Annotations that effectively contribute to semantic interpretation

Harry Bunt

Abstract This chapter presents a new perspective on the use of semantic annotations. It is argued that semantic annotations should themselves have a semantics in order to be really useful. It is shown that, when this is the case, the information in a semantic annotation can be effectively combined with the results of compositional semantic analysis, with the effect of removing some of the underspecification in a compositional interpretation, or narrowing down to one that is appropriate in a given context.

1 Introduction: Functions of Semantic Annotations

Annotations add information to a primary text. In the pre-digital age, annotations took the form of bibliographical, historical, or interpretative notes in the margin or in footnotes. In the digital age, annotations take on a different form, but their function is essentially the same: they add information to a given text.

An annotation that does not add any information would seem not make much sense, but consider the following example of the annotation of a temporal expression using TimeML (Pustejovsky et al., 2003):[1]

(1)
```
<timeml>
  The CEO announced that he would resign as of
  <TIMEX3 tid="t1" type="date" value="2008-12-01"/>
   the first of December 2008
  </TIMEX3>
 </timeml>
```

Tilburg Center for Cognition and Communication (TiCC) and Department of Philosophy, Tilburg University, Netherlands, e-mail: harry.bunt@uvt.nl

[1] For simplicity, the annotations of the events that are mentioned in this sentence and the way they are linked to the date that is mentioned, are suppressed here.

In this annotation, the subexpression (2) adds to the noun phrase *the first of December 2008* the information that his phrase describes the date “2008-12-01”.

(2)
```
<TIMEX3 tid="t1" type="date" value="2008-12-01/>
```

This does not add any information; it rather paraphrases the noun phrase in TimeML. This could be useful if the expression in the annotation language had a formally defined semantics, which could be used directly by computer programs for applications like information extraction or question answering. Unfortunately, TimeML is just a particular form of XML, and as such does not have a semantics.

A case where the annotation of a date as in (1) *does* add something, is the following.

(3) Mr Brewster called a staff meeting today.

In the absence of context information we do not know which date *today* refers to; in this case the annotation (4) would be informative.

(4)
```
<timeml>
  Mr Brewster called a staff meeting
  <TIMEX3 tid=t1 type="date" value="2012-05-14"/>
    today
  </TIMEX3>
</timeml>
```

Note that the annotations in TimeML (1) and (4) are ‘old-fashioned’ in the sense that the `TIMEX3` element is wrapped around the annotated string, so the annotations are inserted in the primary text, similar to the annotations in pre-digital times that were inserted in the same printed text. Modern annotation methods prefer a ‘stand-off’ approach, where annotations are contained in a separate file and point to locations in the primary text. For example, instead of the TIMEX3 element in (1), an element is used as in (5), where the attribute `@target` points to the sequence `#w10...#w14` of word tokens that form the string *the first of December 2008*. In addition to respecting the integrity of the original text, this has the advantage of allowing multiple annotations linked to the same primary text.

(5)
```
<TIMEX3 xml:id="t1" target="#w10...#w14" type="date"
  value="2008-12-01"/>
```

The examples in (1) and (4) illustrate two different functions that semantic annotations may have: *recoding* information contained in a natural language expression in a formal annotation language, and *interpreting* a context-dependent natural language expression. This is for instance also the function of coreference annotations, as illustrated in (6), and of the markup of discourse connectives in the Penn Discourse Treebank (PDTB, Prasad et al., 2008), illustrated in (7).[2]

---

[2] The annotation in (7) uses a modified version of the PDTB representation, following Bunt, Prasad & Joshi (2012).

(6) a. Robin looked at Chris. She seems happy, he thought.

b.
```
<refml>
  <refEntity xml:id="r1" target="#w1" name="robin"/>
  <refEntity xml:id="r2" target="#w4" name="chris"/>
  <refEntity xml:id="r3" target="#w5"
    natGender="female"/>
  <refEntity xml:id="r4" target="#w8"
    natGender="male"/>
  <refLink anaphor="#r3" antecedent="#r2"
   relType="identity"/>
  <refLink anaphor="#r4" antecedent="#r1"
   relType="identity"/>
</refml>
```

This annotation provides the information that *She* is interpreted as indicating Chris (and thus that Chris is a female person; from which it follows that *he* does not refer to Chris but rather to Robin, and that Robin is a male person).

The annotations (4) - (5) and (6) are especially useful if the information which they contain about the interpretation of deictic and anaphoric expressions can be combined effectively with the interpretation of the rest of the sentence. Applying a syntactic parser and a compositional semantic analyzer to the sentence (3), for example, will lead to a semantic representation which leaves the date indicated by *today* unspecified. Such a representation is underspecified in the sense that it does not contain sufficient information to compute its truth value. The information in an underspecified semantic representation (USR) and that in a semantic annotation can be effectively combined if the annotation has a well-defined semantics of its own, so once again we see that the usefulness of a semantic annotation depends on whether it has a formal semantics.

There is a third function that semantic annotations may have, namely to make explicit how two subexpressions of a natural language expression are semantically related, or what is the function of a subexpression. This is illustrated in (7) for the function of a discourse connective (temporal or causal sense of *since*); in (8) (discussed in Section 3.3.2) for the implicit coherence relation connecting two sentences in a discourse; in (9) (discussed in more detail in Section 3.3) for the function of a temporal expression (*at six o'clock* indicating the time of occurrence of the *set*-event or the time at which the alarm is to sound); and in (10) for the semantic role of the referent of a noun phrase.

