Situated Dialogue Processing for Human-Robot Interaction

Geert-Jan M. Kruijff, Pierre Lison, Trevor Benjamin, Henrik Jacobsson, Hendrik Zender, Ivana Kruijff-Korbayová[1]

DFKI GmbH, Saarbrücken Germany gj@dfki.de

1 Introduction

Talking robots. They speak to our imagination – C3PO, Sonny, R2-D2. We are fascinated by the idea of technology we can talk to, work with, all taking place in the world we live in, the places we inhabit.

dialogue

And that is, in a nutshell, what makes it challenging to build such systems – or even just the capabilities for robots to conduct a dialogue , to interact with a human. Because understanding dialogue is about more than just understanding the speech signal, words, or the utterance. For a robot to understand dialogue when talking with a human, it ultimately needs to understand how that dialogue relates and refers to the world we live in.

Which brings us to a fundamental point we would like to make: The meaning we communicate is based in how we understand the world we talk about. Our awareness of the situations to which we refer influences how we understand dialogue – and, through dialogue, we can further our understanding of those situations. Dialogue, language, is a conduit to the world around us.

Problem is, the world around us provides a robot with very rich, perceptual experiences. And the combinatoric system of language makes it possible for us to talk about that world in a wide variety of ways. So, when it comes to

human-robot interaction

developing an approach to dialogue processing in human-robot interaction , how could a robot possibly figure out what an utterance really is supposed to mean in a given context? Particularly – how can it be related to what the robot knows about the world, and how different agents intend to act in there?

We can find a possible answer to that when we look at what we humans do. When we hear an utterance, we do not wait until the end of it and then try to figure out what it may mean. On the contrary. Evidence shows that, as soon as we get the first signals, we start processing, building up partial representations of potential meanings, and linking those potential meanings to the situated context. As the potential meanings of an utterance unfold, connecting them to the context helps us select those meanings which make sense, and discard those that just don't. And we seem to do the same when we try to figure out how to say something.

So, it is not just that dialogue meaning is closely related to how we are aware of the situated context – both levels seem to closely interact when constructing meaning representations. Through that interaction, we can focus on constructing those meanings that make sense in that context, and omit the potential but otherwise irrelevant ones.

situation awareness

The idea that situation awareness and dialogue processing are closely coupled, forms the hypothesis underlying the work we discuss in this chapter. If dialogue understanding is based in situation awareness, we propose that a

bi-directionality

bi-directional connection between dialogue processing and the various processes for building up and maintaining situation awareness, is fundamental to ensure the system constructs meanings which are interpretable in the situated contexts to which dialogue refers. The bi-directional nature of this connection

means that these processes can interchange and interconnect information, to help focusing and completing understanding.

This sets the approach we will develop apart from more traditional views on the nature of language processing. Received wisdom has it that language is processed in a strictly modular fashion. First we process audio, and word formation. Then we create syntactic structures. Then we see how we can attach some meaning to the structures we got. And finally, waste-basket style for all we could never explain before, we apply "pragmatics" to figure out how those utterance meanings fit in a context. And most implemented dialogue systems go about things pretty much the same way.

modularity

Thing is, the idea that we want to situate dialogue meaning, and understand and produce dialogue in a close coupling with situation awareness, it does not put context *last* – it puts context *first*. Context is not an afterthought – it is the very ground on which we construct interpretations, be that dialogue context or situated context. And because we do so, there is little sense in adopting modularity in our designs, in the sense as suggested in the 1970s by Fodor and Chomsky. We will still distinguish different levels at which we construct our representations, and modularize that way. But the interpretations these structures represent will not be built in isolation from other levels, nor from the context in which they are formulated. Instead, what happens at one level is influenced, guided, by what happens in the rest of the system, *in parallel*.

context

Throughout this chapter, we will discuss how this idea of bi-directionality can be implemented, and what its consequences are on how we can deal with "real life" situations in human-robot interaction. The settings for our discussions will be the scenarios we have investigated in the CoSy project, including "home tour"-like interactive spatial exploration, object manipulation, and interactive visual learning. To achieve bi-directionality between the dialogue system and situation awareness, we have completely integrated our system with those discussed in other chapters. The system can connect its content to information about the local visuospatial scene and the overall spatial organization of the environment, as well as to action plans and motivations, and use the way these connections work out to focus speech recognition, to select utterance meanings, and to resolve and produce referring expressions. We show that including context in dialogue processing leads to (statistically) significant improvements in speech recognition and utterance comprehension.

The organization of this chapter reflects the basic point we are trying to make about dialogue meaning, and the hypothesis about bi-directional processing. Rather than discussing a system implementation module-by-module, (reminiscent of that modularity view we would like to avoid), we look at how we can bring about bi-directional processing. What it means for the way we process language. How processing becomes more complex as we gradually up the complexity of language and the situated contexts it refers to. And how bi-directionality helps – helps in what the system needs to do, given what people have observed in user studies, and in what it does do, verified in evaluations.

We first set up a background against which our approach can be placed, discussing in §2 several approaches in cognitive sciences and AI. In §3 we then start off by discussing the basics for contextualized language processing, showing how bi-directionality influences first of all the design of the processes and representations we adopt. We continue in §4 with looking into dialogues about what the robot can see, connecting dialogue meaning with an understanding of local visuo-spatial scenes. We discuss how we can talk about those scenes, for example when a human tries to teach the robot more about the objects it sees, and how bi-directionality helps focusing speech recognition, utterance analysis, reference resolution, and producing references to objects.

Of course, we do not always need to be talking about what is in front of us. The Explorer scenario provides us with a setting in which we often discuss places we can visit, or where we can find objects in the world – without necessarily being right there and then. In §5 we present how we can go beyond the current situated context, and use information about the larger world around the robot to talk about other places. One interesting challenge bi-directionality helps addressing is in resolving and producing references to such places.

indexicality
intentionality

In §6 we take meaning beyond having mostly a referential, *indexical* nature. We look into how particularly speech acts like questions and commands express intentionality , and how we can relate that to processes for motivation and planning. Bi-directionality enters the dialogue processing picture again by indicating which potential utterance interpretations correspond to possible plans, and which ones do not.

In retrospect, what do we contribute? For one, we discuss here a system for situated dialogue processing in human-robot interaction which provides a wider, and deeper, coverage than many other systems. The robot understands more – in terms of what you can talk about, and how that relates to the world. But there is a more fundamental contribution we hope to be making here. We follow out the idea that context matters in situated dialogue processing. Situated context, and dialogue context together. Processing meaning starts and ends in those. We can bring that about by considering processes to be bi-directionally coupled. This coupling enables them to exchange information, complement each others representations, and generally help guide processing by focusing attention on those meanings that are supported by a given situated context. We argue that this affects the way these processes should be designed (incremental processing, in parallel across multiple levels of representation) and what representations should facilitate (over- and under-specification, and packing). And, most importantly, we show what bi-directionality brings. We discuss added functionality over and beyond what any individual level or modality would be capable of providing, and present evaluations which empirically verify the positive effects of bi-directionality on focusing processing. Particularly, bi-directionality leads to significant improvements in speech recognition and utterance comprehension, with a combined

effect of improving more than 56% over a baseline including commercial speech recognition software.

If we would like robots to talk – here is one way we believe we should walk.

2 Background

Language provides us with virtually unlimited ways in which we can communicate meaning. This, of course, raises the question of how precisely we can then understand an utterance as we hear it. Empirical studies in various branches of psycholinguistics and cognitive neuroscience have investigated what information listeners use when trying to understand spoken utterances which are about visual scenes. An important observation across these studies is that interpretation *in context* plays a crucial role in the comprehension of utterance as it unfolds. Following [34] we can identify two important dimensions of the interaction between the purely linguistic, dialogue context, and the situated context. One is the *temporal dimension*. The ways our visual attention are guided appear to be timed closely with how we proceed with understanding an utterance. In empirical studies we can witness this by for example eye movements. The second is the *information dimension*. This indicates that listeners not only use linguistic information during utterance comprehension, but also scene understanding and "world knowledge." Below we discuss aspects of these dimensions in more detail.

2.1 Multi-level integration in language processing

Until the early 1990s, the dominant model of language comprehension was that of a modular, stage-like process. See for example [23]. On this model, a language user would sequentially construct each level of linguistic comprehension – from auditory recognition all the way to pragmatic, discourse-level interpretation. As [69] observe, two hypotheses followed from this view. One hypothesis is that people first construct a local, context-*independent* representation of the communicated meaning. Only once this meaning has been completely constructed, it is interpreted against the preceding dialogue context. Secondly, and related, is the hypothesis that dialogue context-related processing only enters the process of language comprehension at a relatively late stage.

Opposing these hypotheses is the view that language comprehension is an incremental process. In such a process, each level of linguistic analysis is performed in parallel. Every new word is immediately related to representations of the preceding input, across several levels – with the possibility for using the interpretation of a word at one level to co-constrain its interpretation at other levels. A natural prediction that follows from this view is that interpretation against dialogue context can in principle affect utterance comprehension

as the utterance is incrementally analyzed, assisting in restricting the potential for grammatical forms of ambiguity. [18, 5] phrased this as a *principle of parsimony*: those grammatical analyses are selected that for their reference resolution impose the least presuppositional requirements on a dialogue context.

Since then, various studies have investigated further possible effects of dialogue context during utterance comprehension. Methodologically, psycholinguistic studies have primarily investigated the effects of dialogue context by measuring *saccadic eye movements* in a visual scene, based on the hypothesis that eye movements can be used as indications of underlying cognitive processes [63, 42]. Alternatively, cognitive neuroscience-based studies use event-related brain potentials (ERPs) to measure the nature and time course of the effects of dialogue context on human sentence comprehension [65].

Both lines of study have found that lexical, semantic and discourse-level integrative effects occur in a closely time-locked fashion, starting already at the phoneme or sub-word level; see [1], and [68, 69, 70]. Particularly, a range of dialogue-level integrative effects have observed. Referential binding has been shown to play a role in the constraining various types of local syntactic ambiguities, like garden path-constructions [18, 5, 2], and relative clauses [56, 55]; [66, 68, 69]. These effects primarily concern a *disambiguation* of already built structures. Integrating semantic and dialogue-level information during utterance comprehension also has important *anticipatory* effects. [62, 19]; [67] observe how contextual information influences what lexical meanings can be anticipated, priming phonological understanding and lexical access. Contextual information can even override disprefered lexical meaning [45].

Anticipatory effects indicate that utterance comprehension is thus not only an incremental process of constructing and then disambiguating. Anticipation enables context-dependent phonological recognition, lexical retrieval, and syntactic construction - without there being a need to generate and test all combinatory possible constructions. Incrementality and anticipation based on multi-level integration appears to give rise to a process in which comprehension arises through a convergence based on constraining and co-activation. Dialogue context and the interpretative contexts which are delineated during utterance comprehension converge to become functionally identical [69]. As a result, ambiguity need not even arise, or is at least being much more limited a priori through context.

An important issue in all of the above remains of course the degree to which integrative effects indeed should commit to a certain understanding. Garden path sentences are a good example. They show that overcommitment risks the need for re-interpretation – an issue for *cognitive control* [11, 30, 46].

2.2 Language processing and situational experience

We already noted before that humans integrate *linguistic* and *non-linguistic* information when processing an utterance. Below we discuss studies which

investigate how categorical and contextual information from situation awareness can effect utterance comprehension. These studies use eye-trackers to monitor where people look at in a scene, and when.

[3] present a study revealing that listeners focus their attention on objects before these objects are referred to in the utterance. For example, consider a scene with a cat, a mouse, and a piece of cheese. When someone hears "The cat chases the mouse", her gaze already moves to the mouse in the scene before she has actually heard that word; similarly for "The mouse eats the cheese." Knowing that cats typically chase mice (not cheese), and that the argument structure of *chase* reflects this, the listener *expects* that the next object to be mentioned will be the mouse, and directs gaze to that object. We thus see an anticipatory effect arising from the online integration of lexico-semantic information (verbal argument structure), situational context (the present objects, and the reported action), and categorical knowledge (prototypical object-action relations).

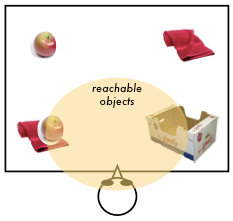

Fig. 1. Put, apple, towel, box

Not only world knowledge can influence online utterance comprehension, also scene understanding can. For example, consider the situation in Figure 1. [63] show that, once the listener has heard "Put the apple on the towel ..." she faces the ambiguity of whether to put the (lone) apple onto the (empty) towel, or to take the apple that is on the towel and put it somewhere else. The ambiguity is revealed as visual search in the scene. Only once she has heard the continuation "... into the box" this ambiguity can be resolved. Interestingly, in [63] the listener cannot directly manipulate the objects. If this is possible (cf. Figure 1), [16] show that also reachability plays a role in comprehending the utterance. Because only one apple is reachable, this is taken as the preferred referent, and as such receives the attention. This underlines the effect *physical embodiment* may have on language comprehension.

Scene understanding also concerns the *temporal projection* towards possible future events [22]. [4, 31] show how such projection can also affect utterance comprehension. These studies used a scene with a table, and beside it a glass and a bottle of wine. Investigated was where listeners look when they hear "The woman will put the glass on the table. Then, she will pick up the wine, and pour it carefully into the glass." It turns out that after hearing the "pour-

ing" phrase, listeners look at the table, not the glass. Listeners thus explicitly project the result of the picking action into the scene, imagining the scene in which the glass is on the table.

These studies reveal that the interaction between vision and language is not *direct*, but *mediated* [4]. Categorical understanding plays an important role in the sensorimotoric grounding of language. This is further underlined by studies like [25, 21], following up on the idea of category systems as mediating between perceptual modalities and language [24, 9]. These studies show how categorical understanding gives rise to expectations based on affordances, influencing comprehension of spatial or temporal aspects of action verbs.

In conversational dialogue [29, 48] gaze has been shown to be automatically aligned in simple collaborative interaction. The time intervals between eye-fixations during production and comprehension of a referring expression are shorter than in monologue. This is further evidence for the relevance of visual common ground of interlocutors and how that accelerates the activation of jointly relevant concepts.

2.3 Situated language processing in AI/HRI

Studies on how humans process visually situated dialogue show an important aspect of "grounding" is based on how we can resolve a referent in the world for an object reference. In establishing referents, listeners use visual and spatio-temporal properties of objects, and combine these properties with various forms of salience.

What have we achieved so far in building AI systems that can relate language to the world? Roy & Reiter present a comprehensive overview of existing approaches (up to 2005) in [50]. They identify several important issues: What are suitable *representations* to mediate between language and sensori-motoric experience? How can linguistic representations be associated with perceptual and action categories? How does "context" come into play? It is easy to see how we can relate these issues to observations in human sentence processing. And to an extent they have been addressed in implemented systems to date.

One of the earliest systems which connected incrementally built utterance analyses to a visual world was Winograd's SHRDLU [73]. Among more recent approaches, the most developed are those by Gorniak & Roy, and Steels *et al.* Gorniak & Roy [26, 27] present an approach in which utterance meaning is probabilistically mapped to visual and spatial aspects of objects in the current scene. Recently, they have extended their approach to include action-affordances [28]. Their focus has primarily been on the grounding aspect. A similar comment can be made for SHRDLU. Steels *et al* [61, 60, 59] have developed an approach where the connection between word meaning and percepts is modeled as a semiotic network, in which abstract categories mediate between language and the visual world.

Although they use an incremental approach to constructing utterance meaning, grounding meanings in the social and physical context as they are

construed, the (im)possibility to ground alternative meanings does not feed back into the incremental process to prune inviable analyses. Furthermore, the approaches focus entirely on category-based mediation (the "what" dimension). They omit most of the spatio-temporal dimension of interpretation (the "where/when" and "how"), restricting them to visual information about the currently perceivable scene.

The approach we present here improves on these approaches along these lines. Like Scheutz et al [51, 13], we develop a model for incremental utterance processing in which the analyses are pruned if it is impossible to ground them in the situated contexts referred to. Furthermore, grounding is not restricted to the visual scene, but is extended to include the larger spatio-temporal context.

3 Talking

What does it take to make a robot talk? Specifically, what does it take to make a robot process situated dialogue?

