Incremental Construction of Robust but Deep Semantic Representations for Use in Responsive Dialogue Systems

Andreas PELDSZUS[1] *David SCHLANGEN*[2]
(1) Department of Linguistics, Potsdam University, Germany
(2) Faculty of Linguistics and Literary Studies, Bielefeld University, Germany
peldszus@uni-potsdam.de, david.schlangen@uni-bielefeld.de

ABSTRACT
It is widely acknowledged that current dialogue systems are held back by a lack of flexibility, both in their turn-taking model (typically, allowing only a strict back-and-forth between user and system) and in their interpretation capabilities (typically, restricted to slot filling). We have developed a component for NLU that attempts to address both of these challenges, by a) constructing *robust but deep meaning representations* that support a range of further user intention determination techniques from inference / reasoning-based ones to ones based on more basic structures, and b) constructing these representations *incrementally* and hence providing semantic information on which system reactions can be based concurrently to the ongoing user utterance. The approach is based on an existing semantic representation formalism, Robust Minimal Recursion Semantics, which we have modified to suit incremental construction. We present the modifications, our implementation, and discuss applications within a dialogue system context, showing that the approach indeed promises to meet the requirements for more flexibility.

KEYWORDS: Incremental Processing, Semantics Construction, Dialogue Systems, Dialogue, Natural Language Understanding, Spoken Language.

1 Introduction

To advance beyond the domains that currently are covered by spoken dialogue systems—acquisition of information for database queries—into more collaborative domains such as explored in the pioneering work by Allen *et al.* (Allen et al., 1995; Ferguson and Allen, 1998), progress in at least three areas will be required. First, as is well known in the study of conversational behaviour, collaboration extends to the contruction of the dialogue contributions themselves (Clark, 1996), something that is precluded by the strict back-and-forth turn-taking model of current dialogue systems. Second, less clearly scripted (and scriptable) domains require deeper interpretation of contributions (Allen et al., 2005). Finally, the problems of coverage that the deep representation-based BDI (beliefs, desires, intentions) approach ran into (discussed for example in (Jurafsky, 2003)) suggest that complementary principled reasoning mechanisms such as recently explored in the field of Artificial Intelligence (e.g., (Domingos et al., 2006; Zettlemoyer et al., 2009)) will need to be applied.

The first of these areas, more flexible turn-taking through incremental processing, has received much attention recently (Aist et al., 2007; Skantze and Schlangen, 2009; Baumann et al., 2009; Buß and Schlangen, 2010; Skantze and Hjalmarsson, 2010; DeVault et al., 2011), and has shown improvements in the perceived naturalness of the resulting systems. However, the implemented systems still followed the simpler information-transaction model, and used shallow meaning representations. In this paper, we present our work towards connecting attempts to improve such temporal flexibilty with the use of deeper representations.

We base our work on an existing semantic representation formalism (RMRS, (Copestake, 2006)) that is designed to capture as much or as little semantic information as could be recovered from an input. Hence, even though it has previously only been used in the much different application of monological information extraction (Schäfer, 2007), this formalism fits well the requirement of robustness for spoken dialogue systems, where input to interpretation may be deviant from standard syntax both in actual fact, through speech disfluency, as well as practically, because of speech recognition problems. We modify this formalism so that it is suitable for incremental semantic construction (Section 4). We present our implementation that combines incremental semantic construction with top-down incremental parsing (Section 5), and describe how we have applied it in applications that begin to make use of hybrid reasoning techniques (Section 6).

2 Related Work

Incremental semantic construction has been tackled occasionally in the literature. As mentioned in the introduction, many of the extant approaches represent meaning in domain-specific semantic frames which are then filled incrementally, often using shallow probabilistic models (see e.g. (Sagae et al., 2009; Heintze et al., 2010)). As our focus in this work is on providing deep representations, the more relevant work is that taking a theoretical, linguistic perspective.

The Dynamic Syntax (DS) grammar formalism (Kempson et al., 2001), for example, presents an incremental parsing-directed approach to semantic construction. Instead of using syntactic trees, the grammar here is mainly specified by lexical actions that directly build a propositional tree. DS offers sophisticated syntax-semantic theorizing and models ellipsis and anaphora phenomena. Recently DS has been applied to sequences of dialog utterances (Gargett et al., 2009); a first outline of a dialog system implementing DS has been presented by Purver et al. (2011). However, coming from a theoretical linguistics background, the approach still centers around notions of grammaticality, whereas in practical applications, robustness is perhaps the more important notion. Moreover, DS is somewhat ‘monolithic’, making it hard to substitute, say, the grammar, the lexicon or the semantic composition by an alternative while keeping the rest. We aim to remain more theory-neutral towards grammar and (base-language) semantics in our approach.

PTT (Poesio and Traum, 1997; Poesio and Rieser, 2010) is another relevant approach; here, use is made of tree adjoining grammars and accompanying semantic rule patterns for construction of representations. The main focus of that theory however seems on the incremental construction of discourse structure and on how that enriches the semantics. The semantic construction is only worked out for small examples, and, to our knowledge, not implemented yet.

The work presented here shares many concerns with that of (Allen et al., 2005). However, again in our re-use of existing formalisms and a more standard base-grammar, we strive for more theory-independence. Moreover, our approaches differ in how underspecification is used (where we allow underspecification of predicate-argument relations) and in the way the methods are applied, as will be described below in Section 6.

In (Peldszus et al., 2012), we have described an application of the work presented in the present paper, where the incremental representations generated by the parser were evaluated against the current environment, providing feedback to the parser about which derivation to expand first, thus improving its accuracy. In (Kennington and Schlangen, 2012), we have used the representations as input to a probabilistic reasoning model. While those papers focus on particular applications within a dialogue processing context (see also the brief discussion below in Section 6), the current paper focusses on the representation and construction in itself, properly developing it out of the tradition of principled syntax / semantics interfaces.

3 Using Underspecification to Represent Meaning Increments

Underspecification of semantic representations was introduced in the 1990s as a means to capture ambiguities—mostly those arising out of quantifier scope—efficiently by letting the syntax/semantics interface leave those semantic aspects unspecified which syntax cannot decide. A variety of formalisms was proposed (*inter alia*, (Reyle, 1993; Bos, 1996; Pinkal, 1996; Deemter and Peters, 1996)) which all realised the same basic idea of letting the grammar produce *descriptions* of so-called *base language* logical formulae. The descriptions themselves are formulae, but in a special, tailor-made language; their models then are the base language formulae that do the normal job of representing meaning. In the discourse theory SDRT (Asher and Lascarides, 2003), an underspecification formalism formed the basis of the interface between compositional semantics and discourse implications; this theory was used in (Schlangen, 2003; Schlangen and Lascarides, 2003) to deal with what could be seen as a restricted case of the current phenomenon, namely intentionally non-sentential dialogue utterances.

Here, we use for the representation of meaning increments (that is, the contributions of new words and syntactic constructions) as well as for the resulting logical forms the formalism *Robust Minimal Recursion Semantics* (Copestake, 2006). In this section, we will first introduce the basic features of this formulism and then argue that it is suitable for describing meaning increments in a principled and well-founded way.