(7) a. 1. *since* as a temporal discourse connective:
The Mountain View, Calif., company has been receiving 1,000 calls a day about the product since *it was demonstrated at a computer publishing conference several weeks ago.*

2. *since* as a causal discourse connective:
It was a far safer deal for lenders since *NWA had a healthier cash flow and more collateral on hand.*

b. Annotation of (7a1):

```
<dRelML>
  <discourseRelation xml:id="dr1"
   arg1="#a1" arg2="#a2" rel="#r1"/>
  <dRelArgument xml:id="a1"target="#w1...#w14"/>
  <dRelArgument xml:id="a2" target="#w16...#w26"/>
  <explRel xml:id="r1" target="#w15"
   sense="succession"/>
</dRelML>
```

(8) Some have raised their cash positions to record levels. [Implicit=because] High cash positions help buffer a fund when the market falls.

(9) Henry set the alarm at six o'clock

(10) a. He drew a gun.

b. First interpretation (a gun is taken out of its holster):

```
<xml>
 <refEntity xml:id="p1" target="#w1" natGender="male"/>
 <event xml:id="e1" target="#w2" pred="draw1"/>
 <refEntity xml:id="p2" target="#w3 #w4" pred="gun"/>
 <semRole event="#e1" participant="#p1"
  relType="agent"/>
 <semRole event="#e1" participant="#p2"
  relType="theme"/>
</xml>
```

c. Second interpretation (a drawing is made of a gun):

```
<xml>
 <refEntity xml:id="p1" target="#w1" natGender="male"/>
 <event xml:id="e1" target="#w2" pred="draw2"/>
 <refEntity xml:id="p2" target="#w3 #w4" pred="gun"/>
 <semRole event="#e1" participant="#p1"
  relType="agent"/>
 <semRole event="#e1" participant="#p2"
  relType="result"/>
</xml>
```

The annotation in (10b) represents the interpretation where a gun was taken out of its holster; the one in (10c) where a drawing was made of a gun.

In sum, a semantic annotation of an expression E in a primary text may have the following functions:

a. **Recoding**: re-expression of the meaning of E in the annotation language;
b. **Contextualization**: specification of the interpretation of a context-specific deictic or anaphoric expressions E;

c. **Explicitation**: representation of an implicit semantic relation or function of one or more subexpressions of E.

Semantic annotations nearly always have the second or third function; this is where their usefulness mainly lies, and what the rest of this chapter will focus on. We have seen that semantic annotations that have the first function do not make much sense if they don't have a semantics, and that the usefulness of semantic annotations having the second or third function also depends on having a formal semantics (see also Bunt & Romary, 2002).

The combination of annotations and USRs is optimally facilitated when the semantics of annotation structures is defined via a translation into the same format as that used in USRs. Bunt (2007a) has shown that an 'interpretation-by-translation' semantics can be defined for TimeML, by means of a systematic, compositional translation of TimeML expressions into Discourse Representation Structures (DRSs, Kamp & Reyle 1993). In Section 3 of this chapter we will show how DRSs interpreting semantic annotations can effectively be combined with underspecified DRSs constructed by a compositional semantic analyzer.

In Section 2 we first consider some work concerned with the design of semantic annotation languages that have a formal semantics.

2 The semantics of semantic annotations

2.1 Interpreting annotations expressed in XML

Attempts to provide a semantics for semantic annotations include the Interval Temporal Logic semantics for TimeML by Pratt-Hartman (2007); the event-based semantics for TimeML by Bunt & Overbeeke (2008a), and other attempts to formally interpret temporal annotations by Katz (2007) and Lee (2008). The most elaborate proposal for a semantics of semantic annotation is formulated in Bunt (2007a) and Bunt & Overbeeke (2008b), where a semantic annotation language is presented with a formal semantics, that integrates temporal information, semantic roles, and coreference relations. These proposals all involve a translation of semantic annotations into first-order logic; however it has been shown to be very hard to achieve this in a satisfactory, compositional manner, where the translation of an annotation structure would be systematically constructed from the translations of its components (see Lee, 2008; Bunt 2011).

Bunt (2011) provides a DRS-based semantics for (a revised version of) ISO-TimeML. , the annotation language that forms part of the ISO 24617-2 standard for the annotation of time and events. While formally equivalent to first-order logic, the representation formalism of DRSs offers an attractive alternative, since it was designed to facilitate the incremental construction of semantic representations. For annotations which are expressed in XML, as is the case for ISO-TimeML annotations, a semantic interpretation via translation into DRSs can exploit the existence of certain structural correspondences between XML expressions and DRSs. Semantic annotations such as (6b), (7b), (10b) and (10c), consist of XML elements of two kinds: (A) those which associate semantic information with a stretch of primary text, that is identified by the value of the `@target` attribute; and (B) those which contain semantic information about an implicit relation (hence having no `@target` attribute) between two stretches of primary data.[3] These two kinds of elements can be translated into DRSs as follows.