Simply put, for a robot to talk it first of all needs to be able to listen. It should be able to process a speech signal, turning it into (possible) sequences of words. Then, turn those words into utterances, and assign a meaning representation to them. These meaning representations should be linked to a model of the preceding dialogue, so that we can figure out how it refers to things that were already said before, and how it moves the dialogue along.

That's one part – listening, comprehending utterances against a model of the dialogue context, and updating that context model as the dialogue continues. The other part is the talking part. Based on how the dialogue has developed so far, the robot should decide how to continue. Then, following up on this decision, it should see how to formulate the utterances to achieve that "goal", and formulate them such that they refer to the situations in a contextually appropriate way. It should be clear to the listener what the robot is referring to, talking about. And once the robot has said what it decided to say, it should of course again update the model of the dialogue context.

In this section we would like to focus on the comprehension side, and sketch how the production side is structured. We would like to start simple here, explaining the basics behind the approach we take to making robots talk. Explain the design decisions as they result from the bi-directionality hypothesis, why we do the things the way we propose to do them. Come the next sections, we will delve into more detail, or where necessary provide references to more technical discussions.

Adopting the bi-directionality hypothesis poses requirements both on how we design our processes, and our representations.

As for processes, we already pointed out earlier that we need to drop certain assumptions regarding modularity. We cannot consider processes in isolation. We need to turn them into "permeable glass boxes." While processing,

it should be possible to take partial results, connect them with information from other processes, and then use the results to guide how to continue processing. One way to design such processes so is to make them *incremental*. In incremental processing , a process proceeds from the "beginning" towards the "end" of a representation it is to process, in a step-wise fashion. After each step, bi-directionality can be used to guide how to take the next step. Because linguistic representations are typically sequential, we can process them incrementally. Much of the benefits of bi-directionality consist therein that they can help processes focus on sensible analyses, discarding those which are not supported by the context.

incremental processing

Each process typically maintains several concurrent hypotheses. Particularly if we look at things from an efficiency point of view, there are several requirements bi-directionality raises for the nature of representations. First of all, we are looking at dialogue, a context in which interpretations develop over time. Utterances refer to the preceding context, adding to or correcting previous information. Connecting information across processes can be more efficient if these relations are clear. It identifies a history of what previously was already interconnected. The way we will address this requirement is by using discourse referents as permanent hooks to relate information to, both at utterance- and at dialogue-level.

discourse referents

Secondly, even though a process may maintain multiple hypotheses, this does not imply that they need not share certain similarities in how they interpret something. Representations should identify how alternative analyses are different, and where there are similarities, so that we can avoid having to check each hypothesis individually. We will address this requirement by packing multiple hypotheses into a single, possibly underspecified graph structure, and determining preference orders over alternatives.

packing

preference orders

Below we will explain these processes and representations in more detail. We start with representations, to make it clear what we are working with, and towards.

Representing an utterance

Loosely speaking, a dialogue is an exchange of utterances between two or more "interlocutors." Usually, this exchange serves a particular purpose. In the context of human-robot interaction, that purpose usually relates to performing tasks in the real-world.

dialogue

The thing is, whereas sentences in a text are usually complete, and grammatically well-formed, this need not be the case with utterances in spoken dialogue. Utterances are often incomplete or grammatically incorrect, and may include self-corrections. "Take the red uh ... no put that green one next to the ... you know, yes, the pyramid." This of course raises the question, what we should consider an utterance to be.

utterance

Most dialogue systems (still) consider an utterance to be like a sentence, and have a definable beginning and end. We adopt a more flexible notion than

that. What we ultimately consider to be an utterance, depends on the context in which linguistically conveyed content is being used. As far as processing within our system is concerned, an utterance is a stream. There are marked points at which it can be further interpreted, either within the dialogue system or beyond it. At such "points," the representation of the utterance provides enough meaning to start off further forms of processing. Each further interpretation modality is thus free in considering when it works with meaning, and thus –ultimately– what it considers an "utterance" to be.

Which brings us to how we represent meaning. We represent meaning as an ontologically richly sorted, relational structure – a logical form [35, 7]. The following is an example of a logical form:

logical form

$$
\begin{aligned}
&@_{w_1}\,\text{:cognition}(\textbf{want} \wedge \langle\textsc{Mood}\rangle\, \mathit{ind} \wedge \langle\textsc{Tense}\rangle\, \mathit{pres} \wedge \\
&\quad \langle\textsc{Actor}\rangle\, (i_1 : \text{person} \wedge \textbf{I} \wedge \langle\textsc{Num}\rangle\, \mathit{sg}) \wedge \\
&\quad \langle\textsc{Event}\rangle\, (p_1 : \text{action-non-motion} \wedge \textbf{put} \wedge \\
&\qquad \langle\textsc{Actor}\rangle\, y_1 : \text{person} \wedge \\
&\qquad \langle\textsc{Patient}\rangle\, (m_1 : \text{thing} \wedge \textbf{mug} \wedge \\
&\qquad\quad \langle\textsc{Delimitation}\rangle\, \mathit{unique} \wedge \langle\textsc{Num}\rangle\, \mathit{sg} \wedge \langle\textsc{Quantification}\rangle\, \mathit{specific} \wedge \\
&\qquad\quad \langle\textsc{Modifier}\rangle\, (r_1 : \text{q-color} \wedge \textbf{red})) \wedge \\
&\qquad \langle\textsc{Result}\rangle\, (t_1 : \text{m-whereto} \wedge \textbf{to} \wedge \\
&\qquad\quad \langle\textsc{Anchor}\rangle\, (r_2 : \text{e-region} \wedge \textbf{right} \wedge \\
&\qquad\quad \langle\textsc{Delimitation}\rangle\, \mathit{unique} \wedge \\
&\qquad\quad \langle\textsc{Num}\rangle\, \mathit{sg} \wedge \\
&\qquad\quad \langle\textsc{Quantification}\rangle\, \mathit{specific} \wedge \\
&\qquad\quad \langle\textsc{Owner}\rangle\, (b_1 : \text{thing} \wedge \textbf{ball} \wedge \\
&\qquad\quad \langle\textsc{Delimitation}\rangle\, \mathit{unique} \wedge \langle\textsc{Num}\rangle\, \mathit{sg} \wedge \langle\textsc{Quantification}\rangle\, \mathit{specific})))) \wedge \\
&\quad \langle\textsc{Patient}\rangle\, (y_1 : \text{person} \wedge \textbf{you} \wedge \langle\textsc{Num}\rangle\, \mathit{sg}) \wedge \\
&\quad \langle\textsc{Subject}\rangle\, i_1 : \text{person})
\end{aligned}
$$

Each node has a unique identifier with an associated ontological sort (e.g. $t1$ of sort $action-motion$), and a proposition (e.g. **want**). Nodes are connected through named relations. These indicate how the content of a single node contributes to the meaning of the whole expression. For example, "you" ($y1$) both indicates the one whom something is wanted of (*Patient*-relation from $w1$), and the one who is to perform the put action (*Actor*-relation from $t1$). Nodes carry additional features, e.g. $i1$ identifies a singular person.

Propositions and relations in such a representation are instances of concepts. This makes it possible for us to interpret logical forms further using ontological reasoning. We use this possibility in reference resolution, and in relating meaning representations to interpretations formed outside the dialogue system.

The relational nature of our representations provides us with several advantages. We build up our representations from elementary propositions as we

illustrated above – sorted identifiers and propositions, features, and relations. An interpretation is thus simply a conjunction of such elementary propositions, and the more we can connect those elementary propositions, the more complete our interpretation becomes. This makes it relatively straightforward to represent partial interpretations. For example, for "take the red ..." receives the following interpretation:

$$
\begin{aligned}
&@t_1 : \text{action-motion}(\mathbf{take} \wedge \langle \textsc{Mood} \rangle \mathit{imp} \wedge \langle \textsc{Tense} \rangle \mathit{pres} \wedge \\
&\quad \langle \textsc{Actor} \rangle (a_1 : \text{entity} \wedge \mathbf{addressee}) \wedge \\
&\quad \langle \textsc{Patient} \rangle (m_1 : \text{thing} \wedge \\
&\qquad \langle \textsc{Delimitation} \rangle \mathit{unique} \wedge \langle \textsc{Num} \rangle \mathit{sg} \wedge \langle \textsc{Quantification} \rangle \mathit{specific} \wedge \\
&\qquad \langle \textsc{Modifier} \rangle (r_1 : \text{q-color} \wedge \mathbf{red})) \\
&\quad \langle \textsc{Subject} \rangle a_1 : \text{entity})
\end{aligned}
$$

The interpretation shows more than just the content for the three words. It also shows that "red" is expected to be the color of the "thing" which is supposed to be taken.

Characteristic for language is that it presents many ways in which we can say things – and interpret them. This inevitably means that we will usually get not just one, but multiple alternative interpretations for an utterance. To keep ambiguity to a minimum, we should look at to what extend these interpretations are indeed different. Where they show overlaps, we should ideally have to deal with those identical parts only once.

Using relational structure and elementary propositions enables us to do so. We represent alternative interpretations as alternative ways in which we can connect content, whereas identical content across interpretations is represented once. The procedure to create such "condensed" representations is called *packing* , after [47, 15]. Figure 2 illustrates the development of the packed packed representation for "here is the ball". At the first step ("take"), 9 logical forms are packed together, with two alternative roots, and several possible ontological sorts for the word "here". The second step reduces the number of alternative interpretations to one single logical form, rooted on the verb "be" with a "presentational" ontological sort. The possible meanings for the determiner is expressed at the dependent node of the "Presented" relation. At this point we have an *overspecified* meaning. Although the delimination is unique, we cannot tell at this point whether we are dealing with a singular object, or a non-singular (i.e. plural) object – all we know it has to be one or the other. This becomes determined in the fourth step ("here is the ball").

packing

In the appendix to this chapter we present a detailed technical discussion of packing.

Representing the interpretation of an utterance in context

The meaning of an utterance goes well beyond what is expressed just by the individual words that make it up. Meaning is about how the utterance

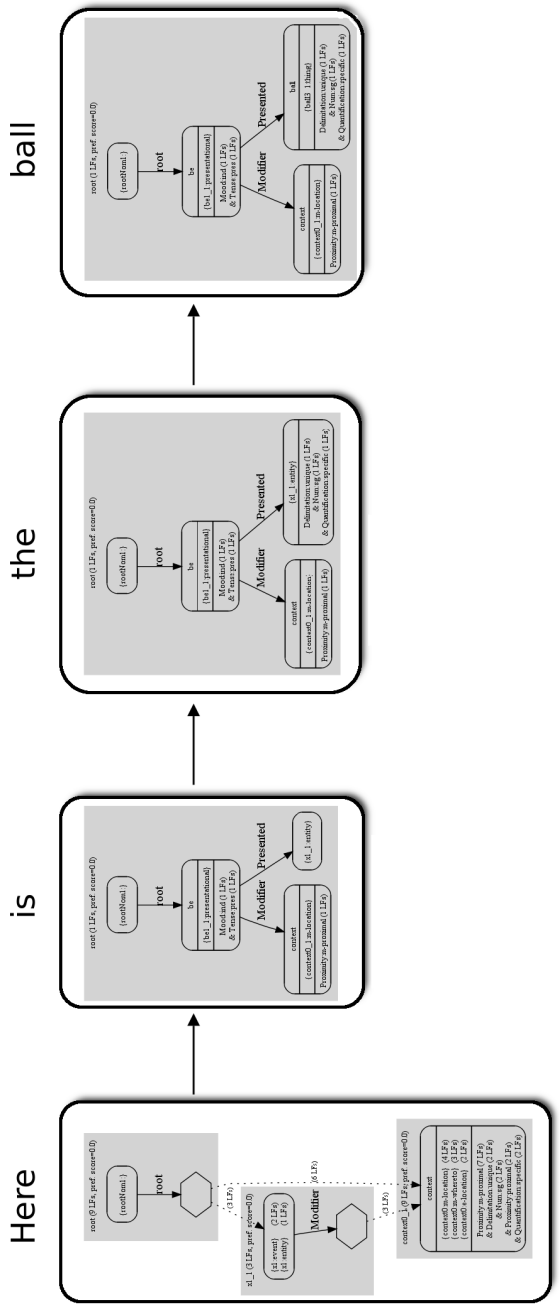

Fig. 2. Example of incremental parsing and packing of logical forms, for the utterance "Here is the ball". Four steps are shown.

relates to the context – the situation (indexically) and to the actions (to be) performed therein (intentionally). How it can be taken to refer to things we already talked about, to beliefs we have, to expectations which may be raised on the basis of what we are saying. How the utterance helps us to further the dialogue, helping to reach a goal – or not.

Referent resolution is the first step we take to relate content from the current utterance, to that of previous utterances in the dialogue context. The purpose here is to establish *co-reference relations*: relations between mentions referring to the same object(s) or event(s). Examples of references to previous objects are pronouns (e.g. "it"), or anaphoric expressions (e.g. "the red mug"). We are using a (simple) algorithm based on referent resolution in the segmented dialogue representation theory of [6].

For each index in a logical form, the algorithm determines potential antecedents in the preceding dialogue, using the model of the dialogue context the system maintains. There are two simple cases. One, we may be talking about a something new. We then create a new (unique) referent identifier, say ant_n, and represent this as a *reference structure* [**NEW** : $\{ant_n\}$]. Two, there is a unique antecedent referent ant_i. We represent this as [**OLD** : $\{ant_i\}$], meaning there is a "discourse old" antecedent ant_i. In both cases we relate the index in the logical form (which only has naming uniqueness within the scope of the logical form) to the built structure.

Complications arise if a reference cannot be ambiguously resolved. A good example of such a situation arises when resolving deictic pronouns like "this". How a deictic pronoun needs to be resolved, depends on the dialogue- and the situated context. If the utterance is not accompanied by a gesture, the preference is to resolve the reference to a preceding antecedent in the dialogue. However, if the utterance is accompanied by a gesture, then this preference may be overridden. It may be that the gesture refers to an object which was mentioned before, just not most recently; or it may refer to an object which has not been talked about at all. To capture these possibilities, we allow for reference structures to specify preference orders over sets of old and new referents. For example, if a deictic pronoun can be resolved to several old antecedents, with ant_i the most preferred, or to a new referent ant_n, then we get

$$[\mathbf{OLD} : ant_i < \{ant_j, ..., ant_k\} < \mathbf{NEW} : \{ant_n\}].$$

Subsequently, information about grounding the utterance in the situated context then can help resolving this ambiguity (e.g. by providing support for a new referent). The example of deictic pronoun nicely illustrates the principle *bi-directional* nature of situated dialogue processing as implemented here. There is no strict pipeline of interpretation processes, invoked at incremental steps. Instead, interpretation processes interact to mutually constrain and complement the interpretations they form.

Another aspect of dialogue-level interpretation regards "speech acts", or dialogue moves. A dialogue move specifies how an utterance "functions in", i.e.

contributes to furthering the dialogue. We determine an utterance's possible dialogue move(s) on the basis of the shape of the logical form, and expectations about possible moves to extend the current dialogue. Figure 3 illustrates a decision tree used to map logical form features to dialogue moves.

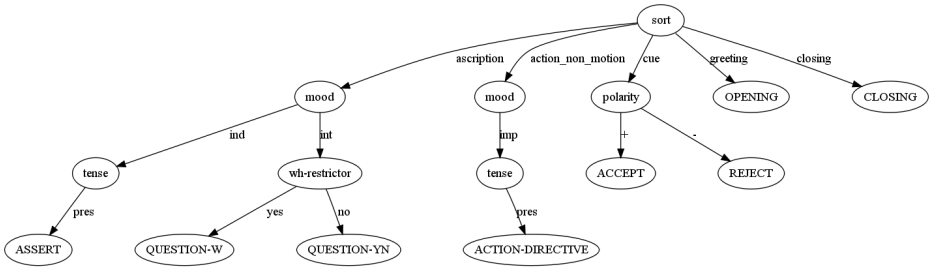

Fig. 3. Example of a decision tree for determining dialogue moves from LF form

Once the dialogue move for an utterance has been determined, the utterance content, and its referent- and event structures are added to the dialogue context model maintained by the system. Figure 4 shows a snapshot of such a model. (We will elaborate on event structures in §6.)

