3 Speech act annotation

Martin Weisser

3.1 Introduction

Any type of linguistic annotation is a highly complex and interpretive process, but none more so than pragmatic annotation. The complexity inherent in this task (Leech et al. 2000) is mainly due to the fact that this type of annotation, unlike, for example, POS (part-of-speech) or semantic tagging/annotation, almost always needs to take into account levels above the individual word and may even need to refer to contextual information beyond those textual units that are commonly referred to as a 'sentence' or 'utterance'. This is also why, in order to be able to produce large-scale and consistently annotated corpora, it is highly advisable to automate such an annotation process.

This chapter, apart from providing an overview of, and guidelines for, representing and annotating pragmatic and other, 'pragmatics-relevant' information, will do so in the form of research into which means may be available to improve existing techniques (e.g. Stolcke et al. 2000 and Weisser 2002, 2003, 2004) for manual and automatic pragmatic annotation of corpora from different domains. One major part of this will be to try and determine the extent to which it is possible to improve the identification and subsequent annotation of pragmatics-related phenomena, based on insights gained from attempts at manual and automatic annotation. In so doing, I shall draw on a small set of materials from a number of annotated corpora that are freely available online, conducting an in-depth analysis of one file from each corpus (filenames provided below), with additional data from the corpora being consulted as and when necessary.

The corpora used are the Trains 93 corpus (Heeman and Allen 1995), the Coconut Corpus (Di Eugenio et al. 1998), the Monroe Corpus (Stent 2000), and the Switchboard Corpus (Godfrey and Holliman 1997).[1] The first three of these represent corpora of *task-driven* dialogues (Leech et al. 2000: 7), where two interlocutors need to work together to achieve a particular task, such

[1] The specific filenames are: d92a-3.1.sgml (Trains 93), s2rec.sgml (Monroe), dave-greg.1.damslcoco.dieugeni.nb (Coconut), and sw_0001_4325.utt (Switchboard).

as transporting goods by train to various places (Trains 93), buying furniture for living and dining rooms on a specific budget (Coconut), and handling emergency situations of various types (Monroe). In contrast, Switchboard is a corpus of approximately 2,400 spontaneous telephone conversations by native speakers of American English, revolving around 52 topics that range from AIDS, via drug testing, gun control, music, to woodworking, and thus cover a wide variety of different everyday-life interests. The version used here is an early version[2] before its official release through the Linguistic Data Consortium. The important difference between the former three corpora and the latter is also that task-driven dialogues tend to be more constrained in the range of options available for interaction by the interlocutors, and therefore easier to model, while unconstrained conversation is generally deemed to be more difficult to handle. I will use the actual application of the marking schemes to these corpora to illustrate how different research projects have 'interpreted' these schemes, and then compare those annotations to automatically annotated data created using my own research tool DART (Weisser 2014).

Another aim will be to try and gain some new insights into whether it may be possible to refine the inferencing process involved in this annotation process (Weisser 2004) in order to gradually move away from using only high-level or relatively neutral generic speech act labels (Allen and Core 1997, Weisser 2004) towards a more precise labelling of speaker intentions. For instance, while an utterance such as *We'll be there at five o'clock* can relatively easily be interpreted by the computer purely as a prediction, which is certainly applicable in domains where arrivals are discussed more 'neutrally', such as in logistical types of planning, in other domains or circumstances it ought to be interpreted as a prediction in combination with a promise. Finally, attempts will be made to point out how the author's annotation approach, which has so far mainly been developed for the annotation of task-oriented domains in English, could be applied to that of wider, less restricted, domains.

Before discussing the research-oriented issues outlined above, however, it is necessary to provide an overview of the features of spoken interaction that may be relevant in representing and annotating spoken interaction, which will be given in the next section. The chapter will then continue with a comparison of the different annotation schemes and their relative merits, followed by a brief case study demonstrating the usefulness of pragmatic annotation, and then end with a conclusion about the current state of development in pragmatic annotation schemes and future prospects.

[2] Available free of charge from www.stanford.edu/~jurafsky/swb1_dialogact_annot.tar.gz.

3.2 Basic representation of dialogues

3.2.1 Basic markup format

When annotating any corpus data, it is first necessary to create a basic *representation* of the data in the form of a transcription that provides enough detail for the features to be annotated, as well as some structural *markup* that allows the relevant units to be recognised. Today, more and more corpora are stored in some form of *XML* (eXtensible Markup Language) format (W3C 2010). It is interesting to note, however, that all of the annotated corpora I refer to in my discussion are either still encoded in *SGML* (Standard Generalized Markup Language: Coconut, Monroe) or in even simpler *plain-text*-based annotation formats (Switchboard). Other corpora, such as the MICASE corpus (Michigan Corpus of Academic Spoken English: Simpson et al. 2002), for example, are frequently marked up according to the guidelines of the *Text Encoding Initiative* (TEI). However, for many purposes, following these guidelines and including all the *markup* required to comply with them represents a kind of overhead that may not be warranted for smaller research projects, and may also make reading and processing the materials to be annotated more difficult. I will therefore begin here by proposing a much more simplified form of basic XML *representation*. This will later be expanded upon in the relevant sections that deal with the different types or levels of *annotation* we may want to add to such data.

The discussion concerning pragmatic data here will focus exclusively on dialogues, although of course we should assume that similar syntactic/functional units to the ones described here also occur in monologues, as well as to some extent even in written language. The basic framework for representing pragmatic data here consists of the highly simplified XML structure illustrated in Figure 3.1. Each syntactic unit within a turn can later be wrapped in one of the syntactic unit tags discussed further below.

3.2.2 Identifying and segmenting pragmatic units for analysis

Amongst the functional units in pragmatic annotation, perhaps the least contentious is the dialogue itself because it can usually be defined by a clear beginning and ending, unless of course we are dealing with an extract or a dialogue that is incomplete, due to interruption. This represents the top level in the hierarchical organisation, as marked through the dialogue element referred to above.

Immediately below this level, we find the individual speaker turns. Turns are generally defined as units of speech where one particular speaker talks for an uninterrupted period of time and then yields the floor to an interlocutor (Sacks et al. 1974: 699), but of course there are also many less straightforward cases where turns either 'include' *backchannels* (which may appear strange at first, but see below), overlapping speech, or non-verbal communication (e.g. in

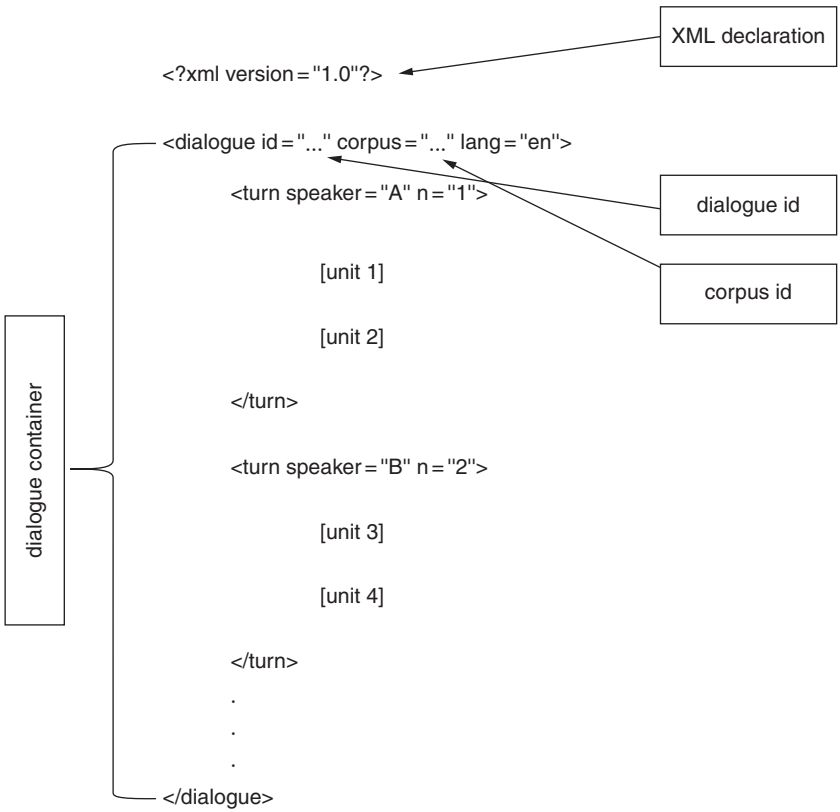

Figure 3.1 XML representation of dialogue structure.

