="JC">
<exclam n="13" sp-act="exclaim" polarity="positive">how interesting
<punct type="stop" /></exclam>
<decl n="14" sp-act="reqConfirm" polarity="positive" mode="tag">
you are a true vulgarian <punct type="comma" /> aren't you <punct
type="question" /></decl>
</turn>
<turn n="13" speaker="KC">
<decl n="15" sp-act="**insult**" polarity="positive" mode="expletive-insult">
you are the vulgarian <punct type="comma" /> you fuck
<punct type="stop" /></decl> (wanda2)

In this excerpt, Kevin Cline is trying to get John Cleese to apologise to him (units 6 & 9), with Cleese initially refusing to do so. In unit 10, Jamie Lee Curtis, who has been locked out of the flat, but can still hear what is going on inside, swears because she can nevertheless anticipate what may happen if Cleese does not respond appropriately. In unit 12, Kline utters another insult in response to Cleese's provocative question. Unit 12 is identified as an instance of insulting by the analysis methodology on the basis of the *expletive* mode because the mode definition for insults, which also triggers the assignment of the speech act *insult* in unit 15, 'expects' a 2nd person (non-possessive) pronoun that is immediately followed by an insulting term, or an insulting verb immediately followed by the pronoun. The tag question in unit 14 again cannot be identified as an indirect insult without a semantic analysis of the trigger word *vulgarian*.

In terms of bringing the dialogue forward, social speech acts, in most cases, perhaps with the exception of *swear* and *insult*, contribute very little or maybe even nothing at all, which is why they differ strongly from other responding speech acts (or non-verbal actions), and should therefore rather be counted towards what Geis (1995: 33) refers to as the level of *interactional*, as opposed to the *transactional, significance*.

7.9 Residual speech acts

To complete our discussion of the DART speech-act taxonomy, it is still necessary to introduce a number residual acts that cannot really be identified automatically by the system, but that may often be usefully applied in post-editing the data to create a finished corpus.

Table 7.17 Residual speech acts

Speech act	Trl NFreqs	Trl Rank	Trl Docs	Trs NFreqs	Trs Rank	Trs Docs	SWBD NFreqs	SWBD Rank	SWBD Docs
selfTalk	–	–	–	–	–	–	–	–	–
thirdParty	–	–	–	–	–	–	–	–	–
uninterpretable	6.5	54	7	1.0	70	1	1.5	75	1
unclassifiable	–	–	–	–	–	–	–	–	–

If a speaker produces an 'aside' not directed at any interlocutor contributing to the verbal interaction, and that aside also does not fit into the dialogue sequence, this can be labelled as *selfTalk*, while speech directed at another party not immediately involved in the current dialogue, that is, occurring in the background, should be labelled *thirdParty*.

A speech act that cannot be interpreted fully as to its presumed intended meaning, generally due to the absence of contextual information, transcription errors, unclear passages, or incoherence, may be marked up as *uninterpretable*. In some cases, however, it may also happen that a unit is perfectly coherent, but its function does not fit any of the existing categories or even comes close to one, in which case the label *unclassifiable* can be used, unless it is possible to coin a new suitable descriptive label. The use of *unclassifiable* is highly advisable, though, when working with a team of annotators so as to avoid any ad hoc coinages and proliferation of new labels. Units marked with this label can then easily be identified in team meetings in order to reach an agreement on whether it may be possible to use an existing label after all, or possibly coin a new one based on consensus.

Out of the four additional speech-act labels, only *uninterpretable* occurs in the three corpora, but the instances are relatively low, with the highest incidence in Trainline. All of the instances there are due to all or parts of the unit being marked as unclear by the original transcriber, and all of them also occur in passages that exhibit overlapping, which may have caused the unintelligibility of one of the speakers. There is only a single instance in Trains where it is impossible to interpret the unit due to incoherence.

(216) <frag n="24" sp-act="**uninterpretable**" polarity="positive" topic="enum" mode="frag">
<pause /> we 1 1 of the 3 engines <punc type="level" /></frag>
<decl n="25" sp-act="stateConstraint" polarity="positive" topic="to" mode="constrain-frag">
<pause /> and <pause /> we need to <pause /> bring that to Corning with a boxcar <backchannel content="right" /> <punc type="level" /></decl>
(d92a-5.1)

CHAPTER 8

Conclusion

As pointed out in the introduction to this book, although pragmatics-related research has been carried out for at least two decades, corpus pragmatics per se is only a very young sub-discipline of corpus linguistics, and most of the research carried out in this area still relies more or less exclusively on traditional methods from corpus linguistics. In the preceding chapters, I have tried to demonstrate how these methods can fruitfully be augmented by creating and exploiting pragmatically annotated corpora in a kind of corpus pragmatics that goes far beyond what has previously been possible. In the course of my discussion, I have not only tried to illustrate the formal requirements and means for creating such corpora, but, by applying the DART methodology to the analysis and comparison of three sets of data from corpora originating in three different domains, also attempted to answer a number of research questions.

The first two of these were which levels of meaning we can distinguish within dialogues pertaining to different domains, and how we can classify and describe these levels of meaning in order to relate them to the identification of pragmatic force and speaker intentions. By discussing the different levels of linguistic meaning that form part of the DART analysis methodology, and illustrating through copious examples drawn from the data how they may each become relevant in determining the pragmatic meaning in all three (sub-)corpora to a greater or lesser extent, I hope to have illustrated that the syntactic form of a unit, along with other 'IFIDs' on the levels of semantico-pragmatics, semantics, and polarity, essentially make it possible to determine the local speech act(s) performed in this unit with a relatively high degree of precision. Furthermore, we have seen that it is also feasible to identify more contextually determined illocutionary potential by looking at various types of information-seeking units or speech acts and their associated responses, the role that specific DMs may play in initiating different stages of a dialogue, or even how simple repetitions echoing the interlocutor's words can either be used for verification purposes or to start verification phases.

In distinguishing between generic and domain-specific content, I have hopefully also made it clear that, the more generic a content element is, the more suitable it becomes in determining pragmatic meaning across different domains, including unconstrained dialogue. This will probably have become most apparent in the relationship between generic topics, such as e.g. times, dates, and places, and certain types of referential speech acts, *referTime, referDate*, and *referPlace*, in which these topics help to create a very specific meaning potential.