A. An XML element of the form `<ENTITY xml:id="id1" target="#m1" attribute_1="val_1" ...attribute_n="val_n"/>` can be translated into a DRS which introduces a discourse referent that corresponds to the value of the attribute `@xml:id`, and for every feature specification `attribute_i="val_i"` contains a corresponding condition of the form $a'_i(x, v'_i)$, where a'_i translates $\texttt{attribute}_i$, x is the newly introduced discourse referent, and v'_i translates `val_i`.
For example, `<refEntity xml:id="#r1" target="#m1" name="robin">` ⇝

x
NAME(x, robin)

.

B. An XML element of the form `<RELATION attribute_1="val_1" ... attribute_n=" val_n"/>` can be translated into a DRS which introduces two discourse referents, and for each feature specification `attribute_i = "val_i"` contains a condition of the form $a'_i(x, y)$, where a'_i is the translation of $\texttt{attribute}_i$, and x and y are the two newly introduced discourse referents.[4]
For example, `<TIME_ANCHORING eventID="#e1" relatedToTime="#t1" relType="before"/>` ⇝

e, t
BEFORE(e, t)

These correspondences make it attractive to interpret annotations expressed in XML via a translation into DRSs. According to the Linguistic Annotation Framework (LAF, ISO 24612:2011), however, an annotation standard should not be defined at the level of representation formats, like XML, but at a more abstract level. The semantics of a annotations should therefore be defined likewise at a more abstract level than that of XML. In the next subsection we will see that systematic correspondences can also be established between abstract annotation structures and DRSs.

---

[3] A relation between two stretches of primary data which is explicitly expressed in the primary text corresponds to an XML element of type A. Here we consider only XML elements of type A which have an XML identifier as value of the attribute `@xml:id`, and elements of type B which have no such identifier. For other cases see Bunt (2013a).

[4] For certain attributes which have a particular status the DRS interpretation of a specification `attribute_i = "val_i"` has to be stipulated separately. An example is the TimeML attribute `@polarity`, of which a specification of the value `negative` gives rise to the negation of the DRS interpreting the rest of the XML element in which it occurs.

2.2 The design of semantic annotation languages

2.2.1 The CASCADES design methodology

The Linguistic Annotation Framework draws a distinction between the concepts of *annotation* and *representation*. The term 'annotation' refers to the linguistic information that is added to segments of primary data, independent of the format in which the information is represented, while the term 'representation' refers to the format in which an annotation is rendered, independent of its content. According to LAF, *annotations* are the proper level of standardization, rather than representations.

In order to (a) comply with the Linguistic Annotation Framework, and (b) satisfy the requirement that semantic annotations should have a semantics, we have developed a methodology for defining languages for semantic annotation, called 'CASCADES' (**C**onceptual analysis, **A**bstract syntax, **S**emantics, and **C**oncrete syntax for **A**nnotation language **DES**ign); Bunt (2010; 2013a.) This approach introduces in the definition of an annotation language a component which specifies the categories of linguistic information that can be used to build semantic annotations, and their possible combinations. This component is called an *abstract syntax*; it specifies the set of possible annotations in abstract, set-theoretical terms. To avoid overloading the term 'annotation', we will use the term 'annotation structure' for the set-theoretical constructs defined by an abstract syntax. Following this approach, the annotation language definition has three parts: (1) an abstract syntax, defining annotation structures; (2) the specification of a representation format for annotation structures, called a 'concrete syntax'; and (3) a semantics. The semantics is defined for the *abstract* rather than the concrete syntax; this has the important advantage that any concrete syntax which specifies a way of representing the annotation structures defined by the abstract syntax inherits the same semantics, from which it follows that alternative representation formats are semantically equivalent, and hence convertible from one to another (see Bunt, 2010; 2013a for formal definitions and proofs).

The distinction between abstract and concrete syntax, which is at the heart of the CASCADES approach, with the definition of a semantics for an abstract syntax, was developed during the project of defining an ISO standard for the annotation of time and events, in order to make this standard compatible with the Linguistic Annotation Framework. More recently, the CASCADES method was developed further (Bunt, 2013a) by specifying in some detail the steps of (1) defining an abstract syntax given a conceptual analysis of the annotation task; (2) defining the semantics of a given abstract syntax; (3) and specifying a XML-based concrete syntax given an abstract syntax. Moreover, steps backward were defined for feedback loops in this process, as visualized in Figure 1. Using these steps, the CASCADES method has been applied in the development of ISO standard 24617-2 for dialogue act annotation, resulting in the 3-part definition of the Dialogue Act Markup Language DiAML (see Bunt et al., 2012; Bunt et al., 2012; and Bunt, 2013b). The approach is currently applied in ISO projects for defining standards for the annotation of discourse relations (see Bunt, Prasad and Joshi, 2012), semantic roles (see Bunt and Palmer, 2013), and spatial information (Pustejovsky et al., 2012).

