Slacker semantics: why superficiality, dependency and avoidance of commitment can be the right way to go

Ann Copestake
Computer Laboratory, University of Cambridge
15 JJ Thomson Avenue, Cambridge, UK
aac@cl.cam.ac.uk

Abstract

This paper discusses computational compositional semantics from the perspective of grammar engineering, in the light of experience with the use of Minimal Recursion Semantics in DELPH-IN grammars. The relationship between argument indexation and semantic role labelling is explored and a semantic dependency notation (DMRS) is introduced.

1 Introduction

The aim of this paper is to discuss work on compositional semantics from the perspective of grammar engineering, which I will take here as the development of (explicitly) linguistically-motivated computational grammars. The paper was written to accompany an invited talk: it is intended to provide background and further details for those parts of the talk which are not covered in previous publications. It consists of an brief introduction to our approach to computational compositional semantics, followed by details of two contrasting topics which illustrate the grammar engineering perspective. The first of these is argument indexing and its relationship to semantic role labelling, the second is semantic dependency structure.

Standard linguistic approaches to compositional semantics require adaptation for use in broad-coverage computational processing. Although some of the adaptations are relatively trivial, others have involved considerable experimentation by various groups of computational linguists. Perhaps the most important principle is that semantic representations should be a good match for syntax, in the sense of capturing all and only the information available from syntax and productive morphology, while nevertheless abstracting over semantically-irrelevant idiosyncratic detail. Compared to much of the linguistics literature, our analyses are relatively superficial, but this is essentially because the broad-coverage computational approach prevents us from over-committing on the basis of the information available from the syntax. One reflection of this are the formal techniques for scope underspecification which have been developed in computational linguistics. The implementational perspective, especially when combined with a requirement that grammars can be used for generation as well as parsing, also forces attention to details which are routinely ignored in theoretical linguistic studies. This is particularly true when there are interactions between phenomena which are generally studied separately. Finally, our need to produce usable systems disallows some appeals to pragmatics, especially those where analyses are radically underspecified to allow for syntactic and morphological effects found only in highly marked contexts.[1]

In a less high-minded vein, sometimes it is right to be a slacker: life (or at least, project funding) is too short to implement all ideas within a grammar in their full theoretical glory. Often there is an easy alternative which conveys the necessary information to a consumer of the semantic representations. Without this, grammars would never stabilise.

Here I will concentrate on discussing work which has used Minimal Recursion Semantics (MRS: Copestake et al. (2005)) or Robust Minimal Recursion Semantics (RMRS: Copestake (2003)). The (R)MRS approach has been adopted as a common framework for the DELPH-IN initiative (Deep Linguistic Processing with HPSG: http://www.delph-in.net) and the work discussed here has been done by and in collaboration with researchers involved in DELPH-IN.

The programme of developing computational compositional semantics has a large number of aspects. It is important that the semantics has a logically-sound interpretation (e.g., Koller and Lascarides (2009), Thater (2007)), is cross-

[1] For instance, we cannot afford to underspecify number on nouns because of examples such as *The hash browns is getting angry* (from Pollard and Sag (1994) p.85).

Proceedings of the 12th Conference of the European Chapter of the ACL, pages 1–9,
Athens, Greece, 30 March – 3 April 2009. ©2009 Association for Computational Linguistics

linguistically adequate (e.g., Bender (2008)) and is compatible with generation (e.g., Carroll et al. (1999), Carroll and Oepen (2005)). Ideally, we want support for shallow as well as deep syntactic analysis (which was the reason for developing RMRS), enrichment by deeper analysis (including lexical semantics and anaphora resolution, both the subject of ongoing work), and (robust) inference. The motivation for the development of dependency-style representations (including Dependency MRS (DMRS) discussed in §4) has been to improve ease of use for consumers of the representation and human annotators, as well as use in statistical ranking of analyses/realisations (Fujita et al. (2007), Oepen and Lønning (2006)). Integration with distributional semantic techniques is also of interest.

The belated 'introduction' to MRS in Copestake et al. (2005) primarily covered formal representation of complete utterances. Copestake (2007a) described uses of (R)MRS in applications. Copestake et al. (2001) and Copestake (2007b) concern the algebra for composition. What I want to do here is to concentrate on less abstract issues in the syntax-semantics interface. I will discuss two cases where the grammar engineering perspective is important and where there are some conclusions about compositional semantics which are relevant beyond DELPH-IN. The first, argument indexing (§3), is a relatively clear case in which the constraints imposed by grammar engineering have a significant effect on choice between plausible alternatives. I have chosen to talk about this both because of its relationship with the currently popular task of semantic role labelling and because the DELPH-IN approach is now fairly stable after a quite considerable degree of experimentation. What I am reporting is thus a perspective on work done primarily by Flickinger within the English Resource Grammar (ERG: Flickinger (2000)) and by Bender in the context of the Grammar Matrix (Bender et al., 2002), though I've been involved in many of the discussions. The second main topic (§4) is new work on a semantic dependency representation which can be derived from MRS, extending the previous work by Oepen (Oepen and Lønning, 2006). Here, the motivation came from an engineering perspective, but the nature of the representation, and indeed the fact that it is possible at all, reveals some interesting aspects of semantic composition in the grammars.

