

Aligning Formal Meaning Representations with Surface Strings for Wide-coverage Text Generation

Valerio Basile

Johan Bos

{v.basile, johan.bos}@rug.nl

Center for Language and Cognition Groningen (CLCG)

University of Groningen, The Netherlands

Abstract

Statistical natural language generation from abstract meaning representations presupposes large corpora consisting of text–meaning pairs. Even though such corpora exist nowadays, or could be constructed using robust semantic parsing, the simple alignment between text and meaning representation is too coarse for developing robust (statistical) NLG systems. By reformatting semantic representations as graphs, fine-grained alignment can be obtained. Given a precise alignment at the word level, the complete surface form of a meaning representations can be deduced using a simple declarative rule.

1 Introduction

Surface Realization is the task of producing fluent text from some kind of formal, abstract representation of meaning (Reiter and Dale, 2000). However, while it is obvious what the output of a natural language generation component should be, namely *text*, there is little to no agreement on what its input formalism should be (Evans et al., 2002). Since open-domain semantic parsers are able to produce formal semantic representations nowadays (Bos, 2008; Butler and Yoshimoto, 2012), it would be natural to see generation as a reversed process, and consider such semantic representations as input of a surface realization component.

The idea of using large text corpora annotated with formal semantic representations for robust generation has been presented recently (Basile and Bos, 2011; Wanner et al., 2012). The need for formal semantic representations as a basis for NLG was expressed already much earlier by Power (1999), who derives semantic networks enriched with scope information from knowledge representations for content planning. In this paper we take

a further step towards the goal of generating text from deep semantic representations, and consider the issue of aligning the representations with surface strings that capture their meaning.

First we describe the basic idea of aligning semantic representations (logical forms) with surface strings in a formalism-independent way (Section 2). Then we apply our method to a well-known and widely-used semantic formalism, namely Discourse Representation Theory (DRT), first demonstrating how to represent Discourse Representation Structures (DRSs) as graphs (Section 3) and showing that the resulting Discourse Representation Graphs (DRGs) are equivalent to DRSs but are more convenient to fulfill word-level alignment (Section 4). Finally, in Section 5 we present a method that generates partial surface strings for each discourse referent occurring in the semantic representation of a text, and composes them into a complete surface form. All in all, we think this would be a first and important step in surface realization from formal semantic representations.

2 Aligning Logic with Text

Several different formal semantic representations have been proposed in the literature, and although they might differ in various aspects, they also have a lot in common. Many semantic representations (or logical forms as they are sometimes referred to) are variants of first-order logic and share basic building blocks such as entities, properties, and relations, complemented with quantifiers, negation and further scope operators.

A simple snapshot of a formal meaning representation is the following (with symbols composed out of WordNet (Fellbaum, 1998) synset identifiers to abstract away from natural language):

$$blue\#a\#1(x) \wedge cup\#n\#1(x)$$

How could this logical form be expressed in natural language? Or put differently, how could we

realize the variable x in text? As simple as it is, x describes “a blue cup”, or if your target language is Italian, “una tazza blu”, or variants hereof, e.g. “every blue cup” (if x happens to be bound by universally quantified) or perhaps as “una tazza azzurra”, using a different adjective to express blueness. This works for simple examples, but how does it scale up to larger and more complex semantic representations?

In a way, NLG can be viewed as a machine translation (MT) task, but unlike translating from one natural language into another, the task is here to translate a formal (unambiguous) language into a natural language like English or Italian. Current statistical MT techniques are based on large parallel corpora of aligned source and target text. In this paper we introduce a method for precise alignment of formal semantic representations and text, with the purpose of creating a large corpus that could be used in NLG research, and one that opens the way for statistical approaches, perhaps similar to those used in MT.

Broadly speaking, alignments between semantic representations and surface strings can be made in three different ways. The simplest strategy, but also the least informative, is to align a semantic representation with a sentence or complete text without further information on which part of the representation produces what part of the surface form. This might be enough to develop statistical NLG systems for small sentences, but probably does not scale up to handle larger texts. Alternatively, one could devise more complex schemes that allow for a more fine-grained alignment between parts of the semantic representation and surface strings (words and phrases). Here there are two routes to follow, which we call the *minimal* and *maximal alignment*.

In maximal alignment, each single piece of the semantic representation corresponds to the words that express that part of the meaning. Possible problems with this approach are that perhaps not every bit of the semantic representation corresponds to a surface form, and a single word could also correspond to various pieces in the semantic representation. This is an interesting option to explore, but in this paper we present the alternative approach, minimal alignment, which is a method where every word in the surface string points to exactly one part of the semantic representation. We think this alignment method forms

a better starting point for the development of a statistical NLG component. With sufficient data in the form of aligned texts with semantic representations, these alignments can be automatically learned, thus creating a model to generate surface forms from abstract, logical representations.