In Chapter 7, and to a somewhat lesser extent also in Chapter 4, I have illustrated what a suitably generic taxonomy of speech acts would need to look like, and which aspects of communicative meaning it can cover. At the same time, we saw in Chapter 4 in which ways the syntax of a unit essentially pre-determines or constrains its illocutionary potential. Together, these two chapters therefore provide the answer to research question number 3, what kind of elements a suitably generic taxonomy of speech acts may need to contain, although the fact that certain speech acts only occurred in specific corpora, but not across all the data, ought to serve as a constant reminder that a truly exhaustive coverage in terms of suitable speech-act labels may only be achievable once a much larger number of corpora from many more different domains will have been annotated and analysed. In addition, although Geoffrey Leech made an invaluable contribution to the original SPAAC speech-act taxonomy, and the labels coined for this taxonomy were already to some extent influenced by the efforts of researchers who contributed to the DRI workshops, etc., the current, much more elaborate, DART taxonomy basically constitutes the long-term work of a single researcher. And even if I have tried my very best to define appropriate and more intuitive labels for the speech acts that can be recognised by DART than previous research in pragmatics has employed, a wide-scale adoption by scholars in corpus pragmatics would either presuppose the acceptance of this taxonomy in the research community, or may require a suitable revision based on a consensus model.

Achieving such a consensus would probably also necessitate an equally wide-scale adoption of DART as a means to annotate large quantities of corpora from diverse domains, as it currently represents the only way of carrying out the large-scale automated pragmatic annotation on all the different pragmatically relevant levels discussed earlier. The feasibility of such an enterprise is something I hope to have demonstrated clearly through the set of corpus data comprising 17,382 c-units, 7,027 turns, and 81,520 words I annotated 'single-handedly' to use as the primary basis for this study. The manual annotation of a corpus of this scale would probably have presented a major obstacle, even for a larger research team, or at least have taken a very long time, and have resulted in many inconsistencies being introduced by the opinions of different annotators. Even using DART, the pre- and post-processing stages of the annotation, especially in contrast to the actual annotation processes that only took a few minutes, were still extremely laborious, and may also have resulted in a number of small inconsistencies and/or errors that are likely to have affected at least parts of the analyses. Nevertheless, at least to some extent, this achievement already constitutes part of the answer to research question number 4, demonstrating to what extent a large-scale automated pragmatic annotation is feasible.

This still leaves the second part of the question unanswered, which was how such an analysis could also incorporate different levels of (in-)directness. In response to this question, I would like to point out that the very fact that it has been possible to create form–function mappings between the locutionary and illocutionary forces for some 120 speech acts for such a large number of units largely automatically already indicates that far more than has commonly been assumed to be indirect in speech is actually highly conventionalised, and should hence probably not be considered indirect at all. In addition, as we have seen in a number of examples from the annotated data, it is clearly possible to take at least a certain amount of contextual information into account to explain the meaning of specific 'indirect' units, such as e.g. when we can deduce that a response to a request for a directive needs to constitute a directive, even if no obvious directive force is indicated locally in the locutionary form.

The final research question I posed was how an annotation like the one produced by DART can truly make corpus-based pragmatics research possible. This is something I hope to have shown throughout the different chapters of this book, partly by explaining the mechanisms and resources necessary to carry out the multi-level annotation of pragmatics-relevant features, but mainly through the analyses presented in Chapter 4 and 7. In these two chapters, I have not only looked at how pragmatic meaning in our three sample corpora is created from two different perspectives, the syntactic and the pragmatic (speech-act) one, but also discussed how the distribution of the relevant features forms an integral part of the functional communicative characteristics of the different domains and speaker strategies represented in the these sets of data. Furthermore, by drawing comparisons between the corpora, and here especially between the two task-oriented corpora and the unconstrained dialogues in Switchboard, I hope to have shown that it is not only possible to annotate data from both restricted and unconstrained domains successfully using the DART methodology, but also that it is equally possible to identify distinctive differences in the frequency distributions that point towards the various purposes and aims inherent in the different types of interaction.

In addition, as DART provides form–function mappings for a multitude of features that already tend to be points of focus in corpus pragmatics, such as DMs or the different pragmatic functions of modal auxiliaries, etc., and it also provides integrated mechanisms for efficiently analysing or even re-annotating these using potentially much larger corpora than have traditionally been used, the methodology would seem to be ideally suited for compiling and analysing a growing set of spoken-language corpora for a variety of different purposes. In order to foster this, I am not only making DART available freely to the research community, and continually developing it further to improve the annotation and analysis functionality, but have also recently outlined how it would be possible to (re-)annotate a variety of different corpora from different areas. These areas currently include the study of pragmatic differences in World Englishes, based on the spoken components of the International Corpus of English (ICE; see Weisser 2017), and the investigation of functional characteristics of spoken language learner data from the Louvain International Database of Spoken English Interlanguage (LINDSEI; see Weisser 2016d). At the same time, apart from in the analyses presented in this book, I have also successfully employed the DART methodology in establishing speaker profiles for agents and their callers covering various aspects of initiative, efficiency, politeness, and directness in the context of call-centre interactions in Britain and the US (Weisser 2016c). These few examples already illustrate the wide appliability of the methodology for different purposes, but it is easy to imagine further contexts in which it could be fruitfully employed, such as e.g. in investigating interactions in healthcare, tracing decision-making strategies in business meetings, or analysing the argumentative strategies used in courtroom interactions.

As I hope to have demonstrated in the discussion above, DART and its associated annotation methodology have now reached a stage of maturity that can provide the basis for their wide-scale adoption in spoken corpus pragmatics, even if their development can by no means be seen as finished. However, the latter is somewhat akin to the continuing development of PoS-tagging theories and taggers that have revolutionised much of corpus linguistics, even though the results of tagging operations without a suitable amount of manual post-processing are still not as good as the designers of these tools would like us to believe (cf. Weisser 2016a, Chapter 7). Thinking beyond pragmatic annotation for spoken language and its by now hopefully obvious possibilities for identifying communicative strategies, though, it should be relatively easy to see that of course very similar mechanisms are also employed in written language, albeit generally in a more complex form that is also likely to require at least slightly different processing mechanisms. Adapting the DART methodology for this therefore, to my mind, represents the next biggest challenge for corpus pragmatics, a challenge I have already begun to face in the shape of the ongoing development of DART's 'big brother', TART, the Text Annotation and Research Tool. More information on TART and its ongoing development will be made available at http://martinweisser.org/ling_soft.html#TART.

