Incremental Semantics for Dialogue Processing: Requirements, and a Comparison of Two Approaches

Julian Hough, Casey Kennington,
David Schlangen
Bielefeld University
{julian.hough,ckennington@cit-ec,
david.schlangen}@uni-bielefeld.de

Jonathan Ginzburg
Université Paris-Diderot
yonatan.ginzburg@
univ-paris-diderot.fr

Abstract

Truly interactive dialogue systems need to construct meaning on at least a word-by-word basis. We propose desiderata for incremental semantics for dialogue models and systems, a task not heretofore attempted thoroughly. After laying out the desirable properties we illustrate how they are met by current approaches, comparing two incremental semantic processing frameworks: Dynamic Syntax enriched with Type Theory with Records (DS-TTR) and Robust Minimal Recursion Semantics with incremental processing (RMRS-IP). We conclude these approaches are not significantly different with regards to their semantic representation construction, however their purported role within semantic models and dialogue models is where they diverge.

1 Introduction

It is now uncontroversial that dialogue participants construe meaning from utterances on at least as fine-grained a level as word-by-word (see Brennan, 2000; Schlesewsky and Bornkessel, 2004, *inter alia*). It has also become clear that using more fine-grained incremental processing allows more likeable and interactive systems to be designed (Skantze and Schlangen, 2009; Skantze and Hjalmarsson, 2010). Despite these encouraging results, it has not been clearly stated which elements of incremental semantic frameworks, either formally or implementationally, are desirable for dialogue models and systems; this paper intends to spell these requirements out clearly.

1.1 The need for incremental semantics in situated dialogue

While traditional computational semantics models the meaning of complete sentences, for interaction this is insufficient for achieving *the construction of meaning in real time as linguistic information is processed*. The motivation for incremental semantics becomes clear in situated dialogue systems, which we illustrate here with a real-world scenario. Imagine interacting with a robot capable of manipulating objects of different colours and shapes on a board, where you can direct the robot's action verbally, and the robot also has the ability to direct your actions. When talking to the robot, natural interactions like the following should be possible (the utterance timings and actions of the two participants are represented roughly at the relative time indicated by their horizontal position):

(1) You: Take... the red cross
Robot: [turns head to board]

(2) You: Take the red cross ... and the blue square
Robot: mhm [takes red cross] [takes blue square]

(3) You: Take the red cross, uh no, that's green.
Robot: [takes green cross]

Proceedings of the 11th International Conference on Computational Semantics, pages 206–216,
London, UK, April 15-17 2015.

(4) You: The big red cross uh no, the one in the corner
Robot: [moves hand over nearby cross] [retracts, moves hand over cross in corner]

(5) You: Take the red ...
Robot: cross?

(6) You: Take the red ...
Robot: what?

(7) You: Take the blue uh ... yes, sorry, the red cross
Robot: Did you mean red?

(8) Robot: What's your name? [makes puzzled face]
You: Take the red cross

However we may not desire the following interactions:

(9) You: Take every no, wait, take every red cross!
Robot: [moves hand over green cross]

(10) You: Take the ...
Robot: okay!

Here we propose incremental semantics should be motivated by modelling and implementing this highly interactive, realistic behaviour, putting immediate requirements in focus. (1) shows the robot should begin signalling attention before the command is over, (2) shows backchannel acknowledgements should be driven by incremental semantic understanding, (3) and (4) show how computing the meaning of a repaired utterance even when the repair is elliptical ('that's green') or anaphoric ('the one') is crucial. The compound contribution (5) shows the need for semantic construction to go across dialogue partners (this does not mean string completion), while in (6), the WH-sluice from the robot relies on the (potentially defeasible) inference that you wanted it to take something. The mid-utterance clarification request (7) and mid-utterance reaction to irrelevant user behaviour in (8) show the possibility for immediate reaction to pragmatic infelicity. While we would like the maximal amount of information possible on a word-by-word basis, (9) shows this should not result in bad predictions. (10) shows how human-robot interaction relying on acoustic cues such as silence detection for 'end-pointing' utterances alone is clearly insufficient– silence is not always an indicator of semantic or dialogue-level completeness, nor is its absence good evidence for a continuation of a unit of meaning (see Schlangen and Skantze, 2011).

We address how to meet these requirements in semantics as follows: Section 2 outlines our proposed desiderata, 3 technically overviews two approaches to incremental semantics, 4 compares the approaches in terms of the desired properties theoretically and practically, and Section 5 concludes with the implications of our findings.

2 Desiderata

We take as our point of departure Milward (1991), who points out the difference between a system's capacity for *strong incremental interpretation* and its ability to access and produce *incremental representation*. While these are important and we still consider them central requirements in terms of *semantic representation construction properties*, there are others we propose below, some directly related to these and others orthogonal to them. We also discuss *semantic model*, *dialogue*, and *computational* desiderata. We explain these in turn and the connections between them. Figure 1 shows some of the desiderata visually for the utterance 'take the red cross' as it is interpreted by a rudimentary interpretation module reasoning about a real-word scene: the action SELECT is inferred upon processing the first word and the referent set indicating the possible objects the user is selecting narrows thereafter word-by-word when relevant information specifies the referent. The parts we are principally concerned with are those on levels two and three in grey, in addition to their interfaces to the rest of the model.