However, aligning semantic representations with words is a difficult enterprise, primarily because formal semantic representations are not flat like a string of words and often form complex structures. To overcome this issue we represent formal semantic representations as a set of tuples. For instance, returning to our earlier example representation for “blue cup”, we could represent part of it by the tuples $\langle \text{blue}\#\text{a}\#1, \text{arg}, x \rangle$ and $\langle \text{cup}\#\text{n}\#1, \text{arg}, x \rangle$. For convenience we can display this as a graph (Figure 1).

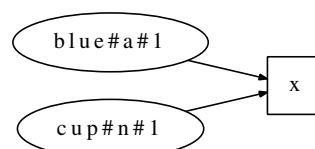


Figure 1: Logical form graph.

Note that in this example several tuples are not shown for clarity (such as conjunction and the quantifier). We show below that we can indeed represent every bit of semantic information in this format without sacrificing the capability of alignment with the text. The important thing now is to show how alignments between tuples and words can be realized, which is done by adding an element to each tuple denoting the surface string, for instance $\langle \text{cup}\#\text{n}\#1, \text{arg}, x, \text{“tazza”} \rangle$, as in Figure 2.

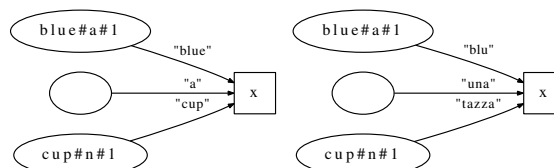


Figure 2: Logical form graphs aligned with surface forms in two languages.

We can further refine the alignment by saying something about the local order of surface expressions. Again, this is done by adding an element to the tuple, in this case one that denotes the local order of a logical term. We will make this clear by continuing with our example, where we add word order encoded as numerical indices in the tuple, e.g. $\langle \text{cup}\#\text{n}\#1, \text{arg}, x, \text{“tazza”}, 2 \rangle$, as Figure 3 shows.

From these graphs we can associate the term

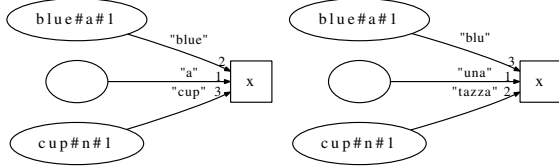


Figure 3: Encoding local word order.

x with the surface strings “a blue cup” and “una tazza blu”. But the way we express local order is not limited to words and can be employed for partial phrases as well, if one adopts a neo-Davidsonian event semantics with explicit thematic roles. This can be achieved by using the same kind of numerical indices already used for the alignment of words. The example in Figure 4 shows how to represent an event “hit” with its thematic roles, preserving their relative order. We call surface forms “partial” or “incomplete” when they contain variables, and “complete” when they only contain tokens. The corresponding partial surface form is then “ y hit z ”, where y and z are placeholders for surface strings.

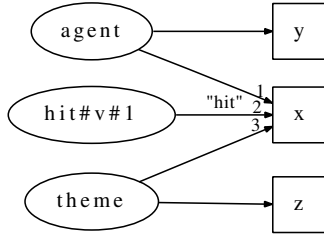


Figure 4: Graph for a neo-Davidsonian structure.

This is the basic idea of aligning surface strings with parts of a deep semantic representation. Note that precise alignment is only possible for words with a lexical semantics that include first-order variables. For words that introduce scope operators (negation particles, coordinating conjuncts) we can’t have the cake and eat it: specifying the local order with respect to an entity or event variable directly and at the same time associating it with an operator isn’t always possible. To solve this we introduce *surface tuples* that complement a semantic representation to facilitate perfect alignment. We will explain this in more detail in the following sections.

3 Discourse Representation Graphs

The choice of semantic formalism should ideally be independent from the application of natural language generation itself, to avoid bias and spe-

cific tailoring the semantic representation to one’s (technical) needs. Further, the formalism should have a model-theoretic backbone, to ensure that the semantic representations one works with actually have an interpretation, and can consequently be used in inference tasks using, for instance, automated deduction for first-order logic. Given these criteria, a good candidate is Discourse Representation Theory, DRT (Kamp and Reyle, 1993), that captures the meaning of texts in the form of Discourse Representation Structures (DRSs).