2 The MRS and RMRS languages

This paper concerns only representations which are output by deep grammars, which use MRS, but it will be convenient to talk in terms of RMRS and to describe the RMRSs that are constructed under those assumptions. Such RMRSs are interconvertible with MRSs.[2] The description is necessarily terse and contains the minimal detail necessary to follow the remainder of the paper.

An RMRS is a description of a set of trees corresponding to scoped logical forms. Fig 1 shows an example of an RMRS and its corresponding scoped form (only one for this example). RMRS is a 'flat' representation, consisting of a bag of **elementary predications** (EP), a set of **argument relations**, and a set of constraints on the possible linkages of the EPs when the RMRS is resolved to scoped form. Each EP has a predicate, a **label** and a unique **anchor** and may have a distinguished (ARG0) argument (EPs are written here as label:anchor:pred(arg0)). Label sharing between EPs indicates conjunction (e.g., in Fig 1, *big*, *angry* and *dog* share the label l2). Argument relations relate non-arg0 arguments to the corresponding EP via the anchor. Argument names are taken from a fixed set (discussed in §3). Argument values may be variables (e.g., e8, x4: variables are the only possibility for values of ARG0), constants (strings such as "London"), or **holes** (e.g. h5), which indicate scopal relationships. Variables have sortal properties, indicating tense, number and so on, but these are not relevant for this paper. Variables corresponding to unfilled (syntactically optional) arguments are unique in the RMRS, but otherwise variables must correspond to the ARG0 of an EP (since I am only considering RMRSs from deep grammars here).

Constraints on possible scopal relationships between EPs may be explicitly specified in the grammar via relationships between holes and labels. In particular **qeq** constraints (the only type considered here) indicate that, in the scoped forms, a label must either plug a hole directly or be connected to it via a chain of quantifiers. Hole arguments (other than the BODY of a quantifier) are always linked to a label via a qeq or other constraint (in a deep grammar RMRS). Variables survive in the models of RMRSs (i.e., the fully scoped trees) whereas holes and labels do not.

[2] See Flickinger and Bender (2003) and Flickinger et al. (2003) for the use of MRS in DELPH-IN grammars.

l1:a1:_some_q, BV(a1,x4), RSTR(a1,h5), BODY(a1,h6), h5 qeq l2,
l2:a2:_big_a_1(e8), ARG1(a2,x4), l2:a3:_angry_a_1(e9), ARG1(a3,x4), l2:a4:_dog_n_1(x4),
l4:a5:_bark_v_1(e2), ARG1(a5,x4), l4:a6:_loud_a_1(e10), ARG1(a6,e2)

_some_q(x4,_big_a_1(e8,x4) ∧ _angry_a_1(e9, x4) ∧ _dog_n_1(x4),_bark_v_1(e2,x4) ∧_loud_a_1(e10,e2))

Figure 1: RMRS and scoped form for 'Some big angry dogs bark loudly'. Tense and number are omitted.

The naming convention for predicates corresponding to lexemes is: _ stem _ major sense tag, optionally followed by _ and minor sense tag (e.g., _loud_a_1). Major sense tags correspond roughly to traditional parts of speech. There are also non-lexical predicates such as 'poss' (though none occur in Fig 1).[3] MRS varies from RMRS in that the arguments are all directly associated with the EP and thus no anchors are necessary.

I have modified the definition of RMRS given in Copestake (2007b) to make the ARG0 argument optional. Here I want to add the additional constraint that the ARG0 of an EP is unique to it (i.e., not the ARG0 of any other EP). I will term this the **characteristic variable** property. This means that, for every variable, there is a unique EP which has that variable as its ARG0. I will assume for this paper that all EPs, apart from quantifier EPs, have such an ARG0.[4] The characteristic variable property is one that has emerged from working with large-scale constraint-based grammars.