Comprehending an utterance in context

When we try to comprehend an utterance, we analyze it at several linguistic levels. As we are dealing with spoken dialogue, the first step is the *automatic speech recognition* [ASR], which takes an audio signal stream as input and produces a word recognition lattice as output. This step is known to be particularly error-prone [44], for several reasons. The first one is the inherent *noise* present in the real-world environments in which our robots are deployed. Since we require the speech recognition system to be *speaker-independent*, we also have to deal with the wide variety of voices, accents and styles of speech of human speakers. And finally, natural spoken dialogue is also characterised by a high proportion of *disfluencies* (filled pauses, speech repairs, corrections, repetitions), and the production of many *partial* or *ill-formed* utterances, all of which negatively affect the performance of the speech recognition.

automatic speech recognition

Our strategy for addressing this issue is to exploit *contextual knowledge* about the situated environment and the dialogue history to prime the utterance recognition. This knowledge is represented in the cognitive architecture as a cross-modal *salience model* of the situated context. It integrates both visual salience (objects perceived in the physical scene) and linguistic salience (previously referred-to objects within the current dialogue). The model is dynamically updated as the environment evolves, and is used to establish expectations about uttered words which are most likely to be heard given the context. The update is realised by continously adapting the word probabilities

salience model

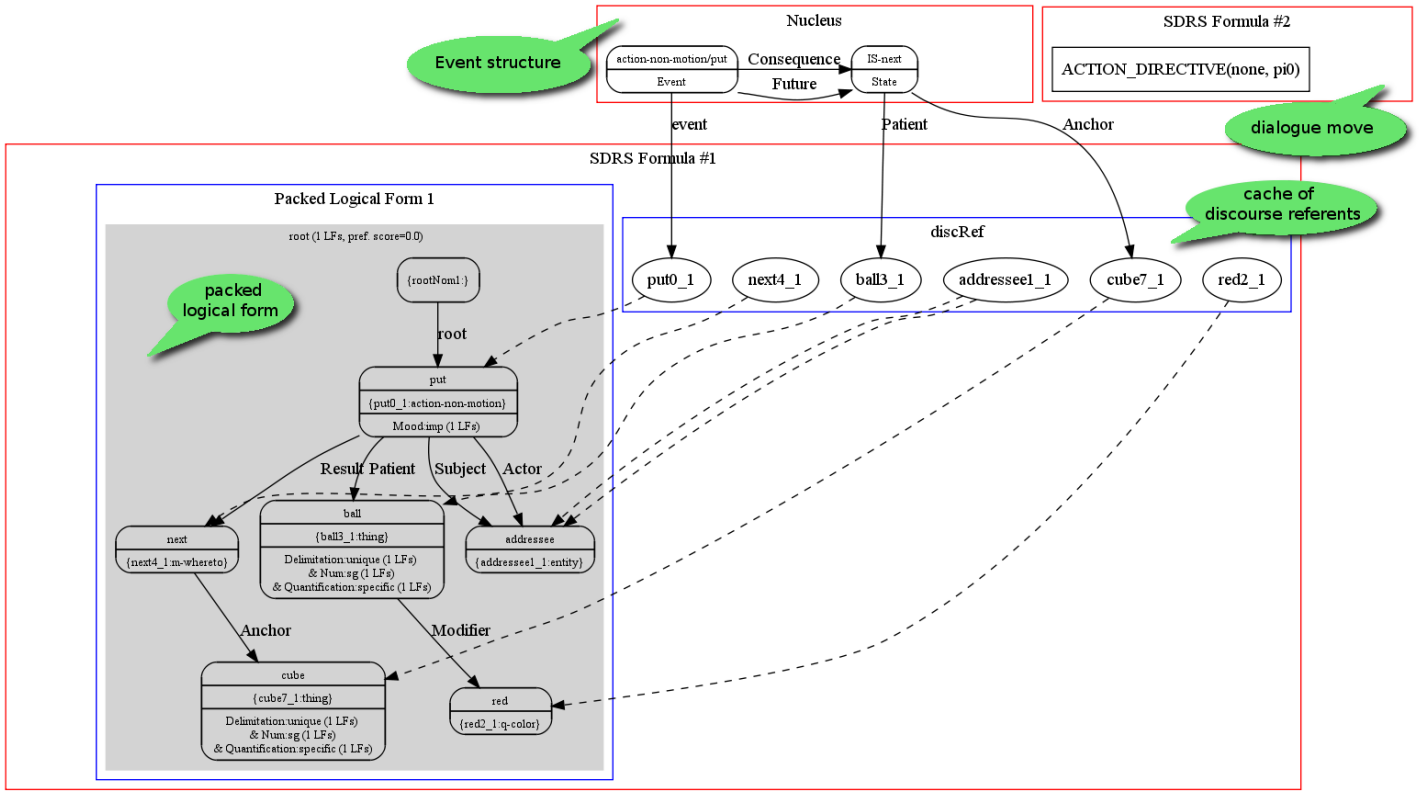

Fig. 4. System visualization of dialogue context model: Utterance content, coreference, event structures, and dialogue moves.

specified in the statistical language model of the speech recognizer. We have shown that this approach yields a statistically significant improvement of the ASR performance compared to a baseline, non context-sensitive model [41].

As soon as the speech recognizer is able to suggest a (partial) recognition hypothesis for the utterance, a *word recognition lattice* is created and inserted into the working memory for subsequent analysis. A word recognition lattice is a packed representation for the set of potential recognition hypothesis, combined with their respective confidence scores. The set of recognition hypotheses can be easily retrieved by traversing the lattice. Figure 5 illustrates a typical example of word lattice.

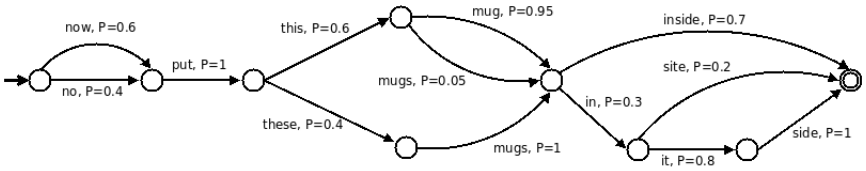

Fig. 5. A typical word recognition lattice

This word recognition lattice is then further processed incrementally – the lowest, incremental level being that of grammatical analysis. For modeling natural language grammar we use the Combinatory Categorial Grammar (CCG) framework [57, 8]. CCG is a lexicalized framework: For each word, there are one or more lexical entries specifying a syntactic category, and a corresponding lexical meaning. A syntactic category defines how the word can be used in forming a larger, grammatical expression. The lexical meaning specifies the word meaning. The meaning of an expression is built up compositionally, in parallel to its syntactical derivation.

Figure 6 illustrates how meaning is built up in parallel to a grammatical derivation. The verb "take" has a syntactic category $s : e/np : p$. This means that it will yield a sentence s if it is combined to the right / with a noun phrase np. The indices e and p relate the syntactic material to the meaning being built: e provides a handle to the index for the verbal meaning, whereas p indicates that the noun phrase will provide the meaning for the Patient [7, 35].

The words "take" and "the" can be combined incrementally into an expression "take the", using function composition (the **B** rule in CCG, cf. [57]). The resulting syntactic category specifies that this expression requires a noun n to its right to yield a complete sentence. The meaning of the determiner "the" circumscribes that of the noun. Finally, "take the" and "mug" are combined into a complete expression, "take the mug".

We have factorized (incremental) grammatical analysis into several, interconnected functions: the incremental parsing process itself, packing/unpacking and pruning of incrementally construed analyses of utterance meaning, and

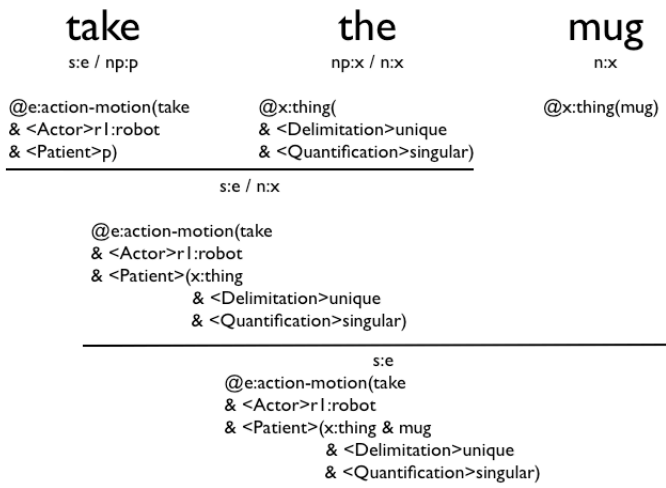

Fig. 6. Incremental analysis of "take the mug"

context-sensitive lexical retrieval. Figure 7 illustrates the interactions between these different functions.

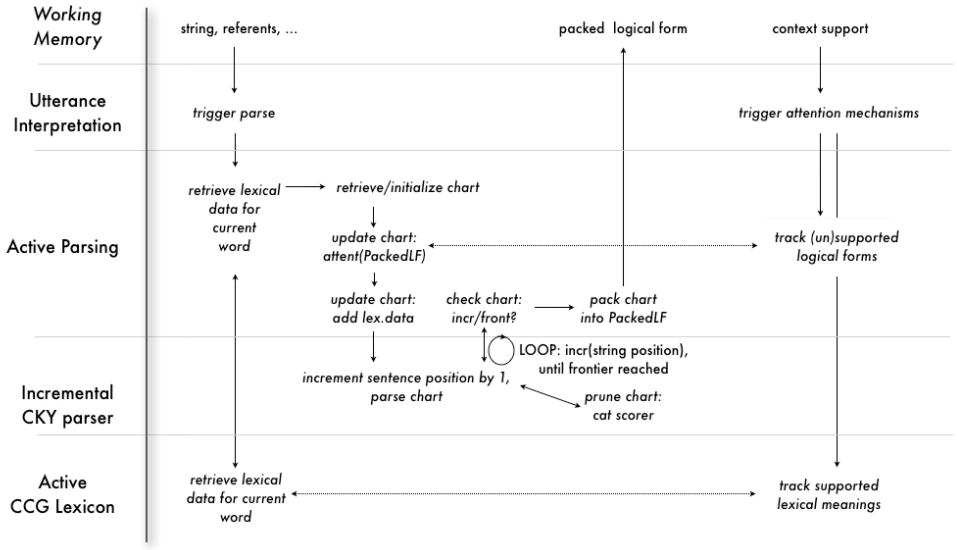

Fig. 7. Context-sensitive utterance interpretation at grammatical level: interactive processes for parsing and lexical retrieval, which can be primed by contextual information.

Parsing begins by retrieving the lexical entries for the first word, and initializing the chart. A chart is a data structure in which all active and completed analysis are stored, marking for each analysis what part of the utterance (from beginning to some position x) it covers. Maintaining partial analyses makes it possible to re-use them at a later point, when constructing analyses that span more of the utterance. (This principle of re-using partial analyses sets chart-based parsing apart from e.g. backtracking, in which analyses are construed every time anew.) The chart is subsequently updated with the lexical entries for the first word, and a parsing process starts. Parsing is based on a bottom-up Early chart parser built for incrementally parsing Combinatory Categorial Grammar. Its implementation relies on basic functionality provided by OpenCCG[1].

Incremental chart parsing creates partial, and integrated analyses for a string in a left-to-right fashion. After each increase in position, the parser checks whether it has reached a *frontier*. A frontier is specified as a type of complete grammatical structure at the right branch of a grammatical derivation. This enables us to specify whether the parser should return after every word, or e.g. after every phrase. At each frontier check, the chart is pruned using a *category scorer*. This scorer ranks the categories for the partial analyses construed so far, possibly pruning them if they are guaranteed not to lead to a complete analysis. (For example, in an incremental analysis, any category requiring an argument to the left \ preceding the beginning of the utterance will never be completed.)

Once incremental parsing stops, a packed logical form is construed, and provided to working memory. This packed representation of possible grammatical interpretations of an utterance provides the basis for further interpretation steps – for example, referent resolution. Depending on the exact fashion in which these processes are synchronized, the next phase of incremental parsing is triggered by the becoming available of further information on working memory (e.g. referents). In this case, the chart is retrieved, and updated with the lexical entries for the current word, and incremental parsing continues as described above.

The advantage of factorizing grammatical analysis into separate inference- and lexical retrieval processes is that the system can use information about the situated- and task-context to prime attention in both processes, possibly asynchronously (i.e. "opportunistically"). Activated categories for objects and events can help to restrict what lexical meanings are retrieved ("activated") for a word. Furthermore, based on what (partial) interpretations can be grounded in the context, unsupported interpretations (analyses) can be removed from the chart.

[1] http://openccg.sf.net

Picking up the right interpretation

Even with the help of these contextual priming/pruning techniques, the outcome of the utterance comprehension process will nevertheless remain in many cases severely ambiguous and underspecified. This is not surprising: ambiguity is known to be extremely pervasive in natural language, at all processing levels (lexical, syntactic, semantic, pragmatic), and contextual priming/pruning alone cannot be expected to resolve all ambiguities. This means that most utterance will still yield tens, if not hundreds, of possible analyses. Without mechanisms for interpretations selection/filtering at our disposal, these ambiguities are inevitably going to hinder any further interpretation.

parse selection

We therefore implemented a robust *parse selection* system able to determine the most probable analysis among a set of alternative interpretations. The parse selection is based on a statistical linear model which explores a set of relevant acoustic, syntactic, semantic and contextual features of the parses, and is applied to compute a *likelihood score* for each of them.

Our approach can therefore be seen as a *discriminative* approach to utterance interpretation: we first generate the possible analyses, and then discriminate amongs them according to various features.

The parameters of this linear model are estimated against an automatically generated corpus of ⟨utterance, logical form⟩ pairs. The learning algorithm is an *averaged perceptron*, a simple and efficient technique for parameter estimation which is known to give very good results for this task [17].

The parse selection can be formalised as a function $F : \mathcal{X} \rightarrow \mathcal{Y}$ where the domain $\mathcal{X}$ is the set of possible input utterances[2], and the range $\mathcal{Y}$ is the set of parses. We assume:

1. A function $\mathbf{GEN}(x)$ which enumerates all possible parses for an input x. In our case, this function simply represents the set of parses of x which are admissible according to the CCG grammar.
2. A d-dimensional feature vector $\mathbf{f}(x, y) \in \Re^d$, representing specific features of the pair (x, y). It incorporates various acoustic, syntactic, semantic or contextual features relevant for discriminating the parses.
3. A parameter vector $\mathbf{w} \in \Re^d$.

The function F, mapping an utterance to its most likely parse, is then defined as:

$$F(x) = \underset{y \in \mathbf{GEN}(x)}{\operatorname{argmax}} \ \mathbf{w}^T \cdot \mathbf{f}(x, y) \tag{1}$$

where $\mathbf{w}^T \cdot \mathbf{f}(x, y)$ is the inner product $\sum_{s=1}^{d} w_s \ f_s(x, y)$, and can be seen as a measure of the "quality" of the parse.

Given the parameters $\mathbf{w}$, the optimal parse of a given utterance x can be therefore easily determined by enumerating all the parses generated by the

[2] or, in the more general case, a set of possible word recognition lattices.

grammar, extracting their features, computing the inner product $\mathbf{w}^T \cdot \mathbf{f}(x, y)$, and selecting the parse with the highest score.

We present evaluation results of parse selection later on in the chapter, after we have discussed how language and visuo-spatial information can be combined.

Producing an utterance in context

Just like in comprehension we put context into the production of dialogue. Planning what to say and how to say it, is all influenced by context – dialogue-, situation-, and action contexts.

Producing one or more utterances is triggered by a communicative goal. This goal can arise within the dialogue system, for example to follow up in a purely communicative way (e.g. matching a greeting with a greeting), or from outside. The differentiation how communicative goals may arise enables us to put dialogue in the service of other modalities, e.g. to help clarify something the robot does not understand [38], and to achieve a continuum between planning action and interaction (as we will explain further in §6).