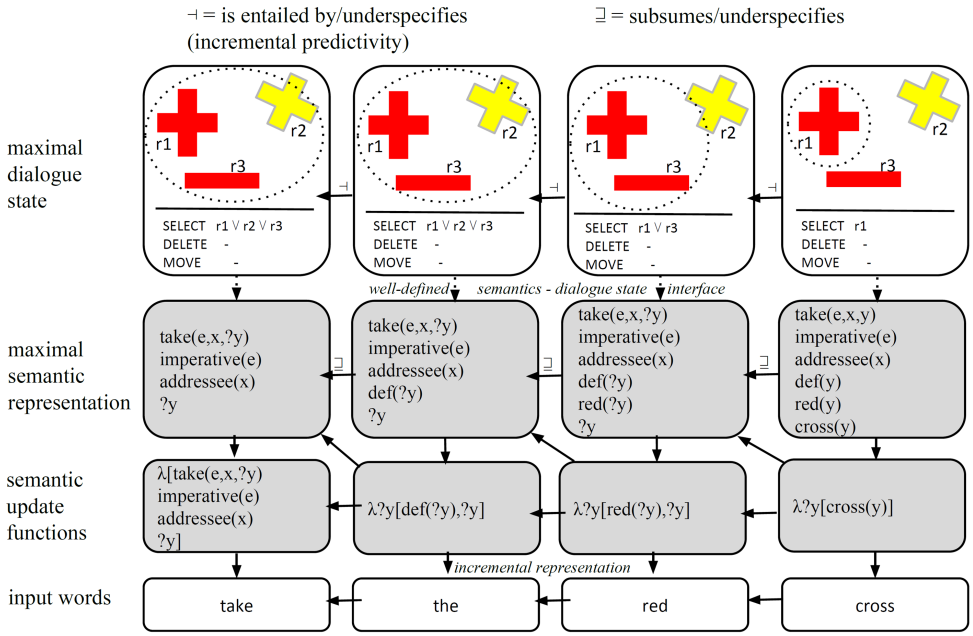

Figure 1: The desired incremental properties of semantics in terms of a dialogue state (level 4, top) idealised scopeless FOL maximal semantic representation with underspecified variables marked'?' (level 3) the update functions (level 2) triggered by input words (level 1). Arrows mean 'triggered by input'.

2.1 Semantic representation construction properties

Strong incremental interpretation In line with Milward (1991), the maximal semantic representation possible should be constructed on a word-by-word basis as it is being produced or interpreted (e.g. a representation such as $\lambda x.like'(john', x)$ should be available after processing "John likes"). The availability of such a representation may, though not necessarily, rely on an interfacing incremental syntactic parsing framework. This is relevant to all examples (1)-(10) for achieving natural understanding and generation. Figure 1 shows the maximal semantic representation at the third level from bottom as idealised scopeless First Order Logic (FOL) formulae with underspecified elements indicated with a '?'.

Incremental representation Again as per (Milward, 1991), assuming a word contributes a minimal amount of semantic representation, the exact contribution each word or substring makes should be available as an increment. However this need not necessarily include all possible information such as semantic dependencies available (e.g. $john'$ attributed to "John" and $\lambda y.\lambda x.like'(y, x)$ attributed to "likes" should be available after processing "John likes"). While strong incremental interpretation is more obviously required for dialogue, the incremental representation requirement becomes stronger when considering the possibility of elements of the input string being revoked in real-time practical dialogue systems– i.e. previous word hypotheses from an ASR output may change (Schlangen and Skantze, 2011). This is also relevant in clarification and repair situations (3), (4) and (7), where on-line computation of the meaning of repaired material relies on identifying its antecedent's semantic representations precisely: access to *how* the incremental information was constructed is essential. Incremental representations are shown as time-linear update functions to the maximal semantic representation as in the second level in Figure 1.

Incremental underspecification and partiality Well-founded underspecification of representation is required– more specifically, structural underspecification, such as that developed in CLLS (Constraint Language for Lambda Structures, Egg et al., 2001). Underspecification should be derivable with incremental representation such as in Steedman (2012)'s Combinatorial Categorial Grammar (CCG) lexicalised model of quantifier scope resolution. As time-linear semantic construction is our central motivation, while we want to capture scope-ambiguous readings of utterances such as 'Every linguist attends a workshop', we add the stipulation that this underspecification be derivable word-by-word. After directly processing a quantifier like 'every' such as in (9), the representation should be as semantically infor-

mative as possible, but no more so; representations should be underspecified enough so as not to make bad predictions for the final structure. Incremental underspecification also means having suitable placeholders for anaphoric and elliptical constructions before they get resolved to their final representation.

Subsumption Dialogue models and systems require well-defined subsumption for incrementally checking representations against domain knowledge, both in understanding and in checking against a semantic goal when generating utterances. One computationally tractable and suitable candidate is Description Logic subsumption, where for two semantic concepts A and B, A *is subsumed by* B, i.e. $B \sqsupseteq A$, iff there is no object belonging to concept A that does not belong to B. The semantic framework should allow subsumption checking from the representation alone– in Figure 1 subsumption holds between maximal semantic representations after each prefix.