Appendix A

The DART speech-act taxonomy (version 2.0)

Speech-act label	(Approximate) function
abandon	abandoning a unit, either choosing not to complete it or due to interruption
hesitate	hesitating before the beginning of a turn/unit
hold	signalling to the interlocutor to hold the line, usually to look up information or to think
muse	deliberating as to how to respond
pardon	signalling misunderstanding/the need for the interlocutor to repeat
phatic	semantically empty discourse-marking expression, such as initial *you know*
acknowledge	signalling decoding, understanding
answer	answering a question
confirm	confirming a request for confirmation
disConfirm	negative response to a request for confirmation
init	initiating a new phase of the dialogue
nominate	indicating that/when someone should speak in a sequence
add	initiating a follow-up in a sequence
attribute	expressing attribution to someone
complete	completing the interlocutor's move
initConclusion	initiating a (logical) conclusion
initCondition	initiating a condition
initConstraint	initiating a constraint
initContrast	initiating a contrast
initCounterExp	initiating something counter to expectation
initFollowUp	initiating one or more additional explanations
initGeneralisation	initiating a generalisation
initOpinion	initiating an opinion
initQ	initiating a query
initReason	initiating a reason

(*Continued*)

Appendix A. (*Continued*)

initSummary	initiating a summary
listSequence	listing items using ordinals
acknowledgeThanks	responding to a thank you
apologise	apologising
bye	saying farewell; closing a dialogue
expressRegret	expressing regret
greet	greeting the interlocutor
identifySelf	identifying the speaker's name/institution
insult	insulting
swear	swearing, possibly insulting
thank	thanking
exclaim	expressing emotion or surprise
expressDisappointment	expressing diappointment
expressDisgust	expressing disgust
expressDislike	expressing dislike
expressIndifference	expressing one's indifference to a proposal, suggestion, or offer
expressLiking	expressing a liking for something
expressPreference	expressing a preference
expressStance	expressing one's attitude, eg. through frankly (speaking), etc.
expressSurprise	expressing surprise
referLiking	referring to a liking
expressConviction	expressing conviction, e.g. through use of no doubt
expressDoubt	expressing doubt
expressOpinion	expressing an opinion/evaluation
expressAwareness	expressing awareness, possibly knowledge of something
expressNonAwareness	negative counterpart to the above
expressUncertainty	expressing uncertainty regarding something
refer	indicating a deictic reference (neutral option)
referAct	referring to an ongoing action/process
referCondition	referring to one or more conditions
referConstraint	referring to one or more constraints
referDate	referring to a date
referDirection	referring to a direction
referDuration	referring to a duration
referHow	referring to how something is done
referOpt	referring to an option

Appendix A. (*Continued*)

referPerson	referring to a person/people (excluding vocatives)
referPlace	referring to a place/places
referProcess	referring to an ongoing action
referThing	referring to a concrete or abstract object
referTime	referring to a specific (point in) time
elab	elaborating the answer to a question
spell	spelling out something
conclude	indicating a (logical) conclusion
explain	providing an explanation
referReason	referring to a reason
stateReason	stating a reason
expressImPossibility	negative counterpart to the above
expressImProbability	negative counterpart to the above
expressPossibility	expressing a possibility
expressProbability	expressing the probability of something occurring
predict	predicting some future event
referPossibility	referring to a possibility
referProbability	referring to a probability
enumerate	enumerating
negate	making a negative-polarity statement
report	reporting what others, including the interlocutor, have said
state	conveying information
stateCondition	stating a condition
stateConstraint	stating a potential constraint
stateDate	stating a date
stateDistance	stating a distance
stateDuration	stating a duration
stateHabit	stating a habit
stateNonConstraint	stating the absence of a constraint
stateNonOpt	stating the absence of a potential option
stateOpt	stating a potential option
stateProcess	stating that a process/action is ongoing
stateTime	stating a time
expressHope	expressing hope that something can or will be done, will happen
expressWish	expressing a wish or desire

(*Continued*)

Appendix A. (*Continued*)

echo	repeating the interlocutor's words for verification
reqConfirm	requesting a confirmation
reqInfo	requesting verbal information
reqModal	requesting permission, advice, etc.
reqOpt	requesting an option
admit	admitting to something
correctSelf	correcting one's own utterance
rejectSelf	rejecting one's own ideas/suggestions
retract	retracting something that has been stated
contradict	contradicting the interlocutor
correct	correcting what the interlocutor has said
disagree	expressing disagreement
disapprove	expressing disapproval
refuse	responding negatively to an offer, etc
reject	rejecting a proposal or suggestion
accept	responding in an active positive way
agree	signalling explicit agreement
approve	expressing appreciation or approval
emphatic	repeating something for emphasis, usu. yes, no, or a DM
reaffirm	indicating that something is (assumed to be) part of the common ground
selfTalk	speaking to oneself (the speaker)
unclassifiable	a speech act not classifiable according to the present scheme
uninterpretable	uninterpretable, due to missing or incoherent information
offer	offering a service to benefit the interlocutor
reqDirect	requesting a directive
stateAttempt	stating that an attempt was/is being made
stateIntent	indicating the speaker's intention
suggest	proposing joint or interlocutor's potential action
direct	eliciting the interlocutor's non-verbal response

References

Aarts, Bas & McMahon, April (eds). 2006. *The Handbook of English Linguistics*. Oxford: Blackwell. doi:10.1002/9780470753002

Adolphs, Svenja. 2008. *Corpus and Context: Investigating Pragmatic Functions in Spoken Discourse* [Studies in Corpus Linguistics 30]. Amsterdam: John Benjamins. doi:10.1075/scl.30

Adolphs, Svenja & Carter, Ronald. 2013. *Spoken Corpus Linguistics: From Monomodal to Multimodal*. London: Routledge.

Aijmer, Karin. 1996. *Conversational Routines in English: Convention and Creativity*. London: Longman.

Aijmer, Karin. 2002. *English Discourse Particles: Evidence from a Corpus* [Studies in Corpus Linguistics 33]. Amsterdam: John Benjamins. doi:10.1075/scl.10

Aijmer, Karin. 2013. *Understanding Pragmatic Markers: A Variational Pragmatic Approach*. Edinburgh: EUP.