Background: (Robust) Minimal Recursion Semantics Minimal Recursion Semantics (MRS) was originally constructed for semantic underspecification (of scope and other phenomena) (Copestake et al., 2005) and then—with the name gaining an additional “R”—adapted to serve the purposes of semantic representation in heterogeneous situations where the results of shallow and deep semantic parsers need to be integrated into a common representation (Copestake, 2006). To this end almost all relational information contained in a logical form is factored into smaller predications, so that, depending on what's required, a fully specified semantic representation can be made less specific by removing some of its predications or by cutting connections between them, or a shallower semantic representation can be expanded monotonically by enrichring it with further statements or by connecting predications.

As in all semantic underspecification formalisms, representing scopal readings is achieved in RMRS by first splitting the logical form into scope-taking and scope-bearing parts (*elementary predications* in RMRS terminology) and then describing the set of admitted embeddings through *scope constraint* statements. The resulting semantic representation is flat, i.e. it can be represented as a list of elementary predications and scope constraints. Removing a scope constraint expands the set of admitted embeddings, while adding one restricts it.

The distinguishing feature of RMRS in comparison with MRS is its ability to underspecify predicate-argument structure. A predicate expression $pred(x, y, z)$ is decomposed further into the "key" predication and *argument relations* which explicitly express which kind of arguments the predication has. Removing such an explicit argument relation hence underspecifies an argument of a predicate expression, while adding one specifies it. Predicates can thus be introduced into the composition process with different specificity: a predicate can be fully specified with fixed arity, i.e. with a defined number of argument positions, and all its arguments given, as in $pred(x, y, z)$; a predicate can have a fixed arity, but leave some argument positions open, which in our simplified illustration could be represented as e.g. $pred(x, y, ?)$; it could be introduced without fixed arity, as illustrated in $pred(?, \dots)$; and arguments can be introduced without knowing which predicates they are arguments of, as in $?(x, y)$. It is even possible to a bind an argument to a predicate without knowing exactly which argument position it is supposed to fill.

RMRS has several other useful features. First, the underlying representation language is a first order logic with generalized quantifiers, event semantics and sortal variables – common formal tools of semantic representation. Also, the representations can be related to each other in a transparent way: Two RMRS structures can then be tested for subsumption (Copestake, 2007a), in order to see whether one structure is a less specific variant of the other. If one subsumes the other, the difference between both can be formulated as an RMRS containing all those statements that could monotonically enrich the less specific one to yield the more specific structure. Furthermore, it is semantically well-founded. A model-theoretic semantics for the language of RMRS has for example been given in Koller and Lascarides (2009). Finally, RMRS is used widely, for example in the LinGO English Resource Grammar (Flickinger, 2000), or in the "Heart of Gold"-architecture (Schäfer, 2007) as a common semantic representation. The RASP parser (Briscoe and Carroll, 2002) is one example of a shallow (yet non-incremental) parser with an RMRS interface.

Representing Meaning Increments with Underspecification Why do we see these underspecification techniques as useful for representing the meaning of an ongoing utterance and of the increments that add to it?

First and foremost, a crucial requirement for incremental semantic representation is that it facilitates extensibility in a technically straightforward manner; as explained above, this is fulfilled even by the standard formulation of RMRS. Ongoing utterances then may raise expectations about how they might be concluded. Even if an utterance is yet incomplete, we can have expectations about what kinds of words may come and fill the necessary but yet open positions. On the other hand, even for a syntactically potentially complete utterance, we have to remain open for supplemental phrases or appositions. A semantic formalism is required that can adequately represent such constraints on possible extensions. However, an incoming word does not necessarily have to determine how it relates to the existing utterance. Sometimes lexical increments might add to the ongoing utterance without already making explicit in which way they connect, as e.g. with the attachment of prepositional phrases when adding the preposition. The semantic formalism should be able to underspecify these connections accordingly and to specify them when evidence is available. Finally, we want to represent the incremental state not only of perfectly planned utterances, but of spontanous natural speech. The semantic formalism should thus ideally offer devices to robustly cope with those phenomena.

RMRS meets these representational desiderata. As an example, Figure 1 shows a growing logical form in a scope-less flat first-order logic.

Words	Predicates
den	?(e,?,x,. . .) def(x)
den winkel	?(e,?,x,. . .) def(x) bracket(x)
den winkel in	?(e,?,x,. . .) def(x) bracket(x) in(x,y)
den winkel in der	?(e,?,x,. . .) def(x) bracket(x) in(x,y) def(y)
den winkel in der dritten	?(e,?,x,. . .) def(x) bracket(x) in(x,y) def(y) third(y)
den winkel in der dritten reihe	?(e,?,x,. . .) def(x) bracket(x) in(x,y) def(y) third(y) row(y)
den winkel in der dritten reihe nehmen	take(e,c,x) def(x) bracket(x) in(x,y) def(y) third(y) row(y)

Table 1: Example of logical forms (flattened into scope-less first-order base-language formulae for convenience) incrementally growing for the utterance ‘den winkel in der dritten reihe nehmen’ (*take the bracket in the third row*)

With every incoming word, the logical form is monotonically enriched, either by adding lexical predicates, by connecting predicates via variable identifications or by specifying underspecified positions. For convenience, we restrict the example to the NP-attachment case; also it should be noted that the base-language logical forms shown in the table correspond to highly factored structures in the RMRS description language. Each of these RMRS structures representing a certain state of the ongoing utterances can be conceived as describing in a well-defined way an infinite set of logical forms that all share a common part (namely, the common lexical prefix).

To give an impression of what the full utterance would look like in the RMRS description language, see example (1). This representation shows elementary predications, argument relations and scope constraints. For a more detailed technical description of the RMRS formalism, we refer the interested reader to (Copestake, 2006).

(1) $\ell_7{:}a_7{:}\_\mathrm{take}(e_7)$, $\mathsf{ARG}_1(a_7,c)$, $\mathsf{ARG}_2(a_7,x_1)$,
$\ell_1{:}a_1{:}\mathrm{def}()$, $\mathsf{BV}(a_1,x_1)$, $\mathsf{RSTR}(a_1,h_1)$, $\mathsf{BODY}(a_1,h_2)$, $h_1 =_q \ell_2$,
$\ell_2{:}a_2{:}\_\mathrm{bracket}(x_1)$,
$\ell_3{:}a_3{:}\_\mathrm{in}(e_3)$, $\mathsf{ARG}_1(a_3,x_1)$, $\mathsf{ARG}_2(a_3,x_4)$,
$\ell_4{:}a_4{:}\mathrm{def}()$, $\mathsf{BV}(a_4,x_4)$, $\mathsf{RSTR}(a_4,h_3)$, $\mathsf{BODY}(a_4,h_4)$, $h_3 =_q \ell_5$,
$\ell_5{:}a_5{:}\_\mathrm{third}(e_5)$, $\mathsf{ARG}_1(a_5,x_4)$,
$\ell_5{:}a_6{:}\_\mathrm{row}(x_4)$

The discussion above has shown how RMRS meets the requirements for representing the content

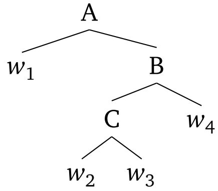

Figure 1: Abstract example tree.

of as-yet (potentially) unfinished utterances. The content that each minimal continuation of an utterance brings with it, *i.e.*, the semantic increment, can be represented in the very same way. We will demonstrate this in the next section.