2.2 Semantic model properties

While the appropriate representation should be available word-by-word as just described, a suitable model and valuation function must reflect their intuitive semantics incrementally, again providing additional desiderata beyond the valuation of fully specified representations.

Interpretation of partial or underspecified representations The partial representations constructed must be evaluable in a consistent way in a given interpretation system. This applies to all examples (1)-(10): for example if the robot responds appropriately before an instruction is over as in (1) it must have computed a meaning representation to the effect *this is a taking event* early in parsing. In recent type-theoretic approaches in computational semantics this kind of valuation is possible if semantic representations are considered types in a type system: inference can be characterized as subtype relation checking either by theorem proving (Chatzikyriakidis and Luo, 2014) or by checking the existence and ordering relations of types on a model (partial order) of types (Hough and Purver, 2014).[1]

Incremental predictivity Related to subsumption is monotonicity (in the sense of monotonic entailment in logic). In general, one would not want the valuation function after the first word to return more specific information than that returned after the second word, nor at the second word evaluate expressions as having a true value which were evaluated as false after the first word, and so on. In general, the total information made available after having consumed a new word should entail the information inferred by the prefix consumed before it is processed– see the top level in Figure 1. However, from a semantic parsing perspective, maintaining robustness while preserving monotonicity for each interpretation requires allowing multiple parse paths due to possible lexical and structural ambiguity, most notably in 'garden path' sentences, and so the output of a semantic parser can update its output non-monotonically, so long as there is a good notion of *predictivity* of future states in time afforded by the semantic model.

Interface and consistency with well-founded reasoning system Well studied logical inference systems like FOL may not be adequate for natural language inference, as evidenced by the logical form equivalence problem (Shieber, 1993).[2] Having said this, consistent logical systems should be in place which reason with the representations.

2.3 Dialogue properties

Incremental illocutionary information Where available syntactically and lexically, information about the type of dialogue move, or illocutionary effects the utterance causes should be made available as soon as possible, as evidenced by (1), in support of Ginzburg (2012)'s approach. This may not generally be lexicalised, and therefore appropriate underspecification should be used instead to interface with the dialogue model. Also, closely related to strong incremental interpretation is the need to allow for *default existential inference*, as in sluices like (6).

[1] Also, while not immediately a natural language model, computationally incremental interpretation can be modelled in terms of projection algebras (Sundaresh and Hudak, 1991), which allow evaluation of partial programs that are consistent with complete programs.

[2] Roughly, Shieber (1993) shows how FOL can have different logical forms equivalent in meaning within a reasoning system, but these equivalences may not ramify in a comparable way in natural language.

Completion and repair potential In dialogue, it is not rare that one participant begins an utterance and another completes it, in the case of compound contributions such as (5)– according to Howes et al. (2011), this happens in 3% of all dialogue contributions (turns). Furthermore (11) from the same authors shows that concatenating contiguous utterances where a speaker completes another's can be ungrammatical, however felicitous at such turn boundaries in real dialogue.

(11) A: Did you burn...
B: myself?

Potential for clarifying semantic content made central in the dialogue framework KoS (Ginzburg, 2012) is another desirable property. Clarification and repair of semantic information requires incremental representation as described above, as parsers and generators must have access to the information as to which part of the semantic construction was triggered by which word.

Interchangeability between parsing and generation Ideally, the representations built up in parsing should be usable by a generation process and vice-versa; akin to the reversible representation approach in (Neumann, 1998). This is not just to deal with compound contributions, but also to be commensurate with the self-monitoring required in generation (Levelt, 1989) without extra overhead.

Well-founded interface to dialogue or discourse models For extrinsic usefulness, incremental semantics should interface with incremental models of discourse and dialogue. While these models are rare, PTT (Poesio and Traum, 1997) and recent extensions of KoS (Ginzburg, 2012) are candidates. For the sub-task of reference resolution, a suitable semantic model should provide word-by-word reference information, relevant to all interactions in our toy domain in (1)-(10). Also, word-by-word access to the dialogue state to compute relevance or coherence allows inferences of pragmatic infelicity like (8).

2.4 Computational properties

Semantic construction stability Related to the predictivity requirement, semantic content already constructed should not be removed and replaced as processing continues unless triggered by revoked input such as a word hypothesis change from ASR input. Stability affects the rest of the dialogue system served by the semantics. This is pertinent in an automatic system which may have different interpretations stored in a beam, where frequent top hypothesis changes may have undesirable effects.

Minimisation of re-computation and efficiency When faced with changing input, one wants to minimise the re-computation of already evaluated parts of the input (the prefix). There are great efficiency benefits if something only has to be evaluated once. For example chart parsing with the Cocke-Younger-Kasami (CYK) algorithm exhibits this property, as it incrementally hypothesises the syntactic structure of a sentence, where partial results of the computation can be stored on a word-by-word basis to maximise efficiency in a dynamic programming chart, and no computation is done more than once. Top-down parsing approaches such as Roark (2001) also have this property.

Well-founded information and probability theoretic properties For training automatic systems, well-understood information theoretic properties of the semantic construction process aid induction of rules from data. This relies on a well understood probability model of the framework in terms of its distributions of structures and update rules.