the form of laughter) produced by the other dialogue participant(s), which may represent certain problems for an automated annotation. An incorrectly marked up backchannel is presented in the following example:

(1) Backchannel incorrectly treated as a separate turn

<turn n=" 17" speaker="u">
right <pause /> **load**
</turn>
<turn n=" 18" speaker="s">
mm…hm
</turn>
<turn n=" 19" speaker="u">
three cars of
</turn>

(Trains d93–20.3)

In the above example, whose XML representation was generated from the original Trains 93 transcription, we can see that speaker *s* contributes to the dialogue through a 'semantically empty' backchannel that is purely phatic in nature. However, if we treated this as a separate turn, as the original transcribers did, it would have to be seen as an interruption of *u*'s turn, as well as of a syntactic unit that could otherwise be identified as an incomplete imperative (*load three cars of . . .*). In terms of the level of politeness existing between the interlocutors, who are 'equal partners' in solving a logistics problem transporting orange juice from one place to another, it would in consequence also have to be interpreted as highly rude if it were indeed an interruption because this would mean that *s* had made an attempt at taking over the turn at an unsuitable *transition relevance place*, or TRP (Sacks et al. 1974: 703). As this is clearly not the case, but this backchannel instead represents a clear case of co-operative and supportive behaviour designed to 'encourage' *u*, it makes far more sense to treat the whole sequence above as one single turn by *s*, which simply contains a backchannel, and which we could more correctly represent as follows:

(2) Corrected representation of a backchannel

```
<turn n=" 17" speaker="u">
 right <pause /> load <backchannel content="mm. . .hm" /> three
 cars of
</turn>
```

(Trains d93–20.3)

Handling backchannels as independent turns seems to be a common problem in most of the annotations in the data I investigated. This may – on the one hand – simply imply a lack of understanding of the functions of backchannels on the part of the transcribers, which needs to be corrected, but it may – on the other hand – represent something that could cause substantial problems for an automated analysis. The solution suggested above, using a so-called 'empty' XML element with a content attribute containing the backchannel, unmistakably signals that it is a form of insertion, rather than an independent unit. In a similar way, longer sections of *overlapping*, a general problem in XML-based annotation, can also be represented by using empty elements containing an indication of the overlapping speaker, respecting the fact that it generally constitutes material that represents a valid turn in its own right.

Below the level of the turn, at the *micro-level*, the situation becomes rather more difficult to describe because different annotation 'traditions' use different labels that may refer to units of varying sizes and functional load. The term *utterance* has common currency in the research community (Leech et al. 2000: 56), but is rather vague and may sometimes refer to units defined

more on prosodic grounds (e.g. *tone-units*), sometimes more on syntactic, semantic, or pragmatic grounds (e.g. *slash-units*, Taylor 1995: 16; *c-units*, Biber et al. 1999: 1070), or purely pragmatic grounds (e.g. *speech acts* or *moves*). Essentially, the problem here is to try and identify an ideal form–function mapping and to see how this can then be incorporated into an annotation scheme. The solution adopted on the SPAAC (Speech-Act Annotated Corpus of Dialogues) project (Leech and Weisser 2003) was to assume that c-units provide a good syntactic basis that may be enriched with information on various other levels, and this is the approach I shall recommend here. We shall see later why such an approach may have distinct advantages over other proposals.

What all approaches, regardless of what they call their respective units of annotation, have in common is that they assume that all 'semantically complete' units, even if they consist of syntactic fragments (e.g. single NPs that answer questions), should have a meaning and pragmatic function that is largely independent of the surrounding meanings and is thus also worth labelling individually. As we have seen above, the issue of identifying the 'right' units is by no means uncontroversial and goes hand-in-hand with that of signalling unit-types such as declaratives versus interrogatives, or prosody in transcriptions through the use of punctuation marks or other indicators. Archer et al. (2008: 647) point out that 'the use of some punctuation marks, e.g. full stop, question mark and ellipsis as sentence delimiter . . . can facilitate further processing', but there still seems to be some reservation about incorporating punctuation into most dialogue corpora available. Recognising the usefulness of such information in order to mark up prosodic information or completeness status, I would propose a tentative, simplified, six-way distinction that employs empty <punct /> tags with a *type* attribute, where the attribute may take on one of the following values:

(i) *comma* exclusively indicates items in a list,
(ii) *stop* signifies final 'declarative' intonation,
(iii) *exclam* indicates an exclamatory nature,
(iv) *query* marks different relatively neutral forms of 'interrogative',
(v) *unsure* signals certain types of 'incredulity' that are often expressed in the form of a fall–rise contour, for instance with the discourse marker *really*,
(vi) and *level* identifies a non-final and non-interrogative prosody that indicates a 'trailing off' or 'please hold' pattern.

Moving back 'upwards' from the micro-level, we can then possibly identify and group together *meso-* or *macro-level* structures or stages within the dialogue, such as *adjacency pairs*, *IRF* (initiation–response–feedback) *sequences*, or *games* and *transactions*, but this is really only possible once we have identified the units at the micro-level in a bottom-up approach, so this will be discussed next.

3.2.3 Existing annotation schemes

Computer-based pragmatics is still largely a developing field, and as yet there exist no real commonly agreed-upon standards as to how exactly this type of research should be conducted, nor exactly which features ought to be annotated. However, there have at least been attempts to define some of the levels and units of annotation/analysis that are required 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 between the years of 1995 and 1998, which led to the development of an annotation scheme called DAMSL (*Dialogue Act Markup in Several Layers*: Allen and Core 1997). Other attempts at reporting on or defining best practice standards in this area are those of Leech et al. (2000) within the EAGLES (*Expert Advisory Group on Language Engineering Standards*) framework and as part of the MATE (*Multilevel Annotation, Tools Engineering*: Klein 1999) project. One of the latest efforts at establishing an ISO (International Organization for Standardization) standard for dialogue annotation, DiAML (*Dialogue Act Markup Language*), unfortunately seems to be going in the wrong direction, at least conceptually from the point of view of a linguist, as Bunt et al. (2010: 2548; my emphasis) 'take a more specific, ***semantic view on dialogue acts as corresponding to update operations on information states***', rather than identifying pragmatically relevant features. Within the field of corpus linguistics, attempts at describing pragmatic annotation (e.g. Archer et al. 2008) have so far remained more on the theoretical level, rather than demonstrating how this can be achieved in practice, a shortcoming I shall try to rectify in this chapter.

One of the major foci in any kind of pragmatic annotation, and the one I will concentrate on here, should be that of identifying and marking up *speech acts* that (roughly) indicate the intentions and interaction strategies of interlocutors. More recently, and predominantly in computer-oriented research, the term *dialogue acts* has also come into vogue (Leech et al. 2000: 6), which, according to Jurafsky (2006: 588), is often understood 'to mean an act with internal structure related specifically to its dialogue function'. However, since this term does not really constitute an improvement over the traditional term *speech act*, and the latter still has more currency with researchers in pragmatics, it is probably preferable to retain the old term.

In the following section, I shall discuss some of the more influential and useful schemes for annotating dialogues, illustrating both their coverage and application, as well as pointing out their potential drawbacks, and providing alternative labelling strategies as and when applicable. To save space, when

no in-depth discussion is required, I shall simply provide the alternative label used in the DART scheme in brackets, prefixed by 'DART'. In order to facilitate further direct comparison between the major annotation schemes discussed below, Table 3A.1 in the Appendix contains a comparison chart, based on examples of the most common/frequent speech act types from Jurafsky et al. (1997).

3.3 A comparison of three main schemes (DAMSL, SWBD DAMSL, and DART)

The DAMSL annotation manual, the highly influential outcome of trying to establish a consensus model during the first two meetings of the DRI, comprises a set of different content levels, or *dimensions* (Allen and Core 1997: 4), to be applied in the form of *utterance-tags*, namely the following:

(i) communicative status,
(ii) information level and status,
(iii) forward-looking communicative function,
(iv) backward-looking communicative function.