4 Incremental Semantics Construction in *i*RMRS

After briefly reviewing how standard RMRS deals with semantic construction, we will describe in this section our modifications that enable incremental construction.

Background: Semantics Constructions in RMRS According to Gottlob Frege's principle of compositionality, the meaning of an expression is a result of the meaning of its parts and the way of combination (Frege, 1897). Syntax-driven semantic construction has typically conceived this principle as follows: The decomposition of an expression into parts is determined by the syntactic tree. Each terminal node is assigned a lexical semantics. The rule expanding a non-terminal node identifies the method of combination of that node's daughters' semantics. In order to compute the meaning of the whole expression, the tree is interpreted bottom-up, inside-outside. As an example, consider the abstract tree in Figure 1. To determine the meaning of the whole string, the combination operation determined by node type A has to be applied to the meaning of the first word and to the intermediate meaning result of node B. This is formally represented in Example (2).

(2) $[[w_1 \ldots w_4]] = OP_A([[w_1]], OP_B(OP_C([[w_2]], [[w_3]]), [[w_4]]))$

The semantic algebra proposed for RMRS (Copestake, 2007b) workes pretty much in this way. Syntactic structures are related to operations of semantic combination, as e.g. scopal combination (equivalent to function application) and intersective combination (equivalent to predicate modification). Those operations combine the two RMRS structures "under construction" by joining their list of elementary predications, argument relations and scope constraints. Additionally, as defined by the applied semantic operation, equations between variables of the joined parts relate their semantic representations. Which variables are equated is determined by the so-called "hooks" and "slots", where one structure (the argument) "hooks" into an open "slot" of the other (the functor) to make it semantically more complete. Thus, the semantic representation can grow monotonically at each combinatory step by simply adding predicates, argument relations and scope constraints and by equating variables according to the hook and slot pair.

Formally, hook and slot are triples $[\ell{:}a{:}x]$ consisting of a label for scope underspecification, an anchor for predicate-argument underspecification and an index variable representing the semantic head of the structure. An RMRS can have multiple slots allowing different equations of its variables. To make the subsequent discussion easier, we will call an RMRS under construction *saturated,* if no open slot is left. Statements identifying variables can either enter the structure explicitly, or be immediately resolved with one variable being substituted by the other. We will call an an RMRS under construction *reduced*, if all equalities are resolved. An RMRS under construction corresponds to a normal RMRS, if it is saturated and reduced.

(Copestake, 2007b) describes this process of semantic construction by tree interpretation for two settings: for a lexicalized setting, where the lexical entries already bring a large part of the slot with them (according to their subcategorization scheme), and for a non-lexicalist setting, where the lexical entries are rather generic and all slots are introduced by rule-semantics. We will focus on the latter setting for the rest of this paper.

In both cases, the slots of an RMRS under construction are organised as a bag of named slots. Open slots can be randomly accessed, i.e. independently of the order of slot introduction, if the semantic combination operation identifies it by its name. However, there is the restriction that a slot with a certain name can only exist once in the bag of open slots.

Adaptations for Incremental Construction In an incremental setting, a proper semantic representation is desired for every single state of growth of the syntactic tree. However this is not easily achieved if the order of semantic combination is kept parallel to a bottom-up traversal of the syntactic tree, as assumed in the RMRS semantic algebra. Consider our abstract example in Figure 1 again and suppose that in the current state of the ongoing utterance only the first two words have been uttered. Following a bottom-up combination order, no proper semantic representation could be given for the utterance so far, because the semantic operation associated e.g. with node C requires an argument that is not yet there. One possible solution to this dilemma would be to assign an adequate underspecified semantics to every projected node, in our example for the nodes of w_3 and w_4. Then, the whole tree could be interpreted as described, yielding a proper semantic representation of the ongoing utterance. Unfortunately, the tree will change with the next incoming word, former projected nodes may be specified, new projected nodes may enter the tree. Consequently, the whole process of finding underspecified semantics for open nodes would start again, and not only the new parts of the tree, but the *whole* tree would be required to be interpreted. Because of these two problems, the need to find underspecified semantics for projected nodes and the need for re-interpretation of already existing parts of the tree, we argue that the bottom-up interpretation in this classic form is not adequate for incremental semantic construction.

For our purposes, it is more elegant to proceed with semantic combination in synchronisation with the syntactic expansion of the tree, i.e. in a top-down left-to-right fashion, circumventing the two problems. Consider example (3): The bracketing already makes obvious that the semantic combination is now left-linearized. Every combinatory step yields a semantic representation that can serve as a starting point for the following combinatory step.

(3) $[[w_1 \dots w_4]] = ((((([[\mathrm{A}]] \lhd [[w_1]]) \lhd [[\mathrm{B}]]) \lhd [[\mathrm{C}]]) \lhd [[w_2]]) \lhd [[w_3]]) \lhd [[w_4]]$

However, in order to define the combination operation signified here with the $\lhd$ symbol, an adjustment to the slot structure of RMRS is required. Left-recursive rules can introduce multiple slots of the same sort before they are filled, which is not allowed in the classic (R)MRS semantic algebra, where only one named slot of each sort can be open at a time. We thus organize the slots as a stack of unnamed slots, where multiple slots of the same sort can be stored, but only the one on top can be accessed. We then define the basic combination operation $\lhd$ equivalent to forward function composition (as in standard lambda calculus, or in CCG (Steedman, 2000)). The basic idea here is the following: When an argument fills the top slot of the functor, the argument's stack of slots itself is pushed onto the functor's stack of slots, so that in the resulting structure the (former) argument's slots get priority over the remaining functor slots. A more formal specification of this operation and our adaptions to RMRS is provided in the appendix for the interested reader.

The stack of semantic slots is thus kept in synchronisation to the stack of syntactic slots. Parallel to the production of syntactic derivations, as the tree is expanded top-down left-to-right, semantic macros are activated for each syntactic rule, composing the contribution of the new increment. If input tokens are matched by the parser, a corresponding generic lexical semantics for that token is added, derived from its lemma and the basic semantic type (individual, event, or underspecified denotations) as determined by its POS tag. This allows for a monotonic semantics construction process that proceeds in lockstep with the syntactic analysis.

A worked example We can now present a small but worked example of the incremental semantic construction process. We directly realize the rule-to-rule hypothesis by annotating every syntactic rule of a toy grammar with a corresponding semantic rule.

For that purpose we first define a few very abstract semantic macros which we call *basic slot manipulators*. None of them contributes an elementary predication or argument relation to the overall representation. Instead they manipulate the slot structure. When the *pop*-combinator $[-]$ is slotted in some RMRS, it merely consumes the top slot without adding anything else; this is required for handling epsilon rules. The *pass*-combinator $[\circ]$ simply restores the slot it has consumed and has no further effect on the representation. It can be seen as a null-semantics and will be the default for any rule without designated rule semantics. The remaining combinators have in common that they add a new slot, besides maintaining the one they have filled. The *equal*-combinator $[=]$ exactly copies the slot, thereby equating labels, anchors and indices, the *plusℓ*-combinator $[+\ell]$ equates labels and indices, and the *plus*-combinator $[+]$ only equates indices. We will use them for modification and adjunction. Note that they are antisymmetric and can be defined for reverse order, e.g. $[+.]$, depending on the linear order of a node's daughter. Other abstract combinators are possible, but these are the ones we will use frequently.