We now describe two current incremental semantic parsing frameworks to illustrate how the above desiderata are met.

3 Two Current Attempts

3.1 DS-TTR

DS-TTR (Purver et al., 2011) integrates Type Theory with Records (TTR, Cooper, 2005) *record type* ('RT' largely from now on) representations with the inherently incremental grammar formalism Dynamic Syntax (DS, Kempson et al., 2001) to provide word-by-word semantic construction. DS-TTR is an action-driven interpretation formalism which has no layer of syntax independent of semantic construction. The trees such as Figure 2 are constructed monotonically through sequences of tree-building

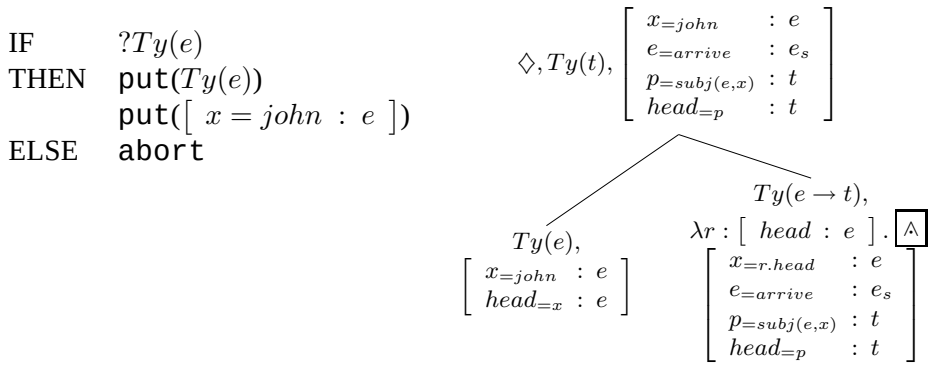

Figure 2: Left: DS-TTR lexical action for 'john'. Right: Final DS-TTR tree for "John arrives"

actions consistent with Logic of Finite Trees (LOFT). The DS lexicon comprises *lexical actions* keyed to words, and also a set of globally applicable *computational actions* (equivalent to general syntactic rules), both of which constitute packages of monotonic update operations on semantic trees, and take the form of IF-THEN-ELSE action-like structures. DS-TTR does not change the LOFT backbone of the DS tree building process, nor does it currently augment the computational actions directly. However, RT formulae are introduced into the lexical actions; for example the lexical action for the word "John" has the preconditions and update effects as in the left-side of Figure 2.

As can be seen on the right side of Figure 2, the DS node types (rather than the RT formulae at the nodes) are terms in the typed lambda calculus, with mother-daughter node relations corresponding to semantic predicate-argument structure. The pointer object, $\Diamond$, indicates the node currently under development. Parsing begins by an initial prediction step on an axiom of a single node with requirement $?Ty(t)$ and then the set of computational actions are Kleene star iterated over to yield a tree set. When a word is consumed, it triggers all possible parses in the current tree set (or those within a given beam-width), and then the set of computational actions are then again iterated over to yield a new tree set.

DS parsing yields an incrementally specified, partial semantic tree as words are parsed or generated, and following Purver et al. (2011) DS-TTR tree nodes are decorated not with simple atomic formulae but with RTs, and corresponding lambda abstracts representing RT λ-functions of type $RT \rightarrow RT$. Using TTR's affordance of *manifest* fields, there is a natural representation for underspecification of leaf node content of DS trees, e.g. $[x : e]$ is unmanifest whereas $[x_{=john} : e]$ is manifest and the latter is a subtype of the former. After every word a RT is compiled to the top node with a simple bottom-up algorithm (Hough and Purver, 2012). DS-TTR tree nodes include a field $head$ in all RTs. Technically, the range of the λ-functions at functor nodes is the asymmetric merge $\boxed{\mathbin{⩚}}$ of their domain RT(s) with the RT in their range. This allows the $head$ field of argument node RTs in β-reduction operations to be replaced by the $head$ field of the function's range RT at the sister functor node in their resulting mother node RT or RT function. On functor nodes semantic content decorations are of the form $\lambda r{:}[l_1 : T_1].r \boxed{\mathbin{⩚}} [l_{2=r.l_1} : T_1]$ where $r.l_1$ is a path expression referring to the label l_1 in r – see the functor node with DS type label $Ty(e \rightarrow t)$ of Figure 2.

Briefly, in DS-TTR generation (Hough and Purver, 2012), surface realisation is done by generating from a goal TTR RT concept. This requires a notion of subsumption which is given by the TTR subtype relation. Generation is driven by parsing and subtype relation checking the goal concept against each tree's top node RT, and consequently meets the desideratum of interchangeability between parsing and generation described above.

3.2 RMRS-IP

While DS-TTR treats both syntactic and semantic construction as one process, Robust Minimal Recursion Semantics with incremental processing (RMRS-IP, Peldszus et al., 2012) splits the task into a top-down PCFG parse followed by the construction of RMRS (Copestake, 2006) formulae using semantic construction rules, operating strictly word-by-word. The current RMRS-IP implementation uses standard top-down non-lexicalised PCFG parsing in the style of Roark (2001), however uses left-factorization of

the standard PCFG grammar rules to delay certain structural decisions as long as possible, employing a beam search over possible parses.