These dimensions are assumed to contribute to the overall function of the syntactic unit in different ways, and each unit may potentially be marked up on all four levels to describe its function(s) exhaustively, but need not necessarily contain tags at all of them if this is not appropriate (Allen and Core 1997: 4). Switchboard (SWBD) DAMSL (henceforth SWBDD) already simplifies this system by explicitly assuming that if a unit should be tagged using any labels from the dimension *communicative status*, then it should no longer be annotated according to any of the other dimensions, thereby reducing the dimensions for all other, communication-relevant, units by one. This makes sense because the dimension marks whether a unit is complete or abandoned/interrupted (DART: *abandon*), and intelligible or unintelligible, or includes information concerning features of contextual relevance, such as *self-talk*. Apart from the special role of dimension (i), all other dimensions seem to be regarded as having equal status, which can be confusing if labels for different levels are applied at the same time (see below for examples), but none of them is in any sense 'prioritised'.

3.3.1 Communicative status and basic annotation formats

The markup format applied to the annotated dialogues using basic DAMSL variants is SGML. Both Trains and Monroe here follow a format where independent units are marked up as utterances, using a single <Utt> tag containing the relevant DAMSL attributes, unit *id*s, as well as references

to the corresponding audio files, while Coconut surrounds each of these units by a number of individual SGML tags representing DAMSL and other additional features:

(3) Trains SGML annotation

<Turn Id=T1 Speaker="s" Speech="-s 0.186421 -e 1.2">
<Utt Id=utt1 Conventional=Opening Info-level=Communication-management
Speech="-s 0.186421 -e 0.651656"> **hello** [sil]
<Utt Id=utt2 Conventional=Opening Influence-on-listener=Info-request-directive Influence-on-speaker=Offer Info-level=Communication-management Speech="-s 0.32 -e 1.2"> **can I help you**

(4) Coconut SGML annotation

<Information-Level_74="Task"><Forward-Comm-Function_91="Influence-on-Listener Directive Action-Directive">
<Forward-Comm-Function_92="Influence-on-Speaker Commit">
<Topic_64><Word-Surface-Features_62="no Matrix lets we">
<Syn-Surface-Features_63="pres imperative">
<Backward-Comm-Function_104="Initiate">[78]: **lets do it**
"</Backward-Comm-Function_104>
</Syn-Surface-Features_63></Word-Surface-Features_62>
</Topic_64></Forward-Comm-Function_92>
</Forward-Comm-Function_91></Information-Level_74>

As can easily be seen from the above examples, the format employed in example (4) is more confusing and, to become properly legible, requires rendering by a dedicated annotation program. Even worse, in Coconut the numbers linked to the tags via underscores do not correspond to the individual textual units, but only enumerate the *n*th occurrence of each tag. While the overall Coconut annotation scheme thus appears less useful, its one advantage is that it incorporates information about the syntactic nature of the relevant unit, which opens up the possibility of investigating correlations between syntactic form and pragmatic function. For instance, searching for imperatives in combination with *let's* (unfortunately frequently incorrectly spelt in Coconut as in example (4)) would quickly reveal that 'let's imperatives' in general do not represent 'Action-Directive(s)', but instead suggestions for future actions. One major disadvantage in the Trains and Monroe output format is that the individual attributes do not always occur in the same positions inside the tags, which makes reading the annotated dialogues more confusing than necessary. This problem may arise because the particular annotation program used to assist the annotation process did not force the annotators to be consistent.

The SWBD-DAMSL format is already far more compact and succinct because essentially all the different dimension attributes are collapsed into one 'speech act' label, as shown in the following example:

(5) SWBD-DAMSL non-SGML annotation

qy A.25 utt1: {C But,} does your sister live in a big community? /
ny B.26 utt1: {F Uh,} yeah /
sd^e B.26 utt2: she lives, – /
sd B.26 utt3: [it's a, + it's a] fairly large community. /[3]

(sw_0001_4325.utt)

Here, the labels *qy* and *ny* refer to a yes/no-question and its corresponding yes-answer, while *sd* marks general declarative statements (as opposed to opinions). The 'tag' ^e indicates an elaboration to the yes-answer. This focus on marking the speech act as the main 'dimension', in combination with the partial integration of syntactic information, already represents an improvement, as the main aim of the annotation is to identify the pragmatic force of the unit.

In order to cater for the annotation of different linguistic levels, as well as to make it possible to establish form–function correlations, the DART scheme replaces the <Utt> element of the DAMSL annotations by more meaningful syntactic element categories. These not only include the traditional declaratives (<decl>), interrogatives (yes/no-questions: <q-yn>; wh-questions: <q-wh>) and imperatives (<imp>), but also terms of address (<address>), discourse markers (<dm>), exclamations (<exclam>), yes-responses (<yes>), no-responses (<no>), and fragments (<frag>), that is, syntactically ungrammatical or incomplete/elliptical syntactic units, all of which tend to be of special importance in spoken interaction. This scheme thus takes into account the lexico-grammatical potential expressed through the different types of syntactic categories, and augments this information through attributes signalling pragmatics (*speech acts*), semantico-pragmatic markers (*modes*), semantics (*topics*), as well as (surface-)polarity. In this way, not only does it become easier to distinguish between these levels but it also becomes easier to model them separately in order to make a purely linguistically motivated, i.e. non-probabilistic,[4] automatic analysis possible. A brief example of what the SWBDD example from above looks like in DART is as follows:

(6) SWBD example represented in DART format

<turn n="25" speaker="A">

[3] The different types of brackets here represent disfluency phenomena annotated in the original Switchboard materials and are not part of the pragmatic annotation. In subsequent examples, these will be omitted for the sake of clarity.

[4] For more information on the unfortunately more prevalent probabilistic methods used in dialogue act recognition, see Jurafsky (2006).

```
<q-yn n="48" sp-act="reqInfo" polarity="positive" mode="constrain-
closed-query">
but does your sister live in a big community <punc type="query" />
</q-yn>
</turn>
<turn n="26" speaker="B">
<yes n="49" sp-act="answer-acknowledge">
uh yeah <punc type="stop" />
</yes>
<decl n="50" sp-act="elab-abandon" polarity="positive" mode=
"abandon">
she lives <punc type="incomplete" />
</decl>
<decl n="51" sp-act="elab-state" polarity="positive" mode="decl">
it's a <comment type="restart" /> it's a fairly large community
<punc type="stop" />
```

As we can see from the above, linguistically relevant information, such as the syntactic type or the speech act, is made explicit and easier to digest in this system than in systems where information is 'hidden' behind dimensions that are less linguistically motivated.

3.3.2 Information level and status categories

DAMSL's original design for marking up transactional dialogues is reflected in the sub-categories for *information level and status*. *Task* is supposed to refer to parts of the dialogue whose function is simply to achieve the task at hand, while *Task-management* describes ways of managing the interaction at a level that is purely concerned with the task, such as keeping track of the stages of the task and particular steps they may involve. The third one, *Communication-management*, essentially comprises features such as the use of discourse markers (henceforth DMs), formulaic expressions such as greetings (DART: *greet*) and farewells (DART: *bye*), and holding directives (DART: *hold*). The fourth level, called *Other-level*, seems to be more or less a garbage category that may, according to its description (Allen and Core 1997: 8), also contain features such as small talk, reminiscent of some of those already discussed under communication status above. In SWBDD, Task is assumed to be the (unmarked) default, but *Task-management* and *Communication-management* can be indicated by the 'tags' ^t and ^c, attached to the 'speech act' label, respectively.

