Computational Models of Dialogue

Jonathan Ginzburg[1] and Raquel Fernández[2]

[1] Dept of Computer Science
King's College London
The Strand, London WC2R 2LS
UK
jonathan.ginzburg@kcl.ac.uk

[2] Institue for Logic, Language & Computation
University of Amsterdam
P.O. Box 94242
1090 GE Amsterdam
The Netherlands
raquel.fernandez@uva.nl

A draft chapter for the Blackwell *Computational Linguistics and Natural Language Processing Handbook*, edited by Alex Clark, Chris Fox and Shalom Lappin. This draft formatted on 28th July 2009.

1 Introduction

Computational study of dialogue, the topic of this article, provides underpinnings for the design of dialogue systems and for models of human performance in conversational settings.[1] Hence, among the central issues are issues pertaining to the information states of the agents participating in a conversation. Some of this information is *public*—available in principle to be grasped and manipulated by the conversational participants, while some of this information is, at the very least, not explicitly made public. The structure and makeup of participant information states—and the extent to which information in them is shared— are issues on which much of the account of dialogue we will present here rides. Linguistic phenomena will provide guidance towards the resolution of these issues: at this point in the state of the art the challenge is to process "real language" with all its fragments, disfluencies, and the like. Such utterances are highly context dependent—to a far higher degree than is the situation with text processing. The participant information states will serve as context; being able to perform this role will, consequently, impose significant constraints on the information states.

One basic task for any theory of dialogue is to account for the coherence of a conversation—a given dialogue move can be coherently followed up by a wide variety of responses, but not by just any response. Coming up with such a theory of coherence presupposes a classification of the space of available moves. This raises a variety of interesting issues, one of the central of which is— can this be done domain independently? It is by now clear that domain dependence cannot be evaded—conversational coherence varies widely across domains. Nonetheless, as we will see, it also seems reasonably clear that there are aspects of coherence which can be explicated in a more or less domain independent way. How to find the proper balance is an important theme we will address at a number of points. After discussing a number of influential taxonomies of dialogue moves, we will concentrate on characterizing in a theory–neutral way the fundamental properties of two of the commonest move types—queries and assertions. From this will emerge a series of benchmarks theories of dialogue need to satisfy.

Metacommunicative Interaction—interaction concerning the ongoing communicative process (e.g. acknowledgements of understanding and clarification requests)— is a fundamental area for dialogue. It was long neglected in formal and computational linguistics. But has now become a much studied

[1] We would like to thank the editors for their very useful comments. Portions of this paper were presented in our course on computational models of dialogue at the 2008 ESSLLI summer school in Hamburg; we would like to thank the participants there for their feedback. Parts of this paper were written while the second author was a research fellow at the Center for the Study of Language and Information, Stanford University. The work of the second author has been partially supported by a Dutch NWO Veni project (grant number 275-80-002).

area, not least because utterances whose main function is metacommunicative are very frequent and play a crucial role in applications. As with queries and assertions, we will proceed initially in a theory–neutral way, gathering benchmarks along the way. Ultimately, one is after a theory which will explicate the coherence of metacommunicative utterances and allow them to be interpreted. This ties in with the final phenomena we will characterize—the non-sentential fragments typical of conversation, many of which occur in metacommunicative utterances. We will address two types: the first are sentential fragments—utterances like 'Bo.', 'Bo?', 'Why?', 'Yes' whose external syntax is non-sentential, but express a complete message in context. The second are disfluencies—self–corrections, hesitations, and the like.

As we mentioned above, the computational study of dialogue provides formal underpinnings for the design of dialogue systems. The second part of this article is devoted to a survey of the most influential paradigms in this area, which we informally evaluate in terms of the benchmarks that will have emerged in the first part of the paper. Dialogue systems are important because they constitute a highly promising technology. We will emphasize also the fact that they serve as a very useful testing ground for dialogue theories.

The third part of the article is devoted to sketching a theory of dialogue, known as KoS, in which meaning and interaction can be modelled. We will show how the lion's share of the benchmarks from the first part of the article can be explicated in a uniform fashion within KoS. We formulate KoS in the framework of Type Theory with Records (Cooper, 2006). This is a framework that simultaneously allows sophisticated semantic modelling using λ–calculus style techniques, while also enabling rich structure to be encoded in a way that resembles typed feature structures. In contrast to typed feature structures, however, Type Theory with Records provides as first class entities both types and tokens. This feature of the framework is of considerable importance for semantics, in particular with respect to modelling metacommunicative interaction.

The final part of the article is devoted to offering pointers to other recent significant directions in research on dialogue, including work on machine learning, multi-party conversation, and multi-modal interaction.

2 The Challenges of Dialogue

A computational theory of dialogue needs to aspire to explicate how conversations start, proceed, and conclude. It should be able to underpin the participation of either a human or an artificial agent in conversations like the following:

```
(1) John:  (1) Okay which one do you think it is?
           (2) Try F1 F1 again and we'll get
    Sarah: (3) Shift and F1?
    Sue:   (4) It's, (5) no.
    John:  (6) No, (7) just F1 F1.
    Sue:   (8) It isn't that.
    John:  (9) F1. (10) Right, (11) and that tells us
    Sue:   (12) It's shift F7.
```

(1) is, in fact, a rather hum drum conversation from the British National Corpus (BNC) (Burnard (2000)) involving three people attempting to print a file some time around 1990. Nonetheless it exhibits features that radically distinguish it from a text and even in several respects from the sort of artificial travel agent or airline booking system/user dialogue routinely described in AI/NLP papers on dialogue in the 1980s and 1990s (e.g. Allen & Perrault (1980); Aust *et al.* (1995a)):

1. **Self answering**: (2) is a case of *self answering*, unexpected on analysis of queries as requests for information (following e.g. (Allen & Perrault (1980)))
2. **Multilogue**: the conversation involves more than two participants, the case handled by the vast majority of all analyses.
3. **Disagreement**: even in this essentially cooperative setting disagreement is rife.
4. **Partial comprehension**: Sarah's (3) is a clarification request, indicating distinct states of semantic processing among participants.
5. **Incomplete utterances**: 3 of the utterances ((2), (4), (11)) are incomplete.
6. **Sentential fragments**: 5 of the utterances ((3), (5), (6), (7),(9)) are not syntactically sentential, yet convey complete illocutionary messages

As with all tasks in NLP, one can perform dialogue processing at a variety of levels, ranging from the very deep, designing agents that can participate in real conversations, through medium, which could involve trying to perform intentional analysis on a conversational participant's contribution, to shallow, which could amount to producing a reasonable paraphrase of (1), for "secretarial purposes", as in office assistants like CALO (Voss & Ehlen, 2007). Notice though that given the fact that form radically underspecifies content in dialogue, even producing such a periphrasis of (1), e.g. something along the lines of (2), involves sophisticated resources—including techniques to resolve (a) the *move type* (or *illocutionary force*) of an utterance, which is rarely signalled explicitly, (b) the content of sentential fragments (on which more below), and (c) the referents of anaphors:

(2) John asked Sue which button did she think one needed to press. He suggested to try F1 F1 once again. Sarah wondered if he meant she should type Shift and F1. Sue was a bit unsure but demurred and John indicated that he meant for her to type F1 F1. Sue disagreed with John that that was what was needed doing. John suggested to try F1, which he thought might indicate something, and then Sue suggested it was shift F7.

Move type resolution: Which one do you think it is?↦ John asked Sarah and/or Sue which button did she think one needed to press.
Sentential fragment resolution + Move type resolution: Shift and F1? ↦ Sarah wondered if he meant she should type Shift and F1.
Anaphora resolution + Move type resolution: It isn't that. ↦ Sue disagreed with John that that was what needed doing.

2.1 Classifying and Characterizing Dialogue Moves

Move Classification

One important task for a theory of dialogue is to explicate the moves or acts that participants can make in a conversation. In so doing there is an inevitable tension between the domain specific and the domain independent conversational possibilities. Some, following Wittgenstein (1953), would come close to denying the existence of domain independent conversational possibilities (e.g. Allwood (1995); Rudnicky (2004)), a position which is understandable for designers of dialogue systems. It is undeniable that knowing how to interact in an unfamiliar setting (shop, court, religious institution, academic lecture, informal meeting with people of different class/ethnic background) often requires considerable guidance. Nonetheless, an emotionally stable adult in an unfamiliar setting might initially miss a trick or even seven, but in at least many cases she is not completely floored and can navigate her way around, albeit with a certain number of stumbles. Moreover, she can acquire the necessary domain knowledge relatively easily, in contrast, for instance, to learning a new language. It thus seems a defensible strategy to try and isolate some domain independent conversational possibilities (e.g. with respect to how questions are asked and responded to or how positive/negative feedback is provided), while acknowledging the possibility that any given domain might involve moves that are specialized in some way. Of course in addition to certain idiosyncrasies about moves, which by analogy with lexical idiosyncrasy needs to be stipulated (e.g. the need to end each turn addressed to a judge in a British court with the word 'm'lud'), one also aspire to find parameters by means of which one can characterize domain specific conversational possibilities (see section 4.6).

Speech act theory (Searle, 1969; Searle & Vanderveken, 1985) emphasizes that there are hundreds of things one could do with words, not fewer than the number of illocutionary verbs that can be used performatively (e.g. 'I declare', 'I name this ship' etc). Without dismissing the significance of performatives, the strategy in most recent taxonomies of the range of moves is far more empiricist, based on the classification of moves observed in corpora. One important empirical basis for such an explication are corpus studies of the range of moves found in conversation. The number of possible moves, based on grammatical cues such as sentence type or discourse particles is reduced to between a dozen (as in the Map Task taxonomy (Carletta *et al.*, 1996)[2] and about twenty in the DAMSL taxonomy (Core & Allen, 1997). The main classes in these taxonomies are given, respectively, in (3a,b):[3,4]

(3) a. **Initiating Moves**: instruct, explain, check, align, query-yn, query-w; **Response moves**: acknowledge, reply-y, reply-n, reply-w, clarify. (From *Map Task Coder's Manual*, (Carletta *et al.*, 1996))

b. **Forward Looking moves**: statement, Influencing-addressee-future-action Info-request, Committing-speaker-future-action, Conventional Opening Closing, Explicit-performative, Exclamation; **Backward Looking moves**: Agreement (incl. accept, reject) Understanding (incl. signal understanding, signal non understanding,) Answer

In line with our earlier remarks, such taxonomies can have no pretenses to the completeness aspired to by e.g. POS taxonomies. Moreover, these taxonomies (and others proposed) have their own biases and different levels of grain, reflecting to some extent researcher biases. Nonetheless, these taxonomies enable coding of corpora at more or less reliable levels of inter-annotator agreement (Core & Allen, 1997; Carletta, 1996). We can draw certain conclusions from this:

- Initiating v. response: one significant dimension distinguishing moves is whether they are initiating or responsive. Initiating moves require more domain–sensitive/agent–particular information for their characterization.

[2] This taxonomy, inspired in part by earlier work by Sinclair & Coulthard (1975), in fact involves classification at a number of levels: the move level, the game level, and the transaction level.

[3] Annotation in DAMSL involves multiple levels, including levels that concern intelligibility/completion, semantic content, *Forward Looking Function*—how the current utterance affects the discourse and its participants, and *Backward Looking Function—how the current utterance relates to the previous discourse.*

[4] Some of the move types in DAMSL are actually supertypes, whose subtypes we have listed in parentheses in (3).

- Metacommunicative interaction: one of the features that distinguishes dialogue from text is the pervasive presence in dialogue of moves that directly concern communication management, primarily acknowledgements of understanding, clarification requests (CRs), and self-corrections. In recent years much more detailed taxonomies of such moves have been provided, including (Novick & Sutton, 1994; Muller & Prevot, 2003) for acknowledgements and (Purver *et al.*, 2001; Rodriguez & Schlangen, 2004) for CRs.

Move Characterization: queries and assertions

In general terms, a dialogue theory should be able to offer answers to the questions in (4) about initiating moves, responsive moves, as well as taking a generation perspective:

(4) a. **Initiating move/Response space conditions**: What contextual conditions characterize initiating (responsive) moves? For a given such context, what are the possible moves?
b. **Generation perspective**: given an agent A with a goal g in a context C, what can A say in C to fulfill g?

We now elaborate on these general tasks. The two main move types (or more precisely super-types) are queries and assertions—they are also the commonest means for interactions with dialogue systems. Hence, the move–related benchmarks we specify primarily concern their characterization. Many of these are modelled on benchmarks formulated in (Bohlin *et al.* (1999)). The benchmarks are loosely and atheoretically formulated, typically of the form 'Accommodate . . . ', this allows 'accommodate' to be understood in various ways, including both from a generation and an interpretive perspective.

The minimal requirement for processing queries is the ability to recognize simple answers:

(5) a. p is a simple answer to q iff p is an instantiation of q or a negation of such an instantiation.
b. For a polar question: $\{r \mid \mathit{SimpleAns}(r, p?\} = \{p, \neg p\}$
c. For a unary *wh*-question: $\{r \mid \mathit{SimpleAns}(r, \lambda b.p(b))\} =$
$\{p(a_1), \ldots, p(a_n), \neg p(a_1), \ldots, \neg p(a_n)\}$

(Q1) Query benchmark1: accommodate simple answers.

Simple answerhood covers a fair amount of ground. But it clearly underdetermines the range of answers coherently concerning a given question that any speaker of a given language can recognize, independently of domain knowledge and of the goals underlying an interaction, a notion dubbed aboutness by Ginzburg (1995). On the polar front, it leaves out the whole gamut of answers to polar questions that are weaker than p or $\neg p$ such as conditional answers 'If r, then p' (e.g. 6a) or weakly modalized answers 'probably/possibly/maybe/possibly not p'(e.g. 6b). As far as wh-questions go, it leaves out quantificational answers (6c-g), as well as disjunctive answers. These missing classes of propositions, are pervasive in actual linguistic use. In some cases they constitute goal fulfilling responses (e.g. (6a,c,d,e,g) below); the answer provided could very well trigger a follow up query (e.g. (7) below):

(6) a. Christopher: Can I have some ice-cream then?
Dorothy: you can do if there is any. (BNC, KBW)
b. Anon: Are you voting for Tory?
Denise: I might. (BNC, KB?, slightly modified)
c. Dorothy: What did grandma have to catch?
Christopher: A bus. (BNC, KBW, slightly modified)
d. Rhiannon: How much tape have you used up?
Chris: About half of one side. (BNC, KB?)
e. Dorothy: What do you want on this?
Andrew: I would like some yogurt please. (BNC, KBW, slightly modified)
f. Elinor: Where are you going to hide it?
Tim: Somewhere you can't have it.(BNC, KBW)
g. Christopher: Where is the box?
Dorothy: Near the window. (BNC, KBW)

(7) a. Anon: Are you voting for Tory? Denise: I might.
Anon: Well are you or aren't you?
b. Dorothy: What did grandma have to catch? Christopher: A bus.
Dorothy: Which bus?
c. Elinor: Where are you going to hide it? Tim: Somewhere you can't have it.
Elinor: But where?

These data lead to:

(Q2a) Query benchmark2a: accommodate non-resolving answers.
(Q2b) Query benchmark2b: accommodate follow up queries to non-resolving answers.

Responses to queries can also contain more information than literally asked for, as exemplified in (8):

(8) A: When is the train leaving? B2: 5:04, platform 12. (Based on an example due to Allen & Perrault (1980)).

This "excess information" should be utilized, leading to:

(Q3) Query benchmark3: accommodate 'overinformative' answers.

Answering a query with a query represents another significant class of possibilities. The commonest such cases are clarification responses, but since these are triggered by essentially *any* move type, we discuss these below as part of a more general discussion of metacommunicative interaction (MCI). One class of query responses are queries that, intuitively, introduce an issue whose resolution is prior to the question asked:

(9) a. A: Who murdered Smith? B: Who was in town?
b. A: Who is going to win the race? B: Who is going to participate?
c. Carol: Right, what do you want for your dinner?
Chris: What do you (pause) suggest? (BNC, KbJ)
d. Chris: Where's mummy?
Emma: What do you want her for? (BNC, KbJ)

(Q4) Query benchmark4: accommodate sub-questions.

One final class of responses, which are of some importance in applications, are "irrelevant responses", whose effect is to indicate lack of interest in the original query:

(10) a. A: Who is the homeowner? B: Who is the supervisor here?
b. Rumpole: Do you think Prof Clayton killed your husband? Mercy Charles: Do you think you'll get him off? (*Rumpole and the Right to Silence*, p. 100)
c. A: Horrible talk by Rozzo. B: It's very hot here.

(Q5) Query benchmark5: accommodate topic changing, "irrelevant" responses.

Moving on to assertions, the most obvious initial task concerns the potential effect their potential acceptance has on context.

(A1) Assertion benchmark1: if accepted, integrate propositional content with existing knowledge base.

One important feature of dialogue, a medium which involves distinct agents, is the possibility for disagreement:

(11) a. A: I'm right, you're wrong. B: No, I'm right, you're wrong.
b. John: No, just F1 F1. Sue: It isn't that.

(A2) Assertion benchmark 2: Accommodate disagreement.

The final two benchmarks are, in a sense, methodological. First, the same basic mechanism seems to regulate queries/assertions, across varying sizes of participant sets:

(12) a. Monologue: self answering (*A: Who should we invite? Perhaps Noam.*)
b. Dialogue: querier/responder (*A: Who should we invite? B: Perhaps Noam.*)

c. Multilogue: multiple discussants (*A: Who should we invite? B: Perhaps Noam. C: Martinu. D: Bedrich. . . .*)

(SC) Scalability benchmark: ensure approach scales down to monologue and up to multilogue.