Logical RMRS forms are built up by semantic construction actions operating on the derived CFG trees. In RMRS, meaning representations of a FOL are underspecified in two ways: First, the scope relationships can be underspecified by splitting the formula into a list of *elementary predications* (EP) which receive a label ℓ and are explicitly related by stating scope constraints to hold between them (e.g. *qeq*-constraints). This way, all scope readings can be compactly represented. Second, RMRS allows underspecification of the predicate-argument-structure of EPs. Arguments are bound to a predicate by anchor variables a, expressed in the form of an *argument relation* ARGREL(a,x). This way, predicates can be introduced without fixed arity and arguments can be introduced without knowing which predicates they are arguments of. RMRS-IP makes use of this form of underspecification by enriching lexical predicates with arguments incrementally– see the right of Figure 5.

Combining two RMRS structures involves at least joining their list of EPs and ARGRELs and of scope constraints. Additionally, equations between the variables can connect two structures, which is an essential requirement for semantic construction. A semantic algebra for the combination of RMRSs in a non-lexicalist setting is defined in Copestake (2007). Unsaturated semantic increments have open slots that need to be filled by what is called the *hook* of another structure. Hook and slot are triples $[\ell{:}a{:}x]$ consisting of a label, an anchor and an index variable. Every variable of the hook is equated with the corresponding one in the slot. This way the semantic representation can grow monotonically at each combinatory step by simply adding predicates, constraints and equations. RMRS-IP extends Copestake (2007) in the organisation of the slots to meet the requirement of strong incremental interpretation, constructing a proper semantic representation for every single state of growth of the syntactic tree. Typically, RMRS composition assumes that the order of semantic combination is parallel to a bottom-up traversal of the syntactic tree. However RMRS-IP proceeds with semantic combination in synchronisation with the syntactic expansion of the tree, i.e. in a top-down left-to-right fashion. This way, no underspecification of projected nodes and no re-interpretation of already existing parts of the tree is required. This, however, requires adjustments to the slot structure of RMRS. 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. Thus slots are organized 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. A basic combination operation equivalent to forward function composition (as in standard lambda calculus, or in CCG) allows combination of substructures in a principled way across multiple syntactic rules without the need to represent slot names.

Each lexical item receives a generic representation derived from its lemma and the basic semantic type (individual, event, or underspecified denotations), determined by its POS tag. This makes the grammar independent of knowledge about what later (semantic) components will actually be able to process ("understand"). 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. This allows for a monotonic semantics construction process that proceeds in lock-step with the syntactic analysis. The stack of semantic slots is always synchronized with the parser's stack.

4 Comparison

We now compare DS-TTR and RMRS-IP in terms of how they meet the desiderata set out in Section 2 and compare their incremental performance extrinsically in a proof-of-concept reference resolution task.

Semantic representation construction properties Figure 5 shows the representation constructed by both formalisms for the utterance 'take the red cross' based on hand-crafted grammars. As can be seen both allow *strong incremental interpretation* after each word. DS-TTR is more predictive after processing 'take' by predicting a second (object) argument, however the RMRS-IP grammar in principle could also have this if its PCFG were extended appropriately. *Underspecification and partiality* in representation is good for both as they exhibit incremental extension of their output formulae word-by-word. The DS tree

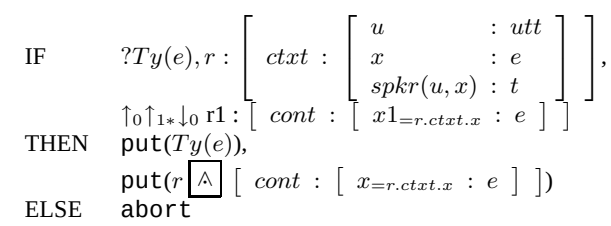

Figure 3: DS-TTR lexical action for 'myself' checks the formula at the subject $Ty(e)$ node, which may not have been constructed by current speaker x but can still reference them

model	metric	**1-6**	**7-8**	**9-14**
RMRS-IP	first-correct (FC)	35.1	23.5	18.4
DS-TTR	(% into utt.)	20.1	20.1	33.1
NGRAM		39.0	23.4	31.7
RMRS-IP	first-final (FF)	43.0	25.5	29.3
DS-TTR	(% into utt.)	23.5	23.3	42.8
NGRAM		46.9	35.5	41.4
RMRS-IP	edit overhead (EO)	7.2	3.3	18.8
DS-TTR		5.8	2.9	17.5
NGRAM		10.4	18.6	9.5

Figure 4: Incremental reference resolution results for utterances of different lengths

keeps a record of the requirements still unsatisfied on its nodes, while in RMRS-IP this is done through the stack of semantic slots (shown in the curly brackets in Figure 5). Both DS-TTR and RMRS-IP allow word-by-word specification of entities (i.e. of the definite description 'the red cross').

In terms of the suitability of the underspecification for ellipsis and anaphora, in DS-TTR the interpretation of strict readings of verb phrase ellipsis (VPE) such as "John likes his donkey and Bill does too" → *Bill likes John's donkey* and sloppy VPE readings, where "John likes his donkey and Bill does too" → *Bill likes his own donkey* is possible incrementally, by different strategies outlined in Kempson et al. (2015). *Wh*-pronouns such as 'who' can be automatically resolved where possible. RMRS has sufficient underspecification to yield similar readings, however this is not operationalised in RMRS-IP parsing.