The following brief discussion of how the Info-level dimension is in fact used in the three DAMSL-annotated corpora will hopefully shed some light on the highly interpretive nature of the scheme. In both Trains and Monroe, *Communication-management* attribute values generally tend to be associated with acknowledging DMs (DART: *acknowledge*), such as *ok* or *right*, essentially in accordance with the DAMSL specifications. However, in both corpora, such DMs

may occasionally also be marked as 'Info-level=Task', especially when they are not explicitly separated from other utterance parts that contain statements, or simply used inconsistently as in the following two examples[5] from Trains:

(7) Trains Info-level

<Utt Id=utt5 Info-level=Task Statement=Assert> and the time now is midnight
<Turn Id=T3 Speaker="s"><Utt Id=utt6 Agreement=Accept ***Info-level=Task*** Response-to="utt4 utt5" Understanding=SU-Acknowledge> ***okay***

(8) Trains Info-level

<Utt Id=utt7 Info-level=Task Statement=Assert>
the [sil] orange warehouse where I need [sil] the oranges from is in [sil] Corning
<Turn Id=T5 Speaker="s">
<Utt Id=utt8 Agreement=Accept ***Info-level=Communication-management*** Response-to="utt7" Statement=None Understanding=SU-Acknowledge> ***[click] [sil] right***

Monroe also seems to mark up DM-like units, such as *I'm sorry*, exclamations, and units where *s* temporarily suspends the interaction in order to evaluate or find information (e.g. *just a second*), as well as confirmatory repeats/rephrases as *Communication-management*, thus exhibiting a justifiably wider scope for that dimension than Trains because these units clearly do contribute towards managing the communication. In Monroe, an additional *None* category is also used for non-verbal responses, such as laughter, although these frequently do occur as acknowledging responses that would certainly merit their being labelled as *Communication-management*.

All three corpora additionally make use of the *Task-management* value, but this is frequently also used in cases where *Communication-management* may equally well be applicable, which indicates that a distinction between the two may be rather difficult to make for human annotators. This, in turn, raises the question whether this particular distinction needs to be made at all, in particular when annotating non-task-oriented dialogues. Nevertheless, some key expressions that for instance occur in Monroe, such as *to recap* or *the plan* could easily be captured by an automated annotation system and mapped onto appropriate semantico-pragmatic labels (see below) that could then be fed into inferencing strategies, thereby making it possible to identify either confirming speech acts or dialogue stages, such as a recapitulation stage towards the end of a booking dialogue, automatically.

[5] In the following, I will present all examples in reduced form, with irrelevant attributes omitted, in order to save space and make the examples easier to read.

Unless their functions are apparent through their lexico-grammar, as in the cases discussed in the next paragraph, the solution for handling communication- or task-management features in DART is to treat them as semantico-pragmatic attributes (*modes*) of the unit. This type of attribute essentially records labels for what Searle (1969: 16) termed 'illocutionary force indicating devices' (or IFIDs for short). These labels, in conjunction with the syntactic information for a unit, are then used as the basis for inferencing processes that try to determine the speech act (attribute) of the unit automatically (Weisser 2010).

The annotation practices in all four corpora predominantly seem not to treat DMs or other initiating/responding short units (such as yes- or no-responses) as units in their own right, but either completely conflate them with other syntactic units, such as in *[SIL] yes i know where three ninety is* (Monroe) or *uh [sil] well we can't [sil] load [sil] oranges into an engine* (Trains), or at best appear to split them off inconsistently. This may lead to a proliferation of different attributes/tags with the result that multiple functions are marked for the conflated unit, thus 'skewing' the impression of its function, while in SWBDD the effect may be that one chooses one single function where marking more than one would have been appropriate. As most of these short units (a) tend to occur in turn-initial positions, (b) are lexico-grammatically easily identifiable, and (c) all fulfil specific text-structuring roles inside the dialogue, it would therefore make more sense to use the separate, syntax-based tags <dm>, <yes>, and <no> to reflect their nature, as is done in DART. These tags can then be endowed with speech act attributes that clarify their individual pragmatic functions, e.g. *init* (short for *initiate* or *initialise*) or *acknowledge* for *dm*, *acknowledge* or *accept* for *yes*, and *negate* or *refuse* for *no*.

The two remaining dimensions, *forward-looking communicative function* and *backward-looking communicative function*, as their names already suggest, are attempts to represent categories that may either reflect how the dialogue is going to progress into the next stage, or relate to something that has occurred in a prior unit or sequence of units, respectively. Their definition in DAMSL, however, is highly problematic, which raises questions regarding their usefulness, as will hopefully become apparent in the following discussion.

3.3.3 'Initiation-indicating' categories

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

(i) Statement: e.g. Assert, Reassert, Other-statement (e.g. expressing wishes or opinions rather than facts).
(ii) Influencing-addressee-future-action: e.g. Info-Request, Open-option (an option that does not directly commit *h* to an action, i.e. a suggestion), Action-directive.

(iii) Committing-speaker-future-action: e.g. Offer, Commit.
(iv) Other-forward-(looking-)function: dummy category for fixed, relatively rare functions such as performatives, exclamatives, conventional-opening, conventional-closing, thanking.

These categories clearly demonstrate that DAMSL was primarily designed to label discourse units according to their purpose on a purely manual basis, and after sometimes seemingly extensive interpretation, as can be seen in the following example from the DAMSL manual, where OO stands for *Open-option* (Allen and Core 1997: 13):

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.

Here, the label *Action-directive* for the unit produced by A appears to be indicating A's authority over B, when in reality A is simply making a suggestion (DART: *suggest*), as indicated by the 'let's imperative'. This positively contradicts the DAMSL definition of the Action-directive speech act as 'obligat[ing] the listener to either perform the requested action or communicate a refusal or inability to perform the action' (Allen and Core 1997: 12), and it does not in fact communicate any directive (DART: *direct*) at all. The first part of the label, *Action*, is also an unfortunate choice because an interlocutor obligated to perform a directive stipulating an action would in fact have to comply by performing the action itself, with an acknowledgement being of secondary importance only. Defining such suggestions as 'directives' consequently also leads to problems in the annotations of our DAMSL-annotated corpora, as in the following examples, where both the unit texts and other relevant information are emphasised.

(9) Erroneously marked Action-directive (Coconut)

<Information-Level_24="Task"><Forward-Comm-Function_41= "Influence-on-Listener Directive ***Action-Directive***"><Syn-Surface-Features_22="pres questionY/N"><Forward-Comm-Function_55= "Influence-on-Speaker ***Offer***"> [28]: ***do you want to buy the green chairs for the living room?*** "

(10) Erroneously marked Action-directive (Monroe)

<Utt Id=utt240 Influence-on-listener=***Action-directive*** Influence-on-speaker=***Commit*** Info-level=Task>
[SIL] and then [SIL] so one ambulance stays at saint mary's [SIL]

At best, example (9) may be interpreted as a suggestion, although it is, in the first instance, a request for information (DART: *reqInfo*) intended to elicit the intention/volition of the interlocutor, but it can by no means be seen as a directive

'forcing' the interlocutor to buy anything. Nor should it be classed as an 'Offer' because that, by definition, implies a commitment only on the part of *s* (DART: *offer*). Example (10) again cannot be a directive, simply because it states something, rather than (as before) exhibiting any directive force at all. In fact, what it does represent is a recapitulation of a joint decision (DART: *confirm*), which may justify the 'Commit' value, albeit this would imply a commitment on *s* and *h*, not just *s* alone. In SWBDD, according to Jurafsky et al. (1997), the *Action-directive* (ad) 'marks imperatives and commands' and '[its] syntactic realization [. . .] may include imperatives, questions ("Do you want to go ahead and start?"), and standard declarative clauses ("You ought to rent the, {F uh,}F X part one")'. Looking at the examples provided here, it becomes apparent that the emphasis on trying to find a consistent functional label summarising expressions of a potential commitment to an action, which may also be expressed in the shape of directives, has somewhat backfired because it not only conflates commitment in the form of agreements and suggested courses of action, but also no longer allows the annotator to make a crucial distinction between 'authority-based', genuine directives and milder forms of influence on *h*, such as suggestions.

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 in the DAMSL manual example above, it is also impossible to detect any kind of direct incentive, let alone an *Offer*, in the traditional sense. In DAMSL, however, the *Offer* tag is 'conditional on the listener's agreement' (Jurafsky et al. 1997: 13); 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 label *Assert* (category (iii)) should be contingent upon the fact that 'the speaker is trying to change the beliefs of the hearer' and 'make claims about the world' (Jurafsky et al. 1997: 10). This appears somewhat too 'philosophical', though, and attributes an unnecessarily high degree of force to any utterance this label is applied to. It would thus perhaps be best to opt for the more neutral label *State* (DART: *state*) instead of *Assert*, which is what SWDB DAMSL does, sub-categorising it into *sd* (statement-non-opinion) and *sv* (statement-opinion). However, whether an opinion is expressed may be open to debate in all cases where an explicit expression of opinion (e.g. *I think/guess/believe/assume*, *in my opinion*) is absent, as in the following example.