With those basic semantic macros at hand, we can then define more specific semantic macros to represent the meaning of a syntatic rule. Sometimes rule semantics are just basic macros, in many other cases they add argument relations or even grammar-specific predicates. A specification of some the semantic macros can be found in the appendix.

We will not go very much into detail about the actual execution of all those semantic operations. However, we want to give an impression of the incremental derivation. As an example, consider the utterance "nimm den winkel in. . . ", a simpler version of the example already introduced in Table 1. Its syntactic tree is shown in Figure 2. In Figure 3, we show how the sequence of semantic combinations unfolds corresponding to the syntactic expansion of the tree. We hide the bracketing for convenience and understand the forward slotfilling combination $\lhd$ to be left-associative. The first line shows the ongoing utterance, the second the decomposition according to the syntactic tree, and the third line shows the more or less abstract semantic macros that are successively combined. Remember that all of those macros are RMRSs under construction and that each combinatory step results in a new one, i.e. a proper semantic structure representing the current state of the process is available any time. Also note that we have a clear and transparent description of the semantic increment.

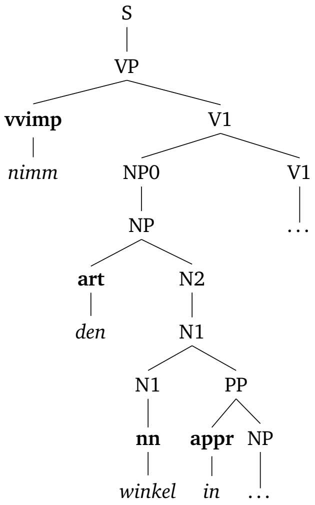

Figure 2: Incremental syntactic derivation of a simple example sentence.

$\llbracket \textit{nimm}\ldots \rrbracket$
$= \llbracket \mathrm{S} \rightarrow \mathrm{VP} \rrbracket \lhd \llbracket \mathrm{VP} \rightarrow \mathrm{vvimp,V1} \rrbracket \lhd \llbracket \textit{nimm} \rrbracket$
$= [\,\circ\,] \lhd [\mathrm{Arg1}] \lhd [\mathrm{adr}] \lhd [=] \lhd \llbracket \textit{nimm} \rrbracket$

$\llbracket \textit{nimm den}\ldots \rrbracket = \llbracket \textit{nimm}\ldots \rrbracket \lhd \llbracket \textit{den} \rrbracket$
$= \ldots \lhd \llbracket \mathrm{V1} \rightarrow \mathrm{NP0,V1} \rrbracket \lhd \llbracket \mathrm{NP0} \rightarrow \mathrm{NP} \rrbracket \lhd \llbracket \mathrm{NP} \rightarrow \mathrm{art,N2} \rrbracket \lhd \llbracket \textit{den} \rrbracket$
$= \ldots \lhd [\mathrm{Arg2}] \lhd [\,\circ\,] \lhd [\mathrm{Q}] \lhd \llbracket \textit{den} \rrbracket$

$\llbracket \textit{nimm den winkel}\ldots \rrbracket = \llbracket \textit{nimm den}\ \ldots \rrbracket \lhd \llbracket \textit{winkel} \rrbracket$
$= \ldots \lhd \llbracket \mathrm{N2} \rightarrow \mathrm{N1} \rrbracket \lhd \llbracket \mathrm{N1} \rightarrow \mathrm{N1,PP} \rrbracket \lhd \llbracket \mathrm{N1} \rightarrow \mathrm{nn} \rrbracket \lhd \llbracket \textit{winkel} \rrbracket$
$= \ldots \lhd [\,\circ\,] \lhd [\dotplus] \lhd [\,\circ\,] \lhd \llbracket \textit{winkel} \rrbracket$

$\llbracket \textit{nimm den winkel in}\ldots \rrbracket = \llbracket \textit{nimm den winkel}\ldots \rrbracket \lhd \llbracket \textit{in} \rrbracket$
$= \ldots \lhd \llbracket \mathrm{PP} \rightarrow \mathrm{appr,NP} \rrbracket \lhd \llbracket \textit{in} \rrbracket$
$= \ldots \lhd [\mathrm{PP}] \lhd \llbracket \textit{in} \rrbracket$

Figure 3: Incremental semantic derivation of a simple example sentence.

5 Implementation in InproTK$_{iRMRS}$

We have implemented this method of semantic construction in the *incremental processing toolkit* (InproTK) (Baumann and Schlangen, 2012), an open-source framework for developing incremental dialogue systems. It realises the abstract model for incremental processing described in (Schlangen and Skantze, 2009), where this processing is conceptualised as consisting of modules that exchange *incremental units*, minimal bits of input and output, that are chained together to form larger units and also linked across modules to allow for revisions.

As mentioned in the introduction, we aim to be theory-neutral if possible, in order to maintain flexibility in case of new emerging linguistic ressources, or newly adopted domains etc. We thus chose to connect more or less "standard" components: A probabilistic top-down parser, a context-free grammar and a common and well-understood semantic representation.

Parser Our parser is a basic version of the approach endorsed by Roark (2001), who presents a strategy for incremental probabilistic top-down parsing and shows that it can compete with high-coverage bottom-up parsers. One of the reasons Roark gives for choosing a top-down approach is that it enables fully left-connected derivations, where at every processing step new increments directly find their place in the existing structure. This monotonically enriched structure can then serve as a context for incremental language understanding, as the author claims, although this part, which we take up here, is not further developed by Roark (2001). The search-space is reduced by using beam search. Due to probabilistic weighing and grammar-transformations, as e.g. the left factorization of the rules to delay certain structural decisions, left recursion poses no direct threat in such an approach. Roark discusses several different techniques for refining his results, such as e.g. including conditioning functions that manipulate a derivation probability on the basis of local linguistic and lexical information; we have for now only implemented the basic strategy. However, in order to cope with spontanous speech and ASR errors, we added three robust lexical operations: *Insertions* consume the current token without matching it to the top stack item. *Deletions* can "consume" a requested but actually non-existent token. *Repairs* adjust unknown tokens to the requested token. These robust operations have strong penalties on the probability to make sure they will survive in the derivation only in critical situations.

Grammar We developed a small grammar (30 rules) covering a "core syntax" of constructions, tailored towards a particular corpus of instructions in task oriented dialogue. These utterances were collected in a Wizard-of-Oz study in the Pentomino puzzle piece domain (which has been used before for example by (Fernández and Schlangen, 2007; Schlangen et al., 2009)). This grammar is hand-written, with weights set according to intuition and manual semantic annotations. With it, we were able to produce semantic representations for a corpus of over 1600 spontaneous dialogue utterances – both for their manually transcriptions as well as for automatic transcription.[1] Although this grammar serves us as a good starting point to experiment with incrementally constructed semantic representations, this obviously is an area for future work. Fortunately, the grammar could easily be substituted by any other context-free grammar, as e.g. one that is induced from a treebank.

Semantic increments in an IU network In the InproTK every increment is represented as an *incremental unit* (IU), which to connected to other units in a network that grows with

[1] We have however only indirectly rated their quality via their interpretability in context (see Peldszus et al., 2012), and so cannot yet give exact numbers for parser perfomance on its own here.