A few concepts from the MRS algebra are also necessary to the discussion. Composition can be formalised as functor-argument combination where the argument phrase's **hook** fills a **slot** in the functor phrase, thus instantiating an RMRS argument relation. The hook consists of an **index** (a variable), an **external argument** (also a variable) and an **ltop** (local top: the label corresponding to the topmost node in the current partial tree, ignoring quantifiers). The syntax-semantics interface requires that the appropriate hook and slots be set up (mostly lexically in a DELPH-IN grammar) and that each application of a rule specifies the slot to be used (e.g., MOD for modification). In a lexical entry, the ARG0 of the EP provides the hook index, and, apart from quantifiers, the hook ltop is the EP's label. In intersective combination, the ltops of the hooks will be equated. In scopal combination, a hole argument in a slot is specified to be qeq to the ltop of the argument phrase and the ltop of the functor phrase supplies the new hook's ltop.

By thinking of qeqs as links in an RMRS graph (rather than in terms of their logical behaviour as constraints on the possible scoped forms), an RMRS can be treated as consisting of a set of trees with nodes consisting of EPs grouped via intersective relationships: there will be a backbone tree (headed by the overall ltop and including the main verb if there is one), plus a separate tree for each quantified NP. For instance, in Fig 1, the third line contains the EPs corresponding to the (single node) backbone tree and the first two lines show the EPs comprising the tree for the quantified NP (one node for the quantifier and one for the N′ which it connects to via the RSTR and its qeq).

3 Arguments and roles

I will now turn to the representation of arguments in MRS and their relationship to semantic roles. I want to discuss the approach to argument labelling in some detail, because it is a reasonably clear case where the desiderata for broad-coverage semantics which were discussed in §1 led us to a syntactically-driven approach, as opposed to using semantically richer roles such as AGENT, GOAL and INSTRUMENT.

An MRS can, in fact, be written using a conventional predicate-argument representation. A representation which uses ordered argument labels can be recovered from this in the obvious way. E.g., l:like_v_1(e,x,y) is equivalent to l:a:like_v_1(e), ARG1(a,x), ARG2(a,y). A fairly large inventory of argument labels is actually used in the DELPH-IN grammars (e.g., RSTR, BODY). To recover these from the conventional predicate-argument notation requires a look up in a semantic interface component (the SEM-I, Flickinger et al. (2005)). But open-class predicates use the ARGn convention, where n is 0,1,2,3 or 4 and the discussion here

[3] In fact, most of the choices about semantics made by grammar writers concern the behaviour of constructions and thus these non-lexical predicates, but this would require another paper to discuss.

[4] I am simplifying for expository convenience. In current DELPH-IN grammars, quantifiers have an ARG0 which corresponds to the bound variable. This should not be the characteristic variable of the quantifier (it is the characteristic variable of a nominal EP), since its role in the scoped forms is as a notational convenience to avoid lambda expressions. I will call it the BV argument here.

only concerns these.[5]

Arguably, the DELPH-IN approach is Davidsonian rather than neo-Davidsonian in that, even in the RMRS form, the arguments are related to the predicate via the anchor which plays no other role in the semantics. Unlike the neo-Davidsonian use of the event variable to attach arguments, this allows the same style of representation to be used uniformly, including quantifiers, for instance. Arguments can omitted completely without syntactic ill-formedness of the RMRS, but this is primarily relevant to shallower grammars. A semantic predicate, such as like_v_1, is a logical predicate and as such is expected to have the same arity wherever it occurs in the DELPH-IN grammars. Thus models for an MRS may be defined in a language with or without argument labels.

The ordering of arguments for open class lexemes is lexically specified on the basis of the syntactic obliqueness hierarchy (Pollard and Sag, 1994). ARG1 corresponds to the subject in the base (non-passivised) form ('deep subject'). Argument numbering is consecutive in the base form, so no predicate with an ARG3 is lexically missing an ARG2, for instance. An ARG3 may occur without an instantiated ARG2 when a syntactically optional argument is missing (e.g., *Kim gave to the library*), but this is explicit in the linearised form (e.g., _give_v(e,x,u,y)).

The full statement of how the obliqueness hierarchy (and thus the labelling) is determined for lexemes has to be made carefully and takes us too far into discussion of syntax to explain in detail here. While the majority of cases are straightforward, a few are not (e.g., because they depend on decisions about which form is taken as the base in an alternation). However, all decisions are made at the level of lexical types: adding an entry for a lexeme for a DELPH-IN grammar only requires working out its lexical type(s) (from syntactic behaviour and very constrained semantic notions, e.g., control). The actual assignment of arguments to an utterance is just a consequence of parsing. Argument labelling is thus quite different from PropBank (Palmer et al., 2005) role labelling despite the unfortunate similarity of the PropBank naming scheme.