A communicative goal specifies a dialogue move, and content which is to be communicated. We formulate this goal as a logical form. As we describe in detail in [36], we then use a planner to expand (and possibly rewrite) this goal logical form into a logical form specifying the content for one or more utterances. The planner uses a collection of systemic-functional grammar networks [10] to decide, on the basis of the provided content, the way this content can be related within the logical form and to the broader context, how to extend a logical form.

systemic-functional grammar networks

Relating content to context is particularly relevant in the case of generating referring expressions. This task of can be paraphrased as finding a description for an entity in the world (the *intended referent*) that refers to the intended referent and only the intended referent. This implies that the description must be chosen in a way that prevents it from referring to another entity in the current *context set*. All entities in the context set except the intended referent form the *contrast set*. The referring expression must thus distinguish the intended referent from the members of the contrast set. A referring expression is a noun phrase (NP) of any degree of complexity. In order to provide enough information to uniquely identify the intended referent, further attributes of the referent need to be expressed, for instance with adjectives or prepositional phrases, which in turn might contain a referring expression NP.

One of the best understood and widely accepted approaches for generating referring expressions, is the *incremental algorithm* of Dale and Reiter[20]. This algorithm needs a knowledge base that describes the *properties* of the domain entities through *attributes* and *values*. A special attribute is an entity's *type*. The algorithm is initialized with the *intended referent*, a *contrast set* (defined as the *context set* without the intended referent) and a list of *preferred attributes*. The algorithm tries to incrementally rule out members

of the contrast set for which a given property of the intended referent does not hold.

In the course of this chapter we describe various instantiations of this algorithm, for producing references to aspects of local contexts (§4), and the larger spatial organization of the environment (§5).

Once content planning has yielded a complete logical form for one or more utterances, we provide content utterance by utterance to a realizer. This realizer uses the same grammar as the parser, to produce a set of possible surface strings expressing that content [72]. We use statistical models, trained over a corpus of "usual" utterances for our domain, to select the best realization [71]. This realization is then provided to the MARY speech synthesis engine, to produce audio output [52].

4 Talking about what you can see

In the previous section we discussed how model meanings for utterances, in a linguistic fashion. We put relations between concept instances, to see how they contribute to the overall meaning – and, by establishing how they relate to preceding utterances, how they contribute to the overall dialogue. Next, let's have a look at how to *situate* that meaning.

We begin by looking at how we could process situated dialogue about things you can see. Both the human and the robot are in the same place, and are talking about objects that are (roughly) in their field of view, often even within reach. Simply put, they are in the same room and that's all they talk about. We call this a *small-scale space* or *closed context* .

small-scale space
closed context

Looking at the literature, you will often find this problem of relating language to "the world" (which, really, mostly means small-scale space) referred to as symbol grounding . There is a linguistic symbol, representing some meaning, and it needs to be "grounded in" how the world is perceived. The degree to which this "grounding" determines the meaning of the linguistic symbol is one issue for discussion – the way we look at interconnecting content across modalities is described in more detail in Chapter 2.

symbol grounding

But there is more to linguistic meaning than just that. Expressing something is an *act*. We convey meaning in a way that makes it clear not just how a listener should understand what we are saying, but also what she is to do with it. There is a purpose, an intention to saying something. This goes beyond trying to understand the dialogue move or speech act of an utterance. Such a move is –often– a reflection of the action to be undertaken in the real world. So when it comes to relating language to the world, we need to do more than connect symbols to perceptions. We also need to connect meanings to how these perceptions are to be acted upon, or dealt with.

intention

Which brings us to another point we want to raise here. Not every bit of meaning is created and presented equal, in a contextual sense. As a dialogue progresses, we build up a collection of references to aspects of the real world.

They form the common ground for the dialogue, a set of mutually held and agreed upon beliefs about the dialogue and the situated context in which that dialogue is set. When we use meanings which refer back to beliefs which are part of the common ground, such meanings provide "old information" on which we can build further, connecting "new information" to it. The point is, there is a difference in which new and old information should be grounded. Whereas the indexicality of old information is generally assumed to hold, for the meanings providing new information it needs to be established. And how that is to be done, couples back to the intentional aspect of an utterance.

common ground

Let us illustrate these ideas on an example, before we discuss how they are dealt with in our approach to situated dialogue processing. One popular setting for human-robot interaction is socially guided learning , in which a robot interacts with a human while trying to learn more about the world. As we describe in Chapters 7 and 10, we have explored several aspects of visual learning in such a setting. For example, when the human would like to explain more about a particular object to the robot, she could say something like "The red mug is big."

socially guided learning

Now what would that mean?

The way she refers to the object in question, "the red mug," makes clear that she assumes that the robot knows what object she is talking about. It is presented as old information, something already talked about and identifiable not just in the dialogue context but also in the situated context. Next comes a bit of information about the size of the object. The property "big" is being attributed to the object, providing new information which the robot presumably did not yet know. Finally, the intention behind attributing such a new property to an already known object is to teach the robot. It should (still) ground "red mug" in its visual models of the scene, and try to update its models so as to be able to classify that object as "big." Separating old from new information thus first of all indicates, what we should be able to ground. The intention clarifies what to do with the new information. Whether or not the robot then succeeds in learning how to classify the "red mug" as "big" determines how it should react to the utterance. Instead, would we have have followed "standard" approaches to grounding, and not make any such distinctions (old, new; indexicality, intentionality), we would just have the robot try to connect all the meanings immediately to what it sees. The almost inevitable outcome of that would have been – "no." Not knowing that the property "big" could be applied to the object, the robot would not be able to ground the meaning "big red mug" to the visual object (using the proposition of applying the predicate "big" to the argument "red mug").

What you see and what you mean

Building up a representation of the possible meanings for an utterance, we relate these meanings to the various models the robot maintains. These models can be of the environment, or of actions and plans set therein. We do so in a

mediated way. We explained already before that we model meaning as ontologically sorted, relational structures – graphs which related concept instances through named relations. We can resolve mentions of these instances against the dialogue context model, so that we know how instances are being talked about and referred to over the course of a dialogue. Grounding such structures using mediation means we use ontologies to mediate the translation of the representations specific to language as a modality, into representations from which we can ultimately construct a-modal representations. (See also Chapter 2, on binding of modality-specific structures and the formation of unions as a-modal representations.)

mediation

Since our graph structures are directed and acyclic, we can use recursion over our structures. At each step, a node in the graph is translated into another graph structure on the basis of its ontological sort (as defined in the ontology used in the grammar). A translation can just cover this single node, or a subgraph governed by that node. It can exclude nominals in that subgraph from further processing, making it possible to collapse or omit parts of the graph. An example of that is the translation of spatial expressions, like "the box to the left of the ball:"

$$
\begin{array}{l}
@b_1 \text{ :thing}(\textbf{box} \wedge \\
\quad \langle \textsc{Delimitation} \rangle \ \textit{unique} \wedge \langle \textsc{Num} \rangle \ \textit{sg} \wedge \langle \textsc{Quantification} \rangle \ \textit{specific} \wedge \\
\quad \langle \textsc{Modifier} \rangle \ (t_1 : \text{m-location} \wedge \textbf{to} \wedge \\
\qquad \langle \textsc{Anchor} \rangle \ (l_1 : \text{e-region} \wedge \textbf{left} \wedge \\
\qquad\quad \langle \textsc{Delimitation} \rangle \ \textit{unique} \wedge \langle \textsc{Num} \rangle \ \textit{sg} \wedge \langle \textsc{Quantification} \rangle \ \textit{specific} \wedge \\
\qquad\quad \langle \textsc{Owner} \rangle \ (b_2 : \text{thing} \wedge \textbf{ball} \wedge \\
\qquad\qquad \langle \textsc{Delimitation} \rangle \ \textit{unique} \wedge \langle \textsc{Num} \rangle \ \textit{sg} \wedge \langle \textsc{Quantification} \rangle \ \textit{specific}))))
\end{array}
$$

Such a representation is translated –more or less– into a structure there is a "Location:left-of" between the box and the ball.

Our approach is set apart from other approaches in several ways. First, we ground meaning as graph structures, not as individual words. Second, we are dealing with instances to which we assign discourse referents. Whenever we ground content, we maintain the connection between the discourse referent, and the content it is grounded to. Next time we have new content pertaining to that discourse referent, we update the translated content rather than that we would provide a whole new graph structure to be anchored in the situated context. Naturally, this connection is also used to inform dialogue about changes of information about content to which referents are attached – or can no longer be attached. Thirdly, any ambiguities present in a packed representation are propagated to content to be grounded. This is one point where it we make use of the mechanism for informing dialogue about the grounding possibilities of discourse referents, and any relations between them. As we will see below, ambiguities which cannot be grounded, can be pruned from the packed representation, not having any contextual support in the current

contextual support

context.

When mapping structures are based on ontological sort, content can be flagged as indexical, intentional, or both. As we describe in Chapters 2, 9 and 10, we assume architecture designs in which we can at least differentiate between a working memory for visuo-spatial information about the current context (the "binding" working memory), and a working memory for structures which will prompt the robot to take actions (the "motivation" working memory). Indexical content is put on binding working memory. (Currently, the architecture designs do not include a form of situated episodic memory. Where that to be the case, the resolved temporal reference of an utterance could be used to establish whether to indeed provide the content to the model of the current scene, or to store it with a future or past episode.)

We store intentional content on motivation working memory, with pointers to the indexical content it is related to. Intuitively, intentional content usually represents processes and ascriptions ("verbs"), with representations of objects they operate on including pointers to their indexical counterparts. For example, consider again the command "take the red mug," which results in the following logical form:

$$
\begin{array}{l}
@t1 : \text{action-motion}(\textbf{take} \wedge \langle \textsc{Mood} \rangle \, imp \wedge \langle \textsc{Tense} \rangle \, pres \wedge \\
\quad \langle \textsc{Actor} \rangle (a_1 : \text{entity} \wedge \textbf{addressee}) \wedge \\
\quad \langle \textsc{Patient} \rangle \, (m_1 : \text{thing} \wedge \textbf{mug} \\
\qquad \langle \textsc{Delimitation} \rangle \, unique \wedge \langle \textsc{Num} \rangle \, sg \wedge \langle \textsc{Quantification} \rangle \, specific \wedge \\
\qquad \langle \textsc{Modifier} \rangle \, (r_1 : \text{q-color} \wedge \textbf{red})) \\
\quad \langle \textsc{Subject} \rangle \, a_1 : \text{entity})
\end{array}
$$

Translating this structure into content to be grounded, binding working memory will end up containing structures for the robot, and the red mug. On the intentional side, we will have a structure for the *take* action, identifying the robot as the Actor of the action, and the red mug as the Patient. The representations for the robot and the mug on motivation working memory refer to their corresponding representations on binding working memory, so that we can situate the intended action.

These translations are based directly on the content of the utterance. We have already seen that they make use of information from the dialogue context, namely resolved discourse referents, to establish what extra-linguistic content to connect to. This primarily concerns indexical information. On the intentional side, information about dialogue move is combined with utterance mood to provide further information about the kind of intention we are dealing with – for example, a command, an assertion, or a question.

We will deal with commands later on, in §6, and focus here on questions and assertions. Particularly in the scenarios we consider in Chapters 9 and 10, questions and assertions have in common that they involve a predication over an argument. If we say "The ball is red" we are effectively stating that we can

predicate having a "red color" over the "ball" object. A question can also be taken to involve such a predication relation, only now we are quantifying that predication. We rephrase a question like "What color is the ball?" to predicate a color over "ball," quantifying that the value of the color attribute. Slightly more complicated, if we ask "Is the ball red?" we want to know whether we can indeed do so – turning quantifying the predication into a higher-order quantification over the truth of the predication (similar to situation theoretic semantics). For formal details, see [38].

Practically, we represent this information through additional relational structure on the working memory dealing with intentions. We connect the structure produced for the predicate to that for the argument, using two relations. One relation indicates whether we have a polar or factual question, i.e. whether we quantify over the truth of, or a value for, the predication. The other relation indicates what it is that the speaker is after – the intended belief state, resulting from answering the question. For example, for "What color is the ball?" we will connect the predicate "color" to the argument "ball" using a relation "Fact-Q", and a relation "SPEAKER-KNOWS:Colour." For polar questions like "Is the ball red?," the latter relation will also indicate the value of the color: "SPEAKER-KNOWS:Colour?red" i.e. the speaker knows whether the ball indeed has a red color.

The argument of SPEAKER-KNOWS can be a pointer to any type of elementary predication in a meaning representation. This makes it possible for to quantify over any type of information represented as predication – be that a sort ("Is this a ball?"), the value of an attribute, or a relation ("Is there a ball to the left of the box?"). Assertions only differ from questions in that we just introduce a relation stating that the hearer knows the asserted information (HEARER-KNOWS, with the same type of structure as SPEAKER-KNOWS).

How we subsequently evaluate whether we can achieve the desired belief state is dependent on the grounding of the various pieces of content. This is driven by the information status of content, as we explain next.

What you all know, and what you are saying

We already differentiate content by its intentional or indexical nature. We make a further division into content which the speaker presents as "old information" belonging to the common ground, and that which is presented as new. Grammatically speaking, we are using a form of information structure [58]. We determine the information status of an object on the basis of its semantic features for delimitation and quantification, and how the content functions in the larger context of an intentional construction [35]. The basic idea is that if an object is established to be old news, we will immediately try to ground it. This way, it can provide the basis for forming the relevant situated context against which we need to consider the intention. Based on the intentional content, we will next evaluate whether the new information can indeed be grounded in relation to the already grounded information.

information structure

In §6 we will explain how this works out for commands like "put the ball to the left of the box." We will see there how information structure interacts with the intention of performing a "put" action on the mentioned objects, and the desired state of the ball being to the left of the box, to help establish how we should evaluate whether this action can indeed be performed. For our current purposes, we will focus on ascriptions – assertions such as "This box is red," or questions like "Is the red box to the left of the ball?"

As we saw above, we build additional structure for question- and assertion-interpretations. This structure reflects a desired update to an interlocutor's belief state. Evaluating a question or an assertion then boils down to establishing whether we can obtain that state. How we evaluate is guided by information structure. When translating meaning to indexical and intentional content, we determine the information status of objects on the basis of using semantic delimitation and quantification. We adopt the overriding exception that we will always interpret the argument of a SPEAKER- or HEARER-KNOWS relation to be "new." Any indexical information with an "old" information status will be written to binding working memory. We do this to establish the relation to common ground. In parallel, we represent the new information and any purely intentional information on the motivation working memory. Evaluation then consists therein to establish whether we can in principle update binding working memory with the new information, or check against the results of grounding whether the new information already holds or could be retrieved. A felicitous update may in this case provide the trigger for learning processes, as we discuss in Chapter 7.

The result is a model of information structure and its interaction with indexical and intentional content that is reminiscent of a dynamic semantics-based approach to information structure [58]. Where we differ is the use of multiple indexical and intentional contexts, and the evaluation of the update on one context relative to the intended use of information as stated in another context.

Using what you see to rank alternative interpretations

As we already outlined in section 3, a *discriminative model* is used to assign a score to each possible semantic interpretation of a given spoken input. The discriminative model includes a wide range of *linguistic* as well as *contextual* features. The linguistic features are defined on the analyses construed at the different processing levels: the *acoustic* level (based on ASR scores), the *syntactic* level (based on the derivational history of the parse), and the *semantic* level (based on substructures of the logical form). As for the contextual features, they are defined using information from both the situated context (the objects in the visual scene) and the dialogue context (previously referred entities in the dialogue history).

Experimental evaluation We performed a quantitative evaluation of our approach to parse selection. To set up the experiments for the evaluation, we

have gathered a corpus of human-robot spoken dialogue for our task-domain, which we segmented and annotated manually with their expected semantic interpretation. The current data set contains 195 individual utterances along with their complete logical form.

Three types of quantitative results are extracted from the evaluation results: *exact-match*, *partial-match*, and *word error rate*. Tables 1, 2 and 3 illustrate the results, broken down by activated features, use of grammar relaxation, and number of recognition hypotheses considered.