Second, as we mentioned at the outset, in moving from domain to domain, there are some aspects that are specific to interacting in that domain and this cannot be avoided. However, we have claimed that human agents adapt well and with relatively little effort can reuse the interactional skills they bring with them from past experience. Hence:

(DA) Domain Adaptability benchmark: reuse interactional procedures from other domains, in so far as possible.

Move Characterization: metacommunication

As we saw earlier, a class of moves whose presence makes itself evident in taxonomies are metacommunicative moves. Such phenomena have been studied extensively by psycholinguists and conversational analysts in terms of notions such as *grounding* and *feedback* (in the sense of Clark (1996) and Allwood (1995), respectively.) and of *repair* (in the sense of Schegloff (1987)). The main claim that originates with Clark & Schaefer (1989) is that any dialogue move m_1 made by A must be grounded (viz acknowledged as understood) by the other conversational participant B before it enters the common ground; failing this, clarification interaction (henceforth *CRification*) must ensue. While this assumption about grounding is somewhat too strong, as Allwood argues, it provides a starting point, indicating the need to interleave the potential for grounding/CRification incrementally; the size of the increments being an important empirical issue. From a semantic theory, we might expect the ability to generate concrete predictions about forms/meanings of MCI utterances in context. More concretely, the adequacy of such a theory requires:

(GCR) Grounding/CRification conditions benchmark: The ability to characterize for any utterance type the update that emerges in the aftermath of successful grounding and the full range of possible CRs otherwise.

Let us make this benchmark more concrete, initially with respect to the content/context of grounding/CRification moves, later with respect to the realization of such moves. There are two main types of MC *inter*actions—acknowledgements of understanding and clarification requests (CRs).[5] A rough idea of the frequency of acknowledgements can be gleaned from the

[5] By far the commonest type of what one might call metacommunicative *intra*actions are *self corrections*, often referred to under the rubric of *disfluencies*, on which more below.

word counts for 'yeah' and 'mmh' in the demographic part of the BNC: 'yeah' occurs 58810 times (rank: 10;10-15% of turns), whereas 'mmh' occurs 21907 times (rank: 30; 5% of turns). Clarification Requests (CRs) constitute approximately 4-5% of all utterances (see e.g. Purver *et al.* (2001); Rodriguez & Schlangen (2004)). Both acknowledgements and CRs, then, constitute central phenomena of interaction, even judged merely in terms of frequency.

An addressee can acknowledge a speaker's utterance, either once the utterance is completed, as in (13a,b), or concurrently with the utterance as in (13c). For conversations where the participants are visible to each other, gesture (head nodding, eye contact etc.) also provides an option by means of which affirmative moves can be made (see Nakano *et al.* (2003).).

(13) a. Tommy: So Dalmally I should safely say was my first schooling. Even though I was about eight and a half. Anon 1: Mm. Now your father was the the stocker at Tormore is that right ? (BNC, K7D)

b. Wizard: Then you want to go north on Speer Boulevard for one and one half miles to Alcott Street.
User: Okay. I want to go right on Speer? (VNS Corpus, Novick & Sutton (1994))

c. A: Move the train ...
B: Aha
A:...from Avon ...
B: Right
A:...to Danville. (Adapted from the Trains corpus)

From this we derive three benchmarks:

(Ack1) Completed Acknowledgements benchmark: accommodate completed acknowledgements.

(Ack2) Incremental Acknowledgements benchmark: accommodate continuation acknowledgements.

(Ack3) Multimodal Acknowledgements benchmark: accommodate gestural acknowledgements.

Although in principle, one can request clarification concerning just about anything in a previous utterance, corpus studies of CRs in both a general corpus Purver *et al.* (2001), as well as task oriented ones Rodriguez & Schlangen (2004); Rieser & Moore (2005) indicate that there are four main categories of CRs:

- **Repetition**: CRs that request the previous utterance to be repeated:

  (14) a. Tim (1): Could I have one of those (unclear)?
  Dorothy (2): Can you have what? (BNC, KW1)

  b. s bust: Great memorial I think really isn't it?
  e bust: Beg pardon?
  s bust: Be a good appropriate memorial if we can afford it. (BNC, KM8)

- **Confirmation**: CRs that seek to confirm understanding of a prior utterance:

  (15) a. Marsha: yeah that's it, this, she's got three rottweilers now and
  Sarah: three? (=Are you saying she's got THREE rottweilers now?)
  Marsha: yeah, one died so only got three now (BNC)
  b. A: Is Georges here?
  B: You're asking if Georges Sand is here.

- **Intended Content**: CRs that query the intended content of a prior utterance:

  (16) a. Tim (5): Those pink things that af after we had our lunch.
  Dorothy (6): Pink things?
  Tim (7): Yeah. Er those things in that bottle.
  Dorothy (8): Oh **I know what you mean.** For your throat? (BNC)
  b. A: Have a laugh and joke with Dick.
  B: Dick?
  A: Have a laugh and joke with Dick.
  B: Who's Dick?

- **Intention recognition**: CRs that query the goal underlying a prior utterance:

  (17) a. X: You know what, the conference might be downtown Seattle. So I may have to call you back on that.
  PT: OK. Did you want me to wait for the hotel then? (Communicator corpus)
  b. Norrine: When is the barbecue, the twentieth? (pause) Something of June.
  Chris: Thirtieth.
  Norrine: A Sunday.
  Chris: Sunday.
  Norrine: Mm.
  Chris: Why? (= *Why do you ask when the barbecue is*)
  Norrine: Becau Because I forgot (pause) That was the day I was thinking of having a proper lunch party but I won't do it if you're going out. (BNC)

The ability to generate and understand such CRs requires correspondingly increasing complexity: from Repetition (which can be done by very simple systems) to Intention recognition, which requires a significantly complex processing architecture. Accordingly, we distinguish:

(CR1) Repetition CR benchmark: Accommodate **Repetition** CRs.
(CR2) Confirmation CR benchmark: Accommodate **Confirmation** CRs.
(CR3) Intended Content CR benchmark: Accommodate **Intended Content** CRs.
(CR4) Intention Recognition CR benchmark: Accommodate **Intention Recognition** CRs.

To conclude our discussion of MCI, let us note some higher level benchmarks. The first is a semantic *non-determinism*, given the fact that an utterance can give rise to distinct updates across participants (grounding in one, CRification in the other):

(SND) Semantic non-determinism: interpretation can lead to distinct updates across conversational participants.

MCI dictates the need for fine-grained utterance representations, given: the emergence of utterance-related presuppositions in the aftermath of grounding (18a,b); the hyperintensional nature of CRification conditions (18c,d)—'lawyer' and 'attorney' are synonymous terms but give rise to distinct CRification conditions; and the existence of syntactic and phonological parallelism conditions on certain CR interpretations (18e,f):

(18) a. A: Banach was born in Łodz. B: It's interesting that the last word you uttered has a letter not on my keyboard.
b. And even rain won't save you this time, Bruce, because you need to win one of the remaining matches. Sorry guys I mentioned 'win' there, you Poms might need to look that word up. (*The Guardian*, test match over by over coverage, 25 Aug 2005).
c. Ariadne: Jo is a lawyer. Bora: A lawyer?/What do you mean a lawyer?/#What do you mean an advocate?/#What do you mean an attorney?
d. Ariadne: Jo is an advocate. Bora: #What do you mean a lawyer?/An advocate?/What do you mean an advocate?/#What do you mean an attorney?
e. A: Did Bo leave? B: Max? (cannot mean: intended content reading: **Who are you referring to?** or **Who do you mean?**)
f. A: Did he adore the book. B: adore? / #adored?

Hence,

(FG) Fine-grained utterance representation benchmark: Provide fine-grained utterance representation to accommodate syntactic and phonological parallelism conditions.

2.2 Fragment Understanding

We distinguish between two classes of non-sentential utterances: sentential fragments and disfluencies.

Sentential fragments

Sentential fragments (SFs) are intuitively complete utterances that lack a verbal (more generally predicative) constituent. SFs include 'short answers', and reprise utterances used to acknowledge or request clarification of prior utterances. Examples of these are provided in boldface in (19):

(19) A: Wasn't he refused the chair in Oxford?
B: **Who?**
A: **Skeat**. Wasn't he refused
B: That's Meak.
A: **Oh Meak, yes.** (London-Lund S.1.9, p. 245)

Estimates of the frequency of SFs are somewhat variable, depending on the classificational criteria applied. de Waijer (2001) provides figures of 40%, 31%, and 30%, respectively, for the percentage of *one word utterances* in the speech exchanged between adults and infant, adult and toddler, and among adults in a single Dutch speaking family consisting of 2 adults, 1 toddler and 1 baby across 2 months. Fernández (2006) cites a figure of 9% for the percentage of utterances lacking a verbal predicate, based on random sampling from (by and large) adult speech in the BNC, a figure that is replicated in other corpus studies she surveys.

There exist a number of recent corpus studies whose taxonomies achieve high coverage. These include Fernández & Ginzburg (2002); Schlangen (2003). The taxonomy of Fernández & Ginzburg (2002) and the distribution it uncovers for the BNC is illustrated in Table 1:

Sentential fragment classes	**Example**	**Total**
Plain Acknowledgement	*A: . . . B: mmh*	599
Short Answer	*A: Who left? B: Bo*	188
Affirmative Answer	*A: Did Bo leave? B: Yes*	105
Repeated Ack.	*A: Did Bo leave? B: Bo, hmm.*	86
Reprise Fragment	*A: Did Bo leave? B: Bo?*	79
Rejection	*A: Did Bo leave? B: No.*	49
Factive Modifier	*A: Bo left. B: Great!*	27
Repeated Aff. Ans.	*A: Did Bo leave? B: Bo, yes.*	26
Helpful Rejection	*A: Did Bo leave? B: No, Max.*	24
Sluice	*A: Someone left. B: Who?*	24
Check Question	*A: Bo isn't here. Okay?*	22
Filler	*A: Did Bo . . . B: leave?*	18
Bare Mod. Phrase	*A: Max left. B: Yesterday.*	15
Propositional Modifier	*A: Did Bo leave? B: Maybe.*	11
Conjunction + frag	*A: Bo left. B: And Max.*	10
Total dataset		**1283**

Table 1. NSUs in a sub-corpus of the BNC

The task of identifying the right SF class can be successfully learned using supervised machine learning techniques Schlangen (2005); Fernández *et al.* (2007). Resolving SF content in context is a more challenging task. Of course the most general benchmark is to achieve comprehensive coverage, relative to a taxonomy such as the above. We can offer some partial benchmarks (as in (SF2) and (SF3)), motivated primarily by frequency: basic answers are crucial in interaction, as reflected in their majoritarian status, similarly with acknowledgements. The reprise fragment benchmark is more challenging: such fragments constitute a very high proportion of CRs, but are frequently ambiguous between uses that have a *confirmation* content and ones that have an *intended content* content (see e.g. (15a) and (16b) above.):

(SF1) Sentential fragment benchmark1: achieve SF wide coverage.
(SF2) Basic Answer Resolution benchmark: accommodate short answers, affirmative answers, and rejection.
(SF3) Reprise Fragment Resolution benchmark: accommodate Reprise fragments, and recognize the potential for ambiguity they exhibit.

SFs are often adjacent to their source. But not always, as illustrated starkly by our initial motivating example (1), repeated here as (20), in which short answers (7) and (9) refer back to the query (1). Data from the BNC (Ginzburg & Fernández, 2005) suggests that this is primarily a feature of short answers in multilogue, though not uncommon in 2 person dialogue either:

```
(20) John:  (1) Okay which one do you think it is?
            (2) Try F1 F1 again and we'll get
     Sarah: (3) Shift and F1?
     Sue:   (4) It's, (5) no.
     John:  (6) No, (7) just F1 F1.
     Sue:   (8) It isn't that.
     John:  (9) F1. (10) Right, (11) and that tells us
     Sue:   (12) It's shift F7.
```

(SF4) Distance benchmark: accommodate long distance short answers.

The final benchmark for SFs concerns their appearance as initiating moves (i.e. without a prior linguistic antecedent or segment initially.). These seem to require a rather sterotypical interactional setting (buying tickets at a train station, querying for directions in a taxi etc). Although such uses do not seem to have been recorded in recent corpus studies, they are clearly not marginal and should be accommodated:

(21) a. Buying a train ticket:
Client: A return to Newcastle please. (=I want a return . . . , please give me a return . . . , . . .)
b. Driver to passenger in a taxi: Where to?

(SF5) Initiating genre sensitive SF benchmark: Accommodate genre sensitive initiating SFs.

Disfluencies

Disfluencies are common in conversation: in the Trains corpus, for instance, 23% of speaker turns contain at least one repair, and 54% of turns with at least ten words contain a repair (Heeman and Allen, 1999). In this area there has been important early work by psycholinguists, most notably Levelt (see e.g. Levelt (1983)), much recent work by speech researchers (e.g. Shriberg (1994)) and corpus-based taxonomies (e.g. Besser & Alexandersson (2007)).

In terms of bare functionality, it is clear that a fundamental benchmark is the ability to be unfazed by disfluencies. In other words, to be able to recognize a disfluency and to effect the appropriate "repair", resulting in a "cleaned up" utterance, as exemplified in (22):

(22) I was *one of the*, I was responsible for all the planning and engineering.
$\mapsto$ I was responsible for all the planning and engineering

(D1) Disfluency benchmark1: Recognize and repair disfluencies

Such an approach using machine learning techniques is demonstrated by Heeman and Allen (1999), who suggest:

> "We propose that these tasks [including detecting and correcting speech repairs, the authors] can be done using local context and early in the processing stream."

Recently, evidence from psycholinguistics has begun emerging that self-corrected material has a long-term processing effect Brennan & Schober (2001); Lau & Ferreira (2005), hence is not being "edited away". It can also bring about linguistic effects in whose interpretation it plays a significant role, for instance anaphora, as in (23a) from Heeman & Allen (1999). In fact, disfluencies yield information: (23a) entails (23b) and defeasibly (23c), which in certain settings (e.g. legal), given sufficient data, can be useful. Moreover, incorporating them in systems' output can improve naturalness (e.g. when speech processing is slow) and improve the user's empathy with the system. Given this, we formulate our second disfluency benchmark:

(23) a. Andy: Peter was, well he was fired.
b. Andy was unsure about what he should say, after uttering 'was'.
c. Andy was unsure about how to describe what happened to Peter.

(D2) Disfluency benchmark2: Explicate disfluency meaning without eliminating disfluencies from context.

3 Approaches to Dialogue System Design

Before presenting a formal framework that is able to account for the various dialogue phenomena described earlier, in this section we briefly describe several important approaches to the design of dialogue systems and evaluate them informally with respect to the benchmarks we have introduced in the previous section. We end with a short description of the Information State approach to dialogue management, closest in spirit to the theory of interaction that we will present in Section 4.

3.1 Basic Architecture of Dialogue Systems

Besides their commercial potential, dialogue systems are also an asset for the dialogue theorist since designing a conversational agent that can communicate naturally with a human can help in the evaluation of theories of dialogue. Of course, for practical reasons researchers do not usually create systems that can talk just about anything. Instead they design systems that are competent only in particular domains and can handle particular tasks—they are task-oriented, domain-dependent conversational systems. This is especially true of commercial systems, which tend to be simpler and less advanced than research prototypes. Applications that involve information retrieval tasks are very common, especially those related to travel planning and management. Other common applications are educational tutoring systems, device management (of in-car or in-home devices), and collaborative problem solving.

To a large extent, the complexity of a system will depend on its application. Most spoken dialogue systems, however, contain the following components: an automatic speech recognizer (ASR) that captures the user's input and converts it to a sequence of words; a natural language understanding (NLU) component that produces a meaningful representation of the input utterance; a dialogue manager (DM) that controls the dialogue flow by integrating the user contributions and deciding what to say next; a source of domain and task knowledge (KB); a natural language generation (NLG) component that chooses the words to express the response together with their prosody; and a text-to-speech (TTS) synthesis engine that outputs a spoken form of the response. Figure 1 shows the basic architecture of a spoken dialogue system. Similar diagrams and much more detailed explanations of the different components can be found e.g. in (McTear, 2004; Delgado & Araki, 2005; Jurafsky & Martin, 2008).

The DM component is often considered the core of a dialogue system. It receives a representation of the input utterance from the NLU module, keeps track of some sort of dialogue state, interfaces with the external knowledge sources, and decides what should be passed to the NLG module. In the remainder of this section, we discuss three main types of dialogue management architectures: Finite-state DMs, frame-based DMs, and inference-based DMs.

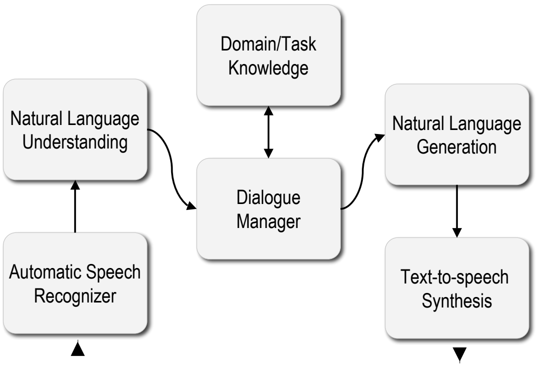

Domain/Task Knowledge
Natural Language Understanding
Dialogue Manager
Natural Language Generation
Automatic Speech Recognizer
Text-to-speech Synthesis

Figure 1. Basic components of a spoken dialogue system

We finish with a sketch of the Information State Update approach to dialogue management.