(11) Erroneous sv example from Switchboard

sv A.47 utt2: but, and they're expensive /

(sw_0001_4325.utt)

(12) Erroneous sv example from Switchboard

sv B.32 utt3: but it was, it was just an incredibly humiliating experience what I went through /

(sw_0002_4330.utt)

Thus, the use of the *sv* label, despite being an interesting attempt at capturing subjective statements, if 'unconstrained', might lead to highly subjective annotations, and it may therefore be best to only mark those units as expressing opinions (DART: *expressOpinion*) where the surface information clearly warrants this. This notion seems to be supported by the fact that the SWBD-DAMSL manual states that 'the distinction was very hard to make by labelers, and accounted for a large proportion of our interlabeler error' (Jurafsky et al. 1997).

The *Reassert* label is supposed to cover units that repeat items already stated in the dialogue. In practice, this seems to be interpreted in the data in one of two ways, either as recapitulating/summarising specific issues as general statements, or in the form of requests for confirmation, as in the following example.

(13) Reassert (type 1)

<Information-Level_16="Task"><Forward-Comm-Function_27="***Statement Reassert***"> [16]: ***2 red chairs*** = ***$100***

(14) Reassert (type 2)

<Utt Id=utt271 Influence-on-listener=Action-directive Influence-on-speaker=Commit Info-level=Task Statement=***Reassert***
[SIL] so that ambulance goes and takes [SIL] the broken leg guy [SIL]

Type 1 (example (13)) is very difficult to identify in an automated annotation, and interpreting it as such is essentially up to the (manual) annotator if the repetition is not verbatim (DART: *echo*). It also does not appear to require any form of acknowledgement, while type 2 (example (14)) is frequently followed by an acknowledging DM and may therefore be interpreted as a request for confirmation (DART: *reqConfirm*) that has simply not been marked by a question tag, which is the obvious choice if *s* wants to make this function explicit. The choice of inclusion in the forward-looking category in fact only makes sense if there is indeed a request for information because this should then trigger a confirming response. In general, though, as with many statements, perhaps most of these utterances will tend to be more backwards-oriented. Due to the inherent difficulty in recognising reasserts, SWBDD abolishes this category.

The main problem with the *Info-Request* category is that it is purely function-based, namely it subsumes everything that 'introduce[s] an obligation to provide an answer' (Allen and Core 1997: 10). Accordingly, it also includes directives, such as 'Tell me the time' (ibid.: 11), where one of the functions may well be to enquire about the present time, but with the further implication that *s* clearly has some degree of authority over the addressee, information that, in DAMSL, would necessitate an additional tag *Action-directive* to clarify the directive nature that is in fact already part of the imperative syntax and could thus easily and more directly be expressed via a syntactic <imp> tag

with a speech act attribute *direct*. As we have seen before, SWBDD seems to subsume such cases in the *ad* label, which makes slightly more sense. Furthermore, it expands the rest of the *Info-request* category into the question (*q*) category, including options for marking declarative (^d) and tag (^g) question status on declarative units. Unfortunately, though, this entails a loss of expressiveness because simply labelling a question as either a yes/no-question (*qy*), a wh-question (*qw*), or an or-question (*qr*) without including additional speech act information does not express anything regarding its function within the dialogue (unless one mistakenly assumes all questions to constitute exclusively requests for information), while using qh for rhetorical questions does not say much about their function or what they may actually look like. This label is thus highly debatable, as its use is prone to rather subjective (over-)interpretation.

The DART solution to handling questions is to have one element for wh-questions (<q-wh>, mode *open* to signal open choice), where the options for speech act labels are request for information (*reqInfo*), request for directive (*reqDirect*), and *suggest* (for cases such as *How/what about* . . .), and one element for yes/no-questions (<q-yn> mode *closed*), with only the first two speech act options.

In SWBDD, the commitment-indicating labels that correspond to DAMSL *Offer* and *Commit* essentially provide 'pro-active' or more initiative-oriented labels. However, in terms of their semantics, there is (again) a confusion between indications of commitment (*cc*) with expressions of intention (DART: *stateIntent*), e.g. 'I'll keep that one in mind' or 'I'll remember that', where *s* clearly is not committing him-/herself in the traditional (interactional) sense because there seems to be no indication that the interlocutor would be affected in any way. In a similar way, units labelled as offers (*co*) in the data I investigated seem to be lacking an element of benefit for the interlocutor, in which case they would better be labelled as *suggest* or even as *hold* because examples such as 'Let me turn off my stereo here' or 'Let me change my channel' (DART: <imp>, speech act *hold*) seem to imply that some form of interruption of the dialogue is necessary until proper verbal interaction can be resumed.

The label *Other-forward-*(*looking-*)*function* (category (iv) above) is in fact a convenience label used in Leech et al. (2000) to group together a number of additional sub-categories. DAMSL, in contrast, 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 the convenience label here for the discussion because it groups together a kind of rag-bag of functions that cannot really be referred to as properly forward-looking.

While a greeting, for instance, may potentially be interpreted as forward-looking because it opens up the dialogue, though without actually contributing

to the content proper, the same can definitely not be said for saying *goodbye*, not even in German or French, where something like *Auf Wiedersehen/-hören* or *Au revoir* at least expresses an inkling of potential future interaction.

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

SWBDD basically retains the basic DAMSL categories for this group, labelling thanking (*ft*), responding to thanks (*fw*; so far missing in DART), and apologising (*fa*). This group mostly contains a number of labels for conventionalised actions, though, such as conventional openings or closings. There is, however, no clear formal definition of what these closings (*fc*) may look like, so that such structures presented in the examples in the manual appear to be rather arbitrarily labelling closing sequences, apart from actually 'masking' a different kind of speech act, such as in '{D Well} good luck with [the, + the] new kid', which stands in contradiction to other examples of 'well-wishing' where, for instance, *I wish you* is clearly listed as an instance of an explicit performative (*fx*). In DART, only the *hereby* category of performatives is in fact labelled as such (DART: *perform*), but it would probably be useful to add a speech act *wishWell* to the taxonomy as a counterpart to the existing *expressWish*, which signals strict volition on the part of *s*, that is, without including any benefit for *h*. DART also has an additional speech act label on the (personal) interaction level, *intro*, which is used to mark up units where speakers introduce themselves or the institutions they may belong to.

Both Coconut and Monroe also use the *Other-forward-function* for DMs and other discourse-managing items whose exact value seems to be difficult to identify, e.g. fillers such as *um* (in initial position in DART: <dm>, *hesitate*), or units that signal a 'holding' phase where one speaker needs to 'stall' in order to verify or look up information or perform some other non-verbal activity. However, as already pointed out, DMs would be much better marked via a syntactic tag that underlines their common discourse-managing feature, and 'holding units' are generally clearly marked lexico-grammatically through expressions such as *just a minute/second* or *hold the line, please*, which makes it possible to identify them automatically as a

special form of directive, and assign the speech act *hold* to whichever syntactic category they may be occurring with.

3.3.4 'Response-indicating' categories

The DAMSL (and partly SWBDD) *backward-looking communicative function* section comprises the following categories.

(i) Agreement: e.g. accept, maybe, reject, hold (*a* in SWBDD, including potential [*am*] and partial agreements [*aap*] and their negative counterparts).
(ii) Understanding: e.g. backchannelling, signal-non-understanding, signal-understanding, repeat-rephrase (*b* in SWBDD).
(iii) Answer: generally signals compliance with a request for information (*n* in SWBDD).
(iv) Information-relation: utterances expressing explicitly how an utterance relates to the previous one.
(v) Antecedents: any utterance may be marked as relating to more than just the preceding one.

Both categories (i) and (ii) here provide food for discussion, as they seem to contain features that essentially belong to the communication-management level, e.g. types of positive or negative responses or acknowledging DMs that form closed, easily identifiable classes. Thus, the question here is really, as pointed out above, whether it would not be better to set up syntactic categories for all such communication-management devices, and then sub-classify them as to whether they may be deemed to be forward-looking (*initiating*) or backward-looking (*acknowledging*, *responding*) through the relevant attributes. A further problem with this group in SWBDD is that the labelling based on them is purely functional, which may cause problems if the same expression can formally represent different categories, such as *yeah*, *uh-huh*, or *okay*, which are sometimes labelled as *b* ('default agreement') versus *ny* ('yes [response] or variations'), or *b* versus *bk* ('acknowledge-answer').