The semantic increment each word contributes is computed as a difference between the formula computed after a given word and that computed at its previous word in both formalisms, therefore both satisfy *incremental representation*. The subtype relation in TTR is *subsumptive* rather than cohersive, giving DS-TTR another one of our desired properties– see Cooper (2005). Subsumption is not defined in RMRS-IP, but due to its monotonicity in valuation it should exhibit similar properties.

Semantic model properties Both formalisms potentially exhibit *incremental predictivity* in terms of valuation in a semantic model. DS-TTR permits the subtype relation to hold between the current RT and the one constructed at the previous word. This allows valuation on a type lattice whereby type judgements hold from one word to the next but become more specified. RMRS formulae can be flattened to FOL with sortal variables, and given this interpretation can be interpreted monotonically. In terms of *interpretation of partial or underspecified representations* and an *interface and consistency with a well-founded reasoning system*, in DS-TTR, supertypes (the dual of subtypes) allow well-defined underspecified RTs, however more work needs to be done on incorporating underspecified scope relations. As RMRS is defined in a semantic algebra allowing underspecification (Copestake, 2007), it is currently more strongly positioned here. Furthermore, the extensive history of reasoning with FOL logical forms puts RMRS-IP at an advantage to work with well understood semantic models.

Dialogue properties DS-TTR makes claims about dialogue modelling beyond those of RMRS-IP to date. For instance, as regards *interchangeability between parsing and generation*, compound contributions are modelled with speaker-hearer switches which build the same RT, which can be further specified by subtyping to a new goal during the speaker switch. The example (5) can also be accounted for in designing lexical actions which interact with context. By assuming a simple dialogue context is maintained that records who is speaking, this allows interaction-oriented lexical actions to be created, such as that for 'myself' as in Figure 3. This also makes self-monitoring and self-repair in generation possible incrementally, including generating repairs in the face of changing goal concepts (Hough and Purver, 2012). Having said this, these are largely made possible by the well-defined subsumption and monotonicity in subtype relations, so this is in principle re-producible in RMRS. In terms of a *well-founded interface to dialogue models*, while DS-TTR has been used as a dialogue model itself, given DS-TTR's output of RTs, other popular models of dialogue can interface with it, most notably KoS (Ginzburg, 2012). RMRS-IP is well positioned to interface with a variety of formalisms that use FOL, and again,

well-founded logical inference in these models puts it at an advantage.

Computational properties Un-enriched PCFGs have well studied information-theoretic properties and complexity, and are learnable from data, however DS-TTR semantic grammars have been proven to be learnable with semantic targets for short utterances (Eshghi et al., 2013), which has not been attempted yet in RMRS-IP. We discuss both formalisms' *semantic construction stability* below.

4.1 Implementation comparison: Reference Resolution task performance

We also compare the frameworks' current parsing implementations in a real-world inference task contingent on the desiderata. This was done in an incremental reference resolution (RR) task using Kennington et al. (2013)'s statistical SIUM model, which learns to associate words (or in our case, semantic representations) with properties belonging to objects in a virtual scene. Both semantic grammars were hand-crafted to achieve coverage of our test corpus of German spoken instructions directed at a manipulator of blocks in the scene. Word-by-word representations from the parsers were used by SIUM to learn which object properties were likely to be in the referred object. Evaluating using a 10-fold cross validation, in addition to utterance-final **RR accuracy** (where the referent hypothesis was the argmax in the distribution over objects produced by SIUM), to investigate incremental performance we use metrics used by the same authors: **first correct (FC)**: how deep into the utterance (in %) does the model predict the referent for the first time?, **first final (FF)**: how deep into the utterance (in %) does the model predict the correct referent and keep that decision until the end?, and **edit overhead (EO)**: how often did the model unnecessarily change its prediction (the only *necessary* prediction happens when it first makes a correct prediction)? Good *semantic construction stability* would mean low EO, and, good *predictivity* should mean short distance between FC and FF (once correct it does not revoke the referent), and in terms of *strong incremental representation* we would want it to make this final choice early on (low FF).

The utterance-final RR accuracy was 0.876 for SIUM using RMRS-IP, out-performing DS-TTR (0.832), and both out-performing a base-line using n-gram features (0.811). In terms of incremental metrics, DS-TTR had good performance in short utterances up to 8 words long, but RMRS-IP, with more robust PCFG parsing strategies and flexible RMRS composition yields better results overall, particularly in longer utterances. DS-TTR showed good stability and predictivity, on average making correct final predictions earlier than RMRS-IP for utterance lengths 1-6 (FF: 23.5% into the utterance vs 43.0%), and lengths 7-8, however falling back significantly for lengths 9-14 (FF: DS-TTR: 42.8% vs. RMRS-IP: 29.3%), which is likely due to bad parses for long utterances. DS-TTR makes more stable choices as the difference between FF and FC is lowest for all but lengths 7-8, and DS-TTR also achieves the lowest edit overhead across all utterance lengths– see Figure 4. Practically, currently RMRS-IP is more robust for long utterances and for utterance-final meaning, while DS-TTR performs better incrementally.