DRSs are capable of effectively representing the meaning of natural language, covering many linguistic phenomena including pronouns, quantifier scope, negation, modals, and presuppositions. DRSs are recursive structures put together by logical and non-logical symbols, as in predicate logic, and in fact can be translated into first-order logic formulas (Muskens, 1996). The way DRSs are nested inside each other give DRT the ability to explain the behaviour of pronouns and presuppositions (Van der Sandt, 1992).

Aligning DRSs with texts with fine granularity is hard because words and phrases introduce different kinds of semantic objects in a DRS: discourse referents, predicates, relations, but also logical operators such as negation, disjunction and implication that introduce embedded DRSs. A precise alignment of a DRS with its text on the level of words is therefore a non-trivial task.

To overcome this issue, we apply the idea presented in the previous section to DRSs, making all recursion implicit by representing them as directed graphs. We call a graph representing a DRS a Discourse Representation Graph (DRG, in short). DRGs encode the same information as DRSs, but are expressed as a set of tuples. Essentially, this is done by reification over DRSs — every DRS gets a unique label, and the arity of DRS conditions increases by one for accommodating a DRS label. This allows us to reformulate a DRS as a set of tuples.

A DRS is an ordered pair of discourse referents (variables over entities) and DRS-conditions. DRS-conditions are basic (representing properties or relations) or complex (to handle negation and disjunction). To reflect these different constructs, we distinguish three types of tuples in DRGs:

- $\langle K, \text{referent}, X \rangle$ means that X is a discourse referent in K (**referent tuples**);
- $\langle K, \text{condition}, C \rangle$ means that C is a condition

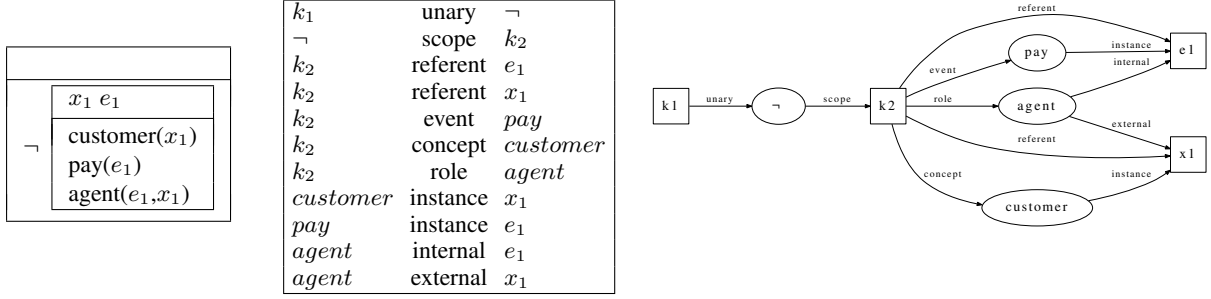


Figure 5: DRS and corresponding DRG (in tuples and in graph format) for “A customer did not pay.”

in K (**condition tuples**), with various subtypes: concept, event, relation, role, named, cardinality, attribute, unary, and binary;

- $\langle C, \text{argument}, A \rangle$ means that C is a condition with argument A (**argument tuples**), with the sub-types internal, external, instance, scope, antecedent, and consequence.

With the help of a concrete example, it is easy to see that DRGs have the same expressive power as DRSs. Consider for instance a DRS with negation, before and after labelling it (Figure 6):

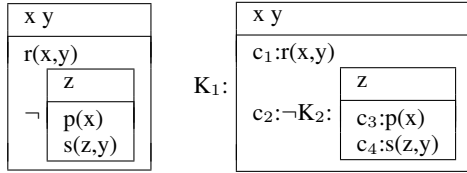


Figure 6: From DRS to DRG: labelling.

Now, from the labelled DRS we can derive the following three referent tuples: $\langle K_1, \text{referent}, x \rangle$, $\langle K_1, \text{referent}, y \rangle$, and $\langle K_2, \text{referent}, z \rangle$; the following four condition tuples: $\langle K_1, \text{relation}, c_1: r \rangle$, $\langle K_1, \text{unary}, c_2: \neg \rangle$, $\langle K_2, \text{concept}, c_3: p \rangle$, and $\langle K_2, \text{relation}, c_4: s \rangle$; and the following argument tuples: $\langle c_1: r, \text{internal}, x \rangle$, $\langle c_1: r, \text{external}, y \rangle$, $\langle c_2: \neg, \text{scope}, K_2 \rangle$, $\langle c_3: p, \text{instance}, x \rangle$, $\langle c_4: s, \text{internal}, z \rangle$, and $\langle c_4: s, \text{external}, y \rangle$. From these tuples, it is straightforward to recreate a labelled DRS, and by dropping the labels subsequently, the original DRS resurfaces again.