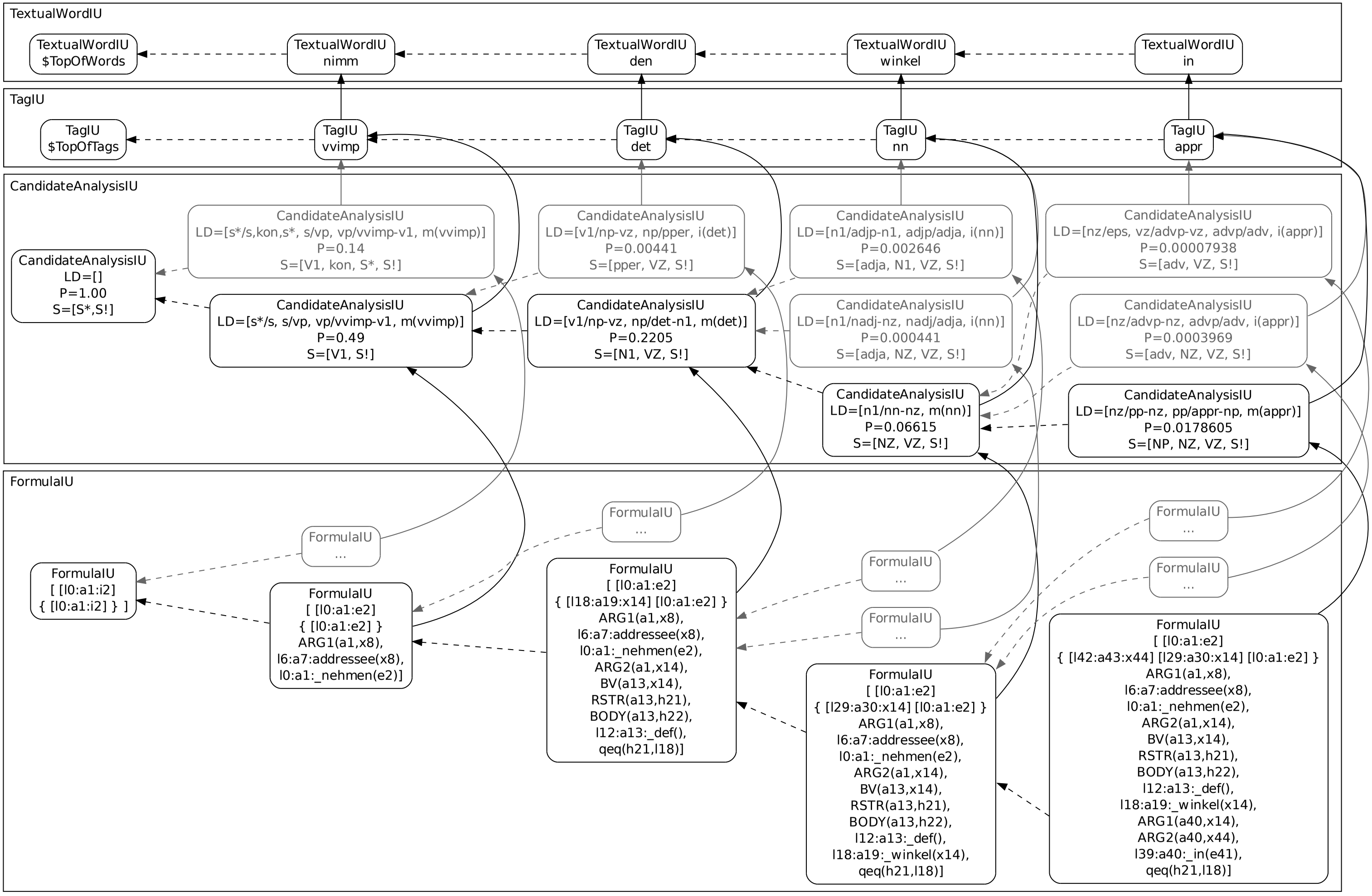
TextualWordIU
TextualWordIU $TopOfWords
TextualWordIU nimm
TextualWordIU den
TextualWordIU winkel
TextualWordIU in
TagIU
TagIU $TopOfTags
TagIU vvimp
TagIU det
TagIU nn
TagIU appr
CandidateAnalysisIU
CandidateAnalysisIU LD=[] P=1.00 S=[S*,S!]
CandidateAnalysisIU LD=[s*/s,kon,s*, s/vp, vp/vvimp-v1, m(vvimp)] P=0.14 S=[V1, kon, S*, S!]
CandidateAnalysisIU LD=[s*/s, s/vp, vp/vvimp-v1, m(vvimp)] P=0.49 S=[V1, S!]
CandidateAnalysisIU LD=[v1/np-vz, np/pper, i(det)] P=0.00441 S=[pper, VZ, S!]
CandidateAnalysisIU LD=[v1/np-vz, np/det-n1, m(det)] P=0.2205 S=[N1, VZ, S!]
CandidateAnalysisIU LD=[n1/adjp-n1, adjp/adja, i(nn)] P=0.002646 S=[adja, N1, VZ, S!]
CandidateAnalysisIU LD=[n1/nadj-nz, nadj/adja, i(nn)] P=0.000441 S=[adja, NZ, VZ, S!]
CandidateAnalysisIU LD=[n1/nn-nz, m(nn)] P=0.06615 S=[NZ, VZ, S!]
CandidateAnalysisIU LD=[nz/eps, vz/advp-vz, advp/adv, i(appr)] P=0.00007938 S=[adv, VZ, S!]
CandidateAnalysisIU LD=[nz/advp-nz, advp/adv, i(appr)] P=0.0003969 S=[adv, NZ, VZ, S!]
CandidateAnalysisIU LD=[nz/pp-nz, pp/appr-np, m(appr)] P=0.0178605 S=[NP, NZ, VZ, S!]
FormulaIU
FormulaIU [[l0:a1:i2] { [l0:a1:i2] }]
FormulaIU ...
FormulaIU [[l0:a1:e2] { [l0:a1:e2] } ARG1(a1,x8), l6:a7:addressee(x8), l0:a1:_nehmen(e2)]
FormulaIU ...
FormulaIU [[l0:a1:e2] { [l18:a19:x14] [l0:a1:e2] } ARG1(a1,x8), l6:a7:addressee(x8), l0:a1:_nehmen(e2), ARG2(a1,x14), BV(a13,x14), RSTR(a13,h21), BODY(a13,h22), l12:a13:_def(), qeq(h21,l18)]
FormulaIU ...
FormulaIU ...
FormulaIU [[l0:a1:e2] { [l29:a30:x14] [l0:a1:e2] } ARG1(a1,x8), l6:a7:addressee(x8), l0:a1:_nehmen(e2), ARG2(a1,x14), BV(a13,x14), RSTR(a13,h21), BODY(a13,h22), l12:a13:_def(), l18:a19:_winkel(x14), qeq(h21,l18)]
FormulaIU ...
FormulaIU ...
FormulaIU [[l0:a1:e2] { [l42:a43:x44] [l29:a30:x14] [l0:a1:e2] } ARG1(a1,x8), l6:a7:addressee(x8), l0:a1:_nehmen(e2), ARG2(a1,x14), BV(a13,x14), RSTR(a13,h21), BODY(a13,h22), l12:a13:_def(), l18:a19:_winkel(x14), ARG1(a40,x14), ARG2(a40,x44), l39:a40:_in(e41), qeq(h21,l18)]