Aijmer, Karin & Rühlemann, Christoph (eds). 2015. *Corpus Pragmatics: A Handbook*. Cambridge: CUP. doi:10.1017/CBO9781139057493

Allen, James. 1995. *Natural Language Understanding*, 2nd edn. Redwood City CA: The Benjamin/Cummings Publishing Company.

Allen, James & Core, Mark. 1997. Draft of DAMSL: Dialog Act Markup in Several Layers. <ftp://ftp.cs.rochester.edu/pub/packages/dialog-annotation/manual.ps.gz>

Alexandersson, Jan, Buschbeck-Wolf, Bianka, Fujinami, Tsutomu, Kipp, Michael, Koch, Stephan, Maier, Elisabeth, Reithinger, Norbert, Schmitz, Birte & Siegel, Melanie. 1997. Dialogue Acts in VERBMOBIL-2. *VM-Report 204*, DFKI GmbH.

Altenberg, Bengt. 1990. Spoken English and the dictionary. In *The London-Lund Corpus of Spoken English: Description and Research*, Jan Svartvik (ed.). Lund: Lund University Press.

Anderson, Anne, Bader, Miles, Bard, Ellen, Boyle, Elizabeth, Doherty, Gwyneth, Garrod, Simon, Isard, Stephen, Kowtko, Jacqueline, McAllister, Jan, Miller, Jim, Sotillo, Catherine, Thompson, Henry & Weinert, Regina. 1991. The HCRC Map Task Corpus. *Language and Speech* 32: 351–366.

Androutsopoulos, Ion & Aretoulaki, Maria. 2003. Natural language interaction. In Mitkov (ed.), 629–648.

Anthony, Laurence. 2014. AntConc (Version 3.4.3) [Computer Software]. Tokyo: Waseda University. <http://www.laurenceanthony.net/>

Ariel, Mira. 2008. *Pragmatics and Grammar*. Cambridge: CUP. doi:10.1017/CBO9780511791314

Atkins, Sue & Rundell, Michael. 2008. *The Oxford Guide to Practical Lexicography*. Oxford: OUP.

Atkinson, J. Maxwell & Heritage, John. 1984/2006. Jefferson's transcript notation. In *Structures of Social Interaction: Studies in Conversation Analysis*, J. Maxwell Atkinson & John Heritage (eds.). Cambridge: CUP. Reprinted in Adam Jaworski & Nicholas Coupland (eds). 2006. *The Discourse Reader*, 2nd edn, 158–165. London: Routledge.

Austin, John Langshaw. 1962. *How to do Things with Words*. Oxford: OUP.

Barnbrook, Geoff. 1996. *Language and Computers: A Practical Introduction to the Computer Analysis of Language*. Edinburgh: EUP.

Beesley, Kenneth & Karttunen, Lauri. 2003. *Finite State Morphology*. Stanford CA: CSLI.

Benson, James & Greaves, William. 1981. Field of discourse: Theory and application. *Applied Linguistics* II(1).

Biber, Douglas, Conrad, Susan & Reppen, Randi. 1998. *Corpus Linguistics: Investigating Language Structure and Use*. Cambridge: CUP. doi:10.1017/CBO9780511804489

Biber, Douglas, Johansson, Stig, Leech, Geoffrey, Conrad, Susan & Finegan, Edward. 1999. *Longman Grammar of Spoken and Written English*. London: Longman.

Blanche-Benveniste, Claire. 1997. *Approches de la langue parlée en français*. Paris: Ophrys.

Bradley, Neil. 1998. *The XML Companion*. Harlow: Addison-Wesley.

Bresnan, Joan. 2000. *Lexical-functional Syntax*. Oxford: Blackwell.

Brown, Gillian & Yule, George. 1983. *Discourse Analysis*. Cambridge: CUP. doi:10.1017/CBO9780511805226

Brown, Penelope & Levinson, Stephen. 1987. *Politeness: Some Universals in Language Usage*. Cambridge: CUP.

Burnard, Lou. 1995. The Text Encoding Initiative: An overview. In Leech, Myers & Thomas (eds), 69–81.

Bussmann, Hadumod. 1996. *Routledge Dictionary of Language and Linguistics*, Gregory Trauth & Kerstin Kazzazi (trans.). London: Routledge.

Butt, Miriam, Holloway King, Tracy, Niño, María-Eugenia & Segond, Frédérique. 1999. *A Grammar Writer's Cookbook*. Stanford CA: CSLI.

Callies, Marcus. 2013. Advancing the research agenda of interlanguage pragmatics: The role of learner corpora. In Romero-Trillo (ed.), 9–36. doi:10.1007/978-94-007-6250-3.

Carletta, Jean, Isard, Amy, Isard, Stephen, Kowtko, Jacqueline, Doherty-Sneddon, Gwyneth & Anderson, Anne. 1996. HCRC Dialogue Structure Coding Manual. HCRC Technical Report 82. Edinburgh: Human Communication Research Centre, University of Edinburgh.

Carroll, John. 2003. Parsing. In Mitkov (ed.), 233–248.

Carstensen, Kai-Uwe, Ebert, Christian, Endriss, Cornelia, Jekat, Susanne, Klabunde, Ralf & Langer, Hagen (eds). 2004. *Computerlinguistik und Sprachtechnologie: Eine Einführung*, 2nd edn. Heidelberg: Spektrum Akademischer Verlag.

Chafe, Wallace. 1984. How people use adverbial clauses. In *Proceedings of the Tenth Annual Meeting of the Berkeley Linguistics Society*, Claudia Brugman & Monika Macaulay (eds), 437–449. Berkeley CA: BLS.

Chafe, Wallace. 1993. Prosodic and functional units of language. In Edwards & Lampert (eds).

Chafe, Wallace. 1995. Adequacy, user-friendliness, and practicability in transcribing. In Leech, Myers & Thomas (eds), 54–61.

Clark, Herbert H. 1996. *Using Language*. Cambridge: CUP. doi:10.1017/CBO9780511620539

Cloeren, Jan. 1999. Tagsets. In van Halteren (ed.), 37–54.

Collins, Peter. 2006. Clause types. In Aarts & McMahon (eds), 72–92. Oxford: Blackwell.