Gram.	Activated Features				Nbest 1			Nbest 5		
Relax.	Sem.	Synt.	Ac.	Cont.	Pr.	R.	F_1	Pr.	R.	F_1
					40.9	45.2	**43.0**	14.4	13.9	14.2
				×	35.2	41.5	38.1	28.8	31.8	30.2
			×		42.8	46.3	44.5	38.1	47.1	42.2
			×	×	41.9	45.8	43.7	43.1	49.4	46.0
	×				59.0	54.3	56.6	30.3	51.3	38.1
	×			×	59.0	54.3	56.6	35.2	55.1	43.0
	×		×		59.0	54.3	56.6	58.3	65.4	61.6
	×		×	×	59.0	54.3	56.6	60.8	66.3	63.4
×					20.9	49.0	29.3	10.7	34.1	16.3
×				×	20.9	49.0	29.3	12.1	39.0	18.4
×			×		27.1	55.5	36.4	27.3	54.6	36.4
×			×	×	21.7	50.0	30.2	27.9	56.2	37.3
×		×			34.1	61.1	43.7	21.0	39.6	27.4
×		×		×	30.2	58.2	39.7	21.9	44.2	29.3
×		×	×		34.1	61.1	43.7	32.8	59.1	42.2
×		×	×	×	32.5	60.0	42.2	32.5	60.0	42.2
×	×				49.6	69.5	57.9	28.9	77.7	42.2
×	×			×	49.6	69.5	57.9	31.0	78.9	44.5
×	×		×		49.6	69.5	57.9	52.1	83.1	64.0
×	×		×	×	49.6	69.5	57.9	53.1	84.4	65.2
×	×	×			52.7	70.8	60.4	29.6	78.1	43.0
×	×	×		×	52.7	70.8	60.4	31.7	79.3	45.3
×	×	×	×		52.7	70.8	60.4	54.6	82.7	65.8
×	×	×	×	×	52.7	70.8	**60.4**	55.6	84.0	**66.9**

Table 1. Exact-match accuracy results, broken down by activated features, use of grammar relaxation, and number of recognition hypotheses considered. For each configuration, we give the precision, recall, and F_1 value (all in percents).

Each line in the tables corresponds to a possible configuration. For each configuration, we analyse the accuracy results on different NBests, and give the precision, recall and F_1 value for each.

The first cell of the first line corresponds to the baseline: no grammar relaxation, no activated features, and use of the first NBest recognition hypothesis. The last line corresponds to the final results with all features, combined with the grammar relaxation mechanism.

Two elements are worth noticing in the results:

1. In each of the three tables, we observe that no configuration is able to beat the results obtained with all activated features. In other words, it shows that all features types are playing a positive role on the task.
2. Likewise, we observe that taking into account more ASR recognition hypotheses has a positive effect on the results: the results obtained using

Gram. Relax.	Activated Features: Sem.	Synt.	Ac.	Cont.	Nbest 1: Pr.	R.	F_1	Nbest 5: Pr.	R.	F_1
					86.2	56.2	**68.0**	73.5	45.8	56.4
				×	85.5	56.0	67.7	81.3	54.2	65.1
			×		86.8	56.4	68.3	84.3	60.4	70.4
			×	×	86.2	56.2	68.1	85.4	60.4	70.7
	×				90.5	57.4	70.3	80.1	66.4	72.6
	×			×	90.5	57.4	70.3	83.3	67.2	74.4
	×		×		90.5	57.4	70.3	88.9	67.1	76.4
	×		×	×	90.5	57.4	70.3	89.5	67.2	76.8
×					75.7	73.3	74.5	71.4	81.9	76.3
×				×	73.7	72.8	73.2	71.7	78.7	75.1
×			×		75.3	73.2	74.2	74.6	73.1	73.8
×			×	×	72.7	72.5	72.6	74.6	74.1	74.4
×		×			80.9	74.6	77.6	76.2	72.1	74.1
×		×		×	80.2	74.4	77.2	78.7	76.2	77.4
×		×	×		80.8	74.6	77.6	80.3	74.5	77.3
×		×	×	×	80.4	74.5	77.3	80.3	75.5	77.8
×	×				86.5	75.8	80.8	80.7	88.4	84.4
×	×			×	86.5	75.8	80.8	80.0	88.3	84.0
×	×		×		86.5	75.8	80.8	86.2	86.7	86.4
×	×		×	×	86.5	75.8	80.8	86.3	87.2	86.8
×	×	×			88.1	76.2	81.7	79.3	88.2	83.5
×	×	×		×	88.1	76.2	81.7	81.7	88.5	85.0
×	×	×	×		88.1	76.2	81.7	87.5	85.4	86.4
×	×	×	×	×	88.1	76.2	**81.7**	87.6	86.0	**86.8**

Table 2. Partial-match accuracy results, broken down by activated features, use of grammar relaxation, and number of recognition hypotheses considered. For each configuration, we give the precision, recall, and F_1 value (all in percents).

Gram. Relax.	Activated Features: Sem.	Synt.	Ac.	Cont.	Nbest 1	Nbest 3	Nbest 5	Nbest 10
					20.5	26.9	29.7	25.9
				×	20.5	23.6	24.6	28.0
			×		20.5	19.7	19.6	19.7
			×	×	20.5	18.7	18.2	18.3
		×			20.5	24.6	25.6	31.2
		×		×	20.5	21.4	23.2	26.1
		×	×		20.5	18.3	18.4	18.1
		×	×	×	20.5	17.3	17.4	17.4
×					19.6	23.6	25.9	23.9
×				×	19.3	20.4	23.3	26.7
×			×		19.7	18.6	18.4	19.3
×			×	×	19.4	18.0	17.6	17.7
×		×			19.4	24.6	26.9	27.9
×		×		×	19.4	22.2	23.9	28.1
×		×	×		19.4	18.8	18.7	18.8
×		×	×	×	19.4	17.8	17.3	17.4
×	×				20.2	22.4	25.5	29.4
×	×			×	20.2	21.0	22.9	26.1
×	×		×		20.2	17.8	17.8	17.8
×	×		×	×	20.2	17.4	17.1	17.1
×	×	×			19.4	21.5	24.3	28.7
×	×	×		×	19.4	19.8	21.9	25.9
×	×	×	×		19.4	16.8	16.7	16.7
×	×	×	×	×	19.4	**16.5**	**15.7**	**15.7**

Table 3. Word Error Rate results, broken down by activated features, use of grammar relaxation, and number of recognition hypotheses considered. For each configuration, we give the error rate (in percents).

five recognition hypotheses are substantially better than those obtained based only on the first hypothesis.

Comparison with baseline Here are the comparative results we obtained:

- Regarding the exact-match accuracy results, the difference between the baseline results and the results with our approach (grammar relaxation and all features activated for NBest 10) is striking: the F_1-measure climbs from 43.0 % to 67.2 %, which means a relative difference of **56.3** %.
- For the partial-match, the F_1-measure goes from 68.0 % for the baseline to 87.3 % for our approach – a relative increase of **28.4** %.
- Finally, the decrease in Word Error Rate is also worth noting: we go from 20.5 % for the baseline to 15.7 % with our approach. The difference is statistically significant (p-value for t-tests is 0.036), and the relative decrease is of **23.4** %.

Using what you see to figure out what is meant

If we have a packed representation that includes alternative interpretations, any indexical ambiguity will end up as alternative relational structures on binding working memory. By monitoring which relational structures can be grounded in the current context, and which ones cannot, we can prune the set of interpretations we maintain for the dialogue. We thus handle examples such as those discussed in [13] through an interaction between binding, and dialogue processing. Below we provide a detailed example of resolving syntactic attachment ambiguities using the situated context. (Lexical ambiguities based in different semantic categories are resolved against visual categories.)

Fig. 8. Situated context for "put the ball near the mug to the left of the box."

Consider the visual scene in Figure 8, and the utterance "put the ball near the mug to the left of the box". Linguistically speaking, this utterance is ambiguous. There are several ways in which we can combine the modifiers "the ball", "near the mug", and "to the left of the box." Is the ball near the mug? Or is "near the mug" the place where the robot is to put the ball, with "the mug" supposedly being located left of the box?

On its own, the utterance is highly ambiguous. But, this ambiguity somehow vanishes when we consider the visual scene in Figure 8. Then it is clear

that there is only one sensible way to understand the utterance. The ball is near the mug, and it should end up to the left of the box (as indicated by the arrow). The system achieves the same disambiguation effects through (incremental) pruning of linguistic interpretations, based on whether they can be grounded in the visuo-spatial situated context.

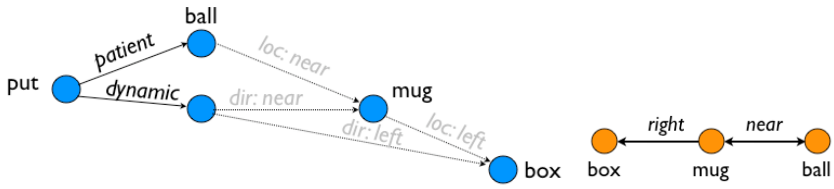

Fig. 9. Ambiguous (complete) packed logical form for "put the ball near the mug to the left of the box" (l.) and the spatial relations for the visual scene (r.)

Figure 9 (right) gives the spatial model for the visual scene in Figure 8 [33]. On the left is the (complete) packed logical form we obtain for "put the ball near the mug to the left of the box". Up to "put the ball near the mug" the modifier "near the mug" remains ambiguous between being the destination for where to put the mug, or specifying a location for "the ball." The visuo-spatial scene provides support for both interpretations, (although planning may prefer the locative reading, as the ball is already near the mug thus pre-empting execution of the action). As soon as "to the left of the box" is being processed, the spatial model invalidates the reading on which the mug is located left of the box. This resolves "to the left of the box" to be the destination of the put action, and (by grammatical inference over the resulting syntactic categories) "near the mug" to be the location modifier of "the ball."

Referring to what you see

A robot isn't just to understand what we are saying. It should also be able to produce dialogue which refers to the environment in meaningful and appropriate ways. In the context of small-scale space, what is particularly important is that the robot can refer to objects and the spatial relations between them.

This presents an interesting challenge. If the robot is to generate any form of spatial language, is needs to construct and maintain a model that explicitly marks the spatial relations between objects in the scene. However, the construction of such a model is prone to the issue of combinatorial explosion both in terms of the number objects in the context (the location of each object in the scene must be checked against all the other objects in the scene) and number of inter-object spatial relations (as a greater number of spatial relations will require a greater number of comparisons between each pair of objects. This becomes particularly problematic when we consider that a scene may be dynamic, requiring the robot to update its models.

We present in [32] a framework that addresses this issue. We provide a way to define the set of objects in the context that may function as a landmark, and then sequence the order in which spatial relations are considered using a cognitively motivated hierarchy of relations. Defining the set of objects in the scene that may function as a landmark reduces the number of object pairs that a spatial relation must be computed over. Sequencing the consideration of spatial relations means that in each context model only one relation needs to be checked and in some instances the agent need not compute some of the spatial relations, as it may have succeeded in generating a distinguishing locative using a relation earlier in the sequence.

A further advantage of our approach stems from the partitioning of the context into those objects that may function as a landmark and those that may not. As a result of this partitioning the algorithm avoids the issue of infinite recursion, as the partitioning of the context stops the algorithm from distinguishing a landmark using its target.

In recapitulation

When it comes to talking about what you see, we discussed above several aspects in which the bi-directionality hypothesis turns up. The possible linguistic meanings we can provide for an utterance are connected to the way the situation is understood, which is coupled back to what meanings are established as contextually supported. We use this mechanism in post-filtering during incremental parsing, in parallel to predictive mechanisms such as parse selection and word lattice re-scoring, and during production in the generation of referring expressions.

We illustrated how we ground meanings, by looking at intentional and indexical aspects, and the information status of content. Instead of grounding all content wholesale word-by-word in visuo-spatial models, as is usually done, we first only ground meaning already part of the common ground, and then evaluate whether new information can be grounded in the sense as indicated by the intention of the utterance. This yields a situated form of dynamic, context-sensitive interpretation of linguistic meaning.

5 Talking about places you can visit

Above we discussed how we process situated dialogue about small-scale space. The human and the robot are in the same location, and talk about things that are in view. Already there we faced the problem to determine what part of that space forms the current context – which objects, and what aspects of spatial organization, we can consider common ground.

large-scale space

This becomes an even bigger issue when we want to talk about *large-scale space* – that kind of "space which cannot be perceived at once" [40]. Discussing aspects of large-scale space, for example where a particular room

is or where the robot could find a specific object, is typical for the Explorer scenario, see [39] and Chapter 9. Most of these referents will, however, not be in view for the interlocutors.

So, whereas in situated dialogue set in small-scale space we can call upon visuo-spatial content stored on a short-term 'binding' working memory, we need to go beyond that in the case of large-scale space. In this section we will discuss how we can integrate content from situated dialogue with ontological reasoning and conceptual spatial mapping [39, 74, 75].

5.1 Talking about places

When it comes to talking about places, various Wizard-of-Oz studies have investigated how humans tend to inform robots about the spatial organization of an environment. For example, [64] discuss a study on how a human presents a familiar indoor environment to a robot, and [53] when a human talks with a robot wheelchair while being seated in it. These studies have yielded various important insights.

The experimental setup in [64] models a typical guided tour scenario. The human guides the robot around and names places and objects. One result of the experiment is the observation that people tend to employ many different strategies to introduce new locations. Besides naming whole rooms ("this is the kitchen" referring to the room itself) or specific locations in rooms ("this is the kitchen" referring to the cooking area), another frequently used strategy was to name specific locations by the objects found there ("this is the coffee machine"). Any combination of these individual strategies could be found during the experiments. Moreover, it has been found that subjects only name those objects and locations that they find interesting or relevant, thus personalizing the representation of the environment that the robot constructs.

In [53], the subjects are seated in a robot wheelchair and asked to guide it around using verbal commands. This setup has a major impact on the data collected. The tutors must use verbal commands containing deictic references in order to steer the robot. Since the perspective of the human tutor is identical to that of the robot, deictic references can be mapped one-to-one to the robot's frame of reference. One interesting finding is that people tend to name areas that are only passed by. This can either happen in a 'virtual tour' when giving route directions or in a 'real guided tour' ("here to the right of me is the door to the room with the mailboxes."). A robust conceptual mapping system must therefore be able to handle information about areas that have not yet been visited.

Next we discuss how we deal with the above findings, combining information from dialogue and commonsense knowledge about indoor environments.

5.2 Representing places to talk about

In Chapter 5, we present our approach to semantic modeling of space. In this approach, we use a multi-layered spatial map that represents space at different

leveles of abstraction. The most abstract layer, the 'conceptual map', characterizes spatial units (e.g. rooms) by assigning them human concepts (e.g. "kitchen"), which can be used to resolve or generate linguistic expressions. The 'conceptual map' is represented as a Description Logics ontology, consisting of a concept taxonomy and a storage of instances, which form the T-Box and A-Box of a Description Logics reasoning framework.[3] The concept taxonomy is a hand-written common sense ontology representing various aspects of an indoor environment (different kinds of areas and other spatial structures, different kinds of objects, agents, and several different relations that can hold between any of these). During run-time the ontology is populated with instances of spatial units and objects through evaluation and interpretation of sensory data (e.g. laser range scans, and visual object detection). A conceptual map that is constructed only from sensory input, e.g. during an autonomous exploration of the robot's environment, will consist of instances of the abstract concept `Area` (corresponding to the units of the topological map layer), which are further specified by the appropriate sub-concepts `Room` and `Corridor` (based on the laser-based semantic place labeling method), and also instances of `Object`, further specified by their respective visual object class, e.g. `Couch` or `TV`. On the basis of this object information, the reasoner can even further specify the area instances, for instance by inferring that a `Room` instance containing some `KitchenObject` instance (e.g. an instance of `Coffeemachine`) is an instance of the more special concept `Kitchen`.

Through this approach, the robot achieves a level of spatial understanding that is already compatible with the linguistic categories that humans use to refer to places in an indoor environment. The conceptual map, however, also holds information about the environment given by human users, for example in a 'guided home tour' interactive mapping set-up.

Our approach to interactive map acquisition accomodates the previously mentioned findings in studies on Human-Augmented Mapping [64] through the following properties:

References to whole rooms or specific locations are used to assert that the instance of the corresponding topological area is of the mentioned concept, even if the reasoner could not infer that knowledge on the basis of the robots own information.