The fact that splitting these categories may lead to confusion is exemplified by the fact that both Trains and Monroe frequently seem to mark the feature 'Agreement=Accept' alongside 'Understanding=SU[6]-Acknowledge', almost by default, when the respective utterances in fact exclusively signal acknowledgement:

(15) Incorrectly marked acceptance (Trains)

<Utt Id=utt5 Info-level=Task Statement=Assert>

[6] Signal Understanding.

and the time now is midnight
<Turn Id=T3 Speaker="s">
<Utt Id=utt6 ***Agreement=Accept*** Info-level=Task Response-to="utt4 utt5" ***Understanding=SU-Acknowledge okay***

(16) Incorrectly marked acceptance (Monroe)

<Utt Id=utt47 Influence-on-listener=Action-directive Info-level=Task > [SIL] if you can find [SIL] the r [SIL] three ninety road where [SIL] it says three ninety
<Turn Id=T34 Speaker="u">
<Utt Id=utt48 ***Agreement=Accept*** Info-level=Communication-management Response-to="utt47" ***Understanding=SU-Acknowledge [SIL] yes i know where three ninety is***

Such erroneous default behaviour could easily be avoided if a single speech act label signalling either whether such a response simply acknowledges the contents of a previous unit or in fact constitutes an acceptance were used in conjunction with an appropriate syntactic tag. In example (16), the confusion may at least partly be due to the fact that the initial *yes* is not treated as a separate unit.

Another problem with having too many separate dimensions in a scheme can be seen in the following example from Monroe:

(17) Problem in using multiple dimensions from Monroe

<Utt Id=utt44 Answer=Yes Response-to="utt43" Statement=Assert> two fifty two is ***horizontal*** [SIL]
<Turn Id=T32 Speaker="u">
<Utt Id=utt45 ***Agreement=Accept*** Info-level=Communication-management Response-to="utt44" ***Understanding=SU-Repeat-rephrase*** [SIL] ***horizontal***

Here, we not only encounter the same problem as above concerning the "Accept" label, but also some overlap with the forward-looking "Reassert" category from above in the "Repeat-rephrase". Actually, such forms of 'backchannelling', where *s* repeats all or part of the previous unit verbatim, are very common forms of communication-management, especially in task-oriented dialogues, and are commonly used as requests for confirmation (DART: *reqConfirm*) that *s* has understood correctly or is simply trying to 'internalise' the information in a kind of self-talk. At other times, as these forms of repeating information occur most frequently in syntactically incomplete units, such as individual words or phrases, and 'echo' all or part of the preceding unit, perhaps it would be best to mark them up inside a syntactic <frag> element with a relatively neutral *echo* speech act attribute. Such labelling can generally be achieved automatically in an initial analysis step, and if further evidence exists to support the assumption that the unit indeed functions as a request for information, for example through a following acknowledging

DM, then this label could be changed accordingly through a suitable automatic inferencing process.

The label *hold* in category (i) is used in a rather misleading way in DAMSL in two respects. First of all, its categorisation here (Allen and Core 1997: 18) suggests that it should signal some form of agreement, but it does not really do so in most cases. Instead, it is generally used for labelling units that do not contain a direct response, but instead include clarification questions, for example, and thus only marginally signal agreement in an indirect way. Thus, it should rather be reserved for the 'stalling' units discussed earlier.

Category (iii), *Answer*, does not present much of a problem, as it is mainly straightforward, apart from perhaps one special feature. In this respect, Allen and Core (1997: 23) state:

> 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.

The special issue hinted at above is that imperatives are also seen as answers when they ought to be seen as directives, even if they occur in response to a question. Furthermore, if all answers were also *assert*s, as Allen and Core claim, then, in conclusion, directives that 'function' as answers would also be *assert*s, which seems somewhat contradictory. Perhaps the key to this confusion lies in the fact that an *answer* is in fact a kind of 'secondary' (or additional) speech act that embodies at least part of what category (iv) from above (*Information-relation*) is supposed to capture. The (DART) solution is thus to allow multiple speech acts to be annotated whenever such information relations are already encoded in the particular types of adjacency pairs of question–answer/response or statement–backchannel/acknowledgements we generally encounter in dialogues. This may even be done automatically, using appropriate inferencing strategies, as can be seen in the earlier DART version of the Switchboard dialogue example, which only includes minor manual corrections.

What exactly category (v) entails 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 and Core 1997: 24), but it seems to concern long-distance relationships, such as recapitulations, or clarifications as to which answer belongs to which question in insertion sequences, perhaps in the form of clarification questions.

3.4 Other schemes

As pointed out above, the schemes discussed here should only be seen as some of the more influential and recent ones, but of course there have been many

other attempts at categorising pragmatic information, starting with Austin's (1962) lists of *explicit performatives* and Searle's broad distinction into *assertives*, *directives*, *commissives*, *expressives*, and *declaratives*, which have certainly influenced all other taxonomies. Amongst those that perhaps deserve further mention are Stiles' (1992: 17) verbal response modes (*disclosure*, *edification*, *advisement*, *confirmation*, *question*, *acknowledgement*, *interpretation*, and *reflection*), rooted in psychology research, and the VERBMOBIL taxonomy for computer-assisted appointment planning, although the former is somewhat limited in its scope and the latter strongly task-oriented. New schemes seem to be regularly based on different versions of (SWBD) DAMSL.

3.5 Using the DART annotation scheme: a brief case study

In order to make it easier to understand the applicability of pragmatic annotation schemes better, I will now present a brief and simplified case study demonstrating how the pragmatic annotation created in DART can be used to research interactive strategies. To investigate the strategies used by efficient call centre agents, I will analyse the behaviour of a British agent whose interactions were recorded for the Trainline corpus, a small corpus of 35 dialogues that forms part of the larger corpus created on the SPAAC (speech-act annotated corpus of dialogues) project (Leech and Weisser 2003).

Upon casual observation, this particular agent, who has to deal with callers who are enquiring about train timetable information and want to carry out bookings, always appears to be in control of the situation. She seems to be particularly skilled at guiding the callers, both in eliciting the relevant personal information from the callers in order to do the bookings, and in asking them for appropriate instructions, which allows her to keep the flow of interaction going smoothly. If we can get this initial subjective impression purely from reading the dialogues, then the question we should ask ourselves in our context is: What exactly are the pragmatic strategies she employs in order to achieve her aims and how can we objectify the analysis of these strategies in order to 'make them countable'?

Of course, quantifying features of pragmatic performance requires a fair degree of interpretation, but we can begin the analysis by looking at the most frequently occurring combinations of syntactic tags + speech act labels, and then turn to individual examples of these for closer examination and interpretation. Figure 3.2 shows all combinations that have a frequency of occurrence above 40 in the data extracted for the agent. This cut-off point was chosen empirically, but as we shall see in the discussion, it will provide us with a very reasonable picture of the most relevant combinations.

The large number of stating declaratives here simply indicates the information-providing nature of the dialogues and is thus fairly unremarkable.

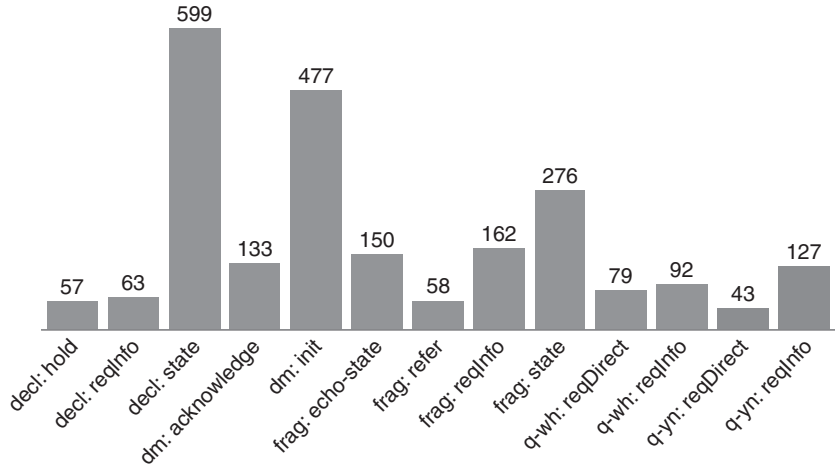

Figure 3.2 Tags + speech acts with a frequency of over 40 occurrences.