Collins, Peter. 2009. *Modals and Quasi-modals in English*. Amsterdam: Rodopi. doi:10.1163/9789042029095

Core, Mark, Ishizaki, Masato, Moore, Johanna, Nakatani, Christine, Reithinger, Norbert, Traum, David & Tutiya, Syun. 1998. *The Report of the Third Workshop of the Discourse Resource Initiative*. Chiba: Chiba University and Kazusa Academia Hall.

Coulthard, Malcolm (ed.). 1977. *An Introduction to Discourse Analysis*. London: Longman.

Coulthard, Malcolm (ed.). 1992. *Advances in Spoken Discourse Analysis*. London: Routledge.

Cresti, Emanuela. 2000. *Corpus di Italiano Parlato*. Firenze: Accademia delle Crusca.

Cresti, Emanuela. 2014. Syntactic properties of spontaneous speech in the language into act theory: Data on Italian complements and relative clauses. In Raso & Mello (eds), 365–410.

Crichton, Charles. (Dir.). 1988. *A Fish Called Wanda*. [Motion Picture]. UK/USA: Metro-Goldwyn-Mayer.

Croft, William & Cruse, D. Alan. 2004. *Cognitive Linguistics*. Cambridge: CUP. doi:10.1017/CBO9780511803864

Davies, Mark. 2009. The 385+ million word *Corpus of Contemporary American English* (1990–2008+): Design architecture, and linguistic insights. *International Journal of Corpus Linguistics* 14(2): 195–190. doi:10.1075/ijcl.14.2.02dav

DeRose, Steven. 1988. Grammatical category disambiguation by statistical optimization. *Computational Linguistics* 14(1): 31–39.

Edwards, Jane. 1993. Principles and contrasting systems of discourse transcription. In Edwards & Lampert (eds), 3–31.

Edwards, Jane & Lampert, Martin (eds). 1993. *Talking Data: Transcription and Coding in Discourse Research*. Hillsdale NJ: Lawrence Erlbaum Associates.

Evans, Vyvyan, Bergen, Benjamin & Zinken, Jörg (eds). 2007. *The Cognitive Linguistics Reader*. London: Equinox.

Fischer, Kerstin (ed.). 2006. *Approaches to Discourse Particles*. Amsterdam: Elsevier.

Fitzpatrick, Eileen (ed.). 2007. *Corpus Linguistics Beyond the Word: Corpus Research from Phrase to Discourse*. Amsterdam: Rodopi.

Flöck, Ilka & Geluykens, Ronald. 2015. Speech acts in corpus pragmatics: A quantitative contrastive study of directives in spontaneous and elicited discourse. In Romero-Trillo (ed.), 7–37.

Francis, Gill & Hunston, Susan. 1992. Analysing everyday conversation. In Coulthard (ed.), 123–161.

Francis, W. Nelson & Kučera, Henry. 1979. *Manual of information to accompany a standard sample of present-day edited American English, for use with digital computers*. First published 1964, revised and augmented 1979. Providence RI: Department of Linguistics, Brown University.

Fraser, Bruce. 1996. Pragmatic markers. *Pragmatics* 6(2): 167–190. doi:10.1075/prag.6.2.03fra

Fraser, Bruce. 1999. What are discourse markers? *Journal of Pragmatics* 31: 931–952. doi:10.1016/S0378-2166(98)00101-5

Fraser, Bruce. 2006. Towards a theory of discourse markers. In Fischer (ed.), 189–204.

Garside, Roger, Leech, Geoffrey & McEnery, Anthony (eds). 1997. *Corpus Annotation: Linguistic Information from Computer Text Corpora*. London: Longman.

Garside, Roger, Leech, Geoffrey & Sampson, Geoffrey (eds). 1987. *The Computational Analysis of English: A Corpus-based Approach*. London: Longman.

Geis, Michael. 1995. *Speech Acts and Conversational Interaction: Toward a Theory of Conversational Competence*. Cambridge: CUP. doi:10.1017/CBO9780511554452

Gibbon, Dafydd. 2000. Computational lexicography. In van Eynde & Gibbon (eds), 1–39.

Gibbon, Dafydd, Mertins, Inge, & Moore, Roger (eds). 2000. *Handbook of Multimodal and Spoken Language Systems*. Dordrecht: Kluwer Academic Publishers.

Gilquin, Gaëtanelle, De Cock, Sylvie, & Granger, Sylviane (eds). 2010. *Louvain International Database of Spoken Interlanguage (LINDSEI)*. Louvain: UCL Presses Universitaires de Louvain.

Godfrey, John & Holliman, Edward. 1997. Switchboard-1 Release 2. Philadelphia: Linguistic Data Consortium. <https://Catalog.ldc.upenn.edu/LDC97S62> (14 September 2017).

Goldberg, Adele. 2007[2003]. Constructions: A new theoretical approach to Language. In *The Cognitive Linguistics Reader*, Vyvyan Evans, Benjamn Bergen & Jorg Zinken (eds). London: Equinox.

Goodwin, Charles. 1996. Transparent vision. In Ochs, Schegloff & Thompson (eds), 370–404.

Gray, Bethany & Biber, Douglas. 2015. Stance markers. In Aijmer & Rühlemann (eds), 219–248.

Gregory, Michael & Carroll, Susanne. 1978. *Language and Situation*. London: Routledge and K. Paul.

Grice, H. Paul. 1989. *Studies in the Way of Words*. Cambridge MA: Harvard University Press.

Haiman, John & Thompson, Sandra A. (eds). 1988. *Clause Combining in Grammar and Discourse* [Typological Studies in Language 18]. Amsterdam: John Benjamins. doi:10.1075/tsl.18

Halliday, Michael & Hasan, Ruqaiya. 1976. *Cohesion in English*. London: Longman.

Haugeland, John. 1997. *Mind Design II*, 2nd edn. Cambridge, MA: The MIT Press.

Hazadiah, M. D. 1991. *The Structure of Topic*. PhD Thesis: University of Birmingham.

Heeman, Peter & Allen, James. 1995. The TRAINS 93 Dialogues. Technical Report. University of Rochester, Rochester NY, USA.

Heritage, John. 2015. *Well*-prefaced turns in English conversation: A conversation analytic perspective. *Journal of Pragmatics* 88: 88–104. doi:10.1016/j.pragma.2015.08.008

Hinrichs Erhard, Kübler, Sandra, Kordoni Valia & Müller Frank. 2000. Robust chunk parsing for spontaneous speech. In *Verbmobil: Foundations of Speech-to-Speech Translation*, Wolfgang Wahlster (ed.), 163–182. Berlin: Springer. doi:10.1007/978-3-662-04230-4_13

Hobbs, Jerry. 2006. Abduction in natural language understanding. In Horn & Ward (eds), 724–741.