It follows from the fixed arity of predicates that lexemes with different numbers of arguments should be given different predicate symbols. There is usually a clear sense distinction when this occurs. For instance, we should distinguish between the 'depart' and 'bequeath' senses of *leave* because the first takes an ARG1 and an ARG2 (optional) and the second ARG1, ARG2 (optional), ARG3. We do not draw sense distinctions where there is no usage which the grammar could disambiguate.

Of course, there are obvious engineering reasons for preferring a scheme that requires minimal additional information in order to assign argument labels. Not only does this simplify the job of the grammar writer, but it makes it easier to construct lexical entries automatically and to integrate RMRSs derived from shallower systems. However, grammar engineers respond to consumers: if more detailed role labelling had a clear utility and required an analysis at the syntax level, we would want to do it in the grammar. The question is whether it is practically possible.

Detailed discussion of the linguistics literature would be out of place here. I will assume that Dowty (1991) is right in the assertion that there is no small (say, less than 10) set of role labels which can also be used to link the predicate to its arguments in compositionally constructed semantics (i.e., argument-indexing in Dowty's terminology) such that each role label can be given a consistent individual semantic interpretation. For our purposes, a consistent semantic interpretation involves entailment of one or more useful real world propositions (allowing for exceptions to the entailment for unusual individual sentences).

This is not a general argument against rich role labels in semantics, just their use as the means of argument-indexation. It leaves open uses for grammar-internal purposes, e.g., for defining and controlling alternations. The earliest versions of the ERG experimented with a version of Davis's (2001) approach to roles for such reasons: this was not continued, but for reasons irrelevant here. Roles are still routinely used for argument indexation in linguistics papers (without semantic interpretation). The case is sometimes made that more mnemonic argument labelling helps human interpretation of the notation. This may be true of semantics papers in linguistics, which tend to concern groups of similar lexemes. It is not true of a collaborative computational linguistics project in which broad coverage is being attempted: names

[5] ARG4 occurs very rarely, at least in English (the verb *bet* being perhaps the clearest case).

can only be mnemonic if they carry some meaning and if the meaning cannot be consistently applied this leads to endless trouble.

What I want to show here is how problems arise even when very limited semantic generalisations are attempted about the nature of just one or two argument labels, when used in broad-coverage grammars. Take the quite reasonable idea that a semantically consistent labelling for intransitives and related causatives is possible (cf PropBank). For instance, *water* might be associated with the same argument label in the following examples:

(1) Kim boiled the water.

(2) The water boiled.

Using (simplified) RMRS representations, this might amount to:

(3) l:a:boil_v(e), a:ARG1(k), a:ARG2(x), water(x)

(4) l:a:boil_v(e), a:ARG2(x), water(x)

Such an approach was used for a time in the ERG with unaccusatives. However, it turns out to be impossible to carry through consistently for causative alternations.

Consider the following examples of *gallop*: [6]

(5) Michaela galloped the horse to the far end of the meadow, ...

(6) With that Michaela nudged the horse with her heels and off the horse galloped.

(7) Michaela declared, "I shall call him Lightning because he runs as fast as lightning." And with that, off she galloped.

If only a single predicate is involved, e.g., gallop_v, and the causative has an ARG1 and an ARG2, then what about the two intransitive cases? If the causative is treated as obligatorily transitive syntactically, then (6) and (7) presumably both have an ARG2 subject. This leads to Michaela having a different role label in (5) and (7), despite the evident similarity of the real world situation. Furthermore, the role labels for intransitive movement verbs could only be predicted by a consumer of the semantics who knew whether or not a causative form existed. The causative may be rare, as with *gallop*, where the intransitive use is clearly the base case. Alternatively, if (7) is treated as a causative intransitive, and thus has a subject labelled ARG1, there is a systematic unresolvable ambiguity and the generalisation that the subjects in both intransitive sentences are moving is lost.

Gallop is an not isolated case in having a volitional intransitive use: it applies to most (if not all) motion verbs which undergo the causative alternation. To rescue this account, we would need to apply it only to true lexical anti-causatives. It is not clear whether this is doable (even the standard example *sink* can be used intransitively of deliberate movement) but from a slacker perspective, at this point we should decide to look for an easier approach.