For the sake of readability we sometimes leave out labels in examples throughout this paper. In addition, we also show DRGs in graph-like pictures, where the tuples that form a DRG are the edges, and word-alignment information attached at the tuple level is shown as labels on the graph edges, as in Figure 9. In such graphs, nodes representing discourse referents are square shaped, and nodes representing conditions are oval shaped.

Note that labelling conditions is crucial to distinguish between syntactically equivalent conditions occurring in different (embedded) DRSs. Unlike Power’s scoped semantic network for DRSs, we don’t make the assumption that conditions appear in the DRS in which their discourse referents are introduced (Power, 1999). The example in Figure 6 illustrates that this assumption is not sound: the condition $p(x)$ is in a different DRS than where its discourse referent x is introduced. Further note that our reification procedure yields “flatter” representations than similar formalisms (Copestake et al., 1995; Reyle, 1993), and this makes it more convenient to align surface strings with DRSs with a high granularity, as we will show below.

4 Word-Aligned DRGs

In this section we show how the alignment between surface text and its logical representation is realized by adding information of the tuples that make up a DRG. This sounds more straightforward than it is. For some word classes this is indeed easy to do. For others we need additional machinery in the formalism. Let’s start with the straightforward cases. Determiners are usually associated with referent tuples. Content words, such as nouns, verbs, adverbs and adjectives, are typically directly associated with one-place relation symbols, and can be naturally aligned with argument tuples. Verbs are assigned to *instance* tuples linking its *event* condition; likewise, nouns are typically aligned to *instance* tuples which link discourse referents to the *concepts* they express; adjectives are aligned to tuples of *attribute* conditions. Finally, words expressing relations (such as prepositions), are attached to the *external* argument tuple linking the relation to the discourse referent playing the role of external argument.

Although the strategy presented for DRG–text

alignment is intuitive and straightforward to implement, there are surface strings that don't correspond to something explicit in the DRS. To this class belong punctuation symbols, and semantically empty words such as (in English) the infinitival particle, pleonastic pronouns, auxiliaries, *there* insertion, and so on. Furthermore, function words such as “not”, “if”, and “or”, introduce semantic material, but for the sake of surface string generation could be better aligned with the event that they take the scope of. To deal with all these cases, we extend DRGs with *surface tuples* of the form $\langle K, \text{surface}, X \rangle$, whose edges are decorated with the required surface strings. Figure 7 shows an example of a DRG extended with such surface tuples.

k_1	unary	$\neg$		
$\neg$	scope	k_2		
k_2	referent	e_1		
k_2	referent	x_1	1	A
k_2	event	pay		
k_2	concept	$customer$		
k_2	role	$agent$		
$customer$	instance	x_1	2	customer
pay	instance	e_1	4	pay
$agent$	internal	e_1	1	
$agent$	external	x_1		
k_2	surface	e_1	2	did
k_2	surface	e_1	3	not
k_2	surface	e_1	5	.

Figure 7: Word-aligned DRG for “A customer did not pay.” All alignment information (including surface tuples) is highlighted.

Note that surface tuples don't have any influence on the meaning of the original DRS – they just serve for the purpose of alignment of the required surface strings. Also note in Figure 7 the indices that were added to some tuples. They serve to express the local order of surface information.

Following the idea sketched in Section 2, the total order of words is transformed into a local ranking of edges relative to discourse referents. This is possible because the tuples that have word tokens aligned to them always have a discourse referent as third element (the *head* of the directed edge, in terms of graphs). We group tuples that share the same discourse referent and then assign indices reflecting the relative order of how these tuples are realized in the original text.

Illustrating this with our example in Figure 7,

we got two discourse referents: x_1 and e_1 . The discourse referent x_1 is associated with three tuples, of which two are indexed (with indices 1 and 2). Generating the surface string for x_1 succeeds by traversing the edges in the order specified, resulting in [A, customer] for x_1 . The referent e_1 associates with six tuples, of which four are indexed (with indices 1–4). The order of these tuples would yield the partial surface string [x₁, did, not, pay, .] for e_1 .

Note that the manner in which DRSs are constructed during analysis ensures that all discourse referents are linked to each other by taking the transitive closure of all binary relations appearing in a DRS, and therefore we can reconstruct the total order from composing the local orders. In the next section we explain how this is done.

5 Surface Composition

In this section we show in detail how surface strings can be generated from word-aligned DRGs. It consists of two subsequent steps. First, a surface form is associated with each discourse referent. Secondly, surface forms are put together in a bottom-up fashion, to generate the complete output. During the composition, all of the discourse referents are associated with their own surface representation. The surface form associated with the discourse unit that contains all other discourse units is then the text aligned with the original DRG.