In contrast to this, however, the relatively high incidence of stating fragments (frag: state; 276), in conjunction with deictic fragments (frag: refer; 58), signals that the agent efficiently avoids providing fully formulated statements where shorter, less redundant, forms of expression suffice. These fragments essentially constitute either summaries/recapitulations of previously discussed/elicited information or clarifying information that is 'tagged on', where the context disambiguates the missing information, e.g. *there's a 6 11 train in the morning* ***arriving 10 31***. Here, the fragmentary non-finite clause with a reduced PP is preferred to the less efficient fully declarative option (*and*) *this train is arriving at 10 31*. A further high number of fragments (frag: echo-state; 150) is used in order to immediately confirm that information, such as a sequence of credit card numbers, has been properly received by the agent, thus eliminating the need to later repeat all the information fully and at length.

The relatively high joint occurrence of interrogatives, both in the form of yes/no- and wh-questions, as well as fragmentary (frag: reqInfo; 162) or declarative questions (decl: reqInfo; 63), again indicates the particular information-seeking nature of the interaction. What is interesting here, though, is that the agent also employs a relatively large number of yes/no-questions that contain requests for directives (q-yn: reqDirect; 43), which represents an especially efficient way of combining providing options for the caller and eliciting what to do next on the part of the agent, e.g. *and do you want the 15 52 for the Virgin value fare*. The fragmentary questions, e.g. *and your initial*, as is the case with the other types of fragments discussed above, again signal efficiency through the absence of redundancy.

The efficiency of this particular agent is also signalled through her use of DMs. While she obviously also adheres to the general rules of politeness and communication-management by using a fair amount of acknowledging DMs (dm: acknowledge; 133), the number of initiating/initialising DMs (dm: init; 477) is positively striking, as it clearly indicates that she controls the interaction by initiating new topics or dialogue sequences in this way.

Although this case study, in order to save space, has by necessity been very brief, just those few simple points will hopefully demonstrate relatively clearly what the advantages of an in-depth analysis using the types of pragmatic annotation discussed above might have to offer in terms of developing our understanding of human interaction further.

3.6 Conclusion

DAMSL, as a consensus model, arrived at through the participation of experts from various fields of linguistics and computer science, represents an interesting attempt to capture the multiple levels of meaning and sometimes also structure that are inherent in spoken interaction. However, as will hopefully have become apparent from the above discussion, applying this scheme in its original form is too difficult, because of the sometimes needless complexity and inherent contradictions. Some of this complexity may be due to the fact that it is a purely functional scheme that by and large neglects aspects of syntax and that the levels of meaning are not appropriately connected to the levels of grammar and meaning established in linguistic theory. The DAMSL-based schemes discussed above, although they provide highly useful starting points for an improved speech act taxonomy and contain lists of many useful patterns that indicate potential speech acts, generally suffer from being too subjective in the motivation for their categories. Even SWBDD, despite presenting some general improvements and extensions in trying to cater for the far wider scope of unrestricted interactive dialogue scenarios, as opposed to purely task-oriented scenarios, still suffers from the drawbacks of the original functional and highly interpretive DAMSL approach. This makes it very difficult to achieve a consistent annotation, and hence also makes SWBDD relatively unsuitable for automated processing. Furthermore, the general tendency pointed out above to subsume DMs and other short units into longer ones means that important contextual cues may be left uninterpreted and conflated by any algorithms used in order to train probabilistic recognition systems (Jurafsky 2006). With the exception of DART, however, the latter unfortunately still seem to represent the norm for almost all types of automated analysis.

As will hopefully have become clear through the above discussion and examples, the DART model not only incorporates information about the

dimensions that form part of other major models, but makes them more easily accessible from a linguistic point of view through the syntactic elements and more semantically or pragmatically oriented types of linguistic information stored in attributes, as well as providing less philosophically motivated and more common-sense labels for speech acts. This approach, along with its implementation, also makes it possible to annotate large sets of data semi-automatically in order to create consistently annotated dialogue corpora, at the same time making it possible to test linguistically motivated hypotheses easily by changing different parts of the underlying linguistic resources, such as the implementation of the grammar for recognising the syntactic categories, different lexica, and so on.

One thing that is still missing from DART, as well as all the other schemes, but would be required to achieve an even better modelling of the relationship between *s* and *h* is the inclusion of information on speaker roles. Although DART already models the different levels of speaker authority discussed above better than DAMSL-based models, more precise information on different levels of authority or equality associated with particular roles in an interaction may possibly also allow us to solve the problem of contextual specificity of particular speech acts via suitable inferencing strategies, for example to help identify whether a suggestion should perhaps be interpreted as an order if the speaker is in a position of authority. At the same time, this might also allow us to gain further insights into different levels of directness and politeness.

REFERENCES

Allen, J. and Core, M. 1997. Draft of DAMSL: Dialog Act Markup in Several Layers. Online at: ftp://ftp.cs.rochester.edu/pub/packages/dialog-annotation/manual.ps.gz.

Androutsopoulos, I. and Aretoulaki, M. 2003. Natural language interaction. In R. Mitkov (ed.), *The Oxford Handbook of Computational Linguistics*. Oxford: Oxford University Press.

Archer, D., Culpeper, J. and Davies, M. 2008. Pragmatic annotation. In A. Lüdeling and M. Kytö (eds), *Corpus Linguistics: An International Handbook*, pp. 613–641. Mouton de Gruyter.

Austin, J. L. 1962. *How To Do Things With Words*. Oxford: Oxford University Press.

Biber, D., Johansson, S., Leech, G., Conrad, S. and Finegan, E. 1999. *Longman Grammar of Spoken and Written English*. London: Longman.

Bunt, H., Alexandersson, J., Carletta, J., Choe, J.-W., Fang, A., Hasida, K., Lee, K., Pethukova, V., Popescu-Belis, A., Romary, L., Soria, C. and Traum, D. 2010. Towards an ISO standard for dialogue annotation. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation* (*LREC 2010*), pp. 2548–2555.

De Eugenio, B., Jordan, P. and Pylkkänen, L. 1998. The COCONUT project: Dialogue Annotation Manual. ISP Technical Report 98–1.

Godfrey, J. and Holliman, E. 1997. Switchboard-1 Release 2. Philadelphia: Linguistic Data Consortium. Online at: https://Catalog.ldc.upenn.edu/LDC97S62.

Godfrey, J., Holliman, E. and McDaniel, J. 1992. SWITCHBOARD: Telephone speech corpus for research and development. In *Proceedings of ICASSP*, vol. 1, pp. 517–520.

Heeman, P. and Allen, J. 1995. The Trains 93 Dialogues: TRAINS Technical Note 94–2. Computer Science Department, The University of Rochester. Online at: ftp://ftp.cs.rochester.edu/pub/papers/ai/94.tn2.Trains_93_dialogues.ps.gz.

Hobbs, J. 2006. Abduction in natural language understanding. In L. Horn and G. Ward (eds), *The Handbook of Pragmatics*. Oxford: Blackwell.

Horn, L. and Ward, G. 2006. *The Handbook of Pragmatics*. Oxford: Blackwell. (Paperback edition of 2004.)

Jurafsky, D. 2006. Pragmatics and computational linguistics. In L. Horn and G. Ward (eds), *The Handbook of Pragmatics*, pp. 578–604. Oxford: Blackwell.

Jurafsky, D., Shriberg, E. and Biasca, D. 1997. Switchboard SWBD-DAMSL Shallow-Discourse-Function Annotation Coder Manual. Online at: www.stanford.edu/~jurafsky/ws97/ics-tr-97-02.ps.

Klein, M. 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'*, pp. 35–41.

Leech, G. and Weisser, M. 2003. Generic speech act annotation for task-oriented dialogues. In D. Archer, P. Rayson, A. Wilson and T. McEnery (eds), *Proceedings of the Corpus Linguistics 2003 Conference*. Lancaster University: UCREL Technical Papers, vol. 16.