3.2 Paradigmatic approaches to Dialogue Management

Finite-state Dialogue Management

The simplest dialogue managers represent the structure of the dialogue as a finite-state transition network. Figure 2 shows a basic finite-state DM for a ticket booking application. We can see that the states in the network are atomic and correspond to system contributions, while the transitions between states correspond to system actions dependent of the user responses. The set of possible paths along the graph represents the set of legal dialogues.

Finite-state DM architectures give rise to conversational agents that fully control the dialogue. The system has the initiative at all times: it utters a series of prompts in a predetermined order, interpreting anything the user says as a direct response to the latest prompt. Any (part of a) user utterance that cannot be interpreted as directly addressing the latest prompt is either ignored or misrecognized. Restricting what the user can say to the latest prompt is often seen as an *advantage* of finite-state architectures by the dialogue system's engineer, as this allows one to simplify the ASR and NLU components of the system. Indeed, finite-state systems tend to use extremely simple understanding components, often limited to language models associated with particular dialogue states and tuned to recognize typical responses to a given prompt (such as city names or dates).

There are a few toolkits that allow fast development of finite-state systems, such as the Nuance Dialog Builder or the CSLU toolkit (McTear, 1998). For a general overview of FSM-based systems see McTear (2004).

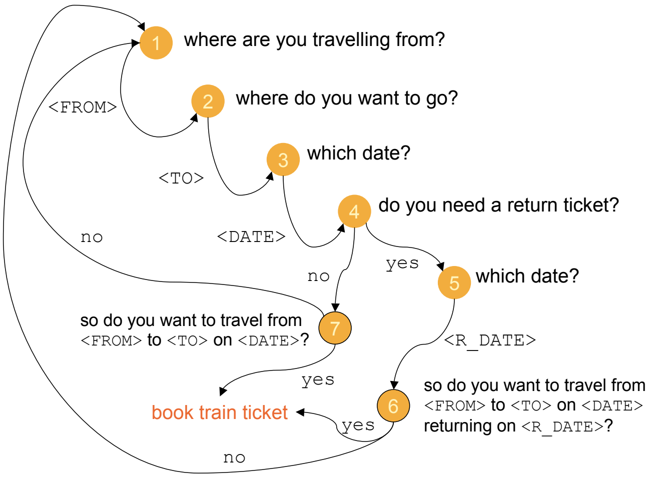

1 where are you travelling from?
2 where do you want to go?
3 which date?
4 do you need a return ticket?
5 which date?
<FROM>
<TO>
<DATE>
no
no
yes
<R_DATE>
7 so do you want to travel from <FROM> to <TO> on <DATE>?
6 so do you want to travel from <FROM> to <TO> on <DATE> returning on <R_DATE>?
yes
yes
book train ticket
no

Figure 2. Finite state machine for a simple ticket booking application

Frame-based Dialogue Management

Frame-based DM offers some advantages over finite-state systems. Although the system's prompts and the range of user contributions that can be handled still need to be determined at design time, frame-based DM allows for more flexibility at the level of the dialogue flow. In frame-based DM, the dialogue states the system keeps track of—so called *frames*—have a richer internal structure than the atomic nodes of finite-state transition networks. A frame typically consists of a series of slots, values and prompts, as exemplified in Figure 3, where each slot corresponds to some bit of information the system needs to get from the user. Again, frame-based systems are especially well-suited for information tasks, where the system needs to find out some information from the user in order to execute some task (such as booking a ticket or retrieving some information from a database).

In finite-state DM the system's contributions are determined by the transition function of the FS network. In contrast, a frame-based dialogue manager includes a control algorithm that determines what to say next given the contents of the frame. The control algorithm keeps track of the slots filled so far and makes sure that filled slots are not revisited. The slots in the frame can be filled in any order and a single user's response can fill in more than one slot. The control algorithm specifies which frame configurations need to be true for a particular prompt to be relevant. This specification can be as general as se-

slot	**value**	**prompt**
`ORIGIN`	`unknown`	`From which city are you leaving?`
`DESTINATION`	`unknown`	`Where are you travelling to?`
`DATE`	`unknown`	`When do you want to travel?`

Figure 3. A simple frame

lecting the first prompt in the frame which has an `unknown` value, or more specific in the form of conditions such as '`If ORIGIN is filled and DESTINATION is unknown, utter DESTINATION prompt, else utter ORIGIN prompt`'.

Thus, although the range of possible contributions is fixed in advance, in contrast to FS systems, the dialogue flow is not completely predetermined at design time but driven by interaction. This increased flexibility in turn requires more complex language models that can deal with multi-slot filling responses.

For a description of some systems that use a frame-based architecture see Aust *et al.* (1995b), Constantinides *et al.* (1998) or Seneff & Polifroni (2000).

Inference-based Dialogue Management

Inference-based DM differs substantially from DM based on frames or fintestate networks. In this approach, which combines planning techniques used in AI with ideas from speech act theory (Austin, 1962; Searle, 1969), dialogue management is considered a planning task driven forward by a *rational agent* (the dialogue system), whose behaviour is determined by inference mechanisms. The approach, developed at the University of Toronto by Perrault and his collaborators (Cohen & Perrault, 1979; Allen & Perrault, 1980), models rational agents in terms of Beliefs, Desires and Intentions (BDI). The latter are formalised as predicates or modal operators in some version of first-order (modal) logic. Agents are also equipped with a set of general rationality axioms and a set of plans and goals, plus a component for automatic plan-based reasoning such as a theorem prover.

Dialogue moves are seen as instances of goal-oriented rational actions, all of which are formalised as plans for goal achievement. A common way of formalising plans is by means of action schemata. These can take different forms, but minimally distinguish between the preconditions required for an action to take place and its effects. Figure 4 shows a couple of examples of possible plans to book a flight and to request some information.

Dialogue managers based on the BDI model of rational agents typically keep track of a repository of shared beliefs or *common ground*, the goal motivating the current dialogue contribution, and information on the status of problem solving (e.g. on whether the preconditions of the current plan are met and its goal has been achieved). Deciding what the system should say

BOOK(S, U, T**)**
`Constraints:` $\mathit{System}(S) \wedge \mathit{User}(U) \wedge \mathit{Ticket}(T)$
`Goal:` $\mathit{Booked}(S, U, T)$
`Preconditions:` $\mathit{Knows}(S, \mathit{Origin}(T)) \wedge \mathit{Knows}(S, \mathit{Dest}(T)) \wedge \ldots$
`Effects:` $\mathit{Booked}(S, U, T)$

INFO_REQUEST(A, B, P**)**
`Constraints:` $\mathit{Speaker}(A) \wedge \mathit{Addressee}(B) \wedge \mathit{Prop}(P)$
`Goal:` $\mathit{Know}(A, P)$
`Preconditions:` $\neg \mathit{Know}(A, P) \wedge \mathit{Desire}(A, \mathit{Know}(A, P) \wedge \mathit{Believe}(A, \mathit{Know}(B, P)) \wedge \ldots$
`Effects:` $\mathit{Believe}(B, \mathit{Desires}(A, \mathit{Know}(A, P)))$

Figure 4. Goal-oriented action schema

next consists in advancing a step in the current plan. For instance, a system that is following a plan to book a flight for the user may decide to utter an INFO_REQUEST move with the goal of satisfying some preconditions of the booking plan, such as knowing the origin and the destination of the trip.

As mentioned earlier, plans are complemented by a set of general rationality axioms. These typically include cooperative axioms stating that agents adopt the intentions of their interlocutors (as long as they do not contradict their own ones). Also note that, as exemplified by the `Effects` of the INFO_REQUEST action scheme in Figure 4, interpreting an utterance amounts to infering the plan-based intentions of the speaker.

Inference-based systems are intended for advanced tasks such as collaborative problem solving. This requires NLU components that are fairly sophisticated since the range of possible user utterances is much less constrained than in purely informational tasks. The TRAINS/TRIPS integreted dialogue system (Allen *et al.*, 1995; Ferguson & Allen, 1998) is one of the most influential systems implementing this approach, but see also Sadek & de Mori (1998). The last chapter of (Allen, 1995) provides a good overview of inference-based DM.

3.3 Comparison of Dialogue Management approaches

In this section we look at how well standard versions of finite state-based, frame-based and inference-based approaches to dialogue management can deal with the benchmarks introduced in Section 2. A summary is shown in Table 5.

Query and Assertion benchmarks

As we mentioned earlier, queries and assertions are the commonest move types in interaction with dialogue systems. All DM approaches we have seen can accommodate direct simple answers to queries and hence meet benchmark Q1. However, accounting for the other query benchmarks is more problematic. The ability to satisfy benchmarks Q2a and Q2b (accommodation of non-resolving answers and follow-up queries to them) in part depends on the sophistication of the NLU and KB components: to interpret a contribution as a non-resolving answer the system needs to be able to reason over some sort of ontology with subtyping (in order to figure out e.g. that 'Germany' may count as an answer to a destination prompt but is probably not specific enough). This capability is standard in inference-based systems, while it is very unlikely to be present in a pure finite-state system, since the main advantage of this approach is the simplification of components by restricting possible user input. Assuming the capability to recognizing non-resolving answers was available, in a finite-state DM sub-queries to such answers could in principle be integrated as additional states. In a frame-based DM, non-resolving answers could be integrated by including a `non-resolving` value type that would trigger follow-up queries relative to each kind of slot. Within the plan-based approach of inference-based DM, an answer is considered 'resolving' if it fullfills the relevant goals in the plan that motivated the question. Goals that are not fully satisfied motivate follow-up queries (*U: I need to travel some time in August. S: And what day in August did you want to travel?*).

Accommodating over-informative answers (benchmark Q3) poses practical problems for finite-state systems. They could in principle be integrated as additional states (e.g. an extra state for answers that include both information about the destination and the origin; another one for those that include destination and date, and so forth), but only as long as they can be predicted at design time. Note however that even if they could be predicted, including them into the finite-state network would easily lead to an explosion of the number of states, which would produce a rather cumbersome structure. Frame-based DMs are better equipped to deal with over-informative answers since multiple slots can be filled in by a single user response. Thus, if the over-informative answer contains information that directly addresses exiting slots, this can be utilized to drive the task forward. In inference-based systems, over-informative answers are seen as a product of domain plan-recognition: they are treated as cooperative responses that help achieve the recognized plan of the interlocutor by providing information that is required to achieved the current goal (e.g. the exchange *U: When is the train leaving? S: At 5:04, platform 12* can be explained by the ability of the system to recognize the user's plan to take the train).

Benchmarks Q4 and Q5 (accommodation of sub-queries and accommodation of topic-changing responses) are highly problematic for finite-state and frame-based DMs. Sub-queries can be handled only to the extent they can be predicted in advance and, as with Q3, this could lead to tractability problems. There are no means for these structured approaches to interpret an irrelevant response as a change of topic. An inference-based system would do slightly better. Regarding sub-queries, it would only be able to acommodate those that are goal-related (such as *U: How much is a ticket to Hamburg? S: When do you want to travel?*). A response that does not match any step in the current plan could potentially be interpreted as topic-changing. However the system would not be able to distinguish this kind of “irrelevance” from situations where the mismatch requires clarification.

We move now to the assertion benchmarks A1 and A2 (integration of propositional content and accommodation of disagreement, respectively). None of them is satisfied by finite-state systems. Benchmark A1 is not satisfied because in a finite-state architecture states do not have any internal structure and therefore there is no propositional or contextual update beyond the information that emanates from the current position in the graph. This also rules out the possibility of accounting for disagreement since there is no propositional content which the agent can disagree about. Frame-based DMs make use of some limited form of contextual update since the control algorithm keeps track of the slots filled so far, but their simple architecture cannot accommodate disagreements. Certainly, inference-based systems satisfy A1 (one of the effects of asserting a propostion P is that P becomes common knowledge or common believe). As for A2, they can accommodate conflicting beliefs and hence some form of diagreement. However, accounting for disagreement in the sense of non-cooperativity is more problematic since the BDI model is basically designed for cooperative tasks without conflicting goals.

The final two benchmarks within this section deal with scalability to monologue and multilogue (SC) and domain adaptability (DA). None of the approaches we have discussed satisfies SC—they are all designed for two-agent dialogue. Finite-state and frame-based DMs are strongly domain-dependent (except perhaps in their metacommunicative behaviour, which we discuss below). In contrast, the BDI model underlying inference-based DMs aims to be a domain-independent theory of rational action. Although it is unclear to what extent procedures employed in actual inference-based systems can effectively be reused, in principle general rationality axioms should be valid across domains.

Metacommunication benchmarks

Given the high number of recognition problems that dialogue systems face due to the poor performance of ASRs, metacommunicative interaction plays an important role in such implemented systems. Finite-state and frame-based architectures usually take a generative perspective, where metacommunicative behaviour comes from the system. This is not surprising since these approaches are highly system-initiating in design. Inference-based systems, on the other hand, have also addressed the problem of interpreting metacommunicative utterances.

The metacommunicative potential of finite-state and frame-based systems in rather similar. What in finite-state systems can be achieved by multiplying the number of states and transitions, in frame-based systems can be implemented by adding extra types of slot values and increasing the complexity of the control algorithm. Finite-state systems usually include states to handle situations when there is no input or no recognition, as well as when there is a need to confirm information provided by the user (as in states 7 and 8 of the transition network in Figure 2). Acknowledgements of completed contributions (benchmark A1) can similarly be integrated as additional states. In a frame-based architecture, slot values (such as `no-match`) and/or confidence scores associated with filled values can be used to decide whether a contribution can be acknowledged or whether there is need to ask for repetition or confirmation. Thus, at least from a generation perspective, finite-state and frame-based DMs meet benchmarks A1, CR1 (repetition CRs) and CR2 (confirmation CRs). However, more complex types of CRs such as those that query the intended content or the intention of a prior utterance (benchmarks CR3 and CR4) cannot be accommodated by these systems.

Satisfying benchmark A2 (accommodation of continuation acknowledgements) would require an incremental architecture not present in any of the systems we have discussed, where transitions to a different state are triggered by full utterances or moves. Gestural acknowledgements (benchmark A3) could in principle be integrated provided that the system is able to process multimodal input and that the gestural acknowledgements acknowledge complete contributions.

Traditionally, inference-based DM has not been too concerned with metacommunication, focussing instead on plan recognition and cooperativity at the task domain. Simple grounding and clarification behaviour such as acknowledgements and repetition/confirmation CRs can in principle be accommodated in a way akin to the strategies we have already discussed (e.g. by using confidence scores or evaluating the output of the NLU component, which is more sophisticated in these systems). To account for other kinds of clarification sub-dialogues, a hierarchical plan structure that incorporates discourse plans—or *metaplans* in the terminology of Litman & Allen (1984)—has been proposed. The idea is that metaplans are performed to obtain knowledge necessary to perform task plans and are inferred when an utterance cannot be interpreted as a step in the current domain plan. For instance, in the following dialogue *S: At 5:04, platform 12. U: Where is it?*, the system would interpret the user's question as a metaplan to find additional information to perform the task plan (presumably taking a train). Thus, in this approach CRs that go beyond asking for repetition or confirmation are only possible in as much they are ultimately related to task plans.

The last two benchmarks related to metacommunication are SND (possibility of different updates across participants, or *semantic non-determinism*) and FG (fine-grained representations). The latter is not satisfied by any of the DM approaches we have considered: dialogue managers across the board get as input some sort of semantic representation. Operating on syntactic and phonological representations would be extremely complicated, if at all possible, in finite-state or frame-based architectures. Inference-based systems could in principle include rich utterance representations (by using a parser that generates the desired output), but it is unclear how a plan-based approach would deal with them. SND is not satisfied either, at least explicitly.

To some extent, any state that leads to a repetition CR implicitly assumes that there is an asymmetry between the user intended utterance and the system's interpretation of it (or lack thereof). But this is not explicitly modelled.

Fragment Understanding benchmarks

We now turn to the last set of benchmarks, which are realted to fragment understanding. Since these benchmarks are directly concerned with how meaning is assigned to fragmentary utterances, they are more tightly linked to the NL modules than the move-related benchmarks (although as we shall see in Section 4, their resolution requires a fair amount of interaction between the linguistic modules and the dialogue manager, which is the module that represents context).

While dialogue systems do not achieve comprehensive coverage of the corpus-based taxonomies of sentential fragments we have mentioned in Section 2.2 (as required by benchmark SF1), they are typically able to accommodate basic fragmentary answers (benchmark SF2). For instance, a state-dependent language model can process short answers, affirmative answers and rejections, which, as long as they are direct simple answers, could be correctly interpreted by a finite-state DM. We have seen examples of this in Figure 2. Similar techniques can be used in frame-based systems where, as mentioned earlier, language models tend to be more complex given the possibility of multi-slot filling.

Genre-sensitive initiating SFs (benchmark SF5) cannot be accommodated by a finite-state DM since the system has the initiative at all times. They can, however, be processed by frame-based systems, where the frame can be seen as encoding the relevant genre. For instance, if a user starts a dialogue with the utterance *To Hamburg, on Tuesday*, a frame-based DM for the travel domain with an appropriate language model could fill in the destination and date slots. However long distance short answers (benchmark SF4) cannot easily be accommodated by finite-state or frame-based DMs.

In inference-based systems the interpretation of basic types of fragments (both responsive and initiating) is achieved by inferring the domain-dependent goals of the speaker (see e.g. Carberry (1990)). However it is not at all clear how long distance short answers could be accommodated in this approach.

Given our discussion of the metacommunication benchmarks above, reprise fragments (benchmark SF3) cannot be successfully accommodated by any of the considered DM approaches.