References to specific objects, and thus omitting naming the whole room, will assert that an instance of the mentioned object type is present, which allows the reasoner to draw further inferences about the current topological area. In the above example, the user only points out that "there is the coffee machine". On the basis of its knowledge that the current area is a `Room` instance, which is asserted to contain a `Coffeemachine` instance, the reasoner now infers the new concept `Kitchen`.

[3] We have used different 3rd party reasoners in our experiments, including RACER, Pellet, and Jena.

Like this, our system can combine sensor-based information and information provided through dialogue with a human user. This allows the system to cope with otherwise incomplete information, and with highly personalized information. Our approach yields a conceptual representation of space that is suitable for understanding linguistic references to spatial entities, and for producing expressions that can be understood by human users.

5.3 Referring to elsewhere

A conversational autonomous mobile robot will inevitably face situations in which it needs to refer to an entity (an object, a locality, or even an event) that is located somewhere outside the current scene. In technical terms, the robot must be able to produce a *referring expression* to an entity in large-scale space [76].

There are conceivably many ways in which a robot might to refer to things in the world, but many such expressions are unsuitable in most human-robot dialogues. Consider the following set of examples:

1. "the location at position $(X = 5.56, Y = -3.92, \theta = 0.45)$"
2. "the mug left of the plate right of the mug left of the plate"
3. "Peter's office no. 200 at the end of the corridor on the third floor of the Acme Corp. building 3 in the Acme Corp. building complex, 47 Evergreen Terrace, Calisota, Planet Earth, (...)"
4. "the area"

These referring expressions are valid descriptions of their respective referents. Still they fail to achieve their *communicative goal*, which is to specify the right amount of information that the hearer needs to uniquely identify the referent. First of all, robots are good at measuring exact distances, humans are not. So the robot should employ qualitative descriptions that make use of the same concepts as a human-produced utterance would. Second, specifying a referent with respect to another referent that is only identifiable relative to the first one leads to infinite recursion instead of the communicative goal. Finally, the robot might have a vast knowledge about facts and entities in the world, but it should not always try to uniquely separate the referent from all entities in the world. At the same time, it is necessary to provide enough information to distinguish the intended referent from those entities in the world that potentially distract the hearer. The following expressions *might* serve as more appropriate variants of the previous examples:

1. "the kitchen around the corner"
2. "the red mug left of the china plate"
3. "Peter's office"
4. "the large hall on the first floor"

The fact that these *might* (or *might not!*) be successful referring expressions points to the importance of knowing what the given context in a situation is. This is especially the case for a mobile robot that operates and interacts in large-scale space. It is thus an important basis to endow the robot with a spatial representation that resembles the way humans conceive of their environment. But it is not enough; the robot must also be able to determine which entities in the world might act as *potential distractors* with respect to the hearer's knowledge.

In the following paragraphs we will show how our multi-layered conceptual spatial map provides a suitable knowledge base for Dale and Reiter's *incremental GRE algorithm*[20]. Furthermore, we will propose a method for a proper construction of the *context set* for successfully referring to entities in large-scale space.

The instances in the ontology are the *entities* of the world model. The conceptual hierarchy provides the taxonomical *type* information of the instances that the GRE algorithm requires. Furthermore, a number of concepts such as `Office`, `Kitchen`, `Corridor`, `Table`, etc. are marked as basic level categories, cf. [14] and [49]. The relations between instances are the *attributes* that the algorithm can use to further specify a referent. In terms of the Dale and Reiter algorithm, we currently use the following list of attributes, ordered by their preference: ⟨ type, topological inclusion, ownership, name ⟩.

Type

We represent an entity's type as the (asserted and inferred) concepts of the corresponding instance. Through ontological reasoning, we can retrieve an instance's most specific concept, its basic level category, and all the instances of a concept.

Topological inclusion

If the current context spans topological units at different hierarchical levels (cf. Figure 10) it is important to specify the intended referent with respect to the topological unit that contains the referent, e.g. when referring to "the kitchen on the 3rd floor", or "the table in the lab". In the ontology the transitive property `topoIncluded(X,Y)` and its inverse property `topoContains(Y,X)` represent topological positions of entities. By constructing a query to the reasoner that only returns those 'topological containers' of an entity that don't contain any other entities which in turn also contain the entity, we assure to only take into account direct topological inclusion despite the transitivity of the ontological properties.

Ownership

Areas in an environment are often referred to by identifying their owners, e.g. "Bob's office". In our ontology instances of `Area` can be related to a

`Person` instance via the `owns(X,y)/isOwnedBy(Y,X)` relation pair. People are instances of the ontological concept `Person`. The name of a person is represented as a string datatype property.

Name

As names are usually (locally) unique, e.g. "the Occam meeting room", or "office 120", they are definitely a highly discriminating attribute for the GRE task. However, names do not seem to be a preferred category for referring to rooms as they seldom contain more useful information than a generic NP + PP referring expression, e.g. "the meeting room on the first floor next to the large hall". On the contrary, such a generic referring expression might even bear additional useful information. Moreover, remembering the inherently artificial name for an entity might involve a higher cognitive load than processing the information encoded in a more generic referential description. For other scenarios though, such as an information desk agent at a hospital, or any other institution in which there is a specific naming scheme, such as e.g. encoding floor number and department, and numbering them in sequential order, the name feature can conceivably be placed in a higher-ranking position in the preference list. In our ontology names for areas are represented as a string datatype property.

Determining the appropriate contrast set

In order to successfully identify a referent it is important to determine a correct and appropriate contrast set. If the contrast set is chosen too small, the hearer might find it difficult to uniquely identify the intended referent with respect to his or her knowledge. If, on the other hand, a too large contrast set is assumed, the generated referring expression might violate *Grice's Maxims*, here the Maxim of Quality, in that it contains too much unnecessary information.

Since the contrast set is defined relative to a context set, the crucial task is hence to determine which part of the environment constitutes the current context. For one, the context needs to include the intended referent. The context must also include the current *referential anchor*, i.e. what is considered the current position of the two interlocutors. In the simple case, this referential anchor is the physical location where the dialogue takes place. But as a dialogue evolves, the referential anchor moves through space and time. Consider the following example dialogue:

> Person A: "Where is the exit?"
> Person B: "You first go down this corridor. Then you turn right. After a few steps you will see the big glass doors."
> Person A: "And the bus station? Is it to the left?"

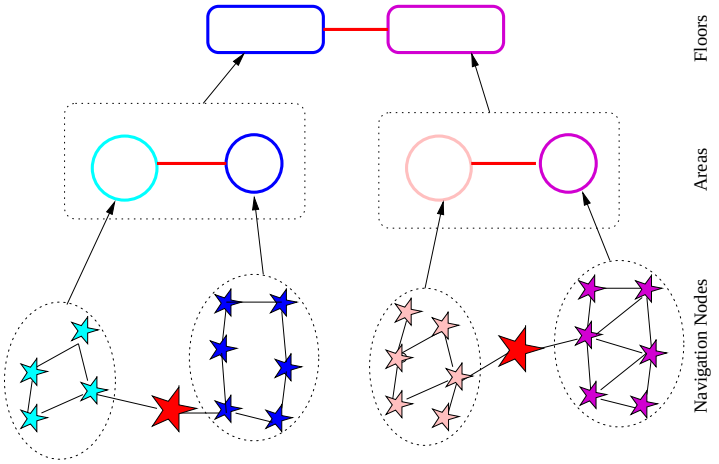

Fig. 10. A topology of places, rooms and floors. Stars depict navigation nodes that denote free and reachable space for our robotic system. The set of navigation nodes is partitioned into distinct spatial areas, such as e.g. rooms. Areas in turn can belong to a floors, which are on the next level of abstraction. Using *topology abstraction*, we construct an appropriate context set for the GRE task.

As can be seen, any utterance in such a collaborative dialogue is grounded in previously introduced discourse referents, both temporally and spatially.

Assuming the last mentioned discourse referent (or the physical location of a conversation) as the referential anchor, the question remains which other entities constitute the current discourse context. In other words: when referring to things, places, and actions in large-scale space, what possible distractors must one rule out in order for a referential description to be successful?

It is a widely accepted theory that humans tend to represent large-scale space in terms of topologies, rather than using exact measures. Following this view, we claim that the context for a dialogue situated in large-scale space can be determined on the basis of a topological representation.

Figure 10 shows a topological segmentation of an indoor space like the one used in our robotic system. The smallest unit in this representation is a graph-like structure of 'place nodes' (distinct places of approx. 1m in diameter that can be reached by the robot) and 'object nodes' (places from which objects are visible). These nodes are linked by edges that denote accessibility or visibility of one node from another. Through a number of processes, cf. Chapter 5, this graph is segmented into distinct areas, corresponding to e.g. rooms, corridors, or special regions within larger spatial structures. This segmentation into areas yields a first topological abstraction of space, in which the information about containment and reachability of its units is preserved, but metric distances don't play a role.

Require: r = intended referent; a = referential anchor
Initialize: $CONTEXT = \{\}$
$CONTEXT = CONTEXT \cup topologicalChildren(a) \cup \{a\}$
if $r \in CONTEXT$ **then**
  return $CONTEXT$
else
  Initialize topological containers to check: $CONTAINERS = \{a\}$
  while $r \notin CONTEXT$ **do**
    if $CONTAINERS = \{\}$ **then**
      return failure
    end if
    for each $c \in CONTAINERS$ **do**
      for each $p \in topologicalParents(c)$ **do**
        $CONTAINERS = CONTAINERS \cup \{p\}$
        $CONTEXT = CONTEXT \cup topologicalChildren(p)$
      end for
    end for
  end while
  return $CONTEXT$
end if

Fig. 11. Topological abstraction algorithm for context generation.

Our process of topology abstraction for determining the context set is depicted in Figure 11. It can be paraphrased as "Start with the referential anchor and check whether the intended referent is a member of the set of the referential anchor and its child nodes. If so, this set is the referential context. If not, construct the set of the referential anchor's parent nodes and their children, and check again. Repeat this procedure of topological abstraction until the intended referent is a member of this growing context set." With respect to the ontological representation of the conceptual spatial map, the function *topologicalChildren(x)* corresponds to a query that matches all instances i for which `topoContains(x,i)` applies. *topologicalChildren(x)* is defined as the set of instances i for which the direct, intransitive variant `direct-topoContains(x,i)` is true.

This means that if an object is located in the same room as the user and the robot, only local landmarks should be considered potential distractors. Likewise, if the robot is to produce a referring expression to a room on a different floor, all known entities inside the building will form the context. Using topological inclusion as the most preferred attribute (after type) will then essentially function as an early pruning of the hierarchically ordered context set.

5.4 Understanding references to elsewhere

A conversational robot should not only be able to produce meaningful speech, but also must be able to understand verbal descriptions given by its users. Similar to the challenge of generating referring expressions to entities in large-scale space, a dialogue system for a mobile robot will have to deal with its

user's referring expressions. The robot essentially needs to match a complex nominal construction with its internal knowledge base. Analogous to the task of generating referring expressions, an appropriate context against which to compare the referential description is crucial.

The first step is to translate the semantic interpretation of an utterance into a query to the ontology reasoner. This is being done through the architectural binding subarchitecture. A Logical Form is translated into a proxy structure, i.e. a number of proxies with well-defined 'concept' features, and labelled relations between them. The subarchitecture that holds the conceptual spatial mapping and reasoning functionality then reads the full relational proxy structure and converts the provided features into attribute-value pairs in the representation of the ontology. The relations are also reconstructed in the ontology language. Iteratively, an ontological description of the referring expression is generated. This description will then serve as a query to the reasoner. Upon such a query the reasoner will return all instances in its knowledge base that fulfill the criteria specified by features and relations.

In order to not overgenerate possible referents, the resulting query needs to be evaluated against a subset of the full knowledge base. The relevant subset is the discourse context. Following the approach described above, the query is first evaluated against the child nodes of the current discourse anchor and the discourse anchor itself. If the reasoner does not find any instances that satisfy the description, the context set is increased using the method of topological abstraction until at least one possible referent can be identified within the context.

6 Talking about things you can do

So far we have seen how bi-directionality figures in situated dialogue processing, when talking about things and space. We are using information about the situated context to predictively filter interpretations in speech recognition and incremental parsing, and later on use grounding of content to further zoom in on those interpretations which are contextually supported. Furthermore, going back and forth between what we know about the scene, the environment, and what we would like to say, we can generate referring expressions which are contextually appropriate.

But how about action? And where do we draw the line between action and interaction, in situated dialogue? We know from human communication that the very situatedness of such dialogues allows us to "overload" actions, giving them also a communicative function. If we want to provide similar possibilities for human-robot interaction, we need to consider how action planning, and dialogue planning, should interact. This takes the bi-directionality even further, from sharing content to sharing decision making processes.

Below we first discuss how action planning and situated dialogue processing interact at content-level, and then close with a discussion on the spectrum between planning for action and interaction.

Things you can do

When we talk about performing an action, it's about more than just the act itself. There are the objects involved, who is to do something. There is what we assume as outset, and what we expect to be the outcome of the action. And it is all these aspects that somehow need to match up in context – in the dialogue context as well as in the situated context.

What we do is to interpret the action further. We follow [43] and assign the action a so-called *event nucleus* . The event nucleus models the action as an event with temporal and causal dimensions. It models what needs to be done before this action could be performed, and what would result from performing this action – in as far as we can determine this linguistically, of course. In [37] we discussed detail how we formally model the event nucleus. Here we will focus on the basic intuitions behind these models, and discuss their interaction with the ideas about indexicality, intentionality, and information structure we discussed already earlier.

event nucleus

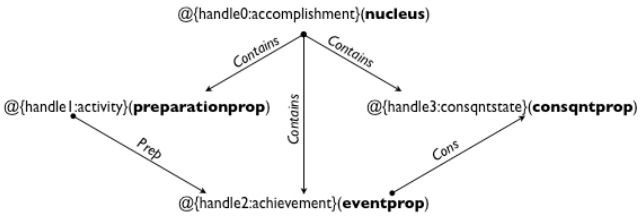

Fig. 12. Event nucleus, from [37]

Figure 12 provides the basic model we use for an event nucleus. Like all our other representations, it is a relational structure, with variables which are ontologically sorted. The sorts model a temporal ontology of types of events [43], whereas the variables themselves gives us the possibility to explicitly refer to these aspects of an event. We resolve any explicit or implicit temporal references to these variables, so that we can establish the temporal relations between events. After all, the ways in which events may be related need to match the order in which we have been talking about them. For example, consider we have a sequence of instructions like "First take the mug, then put it to the left the plate." This sequence already indicates that we first need to do the taking, then putting. Using the event nuclei associated with these actions, we can establish a much richer interpretation though. The consequence of the "take" action is that we have the mug – which is exactly what needs to be

the case for the "put" action. Together with the fact that these actions are to apply to the same object ("the mug" / "it"), we can associate "take" as a preparatory action to "put." Finally, we can establish that being "near the plate" is an intended state for the mug – the goal we are trying to achieve with the put action.

When we look at the nature of the content we provide for this intended state or goal, we again see there is a need to differentiate between the information status of content. We consider content which specifies the kernel (or the root) of the content for the intended state to be "new." Thus, if we have "put [the mug] to the left of the plate" we start out with considering the mug and the plate to be "old" and part of the identifiable common ground. The intended location of the mug, being left of the plate, is new. Before grounding it in binding working memory, we need to establish whether there is a suitable location given the action(s) we want to perform.

We thus follow in principle the same approach as we do for questions and assertions. Considering the command as intentional content, we provide it to motivation working memory with pointers to the indexical content for "mug" and "plate." Action planning in turn tries to establish a suitable location and plans for executing the action, as described in more detail in [12]. Bi-directionality between planning, motivation, binding working memory and dialogue processing subsequently provides feedback on the basis of which we post-filter out any potential linguistic interpretation for which we cannot find a suitable, situated plan, and provide communicative feedback on the command.

Between saying and doing

In situated dialogue, there is no clear division between action and interaction. As a simple example, consider reacting to a put-command like the one above. The robot could listen. Then provide an elaborate response to make clear it has understood: "Okay, let me put the mug to the left of the plate." And then do it. This is possible, but it stands in some contrast to what humans tend to do. There, you often see that someone says "fine" and moves the mug around. Or she even just does the latter. The action *is* the feedback, performing it gets overloaded with a communicative function of providing feedback.