The current ERG captures the causative relationship by using systematic sense labelling:

(8) Kim boiled the water.
l:a:boil_v_cause(e), a:ARG1(k), a:ARG2(x), water(x)

(9) The water boiled.
l:a:boil_v_1(e), a:ARG1(x), water(x)

This is not perfect, but it has clear advantages. It allows inferences to be made about ARG1 and ARG2 of _cause verbs. In general, inferences about arguments may be made with respect to particular verb classes. This lends itself to successive refinement in the grammars: the decision to add a standardised sense label, such as _cause, does not require changes to the type system, for instance. If we decide that we can identify true anti-causatives, we can easily make them a distinguished class via this convention. Conversely, in the situation where causation has not been recognised, and the verb has been treated as a single lexeme having an optional ARG2, the semantics is imperfect but at least the imperfection is local.

In fact, determining argument labelling by the obliqueness hierarchy still allows generalisations to be made for all verbs. Dowty (1991) argues for the notion of proto-agent (p-agt) and proto-patient (p-pat) as cluster concepts. Proto-agent properties include volitionality, sentience, causation of an event and movement relative to another participant. Proto-patient properties include being causally affected and being stationary relative to another participant. Dowty claims that generalisations about which arguments are lexicalised as subject, object and indirect object/oblique can be expressed in terms of relative numbers of p-agt and p-pat properties. If this is correct, then we can,

[6] http://www.thewestcoast.net/bobsnook/kid/horses.htm.

for example, predict that the ARG1 of any predicate in a DELPH-IN grammar will not have fewer p-agt properties than the ARG2 of that predicate.[7]

As an extreme alternative, we could use labels which were individual to each predicate, such as LIKER and LIKED (e.g., Pollard and Sag (1994)). For such role labels to have a consistent meaning, they would have to be lexeme-specific: e.g., LEAVER1 ('departer') versus LEAVER2 ('bequeather'). However this does nothing for semantic generalisation, blocks the use of argument labels in syntactic generalisations and leads to an extreme proliferation of lexical types when using typed feature structure formalisms (one type would be required per lexeme). The labels add no additional information and could trivially be added automatically to an RMRS if this were useful for human readers. Much more interesting is the use of richer lexical semantic generalisations, such as those employed in FrameNet (Baker et al., 1998). In principle, at least, we could (and should) systematically link the ERG to FrameNet, but this would be a form of semantic enrichment mediated via the SEM-I (cf Roa et al. (2008)), and not an alternative technique for argument indexation.

4 Dependency MRS

The second main topic I want to address is a form of semantic dependency structure (DMRS: see `wiki.delph-in.net` for the evolving details). There are good engineering reasons for producing a dependency style representation with links between predicates and no variables: ease of readability for consumers of the representation and for human annotators, parser comparison and integration with distributional lexical semantics being the immediate goals. Oepen has previously produced **elementary dependencies** from MRSs but the procedure (partially sketched in Oepen and Lønning (2006)) was not intended to produce complete representations. It turns out that a DMRS can be constructed which can be demonstrated to be interconvertible with RMRS, has a simple graph structure and minimises redundancy in the representation. What is surprising is that this can be done for a particular class of grammars without making use of the evident clues to syntax in the predicate names. The characteristic variable property discussed in §2 is crucial: its availability allows a partial replication of composition, with DMRS links being relatable to functor-argument combinations in the MRS algebra. I should emphasize that, unlike MRS and RMRS, DMRS is not intended to have a direct logical interpretation.

An example of a DMRS is given in Fig 2. Links relate nodes corresponding to RMRS predicates. Nodes have unique identifiers, not shown here. Directed link labels are of the form ARG/H, ARG/EQ or ARG/NEQ, where ARG corresponds to an RMRS argument label. H indicates a qeq relationship, EQ label equality and NEQ label inequality, as explained more fully below. Undirected /EQ arcs also sometimes occur (see §4.3). The ltop is indicated with a *.

4.1 RMRS-to-DMRS

In order to transform an RMRS into a DMRS, we will treat the RMRS as made up of three subgraphs:

Label equality graph. Each EP in an RMRS has a label, which may be shared with any number of other EPs. This can be captured in DMRS via a graph linking EPs: if this is done exhaustively, there would be $n(n-1)/2$ binary non-directional links. E.g., for the RMRS in Fig 1, we need to link _big_a_1, _angry_a_1 and _dog_n_1 and this takes 3 links. Obviously the effect of equality could be captured by a smaller number of links, assuming transitivity: but to make the RMRS-to-DMRS conversion deterministic, we need a method for selecting canonical links.

Hole-to-label qeq graph. A qeq in RMRS links a hole to a label which labels a set of EPs. There is thus a $1:1$ mapping between holes and labels which can be converted to a $1:n$ mapping between holes and the EPs which share the label. By taking the EP with the hole as the origin, we can construct an EP-to-EP graph, using the argument name as a label for the link: of course, such links are asymmetric and thus the graph is directed. e.g., _some_q has RSTR links to each of _big_a_1, _angry_a_1 and _dog_n_1. Reducing this to a $1:1$ mapping between EPs, which we would ideally like for DMRS, requires a canonical method of selecting a **head** EP from the set of target EPs (as does the selection of the ltop).