Surface forms of discourse referents are lists of tokens and other discourse referents. Recall that the order of the elements of a discourse referent's surface form is reflected by the local ordering of tuples, as explained in the previous section, and tuples with no index are simply ignored when reconstructing surface strings.

The surface form is composed by taking each tuple belonging to a specific discourse referent, in the correct order, and adding the tokens aligned with the tuple to a list representing the surface string for that discourse referent. An important part of this process is that binary DRS relations, represented in the DRG by a pair of internal and external argument tuple, are followed unidirectionally: if the tuple is of the *internal* type, then the discourse referent on the other end of the relation (i.e. following its *external* tuple edge) is added to the list. Surface forms for embedded DRSs include the discourse referents of the events

$$\begin{array}{c}
\frac{x_1 : \text{Michelle} \quad e_1 : x_1 \text{ thinks } p_1}{e_1 : \text{Michelle thinks } p_1} \quad \frac{p_1 : \text{that } e_2 \quad \frac{x_2 : \text{Obama} \quad e_2 : x_2 \text{ smokes .}}{e_2 : \text{Obama smokes .}}}{p_1 : \text{that Obama smokes .}} \\
\hline
k_1 : e_4 \quad e_1 : \text{Michelle thinks that Obama smokes .} \\
\hline
k_1 : \text{Michelle thinks that Obama smokes .}
\end{array}$$

Figure 8: Surface composition of embedded structures.

they contain.

Typically, discourse units contain exactly one event (the main event of the clause). Phenomena such as gerunds (e.g. “the laughing girl”) and relative clauses (e.g. “the man who smokes”) may introduce more than one event in a discourse unit. To ensure correct order and grouping, we borrow a technique from description logic (Horrocks and Sattler, 1999) and invert roles in DRGs. Rather than representing “the laughing girl” as $[\text{girl}(x) \wedge \text{agent}(e,x) \wedge \text{laugh}(e)]$, we represent it as $[\text{girl}(x) \wedge \text{agent}^{-1}(x,e) \wedge \text{laugh}(e)]$, making use of $R(x,y) \equiv R^{-1}(y,x)$ to preserve meaning. This “trick” ensures that we can describe the local order of noun phrases with relative clauses and alike.

To wrap things up, the *composition* operation is used to derive complete surface forms for DRGs. Composition puts together two surface forms, where one of them is complete, and one of them is incomplete. It is formally defined as follows:

$$\frac{\rho_1 : \tau \quad \rho_2 : \Lambda_1 \rho_1 \Lambda_2}{\rho_2 : \Lambda_1 \tau \Lambda_2} \quad (1)$$

where ρ_1 and ρ_2 are discourse referents, τ is a list of tokens, and Λ_1 and Λ_2 are lists of word tokens and discourse referents. In the example from Figure 7, the complete surface form for the discourse unit k_1 is derived by means of composition as formulated in (1) as follows:

$$\frac{x_1 : \text{A customer} \quad e_1 : x_1 \text{ did not pay}}{e_1 : \text{A customer did not pay .}} \\
\hline
k_2 : e_1 \quad k_2 : \text{A customer did not pay .}$$

The procedure for generation described here is reminiscent of the work of (Shieber, 1988) who also employs a deductive approach. In particular our composition operation can be seen as a simplified *completion*.

Going back to the example in Section 4, substituting the value of x_1 in the incomplete surface form of e_1 produces the surface string $[A, \text{customer}, \text{did}, \text{not}, \text{pay}, .]$ for e_1 .

6 Selected Phenomena

We implemented a first prototype using our alignment and realization method and tested it on examples taken from the Groningen Meaning Bank, a large annotated corpus of texts paired with DRSs (Basile et al., 2012). Naturally, we came across phenomena that are notoriously hard to analyze. Most of these we can handle adequately, but some we can’t currently account for and require further work.

6.1 Embedded Clauses

In the variant of DRT that we are using, propositional arguments of verbs introduce embedded DRSs associated with a discourse referent. This is a good test for our surface realization formalism, because it would show that it is capable of recursively generating embedded clauses. Figure 9 shows the DRG for the sentence “Michelle thinks that Obama smokes.”