Finally we come to the disfluency benchmarks. The ability to recognize and repair disfluencies (benchmark D1) depends on the ASR/NLU components of a system. For instance, statistical language models tend to be rather robust for disfluencies. A robust parser can then be applied to their output to extract the relevant information (relative to the latest system prompt, to any slot in a frame, or to the current domain plan). This sort of setting is more common in frame- and inference-based systems than in finite-state ones, but in theory these processing components could be combined with any kind of dialogue manager. In contrast, D2 (accommodation of disfluency meaning without elimination of disfluencies from context) is a much more challenging benchmark that is not met by current systems.

Table 5 summarizes the comparison of the three approaches to dialogue management we have reviewed with respect to the benchmarks introduced in Section 2. For each dialogue management approach (finite-state, frames, and inference-based), the symbol ✓ indicates that the approach safisfies the benchmark in the corresponding row; ∼ that the benchmark could be met with some caveats, as explained in the text above; and — that the benchmark is not met by a standard version of the approach.

3.4 The information State Update Framework

To conclude this section, we shall briefly introduce the main ideas of the information state update (ISU) framework. The approach was developed during the European TRINDI project (Consortium, 2000) as a general framework to implement different kinds of dialogue management models. According to Traum & Larsson (2003), the components of an ISU model are the following:

- A formal representation of the information state (IS) and its components;
- A set of dialogue moves that trigger IS updates;
- A set of update and selection rules that govern how moves change the IS and how changes licence future moves;
- An update strategy for deciding which rules to apply when.

Regardless of the particular model implemented within the framework, what makes the ISU approach attractive is the declarative way in which dialogue states and transitions between states are formulated. In fact, the approach can be seen as a extension of the frame-based architecture, where states can have a much more complex structure than slot-value frames and the procedural rules of the control algorithm are formulated as more general and declarative update and selection rules.

There are some toolkits to implement ISU-based dialogue managers and system architectures, most notably the TrindiKit (Larsson & Traum, 2000) and DIPPER (Bos *et al.*, 2003).[6] GODIS (Cooper *et al.*, 2000) and EDIS (Matheson *et al.*, 2000) are some of the systems implemented using this framework. In the next section we present a theory of dialogue interaction which is ISU-based in spirit.

[6] See `http://www.ling.gu.se/projekt/trindi/trindikit/` and `http://www.ltg.ed.ac.uk/dipper/` for up-to-date information on the toolkits.

Benchmarks	FSMs	Frames	Inference
query and assertion			
Q1 simple answers	✓	✓	✓
Q2a non-resolving answers	~	✓	✓
Q2b follow up queries	~	✓	✓
Q3 overinformative answers	~	✓	✓
Q4 sub-questions	—	—	~
Q5 topic changing	—	—	—
A1 propositional content update	—	~	✓
A2 disagreement	—	—	~
SC scalability	—	—	—
DA domain adaptability	—	—	~
metacommunication			
Ack1 completed acknowledgements	✓	✓	✓
Ack2 continuation acknowledgements	—	—	—
Ack3 gestural acknowledgements	~	~	~
CR1 repetition CRs	✓	✓	✓
CR2 confirmation CRs	✓	✓	✓
CR3 intended content CRs	—	—	—
CR4 intention recognition CRs	—	—	~
SND distinct updates	—	—	—
FG fine-grained representations	—	—	—
fragments			
SF1 wide coverage of SFs	—	—	—
SF2 basic answer resolution	✓	✓	✓
SF3 reprise fragment resolution	—	—	—
SF4 long distance short answers	—	—	—
SF5 genre sensitive initiating SFs	—	✓	✓
D1 recognize and repair disfluencies	✓	✓	✓
D2 keep disfluencies in context	—	—	—

Figure 5. Comparison of dialogue management approaches

4 Interaction and Meaning

In this section we sketch a comprehensive theory of interaction and meaning, indicating how it can be used to fulfill the various benchmarks we specified in earlier sections. This theory is based on the framework KoS Ginzburg (1994, 1996); Ginzburg & Cooper (2004); Larsson (2002); Purver (2006); Fernández (2006); Ginzburg (2009). The latter reference contains a detailed exposition of the theory sketched below. Other comprehensive accounts of a theory of dialogue include work in the PTT framework[7] (e.g. Poesio & Traum (1997, 1998); Matheson *et al.* (2000); Poesio & Rieser (2009)) and work within Segmented Discourse Representation Theory (SDRT) (e.g. Asher & Lascarides (2003, 2008)).

In abstract terms, the model we present here revolves around the information states dialogue participants possess and how these get modified as a consequence of utterances and related interactions. Our exposition proceeds in a number of stages. First, we explicate the proposed structure of information states. We then illustrate how illocutionary interaction can be analyzed—the updates on the information states will be triggered entirely by dialogue *moves.* We then consider domain specificity and how it can be incorporated into this picture—this will involve a minor refinement of the information states. Our final refinement will involve the integration of illocutionary and metacommunicative interaction: this will have two main consequences. Updates will be triggered by utterances—data structures involving parallel representation of phonological, syntactic, semantic, and contextual information—and the information states will be refined slightly to take into account the potential for partial understanding.

Before we enter into all this, however, we introduce briefly the logical formalism in which KoS is formulated, Type Theory with Records.

4.1 Type Theory with Records: the basics

As the underlying logical framework, we use Type Theory with Records (TTR) (Cooper, 2006), a model–theoretic descendant of Martin-Löf Type Theory (Ranta, 1994). This provides a formalism with which to build a semantic ontology, and to write conversational and grammar rules. After introducing TTR, we will explain why we use TTR rather than typed feature structure–based formalisms (see the chapter on Computational Semantics and e.g. (Carpenter, 1992; Penn, 2000)), whose notation is quite similar and which have been used in much work in computational linguistics.

The most fundamental notion of TTR is the typing *judgement* $a : T$ classifying an object a as being of type T. A record is an ordered tuple of the form (24)—each assignment to a field constituting a component of the tuple. Crucially, each successive field can depend on the values of the preceding fields:

(24) a. $$\begin{bmatrix} l_i = k_i \\ l_{i+1} = k_{i+1} \ \ldots \\ l_{i+j} = k_{i+j} \end{bmatrix}$$

[7] PTT is not an acronym, but has some relation to the initials of its progenitors.

b. $\begin{bmatrix} \mathrm{x} = \mathrm{a} \\ \mathrm{y} = \mathrm{b} \\ \mathrm{prf} = \mathrm{p} \end{bmatrix}$

A record type is simply an ordered tuple of the form (25), where again each successive type can depend on its predecessor types within the record:

(25) $\begin{bmatrix} l_i : T_i \\ l_{i+1} : T_{i+1} \ \ldots \\ l_{i+j} : T_{i+j} \end{bmatrix}$

Cooper (2006) proposes that situations and events be modelled as records. Situation and event types are then directly accommodated as record types. The type of a situation with a woman riding a bicycle would then be the one in (26a). A record of this type (a *witness* for this type) would be as in (26b), where the required corresponding typing judgements are given in (26c):

(26) (a) $\begin{bmatrix} \text{x: IND} \\ \text{c1: woman(x)} \\ \text{y: IND} \\ \text{c2: bicycle(y)} \\ \text{time : TIME} \\ \text{loc:LOC} \\ \text{c3: ride(x,y,time,loc)} \\ \end{bmatrix}$ (b) $\begin{bmatrix} \ldots \\ \mathrm{x} = \mathrm{a} \\ \mathrm{c1} = \mathrm{p1} \\ \mathrm{y} = \mathrm{b} \\ \mathrm{c2} = \mathrm{p2} \\ \mathrm{time} = \mathrm{t0} \\ \mathrm{loc} = \mathrm{l0} \\ \mathrm{c3} = \mathrm{p3} \\ \ldots \end{bmatrix}$

(c) a : IND; p1 : woman(a); b : IND; p2 : bicycle(b); t0 : TIME; l0 : LOC;p3 : ride(a,b,t0,l0);

TTR offers a straightforward way for us to model propositions and questions using records, record types, and functions. A proposition is a record of the form in (27a). The type of propositions is the record type (27b) and truth can be defined as in (27c):

(27) a. $\begin{bmatrix} \text{sit} = r_0 \\ \text{sit-type} = T_0 \end{bmatrix}$

b. $\begin{bmatrix} \text{sit : Record} \\ \text{sit-type : RecType} \end{bmatrix}$

c. A proposition $\begin{bmatrix} \text{sit} = r_0 \\ \text{sit-type} = T_0 \end{bmatrix}$ is true iff $r_0 : T_0$

A question can be identified as a propositional abstract, which in TTR amounts to being a function from records into propositions:

(28) a. who ran

b. TTR representation—$(\mathrm{r} : \begin{bmatrix}\mathrm{x : Ind}\\ \mathrm{rest : person(x)}\end{bmatrix})\ \begin{bmatrix}\mathrm{sit} = r_1\\ \mathrm{sit\text{-}type} = \begin{bmatrix}\mathrm{c : run(r.x)}\end{bmatrix}\end{bmatrix}$

That is, a function that maps records $\mathrm{r : T}_{who} = \begin{bmatrix}\mathrm{x : Ind}\\ \mathrm{rest : person(x)}\end{bmatrix}$ into propositions of the form $\begin{bmatrix}\mathrm{sit} = r_1\\ \mathrm{sit\text{-}type} = \begin{bmatrix}\mathrm{c : run(r.x)}\end{bmatrix}\end{bmatrix}$

To explain the motivation for adopting TTR over a typed feature structure–based approach, we illustrate the difference in the respective treatment of utterance representation. In TTR utterance events, like other events, are a kind of record, whereas lexical entries and phrasal rules are explicated as record types. One could, for instance, posit the sound/syntax/meaning constraint in (29a) as a rule of English. For a speech event *se*0, (29b), to be classified as being of this type, the requirements in (29c) will need to be met:[8]

(29) a.

$$\begin{bmatrix}
\textsc{phon} : \texttt{who did jo leave}\\
\textsc{cat} = \mathrm{V[+fin] : syncat}\\
\textsc{c-params} : \begin{bmatrix}\mathrm{s0{:}\ SIT}\\ \mathrm{t0{:}\ TIME}\\ \mathrm{j{:}\ IND}\\ \mathrm{c3{:}\ Named(j,jo)}\end{bmatrix}\\
\mathrm{cont} = (\mathrm{r} : \begin{bmatrix}\mathrm{x : Ind}\\ \mathrm{rest : person(x)}\end{bmatrix})\begin{bmatrix}\mathrm{sit = s0}\\ \mathrm{sit\text{-}type = Leave(j,r.x,t0)}\end{bmatrix} : \mathrm{Questn}
\end{bmatrix}$$

b.

$$\begin{bmatrix}
\textsc{phon} = \mathrm{di\ jo\ liv}\\
\textsc{cat} = \mathrm{V[+fin]}\\
\textsc{c-params} = \begin{bmatrix}\mathrm{s0 = sit0}\\ \mathrm{t0 = time0}\\ \mathrm{j = j0}\\ \mathrm{c3 = c30}\end{bmatrix}\\
\mathrm{cont} = (\mathrm{r} : \begin{bmatrix}\mathrm{x : Ind}\\ \mathrm{rest : person(x)}\end{bmatrix})\begin{bmatrix}\mathrm{sit = s0}\\ \mathrm{sit\text{-}type = Leave(j,r.x,t0)}\end{bmatrix}
\end{bmatrix}$$

[8] A convention we employ here to distinguish phonological tokens and types is to refer to the latter with English words and the former with a mock representation of their pronunciation.

c. hu di jow liv : `who did jo leave`;
sit0 : SIT, time0 : TIME, j0 : IND, c30 : Named(j0,jo)
cont0 = $(\text{r} : \begin{bmatrix}\text{x : Ind}\\ \text{rest : person(x)}\end{bmatrix})\begin{bmatrix}\text{sit} = \text{sit0}\\ \text{sit-type} = \text{Leave(j0,time0)}\end{bmatrix}$: Questn

Specifically: a witness for the type (29a) includes a phonetic token, contextual parameters—a situation, a time, an individual named Jo—and the question entity $(\text{r} : \begin{bmatrix}\text{x : Ind}\\ \text{rest : person(x)}\end{bmatrix})\begin{bmatrix}\text{sit} = \text{sit0}\\ \text{sit-type} = \text{Leave(j0,r.x,time0)}\end{bmatrix}$, a function from records into propositions. Thus, the fact that C-PARAMS represents the type of entities needed to instantiate a meaning is a direct consequence of what it means to be a witness of this type. In addition, the values of the CONT field *are* already the semantic entities. Hence, to take one example, the function in (30a) is of the type in (30b), which is a supertype of the type in (30c). This latter is the type of a question such as (30d). These type assignments enable us to explain the fact that (30c) is intuitively a sub–question of (30a) and to define various notions of answerhood (see e.g. Ginzburg (2005)):

(30) a. $\text{r} : \text{T}_{who} \mapsto \begin{bmatrix}\textit{sit} & = r_1\\ \text{sit-type} & = \text{c: leave(r.x,t)}\end{bmatrix}$

b. $(\text{T}_{who}\ (= \begin{bmatrix}\text{x} & : \text{Ind}\\ \text{rest} & : \text{person(x)}\end{bmatrix}) \rightarrow \text{Prop})$

c. $\text{r} : \text{T}_0 = \begin{bmatrix}\ \end{bmatrix}$

$\mapsto \begin{bmatrix}\textit{sit} & = r_1\\ \text{sit-type} & = \text{c: leave(j,t)}\end{bmatrix}$

d. $(T_0 \rightarrow \mathit{Prop})$

This explanatory state of affairs contrasts with an account of such examples in a typed feature structure–based approach (e.g. Ginzburg & Sag (2000)), given in (31). This AVM *looks* very much like the type (29a), but the appearance in this case is deceiving.

(31)
$$
\begin{bmatrix}
\text{PHON} & \texttt{who did jo leave} \\
\text{CAT} & \text{S} \\
\text{C-PARAMS} & \left\{ \begin{array}{l} \begin{bmatrix} \text{INDEX} & \text{j} \\ \text{RESTR} & \left\{ \textit{named}(\text{Jo})(\text{j}) \right\} \end{bmatrix}, \\ \begin{bmatrix} \text{INDEX} & \text{t} \\ \text{RESTR} & \left\{ \textit{precedes}(\text{t,k}) \right\} \end{bmatrix}, \\ \begin{bmatrix} \text{INDEX} & \text{s} \\ \text{RESTR} & \{\} \end{bmatrix} \end{array} \right\} \\
\text{CONT} & \begin{bmatrix} \textit{question} & \\ \text{PARAMS} & \left\{ \begin{bmatrix} \text{IND} & k \\ \text{RESTR} & \{\textit{person}(k)\} \end{bmatrix} \right\} \\ \text{PROP} & \begin{bmatrix} \text{SIT} & \text{s} \\ \text{SOA} & \text{leave(j,k,t)} \end{bmatrix} \end{bmatrix}
\end{bmatrix}
$$

In (31) CONT is *intended* as representation of the abstract in (32)

(32) $\lambda x_{person(x)} leave(j, x, t)$

But, as Penn (2000) puts it (in discussing a related set of issues), "At this point, feature structures are not being used as a formal device to represent knowledge, but as a formal device to represent data structures that encode formal devices to represent knowledge'.[9] Similarly, C-PARAMS is *intended* as a representation of the contextual parameters that need to be instantiated, but there is no explicit way of modelling this.

This latter point can be amplified. As we discussed in section 2.1, the interaction over grounding of a speaker A's utterance u addressed to B typically leads to two outcomes: either (a) B acknowledges u (directly, gesturally or implicitly) and then responds to the content of u. Alternatively, B utters a clarification question about some unclear aspect of u. As we will see in section 4.7, this interaction can be explicated as an attempt to find a type T_u that uniquely classifies u. This involves *inter alia* recognizing the words used and instantiating the contextual parameters specified in T_u. CRification involves utilizing a partially instantiated content and posing a question constructed from u and T_u. TTR enables a theory of such interaction to be developed:

- **Simultaneous availability of utterance types and tokens**: in TTR both utterance tokens (records) and signs (record types) become available simultaneously in a natural way.

[9] Penn (2000), p. 63.

- **Partially instantiated contents**: a partial witness for C-PARAMS field T_u.c-params is a record r_0 that is extendible to r_1 such that $r_1 : T_u$.c-params. This is exemplified in (33b), where r_0 lacks fields for $j, c3$ from (33a):

(33) a. $T_u\text{.c-params} = \begin{bmatrix} \text{s0: SIT} \\ \text{t0: TIME} \\ \text{j: IND} \\ \text{c3: Named(j,jo)} \end{bmatrix}$

b. $r_0 = \begin{bmatrix} \textsc{phon} = \text{di jo liv} \\ \textsc{cat} = \text{V[+fin]} \\ \textsc{c-params} = \begin{bmatrix} \text{s0} = \text{sit0} \\ \text{t0} = \text{time0} \end{bmatrix} \end{bmatrix}$

c. $r_0 = \begin{bmatrix} \textsc{phon} = \text{di jo liv} \\ \textsc{cat} = \text{V[+fin]} \\ \textsc{c-params} = \begin{bmatrix} \text{s0} = \text{sit0} \\ \text{t0} = \text{time0} \\ \text{j} = \text{j0} \\ \text{c3} = \text{c30} \end{bmatrix} \end{bmatrix}$

- **Constructing clarification questions on the fly**: a crucial ingredient in this modelling is the ability to build functions from utterance tokens and utterance types into types of contexts, characterized in terms of various semantic objects such as propositions and questions. This is straightforward in TTR given the fact that it enables direct use of λ-calculus tools.