5 Conclusion

We have proposed desiderata for incremental semantic frameworks for dialogue processing and compared two frameworks. RMRS-IP and DS-TTR meet semantic representation construction criteria very similarly, however their semantic model, dialogue properties and practical robustness differ currently. In terms of parsimony and familiarity for researchers, RMRS with PCFG parsing combined constitute more widely studied formalisms, however DS takes Montague grammar-like structures with a dynamic tree logic as its backbone, and TTR is a well developed rich type system, so is also semanticist-friendly.

We conclude that the *remit* of incremental semantics for dialogue is what needs to be explored further: the dialogue phenomena that DS-TTR models directly may not be desirable for all applications, while RMRS-IP, although cross-compatible with different well-studied reasoning systems and grammars could be seen as not doing enough dialogical semantics and needs enriching.

Acknowledgements We thank the three IWCS reviewers for their insightful comments. This work is supported by the DUEL project, supported by the Agence Nationale de la Research (grant number ANR-13-FRAL-0001) and the Deutsche Forschungsgemeinschaft (grant number SCHL 845/5-1).

word	DS-TTR top record type	RMRS-IP formula
take	$\begin{bmatrix} e_{=take} & : & es \\ x1 & : & e \\ x_{=addressee} & : & e \\ p2_{=object(e,x1)} & : & t \\ p1_{=subject(e,x)} & : & t \\ p_{=imperative(e)} & : & t \\ head_{=e} & : & es \end{bmatrix}$	$[\ell_0{:}a_0{:}e_0]\ \{\ [\ell_0{:}a_0{:}e_0]\ \}$ $\ell_0{:}a_0{:}\_\text{take}(e_0)$, $\quad \mathsf{ARG}_1(a_0, x_2)$, $\ell_2{:}a_2{:}\text{addressee}(x_2)$
the	$\begin{bmatrix} e_{=take} & : & es \\ r & : & \begin{bmatrix} x & : & e \\ head_{=x} & : & e \end{bmatrix} \\ x1_{=\iota(r.head,r)} & : & e \\ x_{=addressee} & : & e \\ p2_{=object(e,x1)} & : & t \\ p1_{=subject(e,x)} & : & t \\ p_{=imperative(e)} & : & t \\ head_{=e} & : & es \end{bmatrix}$	$[\ell_0{:}a_0{:}e_0]\ \{\ [\ell_7{:}a_7{:}e_4], [\ell_0{:}a_0{:}e_0]\ \}$ $\ell_0{:}a_0{:}\_\text{take}(e_0)$, $\quad \mathsf{ARG}_1(a_0, x_2)$, $\quad \mathsf{ARG}_2(a_0, x_4)$, $\ell_2{:}a_2{:}\text{addressee}(x_2)$, $\ell_4{:}a_4{:}\_\text{def\_q}()$, $\quad \mathsf{BV}(a_4, x_4)$, $\quad \mathsf{RSTR}(a_4, h_1)$, $\quad \mathsf{BODY}(a_4, h_2)$, $\quad h_1 =_q \ell_7$
red	$\begin{bmatrix} e_{=take} & : & es \\ r & : & \begin{bmatrix} x & : & e \\ p_{=red(x)} & : & t \\ head_{=x} & : & e \end{bmatrix} \\ x1_{=\iota(r.head,r)} & : & e \\ x_{=addressee} & : & e \\ p2_{=object(e,x1)} & : & t \\ p1_{=subject(e,x)} & : & t \\ p_{=imperative(e)} & : & t \\ head_{=e} & : & es \end{bmatrix}$	$[\ell_0{:}a_0{:}e_0]\ \{\ [\ell_7{:}a_7{:}e_4], [\ell_0{:}a_0{:}e_0]\ \}$ $\ell_0{:}a_0{:}\_\text{take}(e_0)$, $\quad \mathsf{ARG}_1(a_0, x_2)$, $\quad \mathsf{ARG}_2(a_0, x_4)$, $\ell_2{:}a_2{:}\text{addressee}(x_2)$, $\ell_4{:}a_4{:}\_\text{def\_q}()$, $\quad \mathsf{BV}(a_4, x_4)$, $\quad \mathsf{RSTR}(a_4, h_1)$, $\quad \mathsf{BODY}(a_4, h_2)$, $\quad h_1 =_q \ell_7$, $\ell_7{:}a_{10}{:}\_\text{red}(e_{10})$, $\quad \mathsf{ARG}_1(a_{10}, x_4)$
cross	$\begin{bmatrix} e_{=take} & : & es \\ r & : & \begin{bmatrix} x & : & e \\ p1_{=cross(x)} & : & t \\ p_{=red(x)} & : & t \\ head_{=x} & : & e \end{bmatrix} \\ x1_{=\iota(r.head,r)} & : & e \\ x_{=addressee} & : & e \\ p2_{=object(e,x1)} & : & t \\ p1_{=subject(e,x)} & : & t \\ p_{=imperative(e)} & : & t \\ head_{=e} & : & es \end{bmatrix}$	$[\ell_0{:}a_0{:}e_0]\ \{\ \}$ $\ell_0{:}a_0{:}\_\text{take}(e_0)$, $\quad \mathsf{ARG}_1(a_0, x_2)$, $\quad \mathsf{ARG}_2(a_0, x_4)$, $\ell_2{:}a_2{:}\text{addressee}(x_2)$, $\ell_4{:}a_4{:}\_\text{def\_q}()$, $\quad \mathsf{BV}(a_4, x_4)$, $\quad \mathsf{RSTR}(a_4, h_1)$, $\quad \mathsf{BODY}(a_4, h_2)$, $\quad h_1 =_q \ell_7$, $\ell_7{:}a_{10}{:}\_\text{red}(e_{10})$, $\quad \mathsf{ARG}_1(a_{10}, x_4)$, $\ell_7{:}a_7{:}\_\text{cross}(x_4)$