Leech, G., Weisser, M., Wilson, A. and Grice, M. 2000. Survey and guidelines for the representation and annotation of dialogue. In D. Gibbon, I. Mertins and R. Moore (eds), *Handbook of Multimodal and Spoken Language Systems*. Dordrecht: Kluwer Academic.

Mitkov, R. (ed.) 2003. *The Oxford Handbook of Computational Linguistics*. Oxford: Oxford University Press.

Sacks, H., Schegloff, E. and Jefferson, G. 1974. A simplest systematics for the organization of turn-taking for conversation. *Language* 50 (4): 697–735.

Searle, J. 1969. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge: Cambridge University Press.

Simpson, R., Briggs, S., Ovens, J. and Swales, J. 2002. *The Michigan Corpus of Academic Spoken English*. Ann Arbor, MI: The Regents of the University of Michigan.

Stent, A. 2000. The Monroe Corpus. Technical Report 728/TN 99-2, Computer Science Department, The University of Rochester. Online at: www.cs.rochester.edu/research/cisd/pubs/2000/stent-tr728.pdf.

Stiles, W. 1992. *Describing Talk: A Taxonomy of Verbal Response Modes*. Newbury Park, CA: Sage.

Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Meteer, M. and Van Ess-Dykema, C. 2000. Dialogue act modelling for automatic tagging and recognition of conversational speech. *Computational Linguistics* 26: 339–371.

Taylor, A. 1995. *Dysfluency Annotation Stylebook for the Switchboard Corpus*. University of Pennsylvania, Department of Computer and Information Science (ftp.cis.upenn.edu/pub/treebank/swbd/doc/DFL-book.ps). Original version by Marie Meteer et al.

W3C 2010. XML Technology: W3C. Online at: www.w3.org/standards/xml/ (accessed 11 March 2011).

Weisser, M. 2002. Determining generic elements in dialogue. In *Language, Information and Lexicography*, vols 12, 13, pp. 131–156. Institute of Language and Information Studies, Yonsei University.

2003. SPAACy: A semi-automated tool for annotating dialogue acts. *International Journal of Corpus Linguistics* 8 (1): 63–74.

2004. Tagging dialogues in SPAACy. In J. Véronis (ed.), *Traitement Automatique des Langues: Le Traitement Automatique des Corpus Oraux*, vol. 45, pp. 131–157.

2010. *Annotating Dialogue Corpora Semi-Automatically: A Corpus-Based Approach to Pragmatics*. Unpublished habilitation, University of Bayreuth.

2014. The Dialogue Annotation and Research Tool (DART) (Version 1.0) [Computer Software]. Available from martinweisser.org/ling_soft.html#DART

Appendix 3A

Table 3A.1 can be used to compare the different major annotation schemes discussed in the chapter. It is based on examples from the Switchboard DAMSL annotation manual, and lists the relevant SWBDD labels and abbreviations (columns 3 and 4). Columns 1 and 2 represent attempts to provide the equivalent annotation labels for DART and the original DAMSL scheme. '???' indicates uncertainty as to the applicability of an equivalence option, but may generally specify the closest equivalent option. Forward slashes separate alternatives for DAMSL-based schemes, while commas signify a combination of labels. For DART, syntactic tags are provided if only one syntactic category is concerned, but if a speech act can occur with multiple syntactic categories, only the speech act label is given as the most relevant piece of information. If applicable, additional mode information is also indicated.

Table 3A.1 *Comparison of speech act labels in DART, DAMSL, and SWDB-DAMSL.*

DART (tag +) act (+ mode)	DAMSL tag	SWDB-DAMSL tag	Example
decl + state	(Re-)Assert	STATEMENT (sd)	(Me,) I'm in the legal department.
dm + acknowledge	n/a	BACKCHANNEL/ACKNOWLEDGE (bk)	Uh-huh.
decl + expressOpinion/ approve	Other-statement???	OPINION (sv)	I think it's great.
abandon/uninterpretable	Abandoned / Uninterpretable	ABANDONED (-/)/ UNINTERPRETABLE (%)	(So,) -/
dm + accept	Accept(-part)	AGREEMENT/ACCEPT (aa)	That's exactly it.
decl + acknowledge???	n/a	APPRECIATION (ba)	I can imagine.
q-yn + reqInfo	Info-Request	YES–NO-QUESTION (qy)	Do you have to have any special training?
<comment type="laughter" />, etc.	n/a	NON-VERBAL (x)	<Laughter>,<Throat_clearing>
yes + answer-acknowledge/ accept	Assert/Commit	YES ANSWERS (ny)	Yes.
Bye	Conventional Closing	CONVENTIONAL-CLOSING (fc)	(Well,) it's been nice talking to you.
q-wh + reqInfo	Info-Request	WH-QUESTION (qw)	What did you wear to work today?
no + answer-negate/reject	Answer	No ANSWERS (nn)	No.
dm + acknowledge/agree	Other-forward-function	RESPONSE ACKNOWLEDGMENT (bk)	(Oh,) okay.
state + mode=nonawareness	Assert/Other-statement???	HEDGE (h)	I don't know if I'm making any sense or not.
decl + reqInfo	Info-Request	DECLARATIVE YES–NO-QUESTION (qy^d)	(So) you can afford to get a house?
imp + direct	Action-directive	OTHER (o, fo, bc, by, fw)	(Well) give me a break, you know.
q-yn + reqConfirm	Info-Request	BACKCHANNEL-QUESTION (bh)	Is that right?
decl + state	Assert	QUOTATION (^q)	You can't be pregnant and have cats.
decl + reqConfirm	Info-Request	SUMMARIZE/REFORMULATE (bf)	(Oh,) you mean you switched schools for the kids.
decl + answer	Answer	AFFIRMATIVE NON-YES ANSWERS (na, ny^e)	It is.
q-wh + suggest	Open-option	ACTION-DIRECTIVE (ad)	Why don't you go first.

n/a	Completion	COLLABORATIVE COMPLETION (^2)	Who aren't contributing.
echo	Repeat-rephrase	REPEAT-PHRASE (b^m)	(Oh,) fajitas.
q-wh + reqInfo/suggest	Info-Request, Open-option	OPEN-QUESTION (qo)	How about you?
q-wh + reqInfo	Info-Request	RHETORICAL-QUESTIONS (qr)	Who would steal a newspaper?
decl + state	Hold	HOLD BEFORE ANSWER/ AGREEMENT (^h)	I'm drawing a blank.
no + negate/reject	Reject	REJECT (ar)	(Well,) no.
frag + answer + polarity=negative	Answer, Assert	NEGATIVE NON-NO ANSWERS (ng, nn^e)	(Uh,) not a whole lot.
imp + pardon	Signal-non-understanding	SIGNAL-NON-UNDERSTANDING (br)	Excuse me?
state + answer + mode=nonawareness	Answer	OTHER ANSWERS (no)	I don't know.
Q-wh + greet + mode=intro	Conventional Opening	CONVENTIONAL-OPENING (fp)	How are you?
q-yn + reqInfo + mode=alternative	Info-Request	OR-CLAUSE (qrr)	or is it more of a company?
frag + answer-state + polarity=negative	Answer, (Re-)Assert	DISPREFERRED ANSWERS (arp, nd)	(Well,) not so much that.
thirdParty (direct)	n/a	3RD-PARTY-TALK (t3)	(My goodness,) (Diane,) get down from there.
decl + stateIntent	Offer, Commit	OFFERS, OPTIONS & COMMITS (oo, cc, co)	I'll have to check that out.
selfTalk	Self-talk	SELF-TALK (t1)	What's the word I'm looking for
decl + appreciate	n/a	DOWNPLAYER (bd)	That's all right.
frag + state	Maybe/Accept-part	MAYBE/ACCEPT-PART (aap/am)	Something like that.
reqConfirm + mode=tag	Info-Request	TAG-QUESTION (^g)	Right?
q-wh + reqInfo	Info-Request	DECLARATIVE WH-QUESTION (qw^d)	You are what kind of buff?
decl + apologise	Explicit-performative	APOLOGY (fa)	I'm sorry.
frag + thank	Explicit-performative	THANKING (ft)	(Hey) thanks a lot.