In contrast to these tools, all of which are intrinsic to TTR, typed feature structure–based formalisms can only simulate functions, abstraction, and assignments. Nor do they have types and tokens simultaneously as first class citizens.

4.2 Information States

We analyze conversations as collection of dynamically changing, coupled information states, one per conversational participant. The type of such information states is given in (34a). We leave the structure of the private part unanalyzed here, (for details on this, see Larsson (2002)). The dialogue gameboard (DGB) represents information that arises from publicized interactions. Its structure (or rather a preliminary version suitable for analyzing illocutionary interaction) is given in (34b):

(34) a. TotalInformationState (TIS):

$\begin{bmatrix} \text{dialoguegameboard : DGB} \\ \text{private : Private} \end{bmatrix}$

b. DGB (initial definition)

$$\begin{bmatrix} \text{spkr : Ind} \\ \text{addr : Ind} \\ \text{c-utt : addressing(spkr,addr)} \\ \text{Facts : Set(Prop)} \\ \text{Moves : list(IllocProp)} \\ \text{QUD : poset(Question)} \end{bmatrix}$$

- The spkr/hearer roles serve to keep track of turn ownership.
- FACTS represents the shared knowledge conversational participants utilize during a conversation. More operationally, this amounts to information that a conversational participant can use embedded under presuppositional operators.
- Moves: from within FACTS it is useful to single out LatestMove, a distinguished fact that characterizes the content of the most recent move made. The main motivation is to segregate from the entire repository of presuppositions information on the basis of which coherent reactions could be computed. As we see below (e.g. when discussing greeting interaction), keeping track of more than just the latest move can be useful.
- **QUD**: questions that constitute a "live issue". That is, questions that have been *introduced for discussion* at a given point in the conversation and whose discussion has not yet been *concluded.* There are additional, indirect ways for questions to get added into QUD, the most prominent of which is during metacommunicative interaction (see section 4.7). Being maximal in QUD (max-qud) corresponds to being the current 'discourse topic', and this is a key component of our account

4.3 Illocutionary interaction

To get started, we abstract away from the communicative process, assuming perfect communication. The basic units of change are mappings between dialogue gameboards that specify how one gameboard configuration can be modified into another on the basis of dialogue moves. We call a mapping between DGB types a *conversational rule.* The types specifying its domain and its range we dub, respectively, the *preconditions* and the *effects*, both of which are supertypes of DGB. Notationally a conversational rule will be specified as in (35):

(35) $$\begin{bmatrix} \text{pre(conds) : RType} \\ \text{effects : RType} \end{bmatrix}$$

4.4 Move Coherence

To illustrate how illocutionary interaction can be specified, we consider the example of greetings and partings. An initiating greeting typically occurs dialogue initially. The primary *contextual* effect of such a greeting is simply to provide the addressee with the possibility of reciprocating with a counter-greeting, though of course it has other *expressive* effects (indication of non-hostility etc). The conversational rule associated with greeting is given in (36a). The preconditions state that both Moves and QUD need to be empty, though obviously this does not apply to FACTS. The sole DGB effect a greeting has—remember we are abstracting away from utterance processing for the moment—is to update MOVES with its content. In the sequel we adopt a more economical notation: the preconds can be written as $DGB \wedge PreCondSpec$, where $PreCondSpec$ is a type that includes information specific to the preconditions of this interaction type. The effects can be written as $DGB \wedge PreCondSpec' \wedge ChangePreconSpec$, where $ChangePreconSpec$ represents those aspects of the preconditions that have changed. We notate conversational rules simply as (36b), and the rule for greeting as (36c):

(36) a.
$$\begin{bmatrix} \text{pre} : & \begin{bmatrix} \text{spkr: Ind} \\ \text{addr: Ind} \\ \text{moves} = \text{elist} : \text{list(IllocProp)} \\ \text{qud} = \text{eset} : \text{poset(Question)} \\ \text{facts} = \text{commonground1} : \text{Prop} \end{bmatrix} \\ \text{effects} : & \begin{bmatrix} \text{spkr} = \text{pre.spkr} : \text{Ind} \\ \text{addr} = \text{pre.addr} : \text{Ind} \\ \text{LatestMove} = \text{Greet(spkr,addr):IllocProp} \\ \text{qud} = \text{pre.qud} : \text{list(Question)} \\ \text{facts} = \text{pre.facts} : \text{Prop} \end{bmatrix} \end{bmatrix}$$

b.
$$\begin{bmatrix} \text{pre: PreCondSpec} \\ \text{effects} : \text{ChangePreconSpec} \end{bmatrix}$$

c.
$$\begin{bmatrix} \text{pre} : & \begin{bmatrix} \text{moves} = \text{elist} : \text{list(IllocProp)} \\ \text{qud} = \text{elist} : \text{list(Question)} \end{bmatrix} \\ \text{effects} : & \begin{bmatrix} \text{LatestMove} = \text{Greet(spkr,addr):IllocProp} \end{bmatrix} \end{bmatrix}$$

A countergreeting involves turn change and grounds the original greeting; we capture this potential by the rule in (37):

(37)
$$\begin{bmatrix} \text{pre} : & \begin{bmatrix} \text{LatestMove} = \text{Greet(spkr,addr):IllocProp} \\ \text{qud} = \text{elist} : \text{list(Question)} \end{bmatrix} \\ \text{effects} : & \begin{bmatrix} \text{spkr} = \text{pre.addr: Ind} \\ \text{addr} = \text{pre.spkr: Ind} \\ \text{LatestMove} = \text{CtrGreet(spkr,addr):IllocProp} \end{bmatrix} \end{bmatrix}$$

Parting can be specified in almost analogous terms, with the difference that only QUD needs to be empty—all raised issues have been resolved for current purposes—and that there exists a presupposition that a certain amount of interaction has taken place; see Ginzburg (2009) for details.

4.5 Querying and Assertion

The basic protocol for 2-person querying and assertion that we assume is in (38):

(38)

querying	assertion
LatestMove = Ask(A,q)	LatestMove = Assert(A,p)

A: push q onto QUD;
release turn;
	A: push p? onto QUD;
	release turn

B: push q onto QUD;
take turn;
make q—specific
utterance
take turn.
	B: push p? onto QUD;
	take turn;
	Option 1: Discuss p?
	Option 2: Accept p

	LatestMove = Accept(B,p)

	B: increment FACTS with p;
	pop p? from QUD;

	A: increment FACTS with p;
	pop p? from QUD;

q-specific utterance: an utterance whose content is either a proposition *p* About MAX-QUD(*partial answer*) or a question q_1 on which MAX-QUDDepends (*sub-question*).[10]

Two aspects of this protocol are not query specific:

(1) The protocol is like the one we have seen for greeting—a 2-person turn exchange protocol (2-PTEP).
(2) The specification `make q-specific utterance` is an instance of a general constraint that characterizes the contextual background of reactive queries and assertions.

This latter specification can be formulated as in (39): the rule states that if q is QUD–maximal, then either participant may make a q–specific move. Whereas the preconditions simply state that *q* is QUD–maximal, the preconditions underspecify who has the turn and require that the latest move—the first element on the MOVES list—stand in the *Qspecific* relation to *q*:

[10] For answerhood and dependence plug your favourite semantics of questions (e.g. Groenendijk & Stokhof (1997); Ginzburg & Sag (2000)).

(39) QSpec

$$\left[\begin{array}{ll}
\text{preconds} & : \left[\text{qud} = \left\langle \text{q, Q} \right\rangle : \text{poset(Question)}\right] \\
\text{effects} & : \left[\begin{array}{l}
\text{spkr : Ind} \\
\text{c1 : spkr} = \text{preconds.spkr} \vee \text{preconds.addr} \\
\text{addr : Ind} \\
\text{c2: member(addr,}\left\{\text{preconds.spkr,preconds.addr}\right\}\text{)} \\
\wedge\ \text{addr} \neq \text{spkr} \\
\text{r : AbSemObj} \\
\text{R: IllocRel} \\
\text{Moves} = \left\langle \text{R(spkr,addr,r)} \right\rangle \bigoplus m : \text{list(IllocProp)} \\
\text{c1 : Qspecific(r,preconds.qud.q)}
\end{array}\right]
\end{array}\right]$$

The only query specific aspect of the query protocol in (38) is the need to increment QUD with q as a consequence of q being posed:

(40) Ask QUD–incrementation:

$$\left[\begin{array}{l}
\text{pre} : \left[\begin{array}{l}
\text{q : Question} \\
\text{LatestMove} = \text{Ask(spkr,addr,q):IllocProp}
\end{array}\right] \\
\text{effects} : \left[\text{qud} = \text{[q,pre.qud] : list(Question)}\right]
\end{array}\right]$$

What are the components of the assertion protocol? Not specific to assertion is the fact that it is a 2-PTEP; similarly, the discussion option is simply an instance of QSpec. This leaves two novel components: QUD incrementation with $p?$, which can be specified like (40) *mutatis mutandis*, and acceptance. Acceptance is a somewhat more involved matter because a lot of the action is not directly perceptible. The labour can be divided here in two: first, we have the action brought about by an acceptance utterance (e.g. ‘mmh’, ‘I see’). The background for an acceptance by B is an assertion by A and the effect is to modify LatestMove:

(41) Accept move:

$$\left[\begin{array}{ll}
\text{pre} = & \left[\begin{array}{l}
\text{p : Prop} \\
\text{LatestMove} = \text{Assert(spkr,addr,p):IllocProp} \\
\text{qud} = [\text{p?},\ldots] : \text{list(Question)}
\end{array}\right] \\
\text{effects} = & \left[\begin{array}{l}
\text{spkr} = \text{pre.addr: Ind} \\
\text{addr} = \text{pre.spkr : Ind} \\
\text{LatestMove} = \text{Accept(pre.addr,spkr,p) : IllocProp}
\end{array}\right]
\end{array}\right]$$

The second component of acceptance is the incrementation of FACTS by p. This is not quite as straightforward as it might seem: when FACTS gets incremented, we also need to ensure that p? gets downdated from QUD—only nonresolved questions can be in QUD (resolved questions have a use as "rhetorical questions", see Ginzburg (2009)). In order to ensure that this is the case, we need to check for each element of QUD that it is not resolved by the new value of FACTS. Hence, accepting p involves both an update of FACTS and a downdate of QUD enforced via the function NonResolve—minimally just removing p?, but possibly removing other questions as well:

(42) Fact Update/ QUD Downdate

$$\begin{bmatrix} \text{preconds} & : \begin{bmatrix} \text{p : Prop} \\ \text{LatestMove} = \text{Accept(spkr,addr,p)} \\ \text{qud} = [\text{p?,preconds.qud}] : \text{poset(Question)} \end{bmatrix} \\ \text{effects} & : \begin{bmatrix} \text{facts} = \text{preconds.facts} \cup \left\{\text{p}\right\} : \text{Set(Prop)} \\ \text{qud} = \text{NonResolve(preconds.qud,facts)} : \text{poset(Question)} \end{bmatrix} \end{bmatrix}$$

With this in hand, we can exemplify the framework sketched so far with the example in (43):[11]

(43) A(1): Hi
B(2): Hi
A(3): Who's coming tomorrow?
B(4): Several colleagues of mine (are coming).
A(5): I see.
B(6): Mike (is coming) too.

[11] Utterance (43(3)) is an *initiating* query. Any theory requires some means, typically one that makes reference to the domain in which the interaction takes place of licensing such queries. Here we appeal to the rule Free Speech. This rule, from Ginzburg (2009), is a domain–independent principle that licenses the choice of *any* query or assertion assuming QUD is empty. We discuss how to refine this with a principle that is domain–specific in section 4.6.

Utt.	DGB Update	Rule
	(Conditions)	
initial	MOVES = $\langle\rangle$	
	QUD = $\langle\rangle$	
	FACTS = cg1	
1	LatestMove := Greet(A,B)	greeting
2	LatestMove := CounterGreet(B,A)	countergreeting
3	LatestMove := Ask(A,B,q0)	Free Speech
	QUD : = $\langle q0 \rangle$	Ask QUD–incrementation
4	LatestMove := Assert(B,A,p1)	QSpec
	(About(p1,q0))	
	QUD : = $\langle p1?, q0 \rangle$	Assert QUD–incrementation
5	LatestMove := Accept(A,B,p1)	Accept
	QUD := $\langle q0 \rangle$	Fact update/QUD downdate
	FACTS := cg1 $\wedge$ p1	
6	LatestMove := Assert(B,A,p2)	QSpec
	(About(p2,q0))	
	QUD : = $\langle p2?, q0 \rangle$	Assert QUD–incrementation

We are also now in a position to explain how many of the earlier benchmarks can be met: accommodating *non-resolving answers, follow up queries to non-resolving answers, sub-questions,* and *disagreement* are all fairly immediate consequences of QSpec: the first three follow given that the QUD-maximality of *q* allows a *q-specific* utterance to be made, disagreement is accommodated since asserting *p* makes *p*? QUD-maximal, and *p*?–specific utterances include disagreements. Two other benchmarks can be met due to the mechanism of fact update above: Assertion benchmark1: if accepted, integrate propositional content with existing knowledge base is a direct consequence. Accommodating "overinformative" answers also follows, to a first approximation, given that semantic information does not get "wasted". Full attention to "over informativity" is a long story involving implicature and private parts of information states (on which more below).

We can also say something about the Scaling Up benchmark. Self answering is directly accommodated by QSpec given that it licenses MAX-QUD–specific utterances regardless of who the speaker of LatestMove is. Another consequence of QSpec is the possibility of posing two successive questions by a single speaker, where the second question influences the first; the second query becomes QUD maximal.

(44) a. Ann: What are your shifts next week? Can you remember offhand?
James: Yes. I'm early Monday and Tuesday (pause) and Wednesday (pause) a day off Thursday (pause) Friday (pause) late (BNC, KC2 4968-4971)

b. Ann: Anyway, talking of over the road, where is she? Is she home?
Betty: No. She's in the Cottage. (BNC, KC2 5121-5124)

QSpec also allows for successive assertions p_1, p_2, where p_2 is About p_1?. When the later assertion p_2 is accepted, the issue associated with the earlier assertion p_1 will be downdated iff FACTS (including p_2) resolves p_1?; this is an implicit mechanism for accepting p_1.

Not all successive queries and successive assertions can be dealt with in this way, and some require postulation of additional conversational rules in order to accommodate further rhetorical relations (for more discussion on this see in particular Asher & Lascarides (2003); Prévot (2003)).

4.6 Domain specificity

- DA: reuse interactional procedures across domains, in so far as possible.

So far we have discussed queries and assertions that arise *reactively*. Conventions regulating the *initiating* of such moves, conversation initially and periodically during extended interactions, are less domain independent, far more dependent on the activity conversationalists are enagaged in, and on politeness, prior acquaintance between conversationalists etc. The basic intuition one can pursue is that a move can be made if it *relates to the current activity*.[12] In some cases the activity is very clearly defined and tightly constrains what can be said. In other cases the activity is far less restrictive on what can be said:

(45) a. **Buying a train ticket**: c wants a train ticket: c needs to indicate where to, when leaving, if return, when returning, which class, s needs to indicate how much needs to be paid
b. **Buying in a boulangerie**: c needs to indicate what baked goods are desired, b needs to indicate how much needs to be paid
c. **Buying goods in a minimarket stationed in a petrol station**: c needs to show what she bought, s needs to check if c bought petrol and to tell c how much needs to be paid.
d. **Chatting among friends**: first: how are conversational participants and their near ones?
e. **Buying in a boulangerie from a long standing acquaintance**: combination of (b) and (d).