Figure 4: An example network of incremental units, including the levels of words, POS-tags, syntactic derivations and logical forms.

succeeding processing stages and newly incoming input. An illustration of such a network for our example sentence is shown in Figure 4. In our implementation, we assume IUs of the different processing stages: at the level of words (resulting from ASR or text input), of part-of-speech tags, of syntactic derivations and semantic representations. The different levels are arranged from top to bottom and unfold in time from left to right. Each level contains IUs of its type, shown as rounded boxes in the Figure. Dashed arrows link an IU to its predecessor on the same level. Multiple IUs sharing the same predecessor can be regarded as alternatives. Solid arrows indicate which information from a previous level an IU is grounded in (based on); here, every semantic IU is grounded in a syntactic IU, every syntactic IU in a POS-tag-IU, and so on.

Syntactic derivations ("CandidateAnalysisIUs") are represented by three features: a list of the last parser actions of the derivation (LD), with rule expansions or (robust) lexical matches; the derivation probability (P); and the remaining stack (S), where S* is the grammar's start symbol and S! an explicit end-of-input marker. (To keep the Figure small, we artificially reduced the beam size and cut off alternatives paths, shown in grey.) Semantic representations ("FormulaIUs") are shown by the resulting RMRS. Notice that, apart from the end-of-input marker, the stack of semantic slots (in curly brackets) is always synchronized with the parser's stack.

6 Using *i*RMRS for Dialogue Processing

Schlangen and Skantze (2009) have observed that incremental processing offers the potential not only to speed up responses of dialogue systems, but also to improve their processing, since "higher-level" results based on partial results from lower levels can be fed back to those lower levels and influence their further processing. In (Peldszus et al., 2012), we have shown how this can be realised using the framework detailed in the current paper. In that work, the semantic representations connected to each syntactic derivation—or, more specifically, those of referring expressions—were evaluated against the current dialogue environment in terms of their satisfiability. The result of this test was used as a signal that contributed to the weight of the current derivation and thus it had influence on the order of syntactic expansion. We could show a clear improvement of this processing style. In that work, we made use of the fact that our meaning representations can easily be simplified in a principled way, and used a simple rule-based reference resolution component.

In (Kennington and Schlangen, 2012), we then used our RMRS representations as input for a hybrid, probabilistic logic-based interpretation system, and showed that using these representations as input improved performance compared to a "words-only" model (as is often used in such statistical NLU work, as e.g. in (DeVault et al., 2011; Heintze et al., 2010)). In that work, we could directly transfer *i*RMRS predications into statements in the knowledge base over which the probabilistic reasoning was defined, where those statements could be combined freely with predicates describing the situational context.

These applications were made possible by the property of the framework described here to produce meaning representations at each input increment, which moreover can easily be transferred into shallower variants with loss of information (Peldszus et al., 2012) or into other first-order representation formats (Kennington and Schlangen, 2012). In current work, we are exploring more direct uses of the representations for discourse reasoning. The aim is to formulate discourse expectations, for example not only about the fact that an answer is expected after a question, but also that some aspects of its form can be predicted (for example, in an

NP question, the answer will, possibly implicitly, re-use the main predicate of the question) as *i*RMRS formulae. Annotating the syntactic top-down predictions with such discourse-based content expectations, and making use of the calculus for RMRS subsumption tests mentioned above (Copestake, 2007a), we have a principle mechanism at hand to let such expectations guide interpretation. We are currently evaluating whether this potential advantage translates into a practical improvement.

We cite these applications here as support for our claim that the representation format and construction mechanism described here can form the basis for a variety of work towards more flexible dialogue systems.

Conclusion

We have presented our approach to creating meaning representations for spontaneous spoken utterances. This approach is based on an existing, well-studied representation formalism, RMRS (Copestake, 2006) that can represent various levels of semantic detail, from shallow to deep; we have extended this to suit incremental construction, and so can create meaning representations in lockstep with incremental speech recognition (such as described in (Baumann et al., 2009)) feeding input to incremental parsing. We have described our implementation of such a parser and semantic construction component within an open framework for incremental processing, and have sketched some of the applications that we have already used this in.

While already fully functional within our domain, it remains for future work to extend coverage towards more general coverage. Here we plan to investigate using treebank resources to induce grammar and, the more challenging part, semantic macros for the grammar rules. Also, as sketched above, we are currently investigating using properties of the representation formalism (such as allowing for subsumption tests) to model top-down discourse expectations and evaluate their use for dialogue processing. After the first steps described in the present paper, the aim for that ongoing work is to bring us yet closer towards the goal of increasing both temporal and content-related flexibility of spoken dialogue systems.

Appendix

Definition 1 (Elementary predications). An elementary predication $\ell{:}a{:}R(i)$ consists of a predicate symbol R, a label ℓ, an anchor a, and (optionally) as characteristic variable i an ordinary object language variable (i.e. an individual x, an event e or an underspecified index u).

Definition 2 (Argument relations). An argument relation $\mathsf{REL}(a,v)$ consists of an argument relation symbol REL from a finite set $\{\mathsf{ARG}_{\mathbb{N}}, \mathsf{BV}, \mathsf{RSTR}, \mathsf{BODY}, \mathsf{LEFT}_{i/\ell}, \mathsf{RIGHT}_{i/\ell}\}$, an anchor a, and exactly one argument v, which is either an ordinary object language variable $x/e/u$ or a hole h.

Definition 3 (RMRS structure under construction with a stack of slots). An RMRS structure under construction is a 6-tuple $\langle GT, H, S, R, C, E \rangle$,

- with GT the global top hole h_0,
- with H the hook $[\ell{:}a{:}i]$, consisting of the local top label ℓ, the anchor a and the index i,
- with S the stack of slots of the form $[\ell_n{:}a_n{:}i_n]$,
- with R the bag of elementary predications and argument relations,
- with C the bag of scope constraints and
- with E the set of variable equalities.

Definition 4 (Forward slot filling combination)**.** Given two RMRSs, one being the functor $rmrs_f = \langle GT_f, H_f, S_f, R_f, C_f, E_f \rangle$ with the top slot $\mathrm{top}(S_f) = [\ell_f{:}a_f{:}i_f]$ and one being the argument $rmrs_a = \langle GT_a, H_a, S_a, R_a, C_a, E_a \rangle$ with its hook $H_a = [\ell_a{:}a_a{:}i_a]$, the slot filling combination $rmrs_f \lhd rmrs_a$ yields an RMRS $rmrs = \langle GT, H, S, R, C, E \rangle$, s.t.