Horn, Laurence & Ward, Gregory. 2006[2004]. *The Handbook of Pragmatics*. Oxford: Blackwell. doi:10.1002/9780470756959

Huang, Yan. 2007. *Pragmatics*. Oxford: OUP.

Huddleston, Rodney & Pullum, Geoffrey. 2002. *The Cambridge Grammar of the English Language*. Cambridge: CUP.

Hyland, Ken. 2005. Stance and engagement: A model of interaction in academic discourse. *Discourse Studies* 6(2): 173–191. doi:10.1177/1461445605050365

Ide, Nancy. 2007. Preparation and analysis of linguistic corpora. In Schreibmann, Siemens & Unsworth (eds), 289–305.

Jaworski, Adam & Coupland, Nikolas (eds). 2006. *The Discourse Reader*, 2nd edn. London: Routledge.

Jekat, Susanne, Klein, Marion, Maier, Elisabeth, Maleck, Ilona, Mast, Marion, Quantz, Joachim. 1995. Dialogue Acts in VERBMOBIL. VM-Report 65, DFKI GmbH.

Johansson, Stig. 1995. The approach of the Text Encoding Initiative to the encoding of spoken discourse. In Leech, Myers & Thomas (eds), 82–98.

Johansson, Stig, Leech, Geoffrey & Goodluck, Helen. 1978. *Manual of information to accompany the Lancaster-Oslo/Bergen corpus of British English, for use with digital computers*. Oslo: Department of English, Oslo University.

Jucker, Andreas. 1993. The discourse marker *well*: A relevance-theoretical account. *Journal of Pragmatics* 19(5): 435–452. doi:10.1016/0378-2166(93)90004-9

Jucker, Andreas. 2009. Speech act research between armchair, field and laboratory: The case of compliments. *Journal of Pragmatics* 41: 1611–1635. doi:10.1016/j.pragma.2009.02.004

Jucker, Andreas H. & Smith, Sara W. 1998. *And people just you know like 'wow'*: Discourse markers as negotiation strategies. In Jucker & Ziv (eds), 171–202.

Jucker, Andreas H. & Ziv, Yael (eds). 1998. *Discourse Markers: Description and Theory* [Pragmatics & Beyond New Series 57]. Amsterdam: John Benjamins. doi:10.1075/pbns.57

Jucker, Andreas H. & Ziv, Yael. 1998. Discourse markers: Introduction. In Jucker & Ziv (eds), 1–12.

Jurafsky, Daniel. 2006. Pragmatics and computational linguistics. In Horn & Ward (eds), 578–604.

Jurafsky, Daniel & Martin, James. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River NJ: Prentice Hall.

Jurafsky, Daniel, Shriberg, Liz & Biasca, Debra. 1997. Switchboard SWBD-DAMSL Shallow-Discourse-Function Annotation Coder Manual. <http://www.stanford.edu/~jurafsky/ws97/ics-tr-97-02.ps>

Kallen, Jeffrey & Kirk, John. 2012. *SPICE-Ireland: A User's Guide*. Queen's University Belfast, Trinity College Dublin, and Cló Ollscoil na Banríona.

Klein, Marion. 1999. Standardisation efforts on the level of dialogue act in the MATE Project. In *Proceedings of the ACL Workshop "Towards Standards and Tools for Discourse Tagging"*, 35–41.

Knight, Dawn, Evans, David, Carter, Ronald & Adolphs, Svenia. 2009. HeadTalk, HandTalk and the corpus: Towards a framework for multi-modal, multi-media corpus development. *Corpora* 4(1): 1–32. doi:10.3366/E1749503209000203

Knight, Dawn. 2011. *Multimodality and Active Listenership: A Corpus Approach*. London: Continuum.

Knowles, Gerald. 1987. *Patterns of Spoken English: An Introduction to English Phonetics*. London: Longman.

Leech, Geoffrey. 1983. *The Principles of Pragmatics*. London: Longman.

Leech, Geoffrey & Eyes, Elizabeth. 1997. Syntactic annotation: Treebanks. In *Corpus Annotation: Linguistic Information from Computer Text Corpora*, R. Garside, G. Leech & T. McEnery (eds). London: Longman.

Leech, Geoffrey, Myers, Greg & Thomas, Jenny (eds). 1995. *Spoken English on Computer: Transcription, Mark-up and Application*. London: Longman.

Leech, Geoffrey, Rayson, Paul & Wilson, Andrew. 2001. *Word Frequencies in Written and Spoken English*. London: Longman.

Leech, Geoffrey, Weisser, Martin, Wilson, Andrew & Grice, Martine. 2000. Survey and guidelines for the representation and annotation of dialogue. In *Handbook of Multimodal and Spoken Language Systems*, Dafydd Gibbon, Inge Mertins & Roger K. Moore (eds). Dordrecht: Kluwer.

Leech, Geoffrey & Weisser, Martin. 2003. Generic speech act annotation for task-oriented dialogues. In *Proceedings of the Corpus Linguistics 2003 Conference* [UCREL Technical Papers 16], Dawn Archer, Paul Rayson, Andrew Wilson & Tony McEnery (eds). Lancaster: Lancaster University.

Leech, Geoffrey & Weisser, Martin. 2013. The SPAADIA Annotation Scheme. http://martinweisser.org/publications/SPAADIA_Annotation_Scheme.pdf

Leicher, Sheryl & Maynard, Carson. 2007. Pragmatic annotation of an academic spoken corpus for pedagogical purposes. In Fitzpatrick (ed.), 107–115.

Levinson, Stephen. 1983. *Pragmatics*. Cambridge: Cambridge University Press.

McCarthy, Michael. 1998. Taming the spoken language: Genre theory and pedagogy. *The Language Teacher Online*. <http://www.jalt publications.org/tlt/files/98/sep/mccarthy.html> (1 September 2009).

Mann, William & Thompson, Sandra. 1987. Rhetorical structure theory: Description and construction of text structures. In *Natural Language Generation: New Results in Artificial Intelligence, Psychology and Linguistics*, Gerard Kempen (ed.). Dordrecht: Martinus Nijhoff. doi:10.1007/978-94-009-3645-4_7

Manning, Christopher & Schütze, Hinrich. 1999. *Foundations of Statistical Natural Language Processing*. Cambridge MA: The MIT Press.