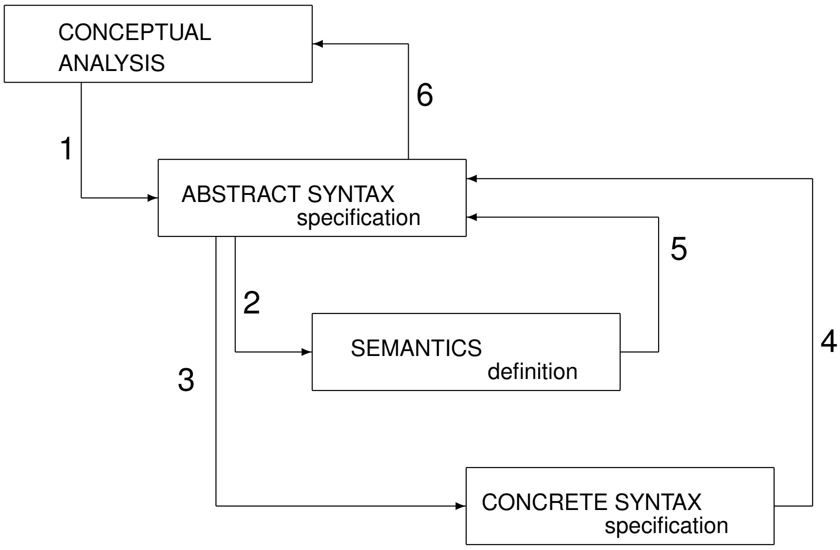

Fig. 1 Steps in the CASCADES model.

In the rest of this section we summarize the application of CASCADES to the definition of an abstract syntax and its semantics for the ISO-TimeML language.

2.2.2 The case of ISO-TimeML

Abstract syntax

An abstract syntax specification consists of two parts, a *conceptual inventory*, specifying the elements from which annotation structures are built up, and a specification of the possible ways of combining these elements into annotation structures.

The conceptual inventory for ISO-TimeML consists of finite sets of elements called *'event types', 'time points', 'tenses', 'aspects', 'temporal relations', 'temporal units', 'aspectual relations',* and *'event-subordination relations'*.

An annotation structure is a set of two kinds of structures, built up from elements of the conceptual inventory: *entity structures* and *link structures*. An entity structure contains information about a segment of primary text; a link structure contains information about the relation between two (or more) segments of primary text. An entity structure is formally a pair $\langle s, a \rangle$, where s identifies a segment of source text[5] and a

[5] Segments of source text may be identified directly (see TEI, 2009) or via the output of another layer of processing, such as a tokeniser.

is a set-theoretical construct whose elements belong to the conceptual inventory. In the case of ISO-TimeML, a is simply an n-tuple of such concepts.[6] A link structure is formally a triple $\langle \epsilon, E, \rho \rangle$, consisting of an entity structure ϵ, a non-empty set E of entity structures, and a relation ρ, which may itself be a structured object.

Four types of entity structure are distinguished and five types of link structure:[7]

(11) a. Types of entity structure:
1. event structure;
2. time point structure;
3. temporal interval structure;
4. time-amount structure.

b. Types of link structure:
1. temporal anchoring structure, anchoring an event (or state; more generally, an eventuality) in time;
2. temporal relation structure, relating a time point or interval to another time point or interval;
3. event-duration structure, relating an event or state to its duration;
4. aspectual structure, describing an aspectual relation between two events;
5. subordination structure, capturing a subordination relation between two events.

Note that, while in general a link structure may relate an entity structure to a set of other entity structures, in ISO-TimeML a link structure always relates an entity structure to a single other entity structure; moreover, the relational component of a link structure in ISO-TimeML is not a structured object but simply a relation.

Semantics

It was noted above that certain correspondences between XML and DRS representations can be used to define a semantics for annotation representations. The same is true for defining a semantics of abstract annotation structures, for the simple reason that both entity structures and link structures are n-tuples, similar to the sequence of attribute-value pairs in an XML element, the significance of an element in an n-tuple being encoded by its position rather than by an XML attribute.[8] Similar to the XML

---

[6] See Bunt (2013b) for more complex entity structures.

[7] The four types of entity structure correspond to four different XML elements in the concrete syntax; the five types of link structure correspond to three relational tags in the concrete syntax, where, following the original TimeML representation format, the `TLINK` tag is used for each of the first three kinds of relation listed in (11b), as well as for representing temporal relations between events. This forms a mismatch between the abstract and the concrete syntax of ISO-TimeML, which should be remedied in the future. In TimeML `TLINK` was also used for relating a temporal interval to its length; ISO-TimeML has the separate `MLINK` tag for this purpose.

[8] In defining a semantics for the above abstract syntax, it was found (Bunt, 2011) that finer distinctions need to be made in the conceptual inventory than those listed in (11). *Date structures* were added as a type of entity structure, and two types of link structure were added: one for linking an interval to its length (*interval measurement structure*) and one for expressing temporal relations between events (*event-temporal relation structures*); the latter two were necessary in order to avoid the semantically problematic overloading that occurs in TimeML of the `TLINK` relation.

- DRS translation sketched in Section 2.1, a mapping from annotation structures to DRSs can be defined as follows:[9]

1. An entity structure $\langle m, s\rangle$, with $s = \langle a_1, \ldots, a_n\rangle$ is mapped into a DRS which introduces a discourse referent x and which contains for each ϵ-component a_i a condition $\pi(x, a'_i)$, where π_i is a predicate that interprets the position of a_i and a'_i is the translation of a_i;
2. A link structure $\langle \epsilon_1, \{\epsilon_2\}, R\rangle$ is interpreted as a DRS that introduces two discourse referents, x_1 and x_2, and which contains a condition of the form $R'(x_1, x_2)$, where R' is a predicate translating the relation R.