- $GT = GT_f = GT_a$
- $H = H_f$
- $S = \text{merge-stacks}(S_a, \mathrm{pop}(S_f))$[2]
- $R = R_f \cup R_a$
- $C = C_f \cup C_a$
- $E = E_f \cup E_a \cup \{\ell_f = \ell_a, a_f = a_a, i_f = i_a\}$

A simple example of slot filling combination in a lexicalist setting

$$\begin{array}{lcl}
[[\mathit{take}\ldots]] & = & [\ell_1{:}a_1{:}e_1]\ \{\,[\ell_3{:}a_3{:}x_3]\,\} \\
 & & \ell_1{:}a_1{:}\text{\_take\_v()},\ \mathsf{ARG}_1(a_1,x_2),\ \mathsf{ARG}_2(a_1,x_3) \\
 & & \ell_2{:}a_2{:}\text{adressee}(x_2) \\
 & & \\
[[\mathit{the}]] & = & [\ell_4{:}a_4{:}x_4]\ \{\,[\ell_5{:}a_5{:}x_4]\,\} \\
 & & \ell_4{:}a_4{:}\text{\_the\_q()},\ \mathsf{BV}(a_4,x_4),\ \mathsf{RSTR}(a_4,h_1),\ \mathsf{BODY}(a_4,h_2),\ h_1 =_q \ell_5 \\
 & & \\
[[\mathit{take}\ldots]] \lhd [[\mathit{the}]] & = & [\ell_1{:}a_1{:}e_1]\ \{\,[\ell_5{:}a_5{:}x_4]\,\} \\
 & & \ell_1{:}a_1{:}\text{\_take\_v()},\ \mathsf{ARG}_1(a_1,x_2),\ \mathsf{ARG}_2(a_1,x_3) \\
 & & \ell_2{:}a_2{:}\text{adressee}(x_2) \\
 & & \ell_4{:}a_4{:}\text{\_the\_q()},\ \mathsf{BV}(a_4,x_4),\ \mathsf{RSTR}(a_4,h_1),\ \mathsf{BODY}(a_4,h_2),\ h_1 =_q \ell_5 \\
 & & \ell_3 = \ell_4,\ a_3 = a_4,\ x_3 = x_4 \\
 & & \\
 & = & [\ell_1{:}a_1{:}e_1]\ \{\,[\ell_5{:}a_5{:}x_3]\,\} \\
 & & \ell_1{:}a_1{:}\text{\_take\_v()},\ \mathsf{ARG}_1(a_1,x_2),\ \mathsf{ARG}_2(a_1,x_3) \\
 & & \ell_2{:}a_2{:}\text{adressee}(x_2) \\
 & & \ell_3{:}a_3{:}\text{\_the\_q()},\ \mathsf{BV}(a_3,x_3),\ \mathsf{RSTR}(a_3,h_1),\ \mathsf{BODY}(a_3,h_2),\ h_1 =_q \ell_5
\end{array}$$

Some basic slotfilling combinators

$$\begin{array}{lcl}
[-] & = & [\ell{:}a{:}u]\ \{\,\}\ . \\
[\,\circ\,] & = & [\ell{:}a{:}u]\ \{\,[\ell{:}a{:}u]\,\}\ . \\
[=] & = & [\ell{:}a{:}u]\ \{\,[\ell{:}a{:}u][\ell{:}a{:}u]\,\}\ . \\
[+] & = & [\ell{:}a{:}u]\ \{\,[\ell_1{:}a_1{:}u][\ell{:}a{:}u]\,\}\ . \\
[+.\,] & = & [\ell{:}a{:}u]\ \{\,[\ell{:}a{:}u][\ell_1{:}a_1{:}u]\,\}\ . \\
[+\ell] & = & [\ell{:}a{:}u]\ \{\,[\ell{:}a_1{:}u][\ell{:}a{:}u]\,\}\ .
\end{array}$$

Some of the semantic macros used in the grammar

$$\begin{array}{lcll}
[\text{Arg1}] & = & [\ell{:}a{:}u] & \{\,[\ell_1{:}a_1{:}x_1][\ell{:}a{:}u]\,\}\ \mathsf{ARG}_1(a,x_1) \\
[\text{Arg2}] & = & [\ell{:}a{:}u] & \{\,[\ell_1{:}a_1{:}x_1][\ell{:}a{:}u]\,\}\ \mathsf{ARG}_2(a,x_1) \\
[\text{Arg3}] & = & [\ell{:}a{:}u] & \{\,[\ell_1{:}a_1{:}x_1][\ell{:}a{:}u]\,\}\ \mathsf{ARG}_3(a,x_1) \\
[\text{adr}] & = & [\ell{:}a{:}x] & \{\,\}\ \ell{:}a{:}\text{addressee}(x) \\
[\text{Q}] & = & [\ell{:}a{:}x] & \{\,[\ell{:}a{:}e_1][\ell_2{:}a_2{:}x]\,\}\ \mathsf{BV}(a,x),\ \mathsf{RSTR}(a,h_1),\ \mathsf{BODY}(a,h_2),\ h_1 =_q \ell_2 \\
[\text{PP}] & = & [\ell{:}a{:}u] & \{\,[\ell_1{:}a_1{:}e_1][\ell_2{:}a_2{:}x_2]\,\}\ \mathsf{ARG}_1(a_1,u),\ \mathsf{ARG}_2(a_1,x_2) \\
[\text{Adj}] & = & [\ell{:}a{:}x] & \{\,[\ell{:}a_1{:}e_1]\,\}\ \mathsf{ARG}_1(a_1,x) \\
[\text{Adv}] & = & [\ell{:}a{:}u] & \{\,[\ell_1{:}a_1{:}e_1]\,\}\ \mathsf{ARG}_1(a_1,u) \\
[\text{Conj}] & = & [\ell{:}a{:}u] & \{\,[\ell_1{:}a_1{:}u_1][\ell{:}a{:}u][\ell_3{:}a_3{:}u_3]\,\}\ \mathsf{LEFT}_i(a,u_1),\ \mathsf{LEFT}_\ell(a,h_1), \\
 & & & \mathsf{RIGHT}_i(a,u_3),\ \mathsf{RIGHT}_\ell(a,h_3),\ h_1 =_q \ell_1,\ h_3 =_q \ell_3
\end{array}$$

[2] The function merge-stacks(A,B) is understoods as yielding a new stack $S = \{a_1,\ldots,a_n,b_1,\ldots,b_n\}$ s.t. $a_1 = \mathrm{top}(A)$ and $b_1 = \mathrm{top}(B)$.

References

Aist, G., Allen, J., Campana, E., Gallo, C. G., Stoness, S., Swift, M., and Tanenhaus, M. K. (2007). Incremental understanding in human-computer dialogue and experimental evidence for advantages over nonincremental methods. In *Proceedings of Decalog 2007, the 11th International Workshop on the Semantics and Pragmatics of Dialogue*, Trento, Italy.

Allen, J., Manshadi, M., Dzikovska, M. O., and Swift, M. (2005). Deep linguistic processing for spoken dialogue systems. In *Proceedings of the 5th Workshop on Deep Linguistic Processing*, pages 49–56, Morristown, NJ, USA. Association for Computational Linguistics.

Allen, J. F., Shubert, L. K., Ferguson, G., Heeman, P., Hwang, C. H., Kato, T., Light, M., Martin, N. G., Miller, B. W., Poesio, M., and Traum, D. R. (1995). The TRAINS project: A case study in building a conversational planning agent. *Journal of Experimental and Theoretical AI*, 7:7–48.

Asher, N. and Lascarides, A. (2003). *Logics of Conversation*. Cambridge University Press, Cambridge.