Variable graph. For the conversion to DMRS, we will rely on the characteristic variable prop-

[7] Sanfilippo (1990) originally introduced Dowty's ideas into computational linguistics, but this relative behaviour cannot be correctly expressed simply by using p-agt and p-pat directly for argument indexation as he suggested. It is incorrect for examples like (2) to be labelled as p-agt, since they have no agentive properties.

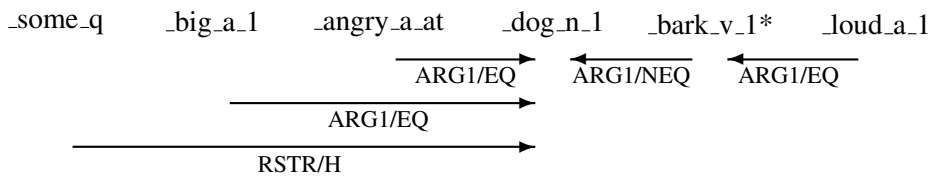

Figure 2: DMRS for 'Some big angry dogs bark loudly.'

erty, that every variable has a unique EP associated with it via its ARG0. Any non-hole argument of an EP will have a value which is the ARG0 of some other EP, or which is unbound (i.e., not found elsewhere in the RMRS) in which case we ignore it. Thus we can derive a graph between EPs, such that each link is labelled with an argument position and points to a unique EP. I will talk about an EP's 'argument EPs', to refer to the set of EPs its arguments point to in this graph.

The three EP graphs can be combined to form a dependency structure. But this has an excessive number of links due to the label equality and qeq components. We need deterministic techniques for removing the redundancy. These can utilise the variable graph, since this is already minimal.

The first strategy is to combine the label equality and variable links when they connect the same two EPs. For instance, we combine the ARG1 link between _big_a_1, and _dog_n_1 with the label equality link to give a link labelled ARG1/EQ. We then test the connectivity of the ARG/EQ links on the assumption of transitivity and remove any redundant links from the label graph. This usually removes all label equality links: one case where it does not is discussed in §4.3. Variable graph links with no corresponding label equality are annotated ARG/NEQ, while links arising from the qeq graph are labelled ARG/H. This retains sufficient information to allow the reconstruction of the three graphs in DMRS-to-RMRS conversion.

In order to reduce the number of links arising from the qeq graph, we make use of the variable graph to select a head from a set of EPs sharing a label. It is not essential that there should be a unique head, but it is desirable. The next section outlines how head selection works: despite not using any directly syntactic properties, it generally recovers the syntactic head.

4.2 Head selection in the qeq graph

Head selection uses one principle and one heuristic, both of which are motivated by the compositional properties of the grammar. The principle is that qeq links from an EP should parallel any comparable variable links. If an EP has two arguments, one of which is a variable argument which links to EP′ and the other a hole argument which has a value corresponding to a set of EPs including EP′, EP′ is chosen as the head of that set.

This essentially follows from the composition rules: in an algebra operation giving rise to a qeq, the argument phrase supplies a hook consisting of an index (normally, the ARG0 of the head EP) and an ltop (normally, the label of the head EP). Thus if a variable argument corresponds to EP′, EP′ will have been the head of the corresponding phrase and is thus the choice of head in the DMRS. This most frequently arises with quantifiers, which have both a BV and a RSTR argument: the RSTR argument can be taken as linking to the EP which has an ARG0 equal to the BV (i.e., the head of the N′). If this principle applies, it will select a unique head. In fact, in this special case, we drop the BV link from the final DMRS because it is entirely predictable from the RSTR link.

In the case where there is no variable argument, we use the heuristic which generally holds in DELPH-IN grammars that the EPs which we wish to distinguish as heads in the DMRS do not share labels with their DMRS argument EPs (in contrast to intersective modifiers, which always share labels with their argument EPs). Heads may share labels with PPs which are syntactically arguments, but these have a semantics like PP modifiers, where the head is the preposition's EP argument. NP arguments are generally quantified and quantifiers scope freely. AP, VP and S syntactic arguments are always scopal. PPs which are not modifier-like are either scopal (small clauses) or NP-like (case marking Ps) and free-scoping. Thus, somewhat counter-intuitively, we can select the head EP from the set of EPs which share a label by looking for an EP which has no argument EPs in that set.