Marcus, Mitchell, Santorini, Beatrice & Marcinkiewicz, Mary Ann. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* 19. (page references refer to downloaded version).

Marshall, Ian. 1987. Tag selection using probabilistic methods. In Garside, Leech & Sampson (eds), 42–56.

Matthiesen, Christian M. I. M. & Thompson, Sandra A. 1988. The structure of discourse and 'subordination'. In Haiman & Thompson (eds), 275–330.

McEnery, Tony & Wilson, Andrew. 1996. *Corpus Linguistics*. Edinburgh: Edinburgh University Press.

McEnery, Tony, Xiao, Richard & Tono, Yukio. 2006. *Corpus-Based Language Studies: An Advanced Resource Book*. London: Routledge.

Mey, Jacob. 1993. *Pragmatics: An Introduction*. Oxford: Blackwell.

Miller, James & Weinert, Regina. 1998. *Spontaneous Spoken Language: Syntax and Discourse*. Oxford: OUP.

Minsky, Marvin. 1974/1997. A framework for representing knowledge. Reprinted in *Mind Design II*, John Haugeland (ed.). Cambridge MA: The MIT Press.

Mitkov, Ruslan (ed.). 2003. *The Oxford Handbook of Computational Linguistics*. Oxford: OUP.

Moneglia, Massimo. 2011. Spoken corpora and pragmatics. *Revista Brasileira de Linguística Aplicada* 11(2): 479–519.

Nelson, Gerald. 2002. *International Corpus of English Markup Manual for Spoken Texts*. London: University College London, Survey of English Usage. <http://ice-corpora.net/ice/manuals.htm> (12 November 2016).

Niemann, Heinrich, Nöth, Elmar, Harbeck, Stefan & Warnke, Volker. 1997. Topic Spotting Using Subword Units. Verbmobil-Report 205. F.A.U. Erlangen-Nürnberg.

Ochs, Elinor, Schegloff, Emanuel A. & Thompson, Sandra A. (eds). 1996. *Interaction and Grammar*. Cambridge: CUP. doi:10.1017/CBO9780511620874

O'Keeffe, Anne, Clancy, Brian & Adolphs, Svenja. 2011. *Introducing Pragmatics in Use*. London: Routledge.

Ooi, Vincent. 1998. *Computer Corpus Lexicography*. Edinburgh: Edinburgh University Press.

Palmer, Frank. 1990. *Modality and the English Modals*, 2nd edn. London: Longman.

Palmer, Frank. 2001. *Mood and Modality*, 2nd edn. Cambridge: CUP. doi:10.1017/CBO9781139167178

Pollard, Carl & Sag, Ivan. 1994. *Head-Driven Phrase Structure Grammar*. Chicago IL: University of Chicago Press.

Power, Richard. 1979. The organization of purposeful dialogues. *Linguistics* 17(1–2): 105–152.

Quirk, Randolph, Greenbaum, Sidney, Leech, Geoffrey & Svartvik, Jan. 1985. *A Comprehensive Grammar of the English Language*. London: Longman.

Raso, Tommaso & Mello, Heliana (eds). 2014. *Spoken Corpora and Linguistic Studies* [Studies in Corpus Linguistics 61]. Amsterdam: John Benjamins.

Roark, Brian & Sproat, Richard. 2007. *Computational Approaches to Morphology and Syntax*. Oxford: OUP.

Romero-Trillo, Jesús (ed.). 2013. *Yearbook of Corpus Linguistics and Pragmatics 2013: New Domains and New Methodologies*. Berlin: Springer. doi:10.1007/978-94-007-6250-3

Romero-Trillo, Jésus (ed.). 2015. *Yearbook of Corpus Linguistics and Pragmatics 2015: Current Approaches to Discourse and Translation Studies*. Berlin: Springer. doi:10.1007/978-3-319-17948-3

Rühlemann, Christoph. 2006. Coming to terms with conversational grammar: 'Dislocation' and 'dysfluency'. *International Journal of Corpus Linguistics* 11(4): 385–409. doi:10.1075/ijcl.11.4.03ruh

Sacks, Harvey, Schegloff, Emanuel A. & Jefferson, Gail. 1974. A simplest systematics for the organization of turn-taking for conversation. *Language* 50(4): 697–735. doi:10.1353/lan.1974.0010

Salton, Gerard & Buckley, Christopher. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management* 24(5): 513–533. doi:10.1016/0306-4573(88)90021-0

Santorini, Beatrice. 1995. *Part of Speech Tagging Guidelines for the Penn Treebank Project*, 3rd revision, 2nd printing. <ftp://ftp.cis.upenn.edu/pub/treebank/doc/tagguide.ps.gz>

Schank, Roger & Abelson, Robert. 1977. *Scripts, Plans, Goals, and Understanding*. Hillsdale NJ: Lawrence Erlbaum Associates.

Schiffrin, Deborah. 1987. *Discourse Markers*. Cambridge: CUP. doi:10.1017/CBO9780511611841

Schiffrin, Deborah. 1994. *Approaches to Discourse*. Oxford: Blackwell.

Schreibmann, Susan, Siemens, Ray & Unsworth, John (eds). 2007. *A Companion to Digital Humanities*. Oxford: Blackwell.

Searle, John. 1969. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge: CUP. doi:10.1017/CBO9781139173438

Searle, John. 1979. *Expression and Meaning: Studies in the Theory of Speech Acts*. Cambridge: CUP. doi:10.1017/CBO9780511609213

Sells, Peter. 1985. *Lectures on Contemporary Syntactic Theories: An Introduction to Government-Binding Theory, Generalized Phrase Structure Grammar, and Lexical-Functional Grammar* [Lecture Notes 3]. Stanford CA: CSLI.

Shannon, Claude. 1948. A mathematical theory of communication. *The Bell System Technical Journal* 27: 379–423, 623–656, July, October 1948. page no. refers to reprint downloadable from <http://plan9.bell labs.com/cm/ms/what/shannonday/shannon1948.pdf> (9 July 2008).

Shannon, Claude & Weaver, Warren. 1963. *The Mathematical Theory of Communication*. Chicago IL: University of Illinois Press.