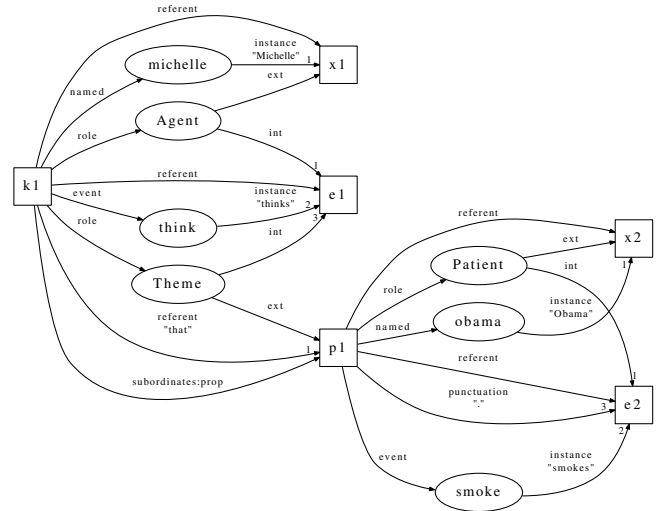


Figure 9: Word-aligned DRG for the sentence “Michelle thinks that Obama smokes.”

Here the surface forms of two discourse units (main and embedded) are generated. In order to generate the complete surface form, first the embedded clause is generated, and then composed

with the incomplete surface form of the main clause. As noted earlier, during the composition process, the *complete* surface form for each discourse referent is generated (Figure 8), showing a clear alignment between the entities of the semantic representation and the surface forms they represent.

6.2 Coordination

Coordination is another good test case for a linguistic formalism. Consider for instance “Subsistence fishing and commercial trawling occur within refuge waters”, where two noun phrases are coordinated, giving rise to either a distributive (introducing two events in the DRS) or a collective interpretation (introducing a set formation of discourse referents in the DRS). We can account for both interpretations (Figure 10).

Note that, interestingly, using the distributive interpretation DRG as input to the surface realization component could result, depending on how words are aligned, in a surface form “fishing occurs and trawling occurs”, or as “fishing and trawling occur”.

6.3 Long-Distance Dependencies

Cases of extraction, for instance with WH-movement, could be problematic to capture with our formalism. This is in particular an issue when extraction crosses more than one clause boundary, as in “Which car does Bill believe John bought”. Even though these cases are rare in the real world, a complete formalism for surface realization must be able to deal with such cases. The question is whether this is a separate generation task in the domain of syntax (White et al., 2007), or whether the current formalism needs to be adapted to cover such long-distance dependencies. Another range of complications are caused by discontinuous constituents, as in the Dutch sentence “Ik heb kaartjes gekocht voor Berlijn” (literally: “I have tickets bought for Berlin”), where the prepositional phrase “voor Berlijn” is an argument of the noun phrase “kaartjes”. In our formalism the only alignment possible would result in the sentence “Ik heb kaartjes voor Berlijn gekocht”, which is arguably a more fluent realization of the sentence, but doesn’t correspond to the original text. If one uses the original text as gold standard, this could cause problems in evaluation. (One could also benefit from this deficiency, and use it to generate

more than one gold standard surface string. This is something to explore in future work.)

6.4 Control Verbs

In constructions like “John wants to swim”, the control verb “wants” associates its own subject with the subject of the infinitival clause that it has as argument. Semantically, this is realized by variable binding. Generating an appropriate surface form for semantic representation with controlled variables is a challenge: a naive approach would generate “John wants John to swim”. One possible solution is to add another derivation rule for surface composition dedicated to deal with cases where a placeholder variable occurs in more than one partial surface form, substituting a null string for a variable following some heuristic rules. A second, perhaps more elegant solution is to integrate a language model into the surface composition process.

7 Related work

Over the years, several systems have emerged that aim at generate surface forms from different kind of abstract input representations. An overview of the state-of-the-art is showcased by the submissions to the Surface Realization Shared Task (Belz et al., 2012). Bohnet *et al.* (2010), for instance, employ deep structures derived from the CoNLL 2009 shared task, essentially sentences annotated with shallow semantics, lemmata and dependency trees; as the authors state, these annotations are not made with generation in mind, and they necessitate complex preprocessing steps in order to derive syntactic trees, and ultimately surface forms. The format presented in this work has been especially developed with statistical approaches in mind.

Nonetheless, there is very little work on robust, wide-scale generation from DRSs, surprisingly perhaps given the large body of theoretical research carried out in the framework of Discourse Representation Theory, and practical implementations and annotated corpora of DRSs that are nowadays available (Basile et al., 2012). This is in contrast to the NLG work in the framework of Lexical Functional Grammar (Guo et al., 2011).