Trying to operationalize activity relevance presupposes that we can classify conversations into various *genres*, a term we use following Bakhtin (1986) to denote a particular type of interactional domain. There are at present remarkably few such taxonomies (though see Allwood (1999) for an informal one.) and we will not attempt to offer one here. However we can indicate how to classify a conversation into a genre. One way is by providing a description

[12] The approach sketched here is inspired by work in Larsson (2002), work implemented in the GODIS system.

of an information state of a conversational participant who has *successfully* completed such a conversation. Final states of a conversation will then be records of type T for T a subtype of DGB$_{fin}$, here Questions No (longer) Under Discussion (QNUD) denotes a list of issues characteristic of the genre which will have been resolved in interaction:

(46) $$\text{DGB}_{fin} = \begin{bmatrix} \text{Facts : Prop} \\ \text{QNUD} = \text{list : list(question)} \\ \text{Moves : list(IllocProp)} \end{bmatrix}$$

In (47) we exemplify two genres, informally specified in (45):

(47) a. CasualChat:

$$\begin{bmatrix} \text{A : Ind} \\ \text{B : Ind} \\ \text{t: TimeInterval} \\ \text{c1 : Speak(A,t)} \vee \text{Speak(B,t)} \\ \text{facts : Set(Prop)} \\ \text{qnud : list(question)} \\ \text{c2:} \left\{ \lambda P.P(A), \lambda P.P(B) \right\} \subset \text{qnud} \\ \text{moves : list(IllocProp)} \end{bmatrix}$$

b. BakeryChat:

$$\begin{bmatrix} \text{A : Ind} \\ \text{B : Ind} \\ \text{t: TimeInterval} \\ \text{c1 : Speak(A,t)} \vee \text{Speak(B,t)} \\ \text{facts : Set(Prop)} \\ \text{qnud : list(question)} \\ \text{c2:} \left\{ \begin{array}{l} \lambda P.P(A), \lambda P.P(B),\ \lambda x.\text{InShopBuy(A,x)}, \\ \lambda x.\text{Pay(A,x)} \end{array} \right\} \subset \text{qnud} \\ \text{moves : list(IllocProp)} \end{bmatrix}$$

We can then offer the following definition of *activity relevance*: one can make an initiating move m0 if one believes that that the current conversation updated with m0 is of a certain genre G0. Making move $m0$ given what has happened so far (represented in *dgb*0) can be *anticipated* to conclude as final state *dgb*1 which is a conversation of type G0:

(48) m0 is relevant to G0 in dgb0 for A iff there exists dgb1 such that $dgb0 \sqsubset dgb1$, and such that dgb1 : G0

4.7 Metacommunicative interaction

A theory of MCI needs to meet the high level benchmarks we formulated earlier, specifically those concerning Semantic non-determinism and Fine-grained utterance representation. KoS is already equipped to address the first challenge due to the fact that each conversational participant is associated with a distinct DGB—concrete exemplification of this is offered towards the end of this section. Therefore there is no single *context* in conversation but rather *coupled and potentially mismatched* dialogue gameboards. Only one modification is required to the structure of the DGB, the postulation of a field **Pending**, whose members are ungrounded utterances. For reasons we discuss shortly the type of **Pending** (and concomitantly that of **Moves**) is a list of *locutionary propositions*, propositions consisting of an utterance record and a (grammatical) type which classifies it. This leads to a new definition of DGB type:

(49) DGB =

$$\begin{bmatrix} \text{spkr : Ind} \\ \text{addr : Ind} \\ \text{c-utt : addressing(spkr,addr)} \\ \text{Facts : Set(Prop)} \\ \text{Pending : list(locProp)} \\ \text{Moves : list(locProp)} \\ \text{QUD : poset(Question)} \end{bmatrix}$$

In the immediate aftermath of a speech event u, **Pending** gets updated with a record of the form $\begin{bmatrix} \text{sit} = \text{u} \\ \text{sit-type} = \text{T}_u \end{bmatrix}$ (of type *locutionary proposition* (LocProp)). Here T_u is a grammatical type for classifying u that emerges during the process of parsing u. In the most general case it should be thought of as a *chart* (Cooper, 2009), but in the cases we consider here it can be identified with a *sign* in the sense of Head Driven Phrase Structure Grammar (HPSG). The relationship between u and T_u—describable in terms of the proposition $p_u = \begin{bmatrix} \text{sit} = \text{u} \\ \text{sit-type} = \text{T}_u \end{bmatrix}$— can be utilized in providing an analysis of grounding/CRification conditions:[13]

(50) a. Grounding: p_u is true: the utterance type fully classifies the utterance token.
b. CRification: T_u is weak (e.g. incomplete word recognition); u is incompletely specified (e.g. incomplete contextual resolution).

[13] A particularly detailed theory of grounding has been developed in the PTT framework, e.g. Poesio & Traum (1997); Poesio & Rieser (2009).

Postulating that Pending be of type LocProp allows us to meet the Fine-grained utterance representation benchmark: T_u provides the fine-grain and the information needed to capture syntactic/phonological parallelism; u is necessary to instantiate the contextual parameters of T_u, as well as to provide the sub-utterance tokens that figure in CRs (on the latter see the discussion concerning example (68)).[14] We can also formulate the following utterance processing protocol, which interleaves illocutionary and metacommunicative interaction:

(51) **Utterance processing protocol**
For an agent A with IS I: if a locutionary proposition $p_u = \begin{bmatrix} \text{sit} = \text{u} \\ \text{sit-type} = \text{T}_u \end{bmatrix}$ is Maximal in PENDING:
(a) If p_u is true, try to integrate p_u in A.DGB using a Moves update rule.
(b) Otherwise: try to accommodate p_u as a CR to LatestMove.
(c) If (a) and (b) fail, seek a witness for T_u by asking a CR: introduce a clarification issue derivable from p_u as the maximal element of QUD; use this context to formulate a clarification request.

A full theory of MCI involves a compositional analysis of (a somewhat more sophisticated version of) this protocol using update rules entirely akin to those used for illocutionary interaction in section 4.3. We concentrate here on elucidating how a CR gets asked and which are the available CRs. Given that any sub-utterance of a given utterance is potentially clarifiable, one prerequisite at the level of utterance representation is the accessibility of all sub-utterances. We achieve this by positing that the field C-PARAMS of a given utterance type is a record type specifying two kinds of witnesses: (a) sub-utterance tokens, characterized in terms of their morphosyntactic properties, and (b) referents, specified in terms of their semantic contribution. Repetition and meaning–oriented CRs are specified by means of a particular class of conversational rules—Clarification Context Update Rules (CCURs). Each CCUR specifies an accommodated MaxQUD built up from a sub-utterance u1 of the target utterance *MaxPending*. Common to all CCURs is a license to follow up *MaxPending* with an utterance which is *co-propositional* with MaxQud.[15] In

[14] This argumentation carries over to identifying the type of LatestMove as LocProp—this information is required to enable A to integrate a CR posed by B concerning A's latest utterance. Data pointing towards the preservation of non-semantic structure in the longer term comes from alignment phenomena (Garrod & Pickering, 2004). However, the extent to which this is the case or only content is preserved in context long term is very much an open question.

[15] Two utterances u_0 and u_1 are *co-propositional* iff the questions q_0 and q_1 they contribute to QUD are co-propositional.

(i) qud-contrib(m0.cont) is m0.cont if m0.cont : Question
(ii) qud-contrib(m0.cont) is ?m0.cont if m0.cont : Prop

the current context co-propositionality amounts to: either a CR which differs from MaxQud at most in terms of its domain, or a correction—a proposition that instantiates MaxQud.

To make this concrete, we consider one specific CCUR Parameter identification, used to specify *intended content* CRs. (52) indicates that given u0, a sub–utterance token of MaxPending, one may accommodate as MaxQUD the issue 'What did spkr mean by u0'. Concomitantly, the next move must be co-propositional with this issue:

(52) Parameter identification

$$\left[\begin{array}{ll} \text{preconds :} & \left[\begin{array}{l}\text{Spkr : Ind}\\ \text{MaxPending : LocProp}\\ \text{u0} \in \text{MaxPending.sit.constits}\end{array}\right]\\ \text{effects:} & \left[\begin{array}{l} \text{MaxQUD} = \text{What did spkr mean by u0? : Question}\\ \text{LatestMove : LocProp}\\ \text{c1: CoProp(LatestMove.cont,MaxQUD)}\end{array}\right]\end{array}\right]$$

(52) underpins CRs such as (53b,c) as follow ups to (53a):

(53) a. A: Is Bo here?
b. B: Who do you mean 'Bo'?
c. B: Bo? (= Who is 'Bo'?)

We can also deal with corrections, as in (54). B's corrective utterance is co-propositional with λxMean(A,u0,x), and hence allowed by the specification:

(54) B: You mean Jo.

In Figure 6 we provide an illustration of our account of the semantic non-determinism benchmark: the same input leads to distinct outputs on the "public level" of information states. In this case this arises due to differential ability to anchor the contextual parameters. The utterance u0 has three sub-utterances, u1, u2, u3, given in Figure 6 with their approximate pronunciations. A can ground her own utterance since she knows the values of the contextual parameters, which we assume here for simplicity include the speaker and the referent of the sub-utterance 'Bo'. This means that the locutionary proposition associated with u0—the proposition whose situational value is a record that arises by unioning u0 with the witnesses for the contextua parameters and whose type is given in Figure 6—is true. This enables the "canonical" illocutionary update to be performed: the issue 'whether b left'

q_0 and q_1 are co-propositional if there exists a record r such that $q_0(r) = q_1(r)$. This means that, modulo their domain, the questions involve similar answers. For instance 'Whether Bo left', 'Who left', and 'Which student left' (assuming Bo is a student.) are all co-propositional.

becomes the maximal element of QUD. In contrast, let assume that B lacks a witness for the referent of ‘Bo’. As a result, the locutionary proposition associated with u0 which B can construct is not true. Given this, uses the CCUR parameter identification to build a context appropriate for a clarification request: B increments QUD with the issue λxMean(A,u2,x), and the locutionary proposition associated with u0 which B has constructed remains in Pending.

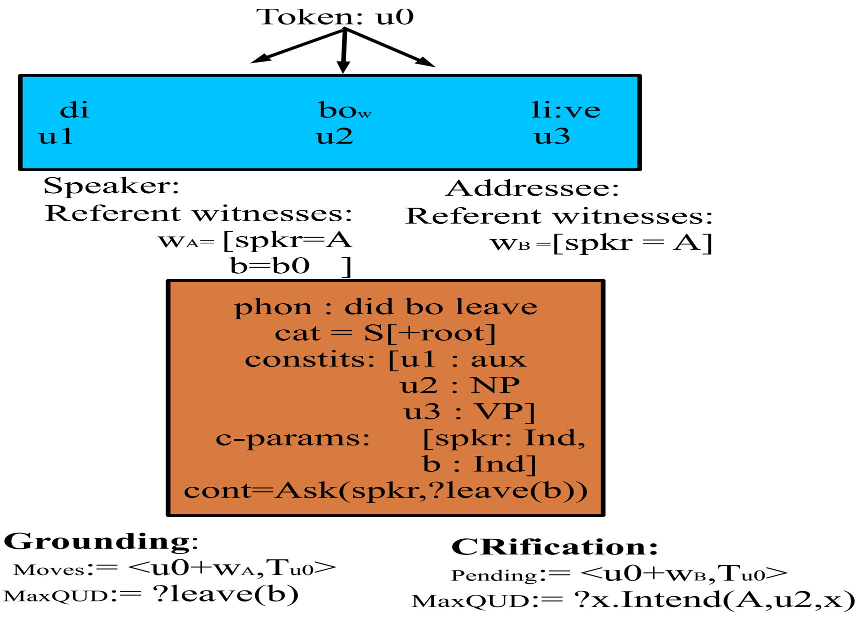


Figure 6. A single utterance gives rise to distinct updates of the DGB for distinct participants

To conclude our discussion of the basics of MCI, we consider briefly relevance CRs and topic changing, “irrelevant responses” (the latter our benchmark Q5). The basic trigger for both is the condition in (55), where the content of an utterance stands in the Irrelevant relation to a dgb:

(55) Irrelevant(u.cont,dgb)

Irrelevant(p,dgb0) here relates an illocutionary proposition *p*, the content of the "irrelevant" move, to a DGB dgb0 just in case there is no update rule U such that U(dgb0).LatestMove.cont = p. For instance, given what we have said here, an irrelevant follow up to an utterance *u* which expresses a query *q* is an utterance which is neither *q*–specific nor a clarification request triggered by *u*:

(56) a. LatestMove = u; u.content = Ask(A,q),
b. *p* is not q–specific
c. *p* is not CoPropositional with any question q0 that satisfies q0 = CCUR1.qud(u) for some CCUR CCUR1

The potential for CRs concerning the relevance of an utterance is already, with one potentially significant caveat, accommodated by the rule parameter identification we saw above. The one significant difference of relevance CRs is that the trigger is typically the irrelevance of a *fully instantiated utterance.* The answer to such a CR will not in general be represented in the DGB, in contrast to other CRs, where it could be found in C-PARAMS or PHON of the responder.

This means that we need to offer an alternative definition for the Mean predicate to the one appropriate for semantically oriented CRs. What we would need would be a definition along the following lines—identifying the speaker meaning with the maximal element of the agenda of the utterance's speaker:

(57) Given u.sit.cont : IllocProp, Mean(A,u,c) iff u.c-param.spkr = A and A.private.maxagenda = c

As for irrelevance implicatures, we can offer a "short circuited" version of the Gricean account—irrelevance is a means of non-grounding the previous utterance, itself an instance of a more general process of ignoring commonly perceived events. The short circuited version takes the form of the update rule in (58)—given that MaxPending is *irrelevant* to the DGB, one can make MaxPending into LatestMove while updating Facts with the fact that the speaker of MaxPending does not wish to discuss MAX-QUD:

(58)
$$\begin{bmatrix} \text{preconds:} & \begin{bmatrix} \text{dgb : DGB} \\ \text{c: IrRelevant(maxpending}^{content}\text{,dgb)} \end{bmatrix} \\ \text{effects :} & \begin{bmatrix} \text{LatestMove = pre.pending : LocProp} \\ \text{Facts = pre.Facts} \cup \\ \left\{ \neg \text{ WishDiscuss(pre.spkr,pre.maxqud)} \right\}. \end{bmatrix} \end{bmatrix}$$

Note that this does not make the *unwillingness to discuss* be the *content* of the offending utterance; it is merely an inference. Still this inference will allow MAX-QUDto be downdated, via a slightly refined version of fact update/question downdate—if information is accepted indicating negative resolution of ?WishDiscuss(q), then *q* may be downdated from QUD.

4.8 Disfluencies

The set up for metacommunicative *inter*action described in the previous section extends straightforwardly to yield an account of *self-correction*, and other disfluencies. The sole, but significantly consequential, modification such an account presupposes is to the structure of PENDING. This now needs to incorporate also utterances that are *in progress*, and hence, incompletely specified semantically and phonologically. This, in turn, requires the use of types that characterize utterances word by word (or minimally constituent by constituent), as e.g. in Combinatory Categorial Grammar (Steedman, 1999), Type Logical Grammar (Morrill, 2000), Dynamic Syntax (Kempson *et al.*, 2000), PTT (Poesio & Traum, 1997) or by abstraction from a "standard" grammar (as one could implement in HPSG$_{TTR}$, that version of HPSG whose logical underpinning is TTR.). A variety of issues arise, in consequence, issues that are still very much open, including monotonicity in processing, and the nature of incremental denotations. Fortunately the account of disfluencies can be formulated without making commitments on these issues.

Incrementalizing PENDING has the independent consequence of enabling us to account for the Incremental Acknowledgements benchmark, (inspired by examples such as 13c) (Ack2). We can formulate a lexical entry for *'mmh'*, which enables a speaker to acknowledge the current addressee's most recently ungrounded utterance, regardless of whether it is complete (in which case its content would be an IllocProp) or not:

(59)
$$\begin{bmatrix} \textsc{phon} : \langle \text{ mmh } \rangle \\ \textsc{cat} = \textit{interjection} : \text{syncat} \\ \text{c-params} : \begin{bmatrix} \text{spkr} : \text{IND} \\ \text{addr} : \text{IND} \\ \text{MaxPending} : \text{LocProp} \\ \text{c2} : \text{address(addr,spkr,MaxPending)} \end{bmatrix} \\ \textsc{cont} = \text{Understand(spkr,addr,MaxPending)} : \text{IllocProp} \end{bmatrix}$$

The basic intuition behind this account of disfluencies is an analogy to CRification: in the latter a CR provides the potential for an answer, which allows the original poser of the CR to fix his utterance. For self corrections *editing phrases* (EditPs) (long silences, discourse particles like 'No ...', 'um' etc) correspond to CRs, whereas the *alternation*, that sub-utterance with the correcting material corresponds to an answer to a CR. There are two remaining steps: first provide for the coherence of the EditP. This is simple to do: all we need to say is that an EditP can be interpolated at any point where PENDING is non-empty. Finally, take as input a state where the LatestMove is an EditP and specify as output a new state in which the MaxQUD is *What did spkr mean to utter at u0?* and where the new utterance has to be an instantiation of MaxQud (propositional or polar question):[16]

(60) Utterance identification:

$$\text{Input: } \begin{bmatrix} \text{Spkr : Ind} \\ \text{MaxPending : LocProp} \\ \text{LatestMove} = \text{EditP(Spkr,MaxPending) : IllocProp} \\ \text{u0} \in \text{MaxPending.sit.constits} \end{bmatrix}$$

$$\text{Output: } \begin{bmatrix} \text{MaxQUD} = \text{What did spkr mean to say at u0? : Question} \\ \text{LatestMove : LocProp} \\ \text{c2: InstPropQ(LatestMove.cont,MaxQUD)} \end{bmatrix}$$

The same mechanism that updates the DGB after a CR and effects an update of information concerning a given utterance applies here. It ensures that the alteration of the original sub-utterance replaces or reinforces the repaired sub-utterance in PENDING. At the same time, the presupposition concerning the latter's taking place will remain in FACTS. We thereby meet

(61) D2: Explicate disfluency meaning without eliminating disfluencies from context.

4.9 Sentential Fragments

The approach we pursue here to sentential fragments is constructional, i.e. from a grammatical point of view we treat such constructions as *sui generis*, not as underlyingly canonical sentences, as is common in generative linguistics. The fundamental argument for this strategy is the existence of a wide array of mismatches between the syntactic and semantic properties of sentential fragments and putative sentential correlates (for extensive argumentation, see (Ginzburg & Sag, 2000; Schlangen, 2003; Fernández, 2006; Ginzburg, 2009)). (62) exemplifies this claim—(62a) shows the distinct distribution of a direct sluice and of its putative canonical correlate; (62b) shows a similar datum for a short answer and its putative canonical correlate; finally (62c) illustrates that elliptical exclamatives cannot be embedded, in contrast to sentential exclamatives:

[16] Some evidence towards the reality of the MAX-QUDpostulated in this CCUR is provided by examples such as the following attested example:

(i) Hmm. Lots of people are texting in and getting involved on 606, and, er, what's the word? Backtracking, that's it. (From a BBC webcast of a football match, Nov 12, 2008.)

(62) a. A: Somebody stood outside the room. B: Who? / #Who the hell? / Who the hell stood outside the room?

b. Who stood outside the room? Not Bo. / #Not Bo stood outside the room.

c. A: What a shot! / *It's amazing what a shot. /It's amazing what a shot she made.