For example, ISO-TimeML annotation of the temporal information in the sentence

(12) *John called at midnight*

uses in its abstract annotation structure an entity structure ϵ_1 for the *call* event and an entity structure ϵ_2 for the time point *midnight*, while the temporal anchoring relation between the event and the time point gives rise to a link structure L_1 connecting the two.

The entity structure for an event contains an n-tuple $\langle a_1, ..., a_n\rangle$, with $1 \leq n \leq 6$, depending on the types of information which are available or relevant about the event. In this example only an event type and a tense are relevant, so the n-tuple is a pair $\langle$*event type, tense*$\rangle$.

The entity structure for the time specification is a pair $\langle s$,*time zone, clock time*$\rangle$; in this chapter we will suppress the use of time zones, which is not relevant here. The semantics maps the entity and link structures to mini-DRSs as follows:

(13) $\epsilon_1 \leadsto$

e_1
type(e_1, *call*) tense(e_1, *past*)

$\epsilon_2 \leadsto$

t_1
clocktime(t_1, 2400)

$L_1 \leadsto$

e_2, t_2
at-time(e_2, t_2)

Merging these DRSs results in (14) for the annotation structure $\langle\{\epsilon_1, \epsilon_2\}, \{L_1\}\rangle$:

(14)

e, t
type(e, *call*) tense(e, *past*) clocktime(t, 2400) at-time(e, t)

[9] For more details see Bunt (2011a; 2013a) .

This says that a call event occurred in the past, at 24:00 o'clock.

Concrete syntax
The XML-based ISO-TimeML-ics representation format, defined by the concrete syntax, is an *ideal* format (Bunt, 2010) in the sense that (a) every annotation structure, defined by the abstract syntax, can be represented in that format; and (b) every ISO-TimeML-ics expression represents only one annotation structure, defined by the abstract syntax. The semantics of an ISO-TimeML-ics representation is therefore defined simply as the semantics of the abstract annotation structure that it represents.

3 Combining semantic annotations and semantic representations

In this section we consider the use of semantic annotations for making the interpretation of a given sentence or text more specific than its purely compositional semantic analysis, by specifying the interpretation of a deictic or an anaphoric expression, or by adding disambiguating information, or by specifying semantic relations between textual elements.

For the representation of ambiguous or underspecified meanings, as the result of purely compositional semantic analysis, we will use an extended form of DRSs. In an overview of representation techniques, Bunt (2007b) shows that underspecified representation of a wide range of semantic phenomena is possible by using *labels* with scope constraints, as in UDRT (Reyle, 1993), or *hole variables* or *handles* as in Hole Semantics (Bos, 1996) and in Minimal Recursion Semantics (Copestake et al., 1996), in combination with *metavariables*, as proposed e.g. by Pinkal (1999). Labels, holes and handles are particularly useful for the representation of structural ambiguities, like relative quantifier scoping, while metavariables are suitable for representing local ambiguities, like anaphora, deixis, metonymy, and sense ambiguities. DRSs with labels and metavariables therefore form a powerful formalism for underspecified semantic representation. The usefulness of DRSs for defining the semantics of semantic annotations having been noted already, we will in the rest of this chapter use (extended) DRSs for both purposes.

3.1 Contextualization

The annotation of coreference relations can be used to effectively reduce the underspecificity in an semantic representation due to the occurrence of anaphoric expressions. Example (15) illustrates this. The USR in (15b) representing the result of compositional semantic analysis of the sentence *John saw Bill when he left the house* introduces a discourse referent (z) as the individual who left the house, allowing z to denote John or Bill. The annotation, in the form of an abstract annotation structure in (15c1) and in concrete XML representation form in (15c2), stipulates that the referential entities corresponding to *Bill* and *he* are identical.

(15) a. John saw Bill when he left the house.

b. Underspecified semantic representation:

x, y, z, e_1, e_2, t_1, t_2
name(x, *john*) name(y, *bill*) see(e_1, x, y, t_1) lefthouse(e_2, z, t_2)

c. Annotation of coreference, with its representation and interpretation:

c1. *Annotation structure*: $a = \langle\{\epsilon_1, \epsilon_2, \epsilon_3\}, \{L_2\}\rangle$, where
- $\epsilon_1 = \langle m_1, a_1\rangle$: markable m_1 identifies the word token `w1` (*John*); a_1 is an individual named “John”;
- $\epsilon_2 = \langle m_2, a_2\rangle$: markable m_2 identifies the word token `w3` (*Bill*); a_2 is an individual named “Bill”;
- $\epsilon_3 = \langle m_3, a_3\rangle$: markable m_3 identifies the word token `w5` (*he*); a_3 is an individual indicated by “he”;
- $L_2 = \langle \epsilon_2, \{\epsilon_3\}, R_{ID}\rangle$: R_{ID} is the identity relation between individuals.