Baumann, T., Atterer, M., and Schlangen, D. (2009). Assessing and improving the performance of speech recognition for incremental systems. In *Proceedings of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT) 2009 Conference*, Boulder, Colorado, USA.

Baumann, T. and Schlangen, D. (2012). The InproTK 2012 release. In *Proceedings of the NAACL-HLT Workshop on Future directions and needs in the Spoken Dialog Community: Tools and Data (SDCTD 2012)*, pages 29–32, Montreal, Canada. ACL.

Bos, J. (1996). Predicate logic unplugged. In Dekker, P. and Stokhof, M., editors, *Proc. of the Tenth Amsterdam Colloquium*, pages 133–143, Amsterdam, Netherlands.

Briscoe, T. and Carroll, J. (2002). Robust accurate statistical annotation of general text. In *Proc. LREC*, pages 1499–1504.

Buß, O. and Schlangen, D. (2010). Modelling sub-utterance phenomena in spoken dialogue systems. In *Proceedings of the 14th International Workshop on the Semantics and Pragmatics of Dialogue (Pozdial 2010)*, pages 33–41, Poznan, Poland.

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

Copestake, A. (2006). Robust minimal recursion semantics. Technical report, Cambridge Computer Lab. Unpublished draft.

Copestake, A. (2007a). *Invited Talk:* Applying Robust Semantics. In *Proceedings of PACLING 2007 – 10th Conference of the Pacific Association for Computational Linguistics*, pages 1–12, Melbourne.

Copestake, A. (2007b). Semantic composition with (robust) minimal recursion semantics. In *Proceedings of the Workshop on Deep Linguistic Processing*, DeepLP '07, pages 73–80, Stroudsburg, PA, USA. Association for Computational Linguistics.

Copestake, A., Flickinger, D., Pollard, C., and Sag, I. (2005). Minimal recursion semantics: An introduction. *Research on Language & Computation*, 3:281–332.

Deemter, K. v. and Peters, S., editors (1996). *Semantic Ambiguity and Underspecification*. CSLI, Stanford.

DeVault, D., Sagae, K., and Traum, D. (2011). Incremental Interpretation and Prediction of Utterance Meaning for Interactive Dialogue. *Dialogue and Discourse*, 2(1):143–170.

Domingos, P., Kok, S., Poon, H., Richardson, M., and Singla, P. (2006). Unifying logical and statistical AI. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, pages 2–7, Boston, USA.

Ferguson, G. and Allen, J. F. (1998). TRIPS: An Integrated Intelligent Problem-Solving Assistant. In *Proceedings of the National Conference on Artificial Intelligence*, pages 567–573.

Fernández, R. and Schlangen, D. (2007). Referring under restricted interactivity conditions. In Keizer, S., Bunt, H., and Paek, T., editors, *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, pages 136–139, Antwerp, Belgium.

Flickinger, D. (2000). On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28.

Frege, G. (1897). Begriffsschrift – Eine der arithmetischen nachbildete Formalsprache des reines Denkens. Jena, Germany.

Gargett, A., Gregoromichelaki, E., Kempson, R., Purver, M., and Sato, Y. (2009). Grammar resources for modelling dialogue dynamically. *Cognitive Neurodynamics*, 3:347–363.

Heintze, S., Baumann, T., and Schlangen, D. (2010). Comparing local and sequential models for statistical incremental natural language understanding. In *Proc. SIGdial 2010*, pages 9–16, Tokyo, Japan.

Jurafsky, D. (2003). Pragmatics and computational linguistics. In Horn, L. R. and Wards, G., editors, *Handbook of Pragmatics*. Blackwell, Oxford, UK.

Kempson, R., Meyer-Viol, W., and Gabbay, D. (2001). *Dynamic syntax: the flow of language understanding*. Blackwell.

Kennington, C. and Schlangen, D. (2012). Markov logic networks for situated incremental natural language understanding. In *Proceedings of the SIGdial Conference 2012*, Seoul, South Korea.

Koller, A. and Lascarides, A. (2009). A logic of semantic representations for shallow parsing. In *Proc. EACL 2009*, pages 451–459, Athens.

Peldszus, A., Buß, O., Baumann, T., and Schlangen, D. (2012). Joint satisfaction of syntactic and pragmatic constraints improves incremental spoken language understanding. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 514–523, Avignon, France. Association for Computational Linguistics.

Pinkal, M. (1996). Radical underspecification. In Dekker, P., Groenendijk, J., and Stokhoff, M., editors, *Proceedings of the 10th Amsterdam Colloqium*, Amsterdam.

Poesio, M. and Rieser, H. (2010). Completions, coordination, and alignment in dialogue. *Dialogue and Discourse*, 1(1):1–89.

Poesio, M. and Traum, D. (1997). Conversational actions and discourse situations. *Computational Intelligence*, 13(3):309–347.

Purver, M., Eshghi, A., and Hough, J. (2011). Incremental semantic construction in a dialogue system. In Bos, J. and Pulman, S., editors, *Proceedings of the 9th International Conference on Computational Semantics (IWCS)*, pages 365–369, Oxford, UK.

Reyle, U. (1993). Dealing with ambiguities by underspecification: Construction, representation and deduction. *Journal of Semantics*, 10:123–179.

Roark, B. E. (2001). *Robust Probabilistic Predictive Syntactic Processing: Motivations, Models, and Applications*. PhD thesis, Department of Cognitive and Linguistic Sciences, Brown University.

Sagae, K., Christian, G., DeVault, D., and Traum, D. (2009). Towards natural language understanding of partial speech recognition results in dialogue systems. In *Short paper Proc. NAACL-HLT'09)*, Boulder, Colorado, USA.

Schäfer, U. (2007). *Integrating Deep and Shallow Natural Language Processing Components – Representations and Hybrid Architectures*. PhD thesis, Faculty of Mathematics and Computer Science, Saarland University, Saarbrücken, Germany.

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

Schlangen, D., Baumann, T., and Atterer, M. (2009). Incremental reference resolution: The task, metrics for evaluation, and a bayesian filtering model that is sensitive to disfluencies. In *Proceedings of SIGdial 2009, the 10th Annual SIGDIAL Meeting on Discourse and Dialogue*, London, UK.

Schlangen, D. and Lascarides, A. (2003). The interpretation of non-sentential utterances in dialogue. In Rudnicky, A., editor, *Proceedings of the 4th SIGdial workshop on Discourse and Dialogue*, Sapporo, Japan.

Schlangen, D. and Skantze, G. (2009). A general, abstract model of incremental dialogue processing. In *EACL '09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 710–718. Association for Computational Linguistics.

Skantze, G. and Hjalmarsson, A. (2010). Towards incremental speech generation in dialogue systems. In *Proceedings of the SIGdial 2010 Conference*, pages 1–8, Tokyo, Japan.

Skantze, G. and Schlangen, D. (2009). Incremental dialogue processing in a micro-domain. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009)*, Athens, Greece.

Steedman, M. (2000). *The Syntactic Process*. MIT Press, Cambridge, Massachusetts.

Zettlemoyer, L. S., Milch, B., and Kealbling, L. P. (2009). Multi-agent filtering with infinitely nested beliefs. In *Neural Information Processing Systems (NIPS)*.