4.3 Some properties of DMRS

The MRS-to-DMRS procedure deterministically creates a unique DMRS. A converse DMRS-to-MRS procedure recreates the MRS (up to label, anchor

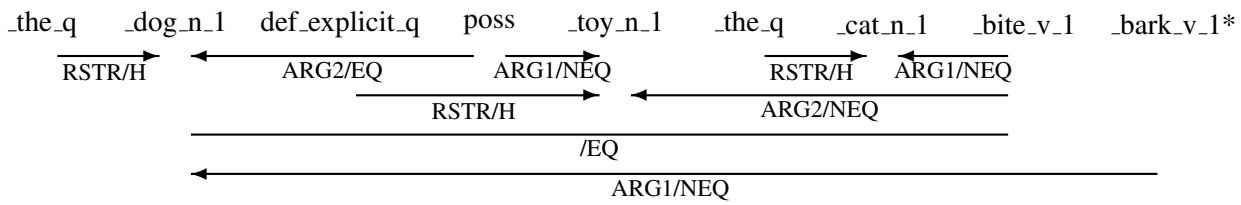

Figure 3: DMRS for 'The dog whose toy the cat bit barked.'

and variable renaming), though requiring the SEM-I to add the uninstantiated optional arguments.

I claimed above that DMRSs are an idealisation of semantic composition. A pure functor-argument application scheme would produce a tree which could be transformed into a structure where no dependent had more than one head. But in DMRS the notion of functor/head is more complex as determiners and modifiers provide slots in the RMRS algebra but not the index of the result. Composition of a verb (or any other functor) with an NP argument gives rise to a dependency between the verb and the head noun in the N′. The head noun provides the index of the NP's hook in composition, though it does not provide the ltop, which comes from the quantifier. However, because this ltop is not equated with any label, there is no direct link between the verb and the determiner. Thus the noun will have a link from the determiner and from the verb.

Similarly, if the constituents in composition were continuous, the adjacency condition would hold, but this does not apply because of the mechanisms for long-distance dependencies and the availability of the external argument in the hook.[8]

DMRS indirectly preserves the information about constituent structure which is essential for semantic interpretation, unlike some syntactic dependency schemes. In particular, it retains information about a quantifier's N′, since this forms the restrictor of the generalised quantifier (for instance *Most white cats are deaf* has different truth conditions from *Most deaf cats are white*). An interesting example of nominal modification is shown in Fig 3. Notice that *whose* has a decomposed semantics combining two non-lexeme predicates def_explicit_q and poss. Unusually, the relative clause has a gap which is not an argument of its semantic head (it's an argument of poss rather than bite_v_1). This means that when the relative clause is combined with the gap filler, the label equality and the argument instantiation correspond to different EPs. Thus there is a label equality which cannot be combined with an argument link and has to be represented by an undirected /EQ arc.

[8]Given that non-local effects are relatively circumscribed, it is possible to require adjacency in some parts of the DMRS. This leads to a technique for recording underspecification of noun compound bracketing, for instance.

5 Related work and conclusion

Hobbs (1985) described a philosophy of computational compositional semantics that is in some respects similar to that presented here. But, as far as I am aware, the Core Language Engine book (Alshawi, 1992) provided the first detailed description of a truly computational approach to compositional semantics: in any case, Steve Pulman provided my own introduction to the idea. Currently, the ParGram project also undertakes large-scale multilingual grammar engineering work: see Crouch and King (2006) and Crouch (2006) for an account of the semantic composition techniques now being used. I am not aware of any other current grammar engineering activities on the ParGram or DELPH-IN scale which build bidirectional grammars for multiple languages.

Overall, what I have tried to do here is to give a flavour of how compositional semantics and syntax interact in computational grammars. Analyses which look simple have often taken considerable experimentation to arrive at when working on a large-scale, especially when attempting cross-linguistic generalisations. The toy examples that can be given in papers like this one do no justice to this, and I would urge readers to try out the grammars and software and, perhaps, to join in.

Acknowledgements

Particular thanks to Emily Bender, Dan Flickinger and Alex Lascarides for detailed comments at very short notice! I am also grateful to many other colleagues, especially from DELPH-IN and in the Cambridge NLIP research group. This work was supported by the Engineering and Physical Sciences Research Council [grant numbers EP/C010035/1, EP/F012950/1].

References

Hiyan Alshawi, editor. 1992. *The Core Language Engine*. MIT Press.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proc. ACL-98*, pages 86–90, Montreal, Quebec, Canada. Association for Computational Linguistics.

Emily Bender, Dan Flickinger, and Stephan Oepen. 2002. The Grammar Matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proc. Workshop on Grammar Engineering and Evaluation, Coling 2002*, pages 8–14, Taipei, Taiwan.