Flat representation of semantic representations, like the DRGs that we present, have also been put forward to facilitate machine translation (Schiehlen et al., 2000) and for evaluation purposes (Allen et al., 2008), and semantic parsing

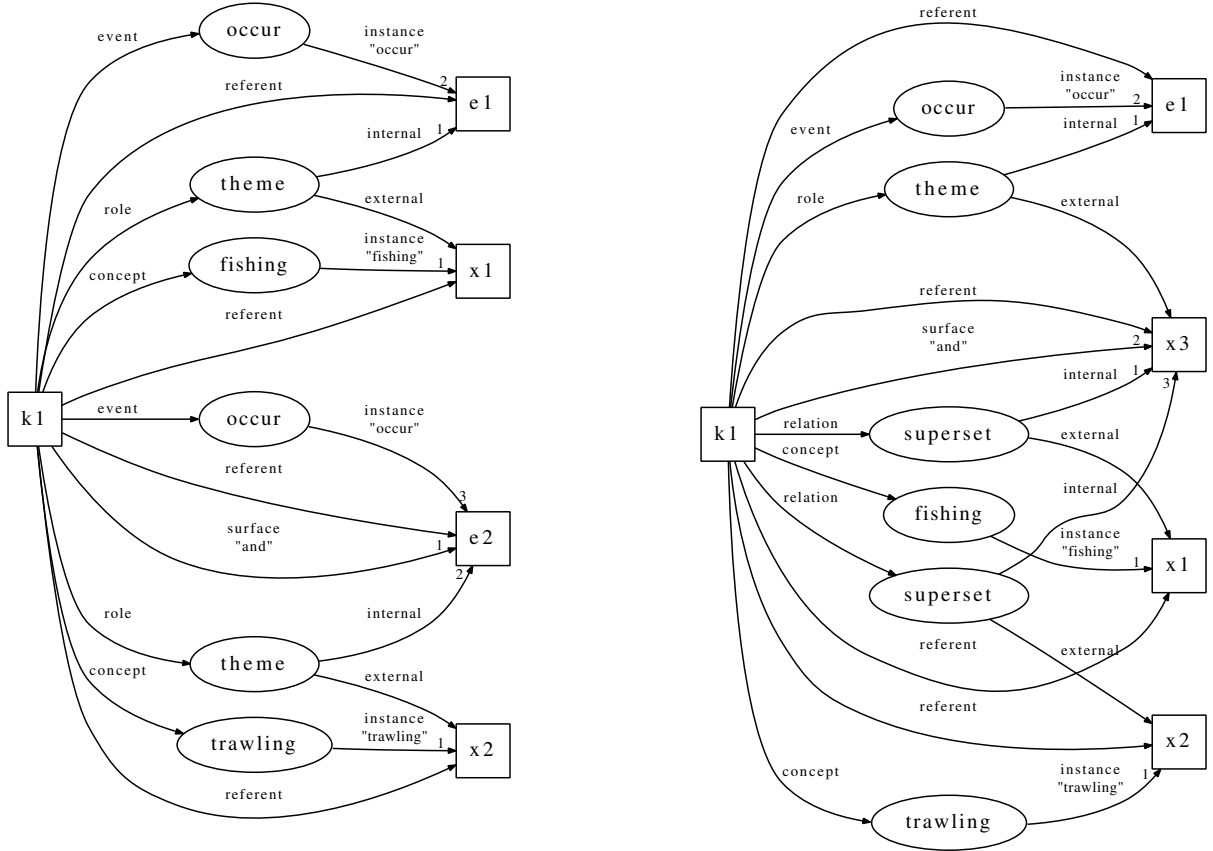


Figure 10: Analysis of NP coordination, in a distributive (left) and a collective interpretation (right).

(Le and Zuidema, 2012) just because they’re easier and more efficient to process. Packed semantic representations (leaving scope underspecified) also resemble flat representations (Copestake et al., 1995; Reyle, 1993) and can be viewed as graphs, however they show less elaborated reification than the DRGs presented in this paper, and are therefore less suitable for precise alignment with surface strings.

8 Conclusion

We presented a formalism to align logical forms, in particular Discourse Representation Structures, with surface text strings. The resulting graph representations (DRGs), make recursion implicit by reification over nested DRSs. Because of their “flat” structure, DRGs can be precisely aligned with the text they represent at the word level. This is key to open-domain statistical Surface Realization, where words are learned from abstract, syntactic or semantic, representations, but also useful for other applications such as learning semantic representations directly from text (Le and Zuidema, 2012). The actual alignment between

the tuples that form a DRG and the surface forms they represent is not trivial, and requires to make several choices.

Given the alignment with text, we show that it is possible to directly generate surface forms from automatically generated word-aligned DRGs. To do so, a declarative procedure is presented, that generates complete surface forms from aligned DRGs in a compositional fashion. The method works in a bottom-up way, using discourse referents as starting points, then generating a surface form for each of them, and finally composing all of the surface form together into a complete text. We are currently building a large corpus of word-aligned DRSs, and are investigating machine learning methods that could automatically learn the alignments.