c2. *Representation of annotation structure:*

```
<xml>
  <refEntity xml:id="r1" target="#w1" name="john"/>
  <refEntity xml:id="r2" target="#w3" name="bill"/>
  <refEntity xml:id="r3" target="#w5"
   natGender="male"/>
  <refLink anaphor="#r3" ante="#r2"
   relType="identity"/>
</xml>
```

d. *Interpretation* of annotation structure:

x, y, z
name(x, *john*) name(y, *bill*) gender(z, *male*) $y = z$

Unification of this interpretation of the coreference annotation with the semantic representation (15b) gives the following fully specified representation:

(16)

$x, y, z, e_1, e_2, t_1, t_2$
name(x, *john*) name(y, *bill*) see(e_1, x, y, t_1) gender(z, *male*) lefthouse(e_2, z, t_2) $y = z$

3.2 Semantic alignment

The example of contextualization in the previous subsection may suggest that the combination of the information in an annotation with that in a USR is simply a matter of DRS merging. This is not quite true, however. Things may be more complicated, and require a process that keeps track of exactly to which segment of source text a component of a semantic annotation applies.

Consider the text fragment (17a), which contains four occurrences of the pronoun *he* and one of *him*, used anaphorically, that are all ambiguous between having Chris or Robin as their antecedent. An underspecified semantic representation of the text is shown in (17b) on the left; on the right the DRS-interpretation of a coreference annotation is shown.

(17) a. Chris saw Robin when he left the house. He was happy. He had phoned him last week and warned that he might be unable to come.

b. Underspecified representation and representation of annotation interpretation (AIR):

USR	AIR
$x, y, z, u, v, w, r,$ e_1, e_2, e_3, e_4	a, b, c, d, f, g, h
name(x, *chris*) name(y, *robin*) see(e_1, x, y, t_1) gender(z, *male*) lefthouse(e_2, z, t_2) when(e_1, e_2) gender(u, *male*) be(u, *happy*) gender(v, *male*) gender(w, *male*) phone(e_3, v, w, t_3) in-time(e_3, *last_week*) (...) gender(r, *male*) come(e_4, r)	name(a, *chris*) name(b, *robin*) $c = b$ gender(c, *male*) gender(d, *male*) $d = a$ gender(f, *male*) $f = b$ gender(g, *male*) $g = a$ gender(h, *male*) $h = b$

The alignment of the elements in the USR and those in the AIR immediately suggest a possible merge of the two DRSs by unifying a with x, b with y, c with z, d with us, and so on, corresponding to the reading:

(18) Chris saw Robin when Robin left the house. Chris was happy. Robin had phoned Chris last week and warned that he [Robin] might be unable to come.

Let us assume that this is the intended reading. From a technical point of view, however, the AIR variable c might just as well unify with (for example) the USR variable u, rather than with z, and similarly the variables d, f, g and h could unify with any of the variables in the USR, giving rise to a different (possibly inconsistent) interpretation of the USR variables that were introduced for anaphoric expressions. These unifications are all possible because the only information about the variables c, d,..., h in the AIR is that they are either equal to the discourse referent a or to the discourse referent b, but that doesn't impose any constraints on how they may unify with the USR variables z, u,..., r. This reveals an inadequacy in the AIR: the interpretation of the anaphoric links in the annotation has lost the information concerning which token of *he/him* corresponds to which discourse referent; in that sense the AIR is not well 'aligned' with the source text.

This can be remedied by treating the information in semantic annotations about their textual anchoring as semantically significant, and taking it along in their interpretation. This information can then be exploited when combining the AIR with the USR, if the USR components are likewise anchored to the source text segments that they interpret. This can be accomplished by replacing discourse referent introductions by pairs, consisting of an identifier of the text segment which gives rise to its introduction, and the discourse referent itself – see (19), which corresponds to the first part of (17).

(19) a. Chris saw Robin when he left the house. He was happy.

b. Tokenization:
`m1="Chris" m2="saw" m3="Robin" m4="when" m5="he"`
`m6="left the house" m7="he" m8="was happy"`

c. Underspecified semantic representation and representation of annotation interpretation:

USR	AIR
⟨m1, x⟩, ⟨m3, y⟩, ⟨m5, z⟩, ⟨m7, u⟩, ⟨m2, e1⟩, ⟨m6, e2⟩, ⟨m2, t1⟩, ⟨m6, t2⟩	⟨m1, a⟩, ⟨m3, b⟩, ⟨m5, c⟩, ⟨m9, d⟩
name(x, *chris*)	name(a, *chris*)
name(y, *robin*)	name(b, *robin*)
see(e_1, x, y, t_1)	$c = b$
gender(z, *male*)	gender(c, *male*)
lefthouse(e_2, z, t_2)	gender(d, *male*)
when(e_1, e_2)	$d = a$
gender(u, *male*)	gender(f, *male*)
be(u, *happy*)	$f = b$

By unifying markable-variable pairs $\langle m, \alpha \rangle$ rather than just the variables, we ensure that effectively only those AIR and USR discourse referents unify that correspond to the same source text segments. Once the unification has been performed, and the anaphors have been resolved, the markables in the conditions can be eliminated, having done their duty, leading to a standard type of DRS as in (20):

(20)

$x, y, z, u, e_1, e_2, t_1, t_2$
name(a, *chris*)
name(b, *robin*)
see(e_1, x, y, t_1)
gender(z, *male*)
$z = x$
lefthouse(e_2, z, t_2)
when(e_1, e_2)
gender(u, *male*)
be(u, *happy*)
$u = x$

3.3 Explicitation

In this section we show how a semantic annotation can be used to make an implicit semantic relation between parts of a sentence or text fragment explicit. Two cases are considered: (a) the semantic role of a prepositional temporal phrase, as either anchoring an event in time or as specifying a time-related participant in the event; (b) the semantic relation between the contents of two sentences in a coherent discourse, when this relation is not expressed in the text.