Emily Bender. 2008. Evaluating a crosslinguistic grammar resource: A case study of Wambaya. In *Proc. ACL-08*, pages 977–985, Columbus, Ohio, USA.

John Carroll and Stephan Oepen. 2005. High efficiency realization for a wide-coverage unification grammar. In *Proc. IJCNLP05, Springer Lecture Notes in Artificial Intelligence, Volume 3651*, pages 165–176, Jeju Island, Korea.

John Carroll, Ann Copestake, Dan Flickinger, and Victor Poznanski. 1999. An efficient chart generator for (semi-)lexicalist grammars. In *Proc. 7th European Workshop on Natural Language Generation (EWNLG'99)*, pages 86–95, Toulouse.

Ann Copestake, Alex Lascarides, and Dan Flickinger. 2001. An algebra for semantic construction in constraint-based grammars. In *Proc. ACL-01*, Toulouse.

Ann Copestake, Dan Flickinger, Ivan A. Sag, and Carl Pollard. 2005. Minimal Recursion Semantics: an introduction. *Research on Language and Computation*, 3(2-3):281–332.

Ann Copestake. 2003. Report on the design of RMRS. DeepThought project deliverable.

Ann Copestake. 2007a. Applying robust semantics. In *Proc. PACLING 2007 — 10th Conference of the Pacific Association for Computational Linguistics*, pages 1–12, Melbourne.

Ann Copestake. 2007b. Semantic composition with (Robust) Minimal Recursion Semantics. In *Proc. Workshop on Deep Linguistic Processing, ACL 2007*, Prague.

Dick Crouch and Tracy Holloway King. 2006. Semantics via F-structure rewriting. In Miriam Butt and Tracy Holloway King, editors, *Proc. LFG06 Conference*, Universitat Konstanz. CSLI Publications.

Dick Crouch. 2006. Packed rewriting for mapping semantics and KR. In *Intelligent Linguistic Architectures Variations on Themes by Ronald M. Kaplan*, pages 389–416. CSLI Publications.

Anthony Davis. 2001. *Linking by Types in the Hierarchical Lexicon*. CSLI Publications.

David Dowty. 1991. Thematic proto-roles and argument selection. *Language*, 67(3):547–619.

Dan Flickinger and Emily Bender. 2003. Compositional semantics in a multilingual grammar resource. In *Proc. Workshop on Ideas and Strategies for Multilingual Grammar Development, ESSLLI 2003*, pages 33–42, Vienna.

Dan Flickinger, Emily Bender, and Stephan Oepen. 2003. MRS in the LinGO Grammar Matrix: A practical user's guide. http://tinyurl.com/crf5z7.

Dan Flickinger, Jan Tore Lønning, Helge Dyvik, Stephan Oepen, and Francis Bond. 2005. SEM-I rational MT — enriching deep grammars with a semantic interface for scalable machine translation. In *Proc. MT Summit X*, Phuket, Thailand.

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

Sanae Fujita, Francis Bond, Stephan Oepen, and Takaaki Tanaka. 2007. Exploiting semantic information for HPSG parse selection. In *Proc. Workshop on Deep Linguistic Processing, ACL 2007*, Prague.

Jerry Hobbs. 1985. Ontological promiscuity. In *Proc. ACL-85*, pages 61–69, Chicago, IL.

Alexander Koller and Alex Lascarides. 2009. A logic of semantic representations for shallow parsing. In *Proc. EACL-2009*, Athens.

Stephan Oepen and Jan Tore Lønning. 2006. Discriminant-based MRS banking. In *Proc. LREC-2006*, Genoa, Italy.

Martha Palmer, Dan Gildea, and Paul Kingsbury. 2005. The Proposition Bank: A corpus annotated with semantic roles. *Computational Linguistics*, 31(1).

Carl Pollard and Ivan Sag. 1994. *Head-driven Phrase Structure Grammar*. University of Chicago Press, Chicago.

Sergio Roa, Valia Kordoni, and Yi Zhang. 2008. Mapping between compositional semantic representations and lexical semantic resources: Towards accurate deep semantic parsing. In *Proc. ACL-08*, pages 189–192, Columbus, Ohio. Association for Computational Linguistics.

Antonio Sanfilippo. 1990. *Grammatical Relations, Thematic Roles and Verb Semantics*. Ph.D. thesis, Centre for Cognitive Science, University of Edinburgh.

Stefan Thater. 2007. *Minimal Recursion Semantics as Dominance Constraints: Graph-Theoretic Foundation and Application to Grammar Engineering*. Ph.D. thesis, Universität des Saarlandes.