Simpson, Rita, Briggs, Sarah, Ovens, Janine & Swales, John. 2002. *The Michigan Corpus of Academic Spoken English*. Ann Arbor MI: The Regents of the University of Michigan.

Sinclair, John. 1992. Priorities in discourse analysis. In Coulthard (ed.), 79–88.

Sinclair, John. 2005. Corpus and text – Basic principles. In *Developing Linguistic Corpora: A Guide to Good Practice*, Martin Wynne (ed.), 1–16. Oxford: Oxbow Books: <http://ota.ox.ac.uk/documents/creating/dlc/chapter1.htm> (26 October 2016).

Sinclair, John & Coulthard, Malcolm. 1975. *Towards an Analysis of Discourse*. Oxford: OUP.

Sinclair, John & Mauranen, Anna. 2006. *Linear Unit Grammar: Integrating Speech and Writing* [Studies in Corpus Linguistics 25]. Amsterdam: John Benjamins. doi:10.1075/scl.25

Sperber, Dan & Wilson, Deirdre. 1995. *Relevance*, 2nd ed. Oxford: Blackwell.

Sperberg-McQueen, Michael & Burnard, Lou (eds). 1994. *Guidelines for Electronic Text Encoding and Interchange (TEI P3)*. Chicago IL: Association for Computers and the Humanities/Association for Computational Linguistics/Association for Literary and Linguistic Computing.

Sproat, Richard. 1992. *Morphology and Computation*. Cambridge MA: The MIT Press.

Stenström, Anna-Brita. 1994. *An Introduction to Spoken Interaction*. London: Longman.
Svartvik, Jan (ed.). 1990. *The London-Lund Corpus of Spoken English: Description and Research*. Lund: Lund University Press.
Szabó, Zoltán (ed.). 2005. *Semantics versus Pragmatics*. Oxford: OUP. doi:10.1093/acprof:oso/9780199251520.001.0001
Taylor, Jasper & Carletta, Jean. 1996. The SGML Representation of the HCRC Map Task Corpus. Technical Report, Human Communication Research Centre, 2 Buccleugh Place, Edinburgh EH8 8LW, Scotland.
van Eynde, Frank & Gibbon, Dafydd. 2000. *Lexicon Development for Speech and Language Processing*. Dordrecht: Kluwer. doi:10.1007/978-94-010-9458-0
van Halteren, Hans (ed.). 1999. *Syntactic Wordclass Tagging*. Dordrecht: Kluwer Academic Publishers. doi:10.1007/978-94-015-9273-4
Verhagen, Arie. 2005. *Constructions of Intersubjectivity: Discourse, Syntax, Cognition*. Oxford: OUP.
Verschueren, Jef. 1999. *Understanding Pragmatics*. London: Arnold.
Weisser, Martin. 2002. Determining generic elements in dialogue. *Language, Information and Lexicography* 12–13: 131–156. 25th December, 2003. Institute of Language and Information Studies, Yonsei University.
Weisser, Martin. 2003. SPAACy – A semi-automated tool for annotating dialogue acts. *International Journal of Corpus Linguistics* 8(1): 63–74. doi:10.1075/ijcl.8.1.03wei
Weisser, Martin. 2004. Tagging dialogues in SPAACy. In *Traitement Automatique des Langues: Le traitement automatique des corpus oraux*, Jean Véronis (ed.), Vol. 45–n° 2/2004. 131–157.
Weisser, Martin. 2009. *Essential Programming for Linguistics*. Edinburgh: Edinburgh University Press.
Weisser, Martin. 2010. Annotating Dialogue Corpora Semi-Automatically: A Corpus-Based Approach to Pragmatics. Habilitation thesis, University of Bayreuth.
Weisser, Martin. 2015. Speech act annotation. In *Corpus Pragmatics: A Handbook*, Karin Aijmer & Christoph Rühlemann (eds), 84–113. Cambridge: CUP. doi:10.1017/CBO9781139057493.005
Weisser, Martin. 2016a. *Practical Corpus Linguistics: An Introduction to Corpus-Based Language Analysis*. Oxford: Wiley-Blackwell. doi:10.1002/9781119180180
Weisser, Martin. 2016b. DART – The dialogue annotation and research tool. *Corpus Linguistics and Linguistic Theory* 12(2): 355–388. doi:10.1515/cllt-2014-0051
Weisser, Martin. 2016c. Profiling agents & callers: A dual comparison across speaker roles and British vs. American English. In *Talking at Work: Corpus-based Explorations of Workplace Discourse*, Lucy Pickering, Eric Friginal & Shelley Staples (eds), 99–26. London: Palgrave Macmillan. Chapter doi:10.1057/978-1-137-49616-4_5
Weisser, Martin. 2016d. Updating the LINDSEI for pragmatic analysis and evaluation. Paper presented at the *Third Asia Pacific Corpus Linguistics Conference (APCLC 2016)*, on 22nd October.
Weisser, Martin. 2017. Annotating the ICE Corpora pragmatically – Preliminary issues & steps. *ICAME Journal* 41: 181–214. doi:10.1515/icame-2017-0008
West, Michael. 1953. *A General Service List of English Words*. London: Longman.
Wichmann, Anne. 2000. *Intonation in Text and Discourse: Beginnings, Middles, and Ends*. Harlow: Pearson Education.
Wilson, Deirdre & Sperber, Dan. 2006. Relevance Theory. In Horn & Ward (eds), 607–632.

Index

This book introduces a methodology and research tool (DART) that make it possible to carry out advanced corpus pragmatics research using dialogue corpora enriched with pragmatics-relevant annotations. It first explores the general use of spoken corpora for pragmatics research, as well as issues revolving around their representation and annotation, and then goes on to describe the resources required for such an annotation process. Based on data from three different corpora, ranging from highly constrained, task-oriented, ones (SPAADIA Trainline & Trains 93) to unconstrained dialogues (Switchboard), it next presents an in-depth discussion and illustration of the potential contributions of syntax, semantics, and semantico-pragmatics towards pragmatic force. This is followed by a description of the largely automatic annotation process itself, and finally an analysis of how a set of more than 110 potential speech acts defined in DART contributes towards establishing the specific communicative characteristics of the three corpora.

ISBN 978 90 272 0047 1

9 789027 200471

JOHN BENJAMINS PUBLISHING COMPANY