3.3.1 Semantic roles

In example (21a) the prepositional phrase *at six o'clock* can be understood as specifying the time that Henry set an alarm clock for waking him up the next morning (as in *Before switching off his bed light, Henry set the alarm clock*), or as specifying the time that the alarm will sound (as in *Henry set the alarm to wake him up at six o'clock*). In order to distinguish the two interpretations, we make use of semantic roles in DRS conditions both in the annotation and in the compositional semantic interpretation (rather than multi-argument event predicates). The semantic role annotation is inspired by the proposals for semantic roles annotation in the LIRICS project (see LIRICS, 2006) and in ISO project 24617-5 (ISO 2013; Bunt and Palmer, 2013).

The USR in (21b) represents the *set* event and its three participants, identifying Henry as the agent and the alarm as the theme, but leaving the semantic role of the time unspecified.

(21) a. Henry set the alarm at six o'clock.
b. Underspecified semantic representation:

⟨m1,x⟩, ⟨m3,y⟩, ⟨m4,t⟩, ⟨m2, e⟩
name(x, *henry*) type(e, *set*) type(y, *alarm*) clocktime(t, 600) agent(e, x) theme(e, y)

c. Annotation of time and events, with its representation and interpretation:

c1. *Annotation structure*: $\alpha = \langle\{\epsilon_1, \epsilon_2\}, \{L_1\}\rangle$, where
- $\epsilon_1 = \langle m_1, \langle et_2, \textit{past}\rangle\rangle$: markable m_1 identifies the word token `w2` (*set*); et_2 is an event type;
- $\epsilon_2 = \langle m_2, ct_{600}\rangle$: markable m_2 identifies token sequence `[w6,w7]` (*six o'clock*); ct_{600} identifies a clock time;
if *at six o'clock* is interpreted as as a temporal specifier of the *set*-event, then
- $L_1 = \langle \epsilon_1, \epsilon_2, R_{at}\rangle$ (relation R_{at} anchoring events in time);
else *at six o'clock* is interpreted as specifying a temporal participant in the *set*-event, and
- $L_1 = \langle \epsilon_1, \epsilon_2, R_{goal}\rangle$ (semantic role relation R_{goal})

c2. *Representation of annotation structure:*
a. For *at six o'clock* as specification of event-time:
```
<xml>
 <event xml:id="e1" target="#w2" pred="set"/>
 <instant xml:id="t1" target="#w6 #w7"
  clockTime="600"/>
 <timeAnchoring event="#e1" time="#t1"
  relType="at"/>
</xml>
```
b. For *at six o'clock* as description of event participant:
```
 <xml>
 <event xml:id="e1" target="#w2" pred="set" />
 <instant xml:id="t1" target="#w6 #w7"
  clockTime="600"/>
 <semRole event="#e1" participant="#t1"
  relType="goal"/>
</xml>
```

c3. *Semantic interpretation* of annotation structure:
a. For *at six o'clock* as specification of event-time:

⟨m2,e⟩, ⟨m4,t⟩
type(e, *set*)
clocktime(t, 600)
at-time(e, t)

b. For *at six o'clock* as description of event participant:

⟨m1,e⟩, ⟨m2,t⟩
type(e, *set*)
clocktime(t, 600)
goal(e, t)

Merging the URS in (21b) with either of the AIRs in (21c3) gives the fully specified semantic representation of either interpretation, as shown in (22):

(22) a.

x, y, t, e
name(x, *henry*)
type(e, *set*)
type(y, *alarm*)
clocktime(t, 600)
agent(e, x)
theme(e, y)
at-time(e, t)

b.

x, y, t, e
name(x, *henry*)
type(e, *set*)
type(y, *alarm*)
clocktime(t, 600)
agent(e, x)
theme(e, y)
goal(e, t)

3.3.2 Implicit discourse relations

Example (23), from the Penn Discourse Treebank, illustrates the use of a semantic annotation for interpreting the relation between sentences in a coherent discourse, when not expressed explicitly. The intended interpretation is that the second sentence provides a reason why the event mentioned in the first sentence occurs.

The underspecified representation shown in (23b) is simply the combined semantic representations of the two sentences in (23a). The annotation in (23c) applies the ISO standard for discourse relation annotation under development as ISO 24617-8 (see Bunt, Prasad & Joshi, 2012). The attribute `@aoType` is used to represent an 'abstract object type' in the sense of Asher (1993), and the attribute `@attribution` is used to represent the source to whom statements in the annotated text are attributed. Semantically, a discourse relation which connects two sentences by establishing a relation between an event expressed in the first sentence and another event expressed in the second, requires the annotation to indicate exactly which events are related, since each of the sentences may mention several events. The attributes `@headID` and `@headPred` are introduced for this purpose;

the first specifies the relevant markable, the second the event type. This information is used to construct a representation of the interpretation as shown in (23c3).