The existence of parallelism between source and NSU on various dimensions necessitates positing one additional contextual parameter, namely an antecedent sub-utterance (of the utterance which is MAX-QUD). Intuitively, this parameter provides a partial specification of the focal (sub)utterance, and hence it is dubbed the *focus establishing constituent* (FEC). Varying roles are played by the FEC: in some cases it is crucial for semantic composition, while in others it plays a disambiguating role via morphosyntactic or phonological parallelism.

Given that their lifetimes are as a rule identical, we can pair QUDs and FECs as part of contextual specification. Concretely this amounts to changing the type of QUD from *list(Questn)* to *list(Info-struc)*, where Info-Struc is the following type:

(63) $$\text{Info-struc} = \begin{bmatrix} \text{q : Questn} \\ \text{fec : set(LocProp)} \end{bmatrix}$$

It also means that FECs get introduced by (minor modifications of) rules we have seen above for incrementing and downdating QUD, namely Ask-QUD incrementation and the CCURs.

With this in hand, we turn to illustrating KOS' approach to sentential fragment grammar and meaning.[17] Sentential fragments are essentially akin to indexicals ('I' : speaker, 'you': addr,'here': speech loc., ...), but whereas the latter resolve to concrete elements of the utterance context, sentential fragment resolution is based on reference to DGB elements:[18]

Yes

Its informal meaning is simply— MAX-QUD's proposition. (64) includes a rudimentary lexical entry for this word which formalizes this intuition:

(64) $$\begin{bmatrix} \text{phon : yes} \\ \text{cat = adv : syncat} \\ \text{max-qud : PolarQuestn} \\ \text{cont = max-qud([]): Prop} \end{bmatrix}$$

[17] See Schlangen (2003) for an alternative approach to NSUs within SDRT.

[18] We have space here only to discuss a small number of cases. In particular, direct sluicing, the most complex non-MCI sentential fragment, would require discussion of our treatment of quantification. For detailed treatments see Fernández (2006); Ginzburg (2009).

Short answers

This construction can be described in the following terms: the content arises by function application of MAX-QUDto the fragment's content; syntactically the fragment must bear an identical syntactic category to the FEC. (65) represents this construction in HPSG$_{TTR}$:

(65) $$\textit{decl-frag-cl} = \begin{bmatrix} \text{cat} = \text{V[+fin]} : \text{syncat} \\ \text{hd-dtr} : \begin{bmatrix} \text{cat} = \text{max-qud.fec.cat} : \text{Syncat} \end{bmatrix} \\ \qquad \wedge \text{ sign} \\ \text{max-qud} : \text{WhQuestn} \\ \text{cont} = \text{max-qud(hd-dtr.cont)} : \text{Prop} \end{bmatrix}$$

Given that the meaning of short answer is directly tied to MAX-QUD, we can fulfill the distance benchmark: accommodate long distance short answers: such answers are predicted to be possible in so far as the corresponding issue is still in QUD. Since QUD consists of elements of type *info-struc*, we can also capture the long distance syntactic parallelism short answers exhibit.

We turn finally to two sentential fragments used in MCI , the *confirmation* and *intended content* readings of Reprise Fragments (RF):

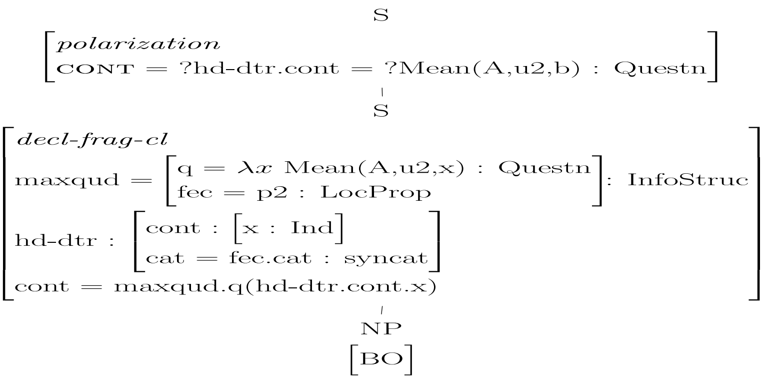
Reprise Fragments: confirmation reading

Assume the utterance to be clarified is (66a). B uses the CCUR parameter identification to build a context as in (66b):

(66) a. A: Did Bo leave?
b. MAX-QUD = λxMean(A,u2,x) ;FEC = A's utterance 'Bo'

Given this, the analysis of the construction is illustrated in (67): the construction *decl-frag-cl* builds the proposition Mean(A,u2,b); the construction *polarization* builds a polar question from this:

(67)


S
polarization
CONT = ?hd-dtr.cont = ?Mean(A,u2,b) : Questn
S
decl-frag-cl
maxqud = [q = λx Mean(A,u2,x) : Questn, fec = p2 : LocProp] : InfoStruc
hd-dtr : [cont : [x : Ind], cat = fec.cat : syncat]
cont = maxqud.q(hd-dtr.cont.x)
NP
BO

Reprise Fragments: intended content reading

Intended content readings of RFs involve a complex mix of a *prima facie* non-transparent semantics and phonological parallelism. Independently of intended content readings, we need to capture the utterance anaphoricity of "quotative" utterances such as (68):

(68) a. A: Bo is coming. B: Who do you mean 'Bo'?
b. D: I have a Geordie accident. J: 'accident' that's funny.

We assume the existence of a grammatical constraint allowing reference to a sub-utterance under phonological parallelism. (69) exemplifies one way of formulating such a constraint: the PHON value is type identical with the PHON value of an utterance identified with the focus establishing constituent, whereas the content is stipulated to be the utterance event associated with the focus establishing constituent:[19]

[19] (69) makes one simplifying assumption: identifying the PHON value of the focus establishing constituent with that of the utterance anaphoric phrase. In practice this should only be the segmental phonological value.

(69) *utt-anaph-ph*

$$\left[\begin{array}{l} \text{tune} = \text{max-qud.fec.sit-type.phon} : \text{Type} \\ \text{phon} : \textit{tune} \\ \text{cat} : \text{syncat} \\ \text{max-qud} : \text{info-struc} \\ \text{cont} = \text{max-qud.fec.sit} : \text{Rec} \end{array}\right]$$

With this in hand, we turn back to consider the issue of how *intended content* RFs arise grammatically. It is worth emphasizing that there is no way to bring about the desired content using *decl-frag-cl*, the short-answer/reprise sluice phrasal type we have been appealing to above, regardless of whether we analyze the NP fragment as denoting its standard conventional content or alternatively as denoting an anaphoric element to the phonologically identical to–be–clarified sub-utterance. This is a prototypical instance of appeal to constructional meaning—a complex content that cannot be plausibly constructed using "standard combinatorial operations" (function application, unification etc) from its constituents. Thus, one way of accommodating *intended content* RF is to posit a new phrasal type, *qud-anaph-int-cl*. This will encapsulate the two idiosyncratic facets of such utterances, namely the MAX-QUD/CONTENT identity and the HD-DTR being an *utt-anaph-ph*:

(70) *qud-anaph-int-cl* =

$$\left[\begin{array}{l} \textsc{max-qud} : \text{InfoStruc} \\ \text{cont=max-qud.q:Questn} \\ \text{hd-dtr: } \textit{utt-anaph-ph} \end{array}\right]$$

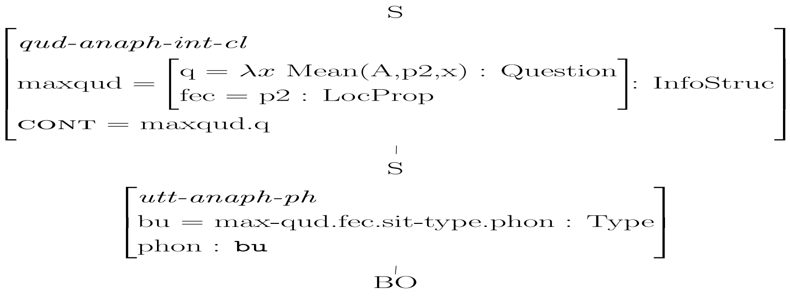
Given this, we can offer the following analysis of (71):

(71) a. A: Is Georges here? B: Georges?
b. B lacks referent for 'Georges'; uses parameter identification to update MAX-QUDaccordingly:

$$\left[\begin{array}{l} \text{spkr} = \text{B} \\ \text{addr} = \text{A} \\ \text{pending} = \left\langle \left[\begin{array}{l} \text{sit} = \text{w0'} \\ \text{sit-type} = \text{IGH} \end{array}\right] \right\rangle \\ \text{maxqud} = \left[\begin{array}{l} \text{q} = \lambda x \ \text{Mean(A,p2,x)} : \text{Question} \\ \text{fec} = \text{p2} : \text{LocProp} \end{array}\right] : \text{InfoStruc} \end{array}\right]$$

Using *qud-anaph-int-cl* yields:

(72)


S
qud-anaph-int-cl
maxqud = [q = λx Mean(A,p2,x) : Question, fec = p2 : LocProp] : InfoStruc
CONT = maxqud.q
S
utt-anaph-ph
bu = max-qud.fec.sit-type.phon : Type
phon : bu
BO

5 Extensions

In the chapter we have surveyed some core phenomena that theories of dialogue need to tackle. We also sketched a unified treatment of these phenomena. For reasons of space we could not enter into discussion of various other highly significant aspects of dialogue. Here we point to some recent work that has tackled these aspects.

5.1 Automatic learning of dialogue management

Recent advances have been made in the application of machine learning (ML) techniques to dialogue management. One of the most common methods used in this line of research is Reinforcement Learning (Sutton & Barto, 1998). In this approach, the conversational skills of a spoken dialogue system are modelled as a Markov Decision Process (MDP) (Levin & Pieraccini, 1997; Levin *et al.*, 1998). The model consists of a finite set of states S, a finite set of actions A, a transition function $T(s', a, s)$ that specifies the probability of transitioning to state s' from state s after performing action a, and a reward function $R(s', a, s)$ that assigns a reward value to each transition. Given this model, the dialogue manager can be seen as a learning agent that learns an *optimal policy* $\pi : S \mapsto A$, that is, a mapping from states to actions that maximizes the overall reward (which is a function, usually a weighted sum, of all reward values obtained).

The use of ML techniques is attractive because it offers the possibility to develop data-driven approaches to dialogue management that bypass the need to handcraft the rules governing the behaviour of a system. Instead of following handcrafted dialogue strategies (in the form of update or inference rules, or as states and transitions in a manually designed finite-state graph), in a Reinforcement Learning (RL) framework the system learns interactively from the rewards it receives. However, appealing as this may be, there are several drawbacks associated with this approach (see e.g. Paek & Chickering (2005) and Paek & Pieraccini (2008)). One of them is that, like most ML methods, dialogue managers based on reinforcement techniques require large amounts of data for training. Collecting and annotating the dialogue corpora required to train the algorithms requires high amounts of time and effort. The dialogue corpora used by the algorithms need to be annotated with detailed information—a process that requires large amounts of time and effort. A related issue, crucial in RL approaches, concerns the modelling of the state space S. Again like all ML approaches, RL faces the problem of selecting the appropriate features for training, i.e. deciding what state variables should be included in the model. This task is for the most part performed manually. Once an initial set of variables has been chosen, the set can be refined with automatic feature selection methods, but the initial candidate variables are selected by hand. Finally, another important parameter that needs to be set and adjusted is the reward function, which directly affects the adopted policy and hence the behaviour of the system. Although there is some research that explores methods to try to infer R from data (e.g. Ng & Russell (2000); Walker & Shannon (2000)), the typical practice is to specify R manually, sometimes taking into account parameters linked to the task at hand or to user satisfaction (Singh *et al.*, 1999, 2002).

In principle, dialogue management policies learned with RL methods can make use of complex sets of variables encoding rich information (such as the dialogue history, filled and confirmed slots, or information about the interlocutor). However, this can easily lead to an explosion of the state space that may be intractable for learning (Sutton & Barto, 1998). Thus, in practice, researchers developing dialogue systems have concentrated on learning limited policies, such as for example confirmation strategies (Singh *et al.*, 2002). Recent work attempts to address the problem of large state spaces to provide more general policies (see e.g. Rieser & Lemon (2008); Henderson *et al.* (2008)).

Models can also take into account uncertain information such as the user's intentions and beliefs. This information is not directly observable by the system but in principle can be inferred from observable variables such as the user's utterance. This can be modelled as a Partially Observable MDP (POMDP) (Zhang *et al.*, 2001; Young, 2006; Williams & Young, 2007). In a POMDP the uncertainty about the current state is represented as a probability distribution over S or a belief state. The reward function thus computes the expected reward over belief states, while a dialogue policy becomes a mapping from n-dimensional belief states to actions (see Kaelbling *et al.* (1996, 1995) for further details).

5.2 Multi-party dialogue

Our discussion has focussed almost exclusively on two person conversations, as has the lion's share of dialogue systems developed so far. However, the general case is *multi-party dialogue* (also known as *multilogue*). A number of multi-party dialogue systems have been developed at the Institute for Creative Technology, including the Mission Rehearsal Exercise project (Swartout *et al.*, 2006), a virtual reality-based training system. Traum (2004) considers some of the basic issues relating multi-party and two person dialogue; based on NSU data, Ginzburg & Fernández (2005) propose some benchmarks that 2 person dialogue theories aspiring to scale up to multi-party need to fulfill and offer general scaling-up transformations applicable to 2-person protocols. Kronlid (2008) refines these transformations, while offering a detailed implementation of a turn-taking algorithm.

5.3 Multi-modal dialogue

Although spoken language is the basis for communication, other modalities such as gesture often play central roles in dialogue. There is an increasing amount of research dedicated to multi-modal communication and to the implementation of systems that can handle some form of multimodal interaction. The simplest multi-modal systems combine speech with other multimodal input and output such as the display of graphics or the recognition of pointing gestures such as mouse clicks. As discussed in the seminal paper by Nigay & Coutaz (1993), the key questions faced by these systems are how information coming from different modalities can be integrated into a single message (e.g. to disambiguate a referring expression by means of a gesture) and how different modaliities can be fused in generating multimodal output. Delgado & Araki (2005) offer a good survey of multimodal interfaces.

A parallel line of research focusses on developing animated characters or embodied conversational agents (Cassell *et al.*, 2000). These are virtual characters that aim at communicating with humans using speech as well as natural facial expressions, hand gestures and body posture.

6 Conclusions

Dialogue is one of the central disciplines of language sciences—languages are first encountered, learned and used in interaction and this has been the case for millenia. And yet, the lion's share of both formal grammar and psycholinguistic work does not presuppose an interactive setting. Dialogue is a flourishing area in NLP and CL, though primarily in the context of developing dialogue systems.

In this paper we have sought to develop an approach to dialogue that combines theoretical and systems perspectives. To do so, we grounded our discussion empirically in two dozen benchmarks, benchmarks concerning the treatment of querying and assertion, domain adaptability and scalability, metacommunication, and the treatment of fragments. We have used these benchmarks to informally evaluate several influential current approaches to the development of dialogue managers for dialogue systems. We then sketched the theory KoS, formulated in the framework of Type Theory with Records, which, with one or two exceptions, fulfills all the benchmarks. KoS involves formulating a rich theory of information states and showing how these get modified in interaction. One of the important features of this theory is that it allows for an interleaving of locutionary (e.g. grounding, clarification, and self-correction) and illocutionary (e.g. querying and assertion) interaction.

KoS provides an existence proof of a theory of dialogue that can satisfy various benchmarks concerning dialogue coherence, while underpinning fairly sophisticated linguistic analysis. As we note in the text, this combination also characterizes a number of other recently developed dialogue frameworks such as PTT and SDRT. It is important to emphasize, nonetheless, that formal/computational work in dialogue is still at a fairly *early* stage. As we noted in section 5, a comprehensive theory of dialogue needs to accommodate the multimodal nature of interaction and the fact that two person dialogue is a particular instance of multi-party dialogue, with the attendant complexity of turn allocation and split attention.

We believe, furthermore, that one of the important areas of development for work in dialogue is embracing both ontogenetic and phylogenetic perspectives. A phylogenetic or evolutionary perspective on language is gaining significant interest among language scientists and is, moreover, rooted in interaction among a community of agents. Nonetheless, such work has, to date, not made much contact with computational work on dialogue. But this is clearly only a matter of time. As discussed in section 5, there is already a flourishing body of work on learning in dialogue, using various machine learning techniques. Such work is significant for practical reasons, not least because it has the promise of allowing domain specificity to be incorporated in a systematic and large scale way. It is significant also because it should provide us with a theory of language learning that captures the fact that interaction between child and caregiver is a vital component in the emergence of linguistic competence. Indeed, once one takes interaction seriously, as pointed out in the article on Unsupervised Learning and Grammar Induction in this volume, could plausibly simplify the task of language learning significantly. An important challenge for future work is fusing machine language techniques with symbolic ones to achieve the robustness of the former with the linguistic sophistication of the latter.

A dialogical perspective is also, as yet, generally lacking from work on complexity and formal language theory (though see Fernández & Endriss (2007) for an example of how the latter can inform work on dialogue.). But for all the reasons we have discussed above, there is nothing intrinsic in these lacunae, and one can confidently expect these to be filled in the coming decade.

References

Allen, James (1995), *Natural Language Understanding*, Benjamin/Cummings, Redwood City.

Allen, James & Ray Perrault (1980), Analyzing intention in utterances, *Artificial Intelligence* 15:143–178.