What we can observe here is the next step in following out the bi-directionality hypothesis. For most of this chapter, we have considered bi-directionality in situating the linguistic content construed in a dialogue, be that during comprehension or production of utterances. Here we go one small step further, and take bi-directionality to the level of decision processes. We go from how to understand communication, to how to direct it in a given situation.

This builds forth on the bi-directional links we have already established, at an intentional level, between action planning and situated dialogue processing. We pointed out how dialogue meaning has two interrelated dimensions, namely an indexical and an intentional one. These dimensions determine how

meaning gets related to content in other modalities in a cognitive architecture. Notably, intentional content is used within the motivational subsystem of the architecture to establish plans for action.

It is on this basis we make a first attempt to establish a spectrum of intentions between planning for action and interaction. Dialogue planning can –naturally– generate dialogue moves from purely linguistic intentions, e.g. those arising at engagement level [54]. At the same time, it can handle intentions which originate outside the realm of dialogue. Typically, these intentions stem from an action plan created on the basis of a communicated intention. The intentions are always accompanied by indexical content, to which they apply. For example, an externally raised need to inquire with the user about a property of a particular object will be modeled as a "question" intention together with references to the object and the property. We can integrate such an external intention as a continuation in the dialogue context model, by using the links we maintain between dialogue referents, and the cross-modally connected indexical and intentional content they figure in. We provide several examples of how this works out in a cross-modal clarification context in [38].

7 Conclusions

We started out with the idea that, when it comes to processing, context is the first and last that situated dialogue is about. It drives what we want to say, it drives how we want to say or understand something, it drives how we want to communicate. The way context comes into play, we hypothesize, is through bi-directionality. Processes can mutually influence each other by sharing information – about the content they are forming, and the decisions they are proposing to make.

Throughout this chapter, we have discussed how that works out. We started out with just talking about ... talking. How we propose the linguistic aspects of processing situated dialogue can be set up, and how information about what is salient in the current context (be that the situated context or the dialogue one) can act as predictor for how to recognize speech or how to interpret an utterance, and how it naturally influences how we want to refer to aspects of the environment.

We then continued by gradually investigating more and more aspects of bi-directionality in situated dialogue processing. We looked at dialogue about the current visual scene. We saw how relating linguistic content to that scene requires distinguishing the indexical aspects of *what* you are talking about, from the intentional aspects of *how* you mean what you are saying – against the background of "old" and "new" information, relative to a common ground already formed. And the other way round, feeding back into the language system, we saw that alternative interpretations could be pruned in post-filtering on the basis of whether they could be grounded in the situated context.

Next, we moved perspective from the scene in front of us to the larger environment around us – where not everything we want to talk about can be seen all at once. This required to establish connections with further models for situation awareness, and ontological reasoning to establish how aspects of the environment could be referred to. Bi-directionality made it possible to complement linguistic content with further information necessary to resolve what it is someone is referring to.

Finally, we put bi-directionality into action. We looked at how communicated content about actions could be enriched with event information, so that we could establish when, how and what was to be done. We established bi-directional links between planning, motivation and dialogue processing. This provided the possibility to create plans on the basis of what was being talked about – and, again, to use information about possible and impossible plans to weed out irrelevant linguistic interpretations. But, bi-directionality between action and dialogue can mean even more than that. We saw that, if we consider action and interaction to be a spectrum, rather than two isolated forms of acting, we can also consider bi-directionality at the level of decision processes.

In retrospect, what do we contribute? We discussed here a system for dialogue in human-robot interaction with which the robot can understand more – what you talk about, and how that relates to the world. More fundamentally, we have argued how one can follow out the idea that context matters in situated dialogue processing. We can involve context, be that situated context or any type of deliberative context, by connecting processes in a bi-directional fashion. This way, processes canexchange information, complement and extend each others content, and generally help guide processing by focusing attention on content which makes sense a given situated context. To make this possible, we designed our processing to be incremental and multi-level, and use rich representations which can easily capture various levels of specificity and ambiguity of content. And, most importantly, we show what bi-directionality brings. For dialogue, involving context helps building the right interpretations, the right expressions – or at least, as we show (in various referenced publications) we can significantly improve performance over not using any context. But there is more. We understand more, say things better, because we have access to information outside of the language system. Information, like ontological reasoning, spatial organization, or planning, which we could not establish using purely linguistic means. Ultimately, when it comes to situated dialogue, what we really understand about dialogue – is about how we understand how dialogue receives its meaning from the environment we experience.

References

1. P.D. Allopenna, J.S. Magnuson, and M.K. Tanenhaus. Tracking the time course of spoken word recognition using eye movements: Evidence for continuous map-

ping models. *Journal of Memory and Language*, 38(4):419–439, 1998.
2. G.T. M. Altmann. Ambiguity in sentence processing. *Trends in Cognitive Sciences*, 2(4), 1988.
3. G.T.M. Altmann and Y. Kamide. Incremental interpretation at verbs: Restricting the domain of subsequent reference. *Cognition*, 73(3):247–264, 1999.
4. G.T.M. Altmann and Y. Kamide. Now you see it, now you don't: Mediating the mapping between language and the visual world. In J.M. Henderson and F. Ferreira, editors, *The Interface of Language, Vision, and Action: Eye Movements and The Visual World*, pages 347–386. Psychology Press, New York NY, 2004.
5. G.T.M. Altmann and M. Steedman. Interaction with context during human sentence processing. *Cognition*, 30(3):191–238, 1988.
6. Nicholas Asher and Alex Lascarides. *Logics of Conversation*. Cambridge University Press, 2003.
7. J. Baldridge and G.J.M. Kruijff. Coupling CCG and hybrid logic dependency semantics. In *Proc. ACL 2002*, pages 319–326, Philadelphia, PA, 2002.
8. J. Baldridge and G.J.M. Kruijff. Multi-modal combinatory categorial grammar. In *Proceedings of EACL'03*, Budapest, Hungary, 2003.
9. L.W. Barsalou. Perceptual symbol systems. *Behavioral & Brain Sciences*, 22:577–660, 1999.
10. John A. Bateman. Enabling technology for multilingual natural language generation: the kpml development environment. *Journal of Natural Language Engineering*, 3(1):15–55, 1997.
11. M.M. Botvinick, T.S. Braver, D.M. Barch, C.S. Carter, and J.D. Cohen. Conflict monitoring and cognitive control. *Psychological Review*, 108(3):624–652, 2001.
12. M. Brenner, N.A. Hawes, J. Kelleher, and J. Wyatt. Mediating between qualitative and quantitative representations for task-orientated human-robot interaction. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, 2007.
13. T. Brick and M. Scheutz. Incremental natural language processing for HRI. In *Proceeding of the ACM/IEEE international conference on Human-Robot Interaction (HRI'07)*, pages 263 – 270, 2007.
14. Roger Brown. How shall a thing be called? *Psychological Review*, 65(1):14–21, 1958.
15. J. Carroll and S. Oepen. High efficiency realization for a wide-coverage unification grammar. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP'05)*, pages 165–176, 2005.
16. C.G. Chambers, M.K. Tanenhaus, and J.S. Magnuson. Actions and affordances in syntactic ambiguity resolution. *Jnl. Experimental Psychology*, 30(3):687–696, 2004.
17. Michael Collins. Parameter estimation for statistical parsing models: theory and practice of distribution-free methods. In *New developments in parsing technology*, pages 19–55. Kluwer Academic Publishers, 2004.
18. S. Crain and M. Steedman. On not being led up the garden path: The use of context by the psychological syntax processor. In D. R. Dowty, L. Karttunen, and A. M. Zwicky, editors, *Natural language parsing: Psychological, computational, and theoretical perspectives*. Cambridge University Press, 1985.
19. D. Dahan and M.K. Tanenhaus. Continuous mapping from sound to meaning in spoke-language comprehension: Immediate effects of verb-based thematic

constraints. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 30(2):498–513, 2004.
20. Robert Dale and Ehud Reiter. Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263, 1995.
21. M. De Vega, D.A. Robertson, A.M. Glenberg, M.P. Kaschak, and M. Rinck. On doing two things at once: Temporal constraints on actions in language comprehension. *Memory and Cognition*, 32(7):1033–1043, 2004.
22. M.R. Endsley. Theoretical underpinnings of situation awareness: A critical review. In M. R. Endsley and D. J. Garland, editors, *Situation awareness analysis and measurement*. Lawrence Erlbaum, 2000.
23. J.A. Fodor. *The Modularity of Mind*. The MIT Press, Cambridge MA, 1983.
24. A.M. Glenberg. What memory is for. *Behavioral & Brain Sciences*, 20:1–55, 1997.
25. A.M. Glenberg and M.P. Kaschak. Grounding language in action. *Psychonomic Bulletin & Review*, 9(3):558–565, 2002.
26. P. Gorniak and D. Roy. Grounded semantic composition for visual scenes. *Journal of Artificial Intelligence Research*, 21:429–470, 2004.
27. P. Gorniak and D. Roy. Probabilistic grounding of situated speech using plan recognition and reference resolution. In *Proceedings of the Seventh International Conference on Multimodal Interfaces (ICMI 2005)*, 2005.
28. P. Gorniak and D. Roy. Situated language understanding as filtering perceived affordances. *Cognitive Science*, 31(2):197–231, 2007.
29. K. Hadelich and M.W. Crocker. Gaze alignment of interlocutors in conversational dialogues. In *Proc. 19th CUNY Conference on Human Sentence Processing*, New York, USA, 2006.
30. B. Hommel, K.R. Ridderinkhof, and J. Theeuwes. Cognitive control of attention and action: Issues and trends. *Psychological Research*, 66:215–219, 2002.
31. Y. Kamide, G.T.M. Altmann, and S.L. Haywood. The time-course of prediction in incremental sentence processing: Evidence from anticipatory eye-movements. *Jnl. Memory and Language*, 49(1):133–156, 2003.
32. J.D. Kelleher and G.J.M. Kruijff. Incremental generation of spatial referring expressions in situated dialog. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 1041–1048, 2006.
33. J.D. Kelleher, G.J.M. Kruijff, and F. Costello. Proximity in context: an empirically grounded computational model of proximity for processing topological spatial expressions. In *Proceedings of ACL/COLING 2006*, 2006.
34. P. Knoeferle and M.C. Crocker. The coordinated interplay of scene, utterance, and world knowledge: evidence from eye tracking. *Cognitive Science*, 2006.
35. Geert-Jan M. Kruijff. *A Categorial-Modal Logical Architecture of Informativity: Dependency Grammar Logic & Information Structure*. PhD thesis, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic, April 2001.
36. G.J.M. Kruijff. Context-sensitive utterance planning for ccg. In *Proceedings of the European Workshop on Natural Language Generation*, Aberdeen, Scotland, 2005.
37. G.J.M. Kruijff and M Brenner. Modelling spatio-temporal comprehension in situated human-robot dialogue as reasoning about intentions and plans. In *Pro-*

ceedings of the Symposium on Intentions in Intelligent Systems, AAAI Spring Symposium Series 2007, Stanford University, Palo Alto, CA, March 2007.
38. G.J.M. Kruijff, M. Brenner, and N.A. Hawes. Continual planning for cross-modal situated clarification in human-robot interaction. In *Proceedings of the 17th International Symposium on Robot and Human Interactive Communication (RO-MAN 2008)*, Munich, Germany, 2008.
39. G.J.M. Kruijff, H. Zender, P. Jensfelt, and H.I. Christensen. Situated dialogue and spatial organization: What, where... and why? *International Journal of Advanced Robotic Systems*, 4(2), 2007.
40. Benjamin J. Kuipers. *Representing Knowledge of Large-scale Space*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1977.
41. P. Lison and G.J.M. Kruijff. Salience-driven contextual priming of speech recognition for human-robot interaction. In *Proceedings of ECAI 2008*, Athens, Greece, 2008.
42. S.P. Liversedge and J.M. Findlay. Saccadic eye movements and cognition. *Trends in Cognitive Science*, 4(1):6–14, 2000.
43. M. Moens and M. Steedman. Temporal ontology and temporal reference. *Journal of Computational Linguistics*, 14:15–28, 1988.
44. R. K. Moore. Spoken language processing: piecing together the puzzle. *Speech Communication: Special Issue on Bridging the Gap Between Human and Automatic Speech Processing*, 49:418–435, 2007.
45. M.S. Nieuwland and J.J.A. Van Berkum. When peanuts fall in love: N400 evidence for the power of discourse. *Journal of Cognitive Neuroscience*, 18(7):1098–1111, 2006.
46. J.M. Novick, J.C. Trueswell, and S.L. Thompson-Schill. Cognitive control and parsing: Reexamining the role of Broca's area in sentence comprehension. *Cognitive, Affective, and Behavioral Neuroscience*, 5(3):263–281, 2005.
47. S. Oepen and J. Carroll. Ambiguity packing in constraint-based parsing: Practical results. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP 2000)*, pages 162–169, 2000.
48. M.J. Pickering and S. Garrod. Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, 27:169–225, 2004.
49. Eleanor Rosch. Principles of categorization. In E. Rosch and B. Lloyd, editors, *Cognition and Categorization*, pages 27–48. Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 1978.
50. D.K. Roy and E. Reiter. Connecting language to the world. *Artificial Intelligence*, 167(1-2):1–12, 2005.
51. M. Scheutz, K. Eberhard, and V. Andronache. A real-time robotic model of human reference resolution using visual constraints. *Connection Science Journal*, 16(3):145–167, 2004.
52. M. Schröder and J. Trouvain. The german text-to-speech synthesis system mary: A tool for research, development and teaching. *International Journal of Speech Technology*, 6:365–377, 2003.
53. H. Shi and T. Tenbrink. Telling rolland where to go: Hri dialogues on route navigation. In *Proceedings of the Workshop on Spatial Language and Dialogue (5th Workshop on Language and Space)*, Delmenhorst, Germany, 2005.
54. C. L. Sidner, C. Lee, C. Kidd, N. Lesh and C. Rich. Explorations in engagement for humans and robots. *Artificial Intelligence*, 166:140-164, 2005.

55. M.J. Spivey and M.K. Tanenhaus. Syntactic ambiguity resolution in discourse: Modeling the effects of referential context and lexical frequency. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 24:15211543, 1998.
56. M.J. Spivey, J.C. Trueswell, and M.K. Tanenhaus. Context effects in syntactic ambiguity resolution: discourse and semantic influences in parsing reduced relative clauses. *Canadian Journal of Experimental Psychology*, 47(2):276–309, 1993.
57. M. Steedman. *The Syntactic Process*. The MIT Press, Cambridge MA, 2000.
58. M. Steedman and I. Kruijff-Korbayová. Discourse and information structure. *Journal of Logic, Language and Information*, 12:249–259, 2003.
59. L. Steels. The symbol grounding problem has been solved. so what's next? In M. De Vega, G. Glennberg, and G. Graesser, editors, *Symbols, embodiment and meaning*. Academic Press, New Haven, 2008.
60. L. Steels. Semiotic dynamics for embodied agents. *IEEE Intelligent Systems*, 21:32–38, 3.
61. L. Steels and J-C. Baillie. Shared grounding of event descriptions by autonomous robots. *Robotics and Autonomous Systems*, 43(2-3):163–173, 2003.
62. M.K. Tanenhaus, J.K. Magnuson, D. Dahan, and G. Chambers. Eye movements and lexical access in spoken-language comprehension: Evaluating a linking hypothesis between fixations and linguistic processing. *Journal of Psycholinguistic Research*, 29(6):557–580, 2000.
63. M.K. Tanenhaus, M.J. Spivey-Knowlton, K.M. Eberhard, and J.C. Sedivy. Integration of visual and linguistic information in spoken language comprehension. *Science*, 268:1632–1634, 1995.
64. E. A. Topp, H. Hüttenrauch, H.I. Christensen, and K. Severinson Eklundh. Bringing together human and robotic environment representations – a pilot study. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, October 2006.
65. J.J.A. Van Berkum. Sentence comprehension in a wider discourse: Can we use erps to keep track of things? In M. Carreiras and C. Clifton Jr., editors, *The on-line study of sentence comprehension: Eyetracking, ERPs and beyond*, pages 229–270. Psychology Press, New York NY, 2004.
66. J.J.A. van Berkum, C.M. Brown, and P. Hagoort. Early referential context effects in sentence processing: Evidence from event-related brain potentials. *Journal of Memory and Language*, 41:147–182, 1999.
67. J.J.A. Van Berkum, C.M. Brown, P. Zwitserlood, V. Kooijman, and P Hagoort. Anticipating upcoming words in discourse: Evidence from erps and reading times. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, 31(3):443–467, 2005.
68. J.J.A. van Berkum, P. Hagoort, and C.M. Brown. Semantic integration in sentences and discourse: Evidence from the n400. *Journal of Cognitive Neuroscience*, 11(6):657–671, 1999.
69. J.J.A. Van Berkum, P. Zwitserlood, C.M. Brown, and P. Hagoort. When and how do listeners relate a sentence to the wider discourse? evidence from the n400 effect. *Cognitive Brain Research*, 17:701–718, 2003.
70. C. Van Petten, S. Coulson, S. Rubin, E. Plante, and M. Parks. Time course of word identification and semantic integration in spoken language. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 25(2):394–417, 1999.