Figure 5: Incremental semantic construction by DS-TTR and RMRS-IP

References

Brennan, S. E. (2000). Processes that shape conversation and their implications for computational linguistics. In *Proceedings of the 38th annual meeting of the ACL*, pp. 1–11. ACL.

Chatzikyriakidis, S. and Z. Luo (2014). Natural language reasoning using proof-assistant technology: Rich typing and beyond. In *EACL 2014 TTNLS Workshop*, Gothenburg, Sweden, pp. 37–45. ACL.

Cooper, R. (2005). Records and record types in semantic theory. *Journal of Logic and Computation 15*(2), 99–112.

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

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

Egg, M., A. Koller, and J. Niehren (2001). The constraint language for lambda structures. *Journal of Logic, Language and Information 10*(4), 457–485.

Eshghi, A., J. Hough, and M. Purver (2013). Incremental grammar induction from child-directed dialogue utterances. In *The Fourth Annual CMCL Workshop*, Sofia, Bulgaria, pp. 94–103. ACL.

Ginzburg, J. (2012). *The Interactive Stance: Meaning for Conversation*. Oxford University Press.

Hough, J. and M. Purver (2012). Processing self-repairs in an incremental type-theoretic dialogue system. In *Proceedings of the 16th SemDial Workshop (SeineDial)*, Paris, France, pp. 136–144.

Hough, J. and M. Purver (2014). Probabilistic type theory for incremental dialogue processing. In *Proceedings of the EACL 2014 TTNLS Workshop*, Gothenburg, Sweden, pp. 80–88. ACL.

Howes, C., M. Purver, P. G. Healey, G. Mills, and E. Gregoromichelaki (2011). On incrementality in dialogue: Evidence from compound contributions. *Dialogue & Discourse 2*(1), 279–311.

Kempson, R., R. Cann, A. Eshghi, E. Gregoromichelaki, and M. Purver (2015). Ellipsis. In S. Lappin and C. Fox (Eds.), *Handbook of Contemporary Semantic Theory* (2nd ed.)., Chapter 3. Wiley.

Kempson, R., W. Meyer-Viol, and D. Gabbay (2001). *Dynamic Syntax: The Flow of Language Understanding*. Oxford: Blackwell.

Kennington, C., S. Kousidis, and D. Schlangen (2013). Interpreting Situated Dialogue Utterances: an Update Model that Uses Speech, Gaze, and Gesture Information. In *SIGdial 2013*.

Levelt, W. J. (1989). *Speaking: From intention to articulation*. MIT press.

Milward, D. (1991). *Axiomatic Grammar, Non-Constituent Coordination and Incremental Interpretation*. Ph. D. thesis, University of Cambridge.

Neumann, G. (1998). Interleaving natural language parsing and generation through uniform processing. *Artificial Intelligence 99*, 121–163.

Peldszus, A., O. Buß, T. Baumann, and D. Schlangen (2012). Joint Satisfaction of Syntactic and Pragmatic Constraints Improves Incremental Spoken Language Understanding. In *Proceedings of the 13th EACL*, Avignon, France, pp. 514–523. ACL.

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

Purver, M., A. Eshghi, and J. Hough (2011). Incremental semantic construction in a dialogue system. In J. Bos and S. Pulman (Eds.), *Proceedings of the 9th IWCS*, Oxford, UK, pp. 365–369.

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

Schlangen, D. and G. Skantze (2011). A General, Abstract Model of Incremental Dialogue Processing. *Dialoge & Discourse 2*(1), 83–111.

Schlesewsky, M. and I. Bornkessel (2004). On incremental interpretation: Degrees of meaning accessed during sentence comprehension. *Lingua 114*(9), 1213–1234.

Shieber, S. M. (1993). The problem of logical-form equivalence. *Computational Linguistics 19*(1), 179–190.

Skantze, G. and A. Hjalmarsson (2010). Towards incremental speech generation in dialogue systems. In *Proceedings of the 11th Annual Meeting of SIGDIAL*, pp. 1–8. ACL.

Skantze, G. and D. Schlangen (2009). Incremental dialogue processing in a micro-domain. In *Proceedings of the 12th Conference of the EACL*, pp. 745–753. ACL.

Steedman, M. (2012). *Taking scope: The natural semantics of quantifiers*. MIT Press.

Sundaresh, R. S. and P. Hudak (1991). A theory of incremental computation and its application. In *Proceedings of the 18th ACM SIGPLAN-SIGACT symposium*, pp. 1–13. ACM.