Allen, James F., Lenhart K. Schubert, George Ferguson, Peter Heeman, Chung Hee Hwang, Tsuneaki Kato, Marc Light, Nathaniel G. Martin, Bradford W. Miller, Massimo Poesio, & David R. Traum (1995), The trains project: A case study in building a conversational planning agent, *Journal of Experimental and Theoretical AI* 7:7–48.

Allwood, Jens (1995), An activity based approach of pragmatics, *Gothenburg Papers in Theoretical Linguistics, 76* Reprinted in Bunt et al (2000) 'Abduction, Belief and Context in Dialogue; Studies in Computational Pragmatics'. Amsterdam, John Benjamins.

Allwood, Jens (1999), The swedish spoken language corpus at göteborg university, in *Proceedings of Fonetik 99*, volume 81 of *Gothenburg Papers in Theoretical Linguistics*.

Asher, Nicholas & Alex Lascarides (2003), *Logics of Conversation*, Cambridge University Press, Cambridge.

Asher, Nicholas & Alex Lascarides (2008), Commitments, beliefs and intentions in dialogue, in Jonathan Ginzburg, Yo Sato, & Pat Healey (eds.), *Proceedings of LonDial, the 12th Workshop on the Formal Semantics and Pragmatics of Dialogue*, Queen Mary, University of London, London.

Aust, H., M. Oerder, F. Seide, & V. Steinbiss (1995a), The Philips automatic train timetable information system, *Speech Communication* 17(3-4):249–262.

Aust, Harald, Martin Oerder, Frank Seide, & Volker Steinbiss (1995b), The Philips automatic train timetable information system, *Speech Communication* 17:249–262.

Austin, John L. (1962), *How to do things with Words*, Harvard University Press.

Bakhtin, M.M. (1986), *Speech Genres and Other Late Essays*, University of Texas Press.

Besser, Jana & Jan Alexandersson (2007), A comprehensive disfluency model for multi-party interaction, in *Proceedings of SigDial 8*, (182–189).

Bohlin, P., J. Bos, S. Larsson, I. Lewin, C. Matheson, & D. Milward (1999), Survey of existing interactive systems, *Deliverable D* 3.

Bos, Johan, Ewan Klein, Oliver Lemon, & Tetsushi Oka (2003), DIPPER: Description and formalisation of an information-state update dialogue system architecture, in *PRoceeding sof the 4th SIGdial workshop on Discourse and Dialogue*, (115–124).

Brennan, Susan E. & Michael F. Schober (2001), How listeners compensate for disfluencies in spontaneous speech, *Journal of Memory and Language* 44:274–296.

Burnard, L. (2000), *Reference Guide for the British National Corpus (World Edition)*, Oxford Universtity Computing Services.

Carberry, Sandra (1990), *Plan Recognition in Natural Language Dialogue*, Bradford Books, MIT Press, Cambridge.

Carletta, Jean (1996), Assessing agreement on classification tasks: the kappa statistics, *Computational Linguistics* 2(22):249–255.

Carletta, Jean, Amy Isard, Stephen Isard, Jacqueline Kowtko, Gwyneth Doherty-Sneddon, & Anne Anderson (1996), Map Task coder's manual, *HCRC Research Paper* RP-82.

Carpenter, B. (1992), *The logic of typed feature structures: with applications to unification grammars, logic programs, and constraint resolution*, Cambridge University Press.

Cassell, Justine, Joseph Sullivan, Scott Prevost, & Elisabeth Churchill (eds.) (2000), *Embodied Conversational Agents*, MIT Press, Cambridge, MA.

Clark, Herb & Edward Schaefer (1989), Contributing to discourse, in *Arenas of Language Use*, CSLI Publications, Stanford, (259–94), reprinted from a paper in Cognitive Science.

Clark, Herbert (1996), *Using Language*, Cambridge University Press, Cambridge.

Cohen, Philip & Ray Perrault (1979), Elements of a plan-based theory of speech acts, *Cognitive Science* 3:177–212.

Consortium, The TRINDI (2000), *The TRINDI Book*, University of Gothenburg, Gothenburg, available from http://www.ling.gu.se/research/projects/trindi.

Constantinides, Paul, Scott Hansma, Chris Tchouand, & Alexander Rudnicky (1998), A schema based approach to dialog control, in *Proceedings of the 5th International Conference on Spoken Language Processing*, Sydney, Australia.

Cooper, Robin (2006), Austinian truth in martin-löf type theory, *Research on Language and Computation* :333–362.

Cooper, Robin (2009), Dialogue and type theory with records.

Cooper, Robin, Staffan Larsson, James Hieronymus, Stina Ericsson, Elisabet Engdahl, & Peter Ljunglof (2000), *GODIS and Questions Under Discussion*, University of Gothenburg, Gothenburg, available from http://www.ling.gu.se/research/projects/trindi.

Core, Mark & James Allen (1997), Coding dialogs with the DAMSL scheme, *Working notes of the AAAI Fall Symposium on Communicative Action in Humans and Machines* .

Delgado, Ramon & Masahiro Araki (2005), *Spoken, Multilingual And Multimodal Dialogue Systems*, John Wiley And Sons Ltd, UK.

Ferguson, George & James Allen (1998), Trips: An integrated intelligent problem-solving assistant, in *Proceedings of the Fifteenth National Conference on AI (AAAI-98)*, (26–30).

Fernández, Raquel (2006), *Non-Sentential Utterances in Dialogue: Classification, Resolution and Use*, Ph.D. thesis, King's College, London.

Fernández, Raquel & Ulle Endriss (2007), Abstract models for dialogue protocols, *Journal of Logic, Language and Information* 16(2):121–140.

Fernández, Raquel & Jonathan Ginzburg (2002), Non-sentential utterances: A corpus study, *Traitement automatique des languages. Dialogue* 43(2):13–42.

Fernández, Raquel, Jonathan Ginzburg, & Shalom Lappin (2007), Classifying ellipsis in dialogue: A machine learning approach, *Computational Linguistics* 33(3):397–427.

Garrod, Simon & Martin Pickering (2004), Toward a mechanistic psychology of dialogue, *Behavioural and Brain Sciences* 27:169–190.

Ginzburg, Jonathan (1994), An update semantics for dialogue, in H. Bunt (ed.), *Proceedings of the 1st International Workshop on Computational Semantics*, ITK, Tilburg University, Tilburg.

Ginzburg, Jonathan (1995), Resolving questions, i, *Linguistics and Philosophy* 18:459–527.

Ginzburg, Jonathan (1996), Interrogatives: Questions, facts, and dialogue, in Shalom Lappin (ed.), *Handbook of Contemporary Semantic Theory*, Blackwell, Oxford, (359–423).

Ginzburg, Jonathan (2005), Abstraction and ontology: questions as propositional abstracts in constructive type theory, *Journal of Logic and Computation* :113–130.

Ginzburg, Jonathan (2009), *The Interactive Stance: Meaning for Conversation*, CSLI Studies in Computational Linguistics, CSLI Publications, Stanford: California, draft available from http://www.dcs.kcl.ac.uk/staff/ginzburg/tis.pdf.

Ginzburg, Jonathan & Robin Cooper (2004), Clarification, ellipsis, and the nature of contextual updates, *Linguistics and Philosophy* 27(3):297–366.

Ginzburg, Jonathan & Raquel Fernández (2005), Scaling up to multilogue: some benchmarks and principles, in *Proceedings of the 43rd Meeting of the Association for Computational Linguistics*, Michigan, (231–238).

Ginzburg, Jonathan & Ivan A. Sag (2000), *Interrogative Investigations: the form, meaning and use of English Interrogatives*, number 123 in CSLI Lecture Notes, CSLI Publications, Stanford: California.

Groenendijk, Jeroen & Martin Stokhof (1997), Questions, in Johan van Benthem & Alice ter Meulen (eds.), *Handbook of Logic and Linguistics*, North Holland, Amsterdam.

Heeman, Peter A. & James F. Allen (1999), Speech repairs, intonational phrases and discourse markers: Modeling speakers' utternaces in spoken dialogue, *Computational Linguistics* 25(4):527–571.

Henderson, James, Oliver Lemon, & Kallirroi Georgila (2008), Hybrid reinforcement / supervised learning of dialogue policies from fixed datasets, *Computational Linguistics* 34(4):487–511.

Jurafsky, D. & J.H. Martin (2008), *Speech And Language Processing*, Prentice Hall.

Kaelbling, Leslie Pack, Michael L. Littman, & Anthony R. Cassandra (1995), Planning and acting in partially observable stochastic domains, *Artificial Intelligence* 101:99–134.

Kaelbling, Leslie Pack, Michael L. Littman, & Andrew W. Moore (1996), Reinforcement learning: A survey, *Journal of Artificial Intelligence Research* 4:237–285.

Kempson, Ruth, Wilfried Meyer-Viol, & Dov Gabbay (2000), *Dynamic Syntax: The Flow of Language Understanding*, Blackwell, Oxford.

Kronlid, Fredrik (2008), *Steps towards Multi-Party Dialogue Management*, Ph.D. thesis, Gothenburg University.

Larsson, Staffan (2002), *Issue based Dialogue Management*, Ph.D. thesis, Gothenburg University.

Larsson, Staffan & David Traum (2000), Information state and dialogue management in the trindi dialogue move engine toolkit, *Natural Language Engineering* 6:323–340.

Lau, Ellen F. & Fernanda Ferreira (2005), Lingering effects of disfluent material on comprehension of garden path sentences, *Language and Cognitive Processes* 20(5):633–666.

Levelt, Willem J. (1983), Monitoring and self-repair in speech, *Cognition* 14(4):41–104.

Levin, E. & R. Pieraccini (1997), A stochastic model of computer-human interaction for learning dialogue strategies, in *Proceedings of Eurospeech*.

Levin, E., R. Pieraccini, & W. Eckert (1998), Using Markov decision processes for learning dialogue strategies, in *Proceedings og IEEE Transactions on Speech and Audio Processing*, volume 8, (11–23).

Litman, Diane & James Allen (1984), A plan recognition model for clarification subdialogues, in *Proceedings of the 10th Anual Meeting of the Association for Computational Linguistics*, Stanford, CA, (302–311).

Matheson, Colin, Massimo Poesio, & David Traum (2000), Modeling grounding and discourse obligations using update rules, in *Proceedings of the 1st Annual Meeting of the North American Chapter of the ACL*, Seattle.

McTear, Michael (1998), Modelling spoken dialogues with state transition diagrams: Experiences with the CSLU toolkit, in *Proceedings of the 5th International Conference on Spoken Language Processing*, (1223–1226).

McTear, Michael (2004), *Spoken dialogue technology: toward the conversational user interface*, Springer Verlag, London.

Morrill, Glyn (2000), Incremental processing and acceptability, *Computational Linguistics* 26(3):319–338.

Muller, P. & L. Prevot (2003), An empirical study of acknowledgement structures, *Proceedings of DiaBruck* 3.

Nakano, Yukiko, Gabe Reinstein, Tom Stocky, & Justine Cassell (2003), Towards a model of face-to-face grounding, in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, (553–561).

Ng, A.Y. & S. Russell (2000), Algorithms for inverse reinforcement learning, in *Proceedings of the 17th International Conference on Machine Learning*, (663–670).

Nigay, Laurence & Joëlle Coutaz (1993), A design space for multimodal systems: concurrent processing and data fusion, in *CHI '93: Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, ACM, New York, NY, USA, ISBN 0-89791-575-5, (172–178), doi:http://doi.acm.org/10.1145/169059.169143.

Novick, David & Stephen Sutton (1994), An empirical model of acknowledgment for spoken-language systems, in *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, (96–101).

Paek, T. & D.M. Chickering (2005), On the Markov assumption on spoken dialogue management, in *Proceedings of the 6th SIGDIAL Workshop on Discourse and Dialogue*, (35–44).

Paek, T. & R. Pieraccini (2008), Automating spoken dialogue management design using machine learning: An industry perspective, *Speech Communication* 50:716–729.

Penn, Gerald (2000), *The Algebraic Structure of Attributed Type Signatures*, Ph.D. thesis, Carnegie Mellon University.

Poesio, Massimo & Hannes Rieser (2009), (prolegomena to a theory of) completions, continuations, and coordination in dialogue, university of Essex and Bielefeld University Ms.

Poesio, Massimo & David Traum (1997), Conversational actions and discourse situations, *Computational Intelligence* 13:309–347.

Poesio, Massimo & David Traum (1998), Towards an axiomatization of dialogue acts, in J. Hulstijn & A. Nijholt (eds.), *Proceedings of TwenDial 98, 13th Twente workshop on Language Technology*, Twente University, Twente, (207–221).

Prévot, Laurent (2003), *Structures sémantiques et pragmatiques pour la modélisation de la cohérence dans des dialogues finalisés*, Ph.D. thesis, Université Paul Sabatier, Toulouse.

Purver, M. (2006), CLARIE: Handling clarification requests in a dialogue system, *Research on Language & Computation* 4(2):259–288.

Purver, Matthew, Jonathan Ginzburg, & Patrick Healey (2001), On the means for clarification in dialogue, in Jan van Kuppevelt & Ronnie Smith (eds.), *Current and New Directions in Discourse and Dialogue*, Kluwer, (235–256).

Ranta, Aarne (1994), *Type Theoretical Grammar*, Oxford University Press, Oxford.

Rieser, Verena & Oliver Lemon (2008), Learning Effective Multimodal Dialogue Strategies from Wizard-of-Oz data: Bootstrapping and Evaluation, in *Proceedings of ACL*.

Rieser, Verena & Joanna Moore (2005), Implications for generating clarification requests in task-oriented dialogues, in *Proceedings of the 43rd Meeting of the Association for Computational Linguistics*, Michigan.

Rodriguez, Kepa & David Schlangen (2004), Form, intonation and function of clarification requests in german task-oriented spoken dialogues, in Jonathan Ginzburg & Enric Vallduvi (eds.), *Proceedings of Catalog'04, The 8th Workshop on the Semantics and Pragmatics of Dialogue*, Universitat Pompeu Fabra, Barcelona.

Rudnicky, Alex (2004), Learning to talk by listening, talk presented at Catalog'04, The 8th Workshop on the Semantics and Pragmatics of Dialogue, Universitat Pompeu Fabra, Barcelona, July 2004.

Sadek, David & Renato de Mori (1998), Dialogue systems, in R. de Mori (ed.), *Spoken Dialogues with Computers*, Academic Press, (523–561).

Schegloff, Emanuel (1987), Some sources of misunderstanding in talk-in-interaction, *Linguistics* 25:201–218.

Schlangen, David (2003), *A Coherence-Based Approach to the Interpretation of Non-Sentential Utterances in Dialogue*, Ph.D. thesis, University of Edinburgh, Edinburgh.

Schlangen, David (2005), Towards finding and fixing fragments: Using machine learning to identify non-sentential utterances and their antecedents in multi-party dialogue, in *Proceedings of the 43rd Meeting of the Association for Computational Linguistics*, Michigan.

Searle, John (1969), *Speech Acts*, Cambridge University Press, Cambridge.

Searle, John & Daniel Vanderveken (1985), *Foundations of Illocutionary Logic*, Cambridge University Press, Cambridge.

Seneff, Stephanie & Joseph Polifroni (2000), Dialogue management in the mercury flight reservation system, in *Workshop On Conversational Systems*.

Shriberg, Elizabeth E. (1994), *Preliminaries to a theory of speech disfluencies*, Ph.D. thesis, University of California at Berkeley, Berkeley, USA.

Sinclair, J.M.H. & R.M. Coulthard (1975), *Towards an Analysis of Discourse: The English Used by Teachers and Pupils*, Oxford University Press.

Singh, S., M. Kearns, D. Litman, & M. Walker (1999), Reinforcement learning for spoken dialogue systems, in *Proceedings of NIPS'99*.

Singh, S., D. Litman, M. Kearns, & M. Walker (2002), Optimizing dialogue management with reinforcement learning: Experiments with the njfun system, *Journal of Artificial Intelligence Research* 16:105–133.

Steedman, Mark (1999), *The Syntactic Process*, Linguistic Inquiry Monographs, MIT Press, Cambridge.

Sutton, Richard & Andrew Barto (1998), *Reinforcement Learning*, MIT Press.

Swartout, W., J. Gratch, R. Hill, E. Hovy, S. Marsella, J. Rickel, & D. Traum (2006), Toward virtual humans, *AI Magazine* 27(2):96.

Traum, D. (2004), Issues in multiparty dialogues, *Lecture Notes in Computer Science* :201–211.

Traum, David & Staffan Larsson (2003), The information state approach to dialogue management, in Jan van Kuppevelt & Ronnie Smith (eds.), *Advances in Discourse and Dialogue*, Kluwer.

Voss, L. & P. Ehlen (2007), The calo meeting assistant, *Proceedings of NAACL-HLT, Rochester, NY, USA* .

de Waijer, Joost Van (2001), ‘the importance of single-word utterances for early word recognition, in *Proceedings of ELA 2001*, Lyon.

Walker, M. & A.T.T. Shannon (2000), An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email, *Journal of Artificial Intelligence Research* 12:387–416.

Williams, JD & SJ Young (2007), Partially observable markov decision processes for spoken dialog systems, *Computer Speech and Language* 21(2):231–422.

Wittgenstein, Ludwig (1953), *Philosophical Investigations*, Basil Blackwell, Oxford, citations from second edition, 1988 reprint.

Young, SJ (2006), Using pomdps for dialog management, in *IEEE/ACL Workshop on Spoken Language Technology (SLT 2006)*, Aruba.

Zhang, Bo, Qingsheng Cai, Jianfeng Mao, & Baining Guo (2001), Planning and acting under uncertainty: A new model for spoken dialogue system, in *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, (572–579).