Surprisingly, given that DRT is one of the best studied formalisms in formal semantics, there isn’t much work on generation from DRSs so far. The contribution of this paper presents a method to align DRSs with surface strings, that paves the way for robust, statistical methods for surface generation from deep semantic representations.

References

- James F. Allen, Mary Swift, and Will de Beaumont. 2008. Deep Semantic Analysis of Text. In Johan Bos and Rodolfo Delmonte, editors, *Semantics in Text Processing. STEP 2008 Conference Proceedings*, volume 1 of *Research in Computational Semantics*, pages 343–354. College Publications.
- Valerio Basile and Johan Bos. 2011. Towards generating text from discourse representation structures. In *Proceedings of the 13th European Workshop on Natural Language Generation (ENLG)*, pages 145–150, Nancy, France.
- Valerio Basile, Johan Bos, Kilian Evang, and Noortje Venhuizen. 2012. Developing a large semantically annotated corpus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC 2012)*, pages 3196–3200, Istanbul, Turkey.
- Anja Belz, Bernd Bohnet, Simon Mille, Leo Wanner, and Michael White. 2012. The surface realisation task: Recent developments and future plans. In Barbara Di Eugenio, Susan McRoy, Albert Gatt, Anja Belz, Alexander Koller, and Kristina Striegnitz, editors, *INLG*, pages 136–140. The Association for Computer Linguistics.
- Bernd Bohnet, Leo Wanner, Simon Mille, and Alicia Burga. 2010. Broad coverage multilingual deep sentence generation with a stochastic multi-level realizer. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 98–106.
- Johan Bos. 2008. Wide-Coverage Semantic Analysis with Boxer. In J. Bos and R. Delmonte, editors, *Semantics in Text Processing. STEP 2008 Conference Proceedings*, volume 1 of *Research in Computational Semantics*, pages 277–286. College Publications.
- Alastair Butler and Kei Yoshimoto. 2012. Banking meaning representations from treebanks. *Linguistic Issues in Language Technology - LiLT*, 7(1):1–22.
- Ann Copestake, Dan Flickinger, Rob Malouf, Susanne Riehemann, and Ivan Sag. 1995. Translation using Minimal Recursion Semantics. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 15–32, University of Leuven, Belgium.
- Roger Evans, Paul Piwek, and Lynne Cahill. 2002. What is NLG? In *Proceedings of the Second International Conference on Natural Language Generation*, pages 144–151.
- Christiane Fellbaum, editor. 1998. *WordNet. An Electronic Lexical Database*. The MIT Press.
- Yuqing Guo, Haifeng Wang, and Josef van Genabith. 2011. Dependency-based n-gram models for general purpose sentence realisation. *Natural Language Engineering*, 17:455–483.
- Ian Horrocks and Ulrike Sattler. 1999. A description logic with transitive and inverse roles and role hierarchies. *Journal of logic and computation*, 9(3):385–410.
- Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT*. Kluwer, Dordrecht.
- Phong Le and Willem Zuidema. 2012. Learning compositional semantics for open domain semantic parsing. Forthcoming.
- Reinhard Muskens. 1996. Combining Montague Semantics and Discourse Representation. *Linguistics and Philosophy*, 19:143–186.
- R. Power. 1999. Controlling logical scope in text generation. In *Proceedings of the 7th European Workshop on Natural Language Generation*, Toulouse, France.
- E. Reiter and R. Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press.
- Uwe Reyle. 1993. Dealing with Ambiguities by Underspecification: Construction, Representation and Deduction. *Journal of Semantics*, 10:123–179.
- Michael Schiehlen, Johan Bos, and Michael Dorna. 2000. Verbmobil interface terms (vits). In Wolfgang Wahlster, editor, *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer.
- Stuart M. Shieber. 1988. A uniform architecture for parsing and generation. In *Proceedings of the 12th conference on Computational linguistics - Volume 2, COLING '88*, pages 614–619, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Rob A. Van der Sandt. 1992. Presupposition Projection as Anaphora Resolution. *Journal of Semantics*, 9:333–377.
- Leo Wanner, Simon Mille, and Bernd Bohnet. 2012. Towards a surface realization-oriented corpus annotation. In *Proceedings of the Seventh International Natural Language Generation Conference, INLG '12*, pages 22–30, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Michael White, Rajakrishnan Rajkumar, and Scott Martin. 2007. Towards broad coverage surface realization with CCG. In *Proceedings of the Workshop on Using Corpora for NLG: Language Generation and Machine Translation (UCNLG+MT)*.