71. M. White. Efficient realization of coordinate structures in combinatory categorial grammar. *Research on Language and Computation*, 4(1):3975, 2006.
72. M. White and J. Baldridge. Adapting chart realization to CCG. In *Proceedings of the Ninth European Workshop on Natural Language Generation*, Budapest, Hungary, 2003.
73. T. Winograd. A process model of language understanding. In R.C. Schank and K.M. Colby, editors, *Computer Models of Thought and Language*, pages 152–186. Freeman, New York, NY, 1973.
74. H. Zender, P. Jensfelt, Ó. Martínez Mozos, G.J.M. Kruijff, and W. Burgard. An integrated robotic system for spatial understanding and situated interaction in indoor environments. In *Proc. of AAAI-07*, pages 1584–1589, Vancouver, BC, Canada, July 2007.
75. H. Zender, P. Jensfelt, Ó. Martínez Mozos, G.J.M. Kruijff, and W. Burgard. Conceptual spatial representations for indoor mobile robots. *Robotics and Autonomous Systems*, 56(6), June 2008. Special Issue "From Sensors to Human Spatial Concepts".
76. Hendrik Zender and Geert-Jan M. Kruijff. Towards generating referring expressions in a mobile robot scenario. In *Language and Robots: Proceedings of the Symposium*, pages 101–106, Aveiro, Portugal, December 2007.

A Packing algorithm

A *packing* mechanism [47, 15] is used by the incremental parser to efficiently represent and manipulate logical forms accross the communication subarchitecture. A packed logical form [PLF] represents content similar across the different analyses of a given input as a single graph, using over- and underspecification of how different nodes can be connected to capture lexical and syntactic forms of ambiguity.

After each incremental step, the resulting set of logical forms is compacted into a single representation, which can then be directly manipulated by various processes, in order, for example, to prune unsupported interpretations. It can also be *unpacked*, ie. the original logical forms can be completely regenerated (this is done by traversing the packed structure).

The packed representations are made of two basic elements: *packing nodes* and *packing edges*. A packing node groups a set of nominals sharing identical properties and named relations under a particular subset of the logical forms. Packing edges are responsible for connecting the different packing nodes together, thus ensuring the correspondence between the packed structure and the set of logical forms it represents.

The packing of logical forms is performed in two main steps:

1. An initial PLF is first constructed on the basis of the set of logical forms (*Step 1* of algorithm 6). To this end, each logical form is traversed and its nominals are used to populate the packed structure.
2. The resulting structure is then compacted by merging particular substructures (*Step 2* of algorithm 6).

A.1 Example

The Figures 13-15 below exemplify a simple case of packing operation. The parsed utterance is "Take the ball to the left of the box". Two distinct readings can be derived, depending on the interpretation of the phrase "to the left of the box". In the first reading (LF_1 in the figure 13), the robot is asked to take the ball and put it to the left of the box - the phrase is thus seen as indicating the *direction* of the move. In the second reading (LF_2) however, "to the left of the box" indicates the *location* of the ball to take.

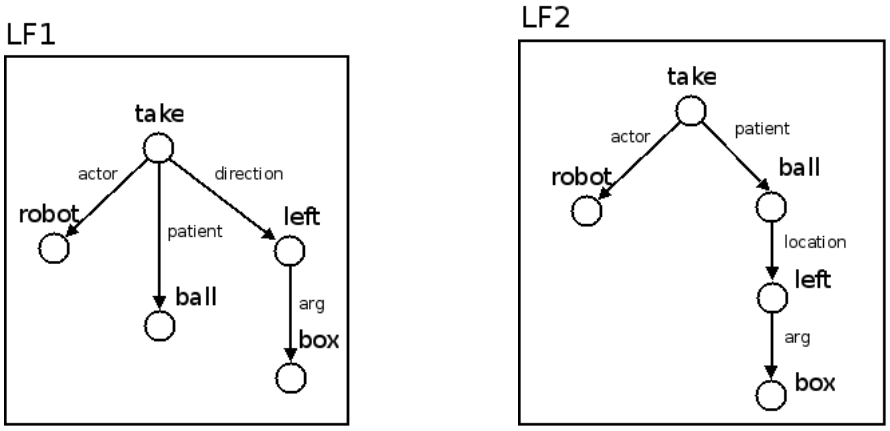

Fig. 13. The two initial logical forms LF_1 and LF_2 retrieved from parsing the utterance "Take the ball to the left of the box"

Figure 14 illustrates the application of the first step of the packing operation. A packing node - drawn in the figure as a square - is created for each nominal. A packing edge is constituted for each relation found in the logical forms. As shown in the figure, some packing edges are shared by both logical forms, whereas others are only evidenced in one of them. An example of the first case is the edge between "take" and "robot", which shared by the two logical forms LF_1 and LF_2. The edge between "take" and "left" illustrates the second case: it is only evidenced in LF_1.

In the example we present here, all packing edges have only one packing node target. In the general case however, several distinct targets can be specified within the same edge.

During the second step, the packed structure is compacted by merging packing nodes. The criteria to decide whether two packing nodes can be merged is the following: if (1) two packing nodes are connected by a packing edge, and if (2) the logical form identifiers for the head node, the edge and the target node are all identical, then the two packing nodes can be merged. For example, the packing node surrounding "take" and the one surrounding

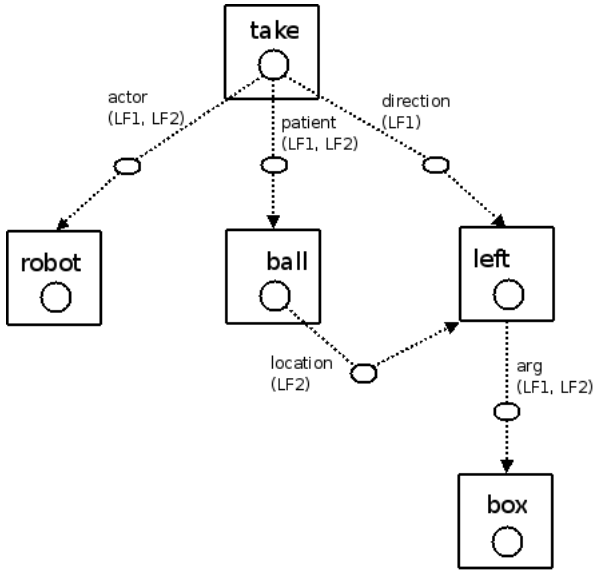

Fig. 14. The resulting packed logical form, before compacting

"robot" can be merged, since the two nodes and the edge between them are present both in LF_1 and LF_2.

The compacting operation is repeated until no more merges are possible. In our case, illustrated in the figure 15, we are left with two packing nodes, one rooted on the nominal "take", and one on "left".

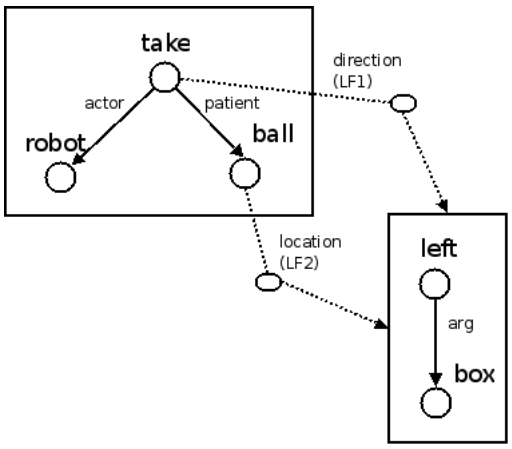

Fig. 15. The final packed logical form, after compacting

A.2 Data structures

We present below the informal specifications of the various data structures used to construct PLFs. See figure 17 for a graphical representation.

PackedLogicalForm:

- id: packed logical form identifier
- packingNodes: set of packing nodes
- root: root packing node

PackingNode:

- id: packing node identifier
- packedNominals: set of packed nominals
- lfIds: set of LF identifiers, enumerating the logical forms in which the nominals included in the packing node are present
- root: root nominal

PackedNominal:

- id: packed nominal identifier
- sort: ontological sort
- prop: logical proposition
- features: set of packed features
- relations: set of internal relations
- packingEdges: set of outgoing packing edges

PackedFeature:

- feature: name of the feature
- value: value of the feature
- lfIds: set of the LF identifiers, enumerating the logical forms in which the feature holds

PackingEdge:

- id: packing edge identifier
- head: head nominal
- mode: edge label
- packingNodeTargets: set of packing node targets

PackingNodeTarget:

- lfIds: set of LF identifiers, enumerating the logical forms in which the edge exists
- target: packing node targeted by the edge

Fig. 16. Data structures used to construct PLFs

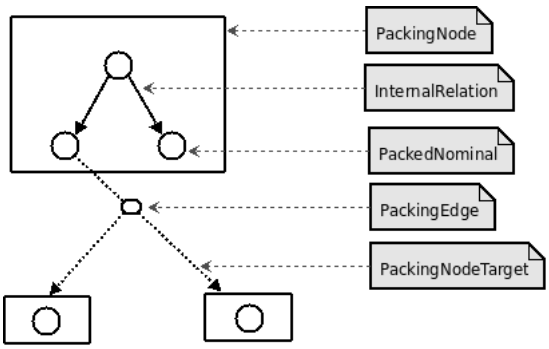

Fig. 17. Graphical representation of the data structures

A.3 Pseudo-code

We finally describe the details of the algorithms used in the packing mechanism we implemented.

Algorithm 1 : `Pack(LFs)` - Packing of a set of logical forms

Require: `LFs` is a set of logical forms (describing the same utterance)

```
% Step 0: Initialization
rootNominal ← ⟨ rootSort, 'root', ∅, ∅, ∅⟩
rootNode ← ⟨ {rootNominal}, ∅, rootNominal ⟩
packingNodes ← {rootNode}
PLF ← ⟨ packingNodes, rootNode ⟩

% Step 1: Construction of the packed logical form
for lf ∈ LFs do
   AddLFInformation(lf, PLF)
end for

% Step 2: Merge of the packed logical form
PLF = MergePackedLogicalForm(PLF)

return PLF
```

Algorithm 2 : `CreateNewNode(nom)` - using the information in `nom`, create (1) a new packing node, (2) a new packed nominal inside it and (3) new packing edges connected to the latter.

Require: A well-formed nominal `nom`

```
newEdges ← ∅
for every relation rel in rels(nom) do
   % A packing edge is defined with a head nominal, a mode ("edge label"), a set of packing
   node targets, and a set of logical form identifiers
   newEdge ← ⟨ head(rel), mode(rel), {target(rel)}, {lfId(nom)}⟩,
   newEdges ← newPackingEdges ∪ {newEdge}
end for

% A packing nominal comprises an ontological sort, a logical proposition, a set of features,
a set of internal relations, and a set of outgoing packing edges
newNom ← ⟨ sort(nom), prop(nom), feats(nom), ∅, newEdges ⟩

% A packing node is a triple comprising a set of packing nominals, a set of LF identifiers,
and a reference to the root nominal
newPackingNode ← ⟨{newNom},{lfId(nom)}, newNom⟩

return newPackingNode
```

Algorithm 3 : `AddLFInformation(lf, PLF)` - Add the information contained in `lf` to the packed logical form.

Require: `lf` is a well-formed logical form

for every nominal `nom` in `nominals(lf)` **do**

if there is no packing node in `PLF` which encapsulates a packed nominal with the ontological sort `sort(nom)` and the logical proposition `prop(nom)` **then**

% *We create a new packing node and its related substructures*
`newPackingNode` ← `CreateNewPackingNode(nom)`

% *We add the packing node to the PLF structure*
`packingNodes(PLF)` ← `packingNodes(PLF)` ∪ {`newPackingNode`}

else
% *We update the existing nominal and its dependent edges*
let `pNom` = the packed nominal with `sort(nom)` and `prop(nom)`
let `pNode` = the packing node encapsulating `pNom`

`pNode` ← `IntegrateNominalToPackingNode(nom, pNode)`
end if

if `nom` is the root nominal in `lf` **then**
% *We establish a connection between the root node and the current one*

let `packingNode` = the packing node which encapsulates `nom` in `PLF`

Add a packing edge between `root(PLF)` and `packingNode`
`lfIds(root(PLF))` = `lfIds(root(PLF))` ∪ {`id(lf)`}
end if

end for

return `PLF`

Algorithm 4 : `IntegrateNominalToPackingNode(nom, pNode)` - integrate the information contained in `nom` to the existing packing node `pNode`

Require: A well-formed nominal `nom`
Require: A well formed packing node `pNode` which already encapsulates a nominal with the same ontological sort and logical proposition as `nom`

let `pNom` = the nominal encapsulated in `pNode`

for every relation `rel` in `rels(nom)` **do**
  if ∃ `edge` ∈ `edges(pNom)` where `mode(rel)` = `mode(edge)` **then**
    % *If there is already a packing edge with same mode, add one packing node target and the LF identifier*
    `targets(edge)`←`targets(edge)` ∪ {`target(rel)`}
    `lfIds(edge)` ← `lfIds(edge)` ∪ {`lfId(nom)`}
  else
    % *Else, we create a new packing edge*
    `newEdge` ← ⟨ `head(rel)`, `mode(rel)`, {`target(rel)`}, {`lfId(nom)`}⟩
    `edges(pNom)` ← `edges(pNom)` ∪ {`newEdge`}
  end if
end for

% *Update the features in the nominal, and the LF identifiers in the packing node*
`feats(pNom)` ← `feats(pNom)` ∪ {`feats(nom)`}
`lfIds(pNode)` ← `lfIds(pNode)` ∪ {`lfId(nom)`}

return `pNode`

Algorithm 5 : `MergePackedLogicalForm(PLF)` - compact the PLF representation by merging nominals

Require: `PLF` a well formed packed logical form

while there are packing nodes in `PLF` which can be merged **do**
  for every packing node `packingNode` ∈ `PLF` **do**
    for every nominal `nom` ∈ `nominals(packingNode)` **do**
      for every edge `edge` ∈ `edges(nom)` **do**
        if `edge` has only one packing node target **then**

          let LFS_{head} = set of logical forms identifiers in `packingNode`
          let LFS_{edge} = set of logical forms identifiers in `edge`
          let LFS_{target} = set of logical forms identifiers in `target(edge)`

          if $LFS_{head} = LFS_{edge} = LFS_{target}$ **then**
            % *If the set of logical forms shared by the two packing nodes (and the packing edge between them) is identical, then they can be merged in one packing node*

            let `targetNom` = the head nominal of `target(edge)`

            Merge `packingNode` and `targetNom` into a single packing node

            Transform `edge` into an internal relation (in the merged packing node) between `nom` and `targetNom`

          end if
        end if
      end for
    end for
  end for
end while
return PLF

A

Appendix Heading

Your text comes here. Separate text sections

A.1 Section Heading

A.1.1 Subsection Heading

Subsubsection Heading

Paragraph Heading

Subparagraph Heading. as required.

Table A.1. Please write your table caption here

first	second	third
number	number	number
number	number	number

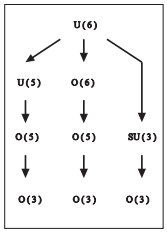

Fig. A.1. Please write your figure caption here

Index