(23) a. Some have raised their cash positions to record levels. Implicit= because High cash positions help buffer a fund when the market falls.

b. Underspecified semantic representation:

$\langle$m1, $x\rangle$, $\langle$m3,$y\rangle$, $\langle$m4,z$\rangle$, $\langle$m5,$u\rangle$, $\langle$m8,v$\rangle$, $\langle$m9,w$\rangle$, $\langle$m2,$e_1\rangle$, $\langle$m6,$e_2\rangle$, $\langle$m7,$e_3\rangle$, $\langle$m10,$e_4\rangle$
type(e_1, *raise*), some(x), agent(e_1, x), cashposition(y), theme(e_1, y), recordlevel(z), goal(e_1, z), type(e_2, *help*), hicashposition(u), instrument(e_2, u), type(e_3, *buffer*), theme(e_2, e_3), fund(v), theme(e_3, v), type(e_4, *fail*), when(e_4, e_3), market(w), theme(e_4, w)

c. Annotation of discourse relations, with representation and interpretation:

c1. *Annotation structure*: $\alpha = \langle\{\epsilon_1, \epsilon_2\}, \{L_1\}\rangle$, where
- $\epsilon_1 = \langle$m$_1$, $\langle et_3$,*past*$\rangle\rangle$ (event type et_3);
- $\epsilon_2 = \langle$m$_2$, $\langle et_4\rangle\rangle$ (event type et_4);
- $L_1 = \langle\epsilon_1, \epsilon_2, R_{reason}\rangle$; R_{reason} is the ‘reason’ relation between events)

c2. *Representation of annotation structure:*

```
<dRelML>
<discourseRelation xml:id="dr1"
  arg1="#a1" arg2="#a2" rel="#r1"/>
<dRelArgument xml:id="a1" target="#w1...#w9"
  aoType="event" headID="#w3"
  headPred="raise" attribution="#at1"/>
<dRelArgument xml:id="a2" target="#w10...#w20"
  aoType="event" headID="#w13"
  headPred="help" attribution="#at1"/>
<implRel xml:id="r1" discRel="reason"
  attribution="#at1"/>
<attributionRep xml:id="at1" aSource="author"/>
</dRelML>
```

c3. *Semantic interpretation* of annotation structure:

$$\begin{array}{|l|}\hline \langle \mathrm{m}_2, e_1\rangle, \langle \mathrm{m}_6, e_2\rangle \\ \hline \text{type}(e_1, \textit{raise}) \\ \text{type}(e_2, \textit{help}) \\ \text{reason}(e_1, e_2) \\ \hline \end{array}$$

Unification of the semantic representations of the two sentences, construed as a single DRS in (23b), with the interpretation of the annotation (and dropping the markables associated with the discourse referents) leads to the representation (24) of the discourse fragment as a whole.

$$(24)\quad \begin{array}{|l|}\hline x, y, z, u, v, w, e_1, e_2, e_3, e_4 \\ \hline \text{type}(e_1, \textit{raise}), \\ \text{some}(x),\ \text{agent}(e_1, x), \\ \text{cashposition(y)},\ \text{theme}(e_1, y), \\ \text{recordlevel}(z),\ \text{goal}(e_1, z), \\ \text{type}(e_2, \textit{help}), \\ \text{hicashposition}(u),\ \text{instrument}(e_2, u), \\ \text{type}(e_3, \textit{buffer}),\ \text{theme}(e_2, e_3), \\ \text{fund}(v),\ \text{theme}(e_3, v), \\ \text{type}(e_4, \textit{fail}),\ \text{when}(e_4, e_3), \\ \text{market}(w),\ \text{theme}(e_4, w), \\ \text{reason}(e_2, e_3) \\ \hline \end{array}$$

4 Conclusions and perspectives

,

In this paper we have indicated how the information, contained in semantic annotations, may effectively be used to resolve ambiguities and to narrow down underspecified meanings. This is possible if the annotations are expressed in an annotation language that has a formal semantics. This is often not the case, but under the influence of efforts of the international organisation for standards ISO, projects are under way that do indeed aim to define such annotation languages. Studies by Pratt-Hartmann, Katz, Lee, and the author have demonstrated the feasibility of doing so for substantial fragments of semantic annotation languages, as illustrated by the annotation language ISO-TimeML of ISO standard 24617-1 (Time and Events) and annotation language DiAML of ISO standard 24617-2 (Dialogue Acts).

This approach opens the possibility to exploit semantic annotations in a computational interpretation process, as we have shown by casting the interpretation of semantic annotations in a DRS-based representation format that is suitable for underspecified semantic representation, allowing a unification-based process for combining the information in semantic annotations with that obtained through compositional semantic analysis.

This is potentially very useful, since semantic annotations are constructed using quite different techniques (machine learning from corpora, exploitation of domain ontologies, searching metadata,..) than the compositional syntactic-semantic analysis techniques that make sentential semantic content explicit. The approach that we have described here therefore makes it possible to effectively combine heterogeneous processes and information sources in order to arrive at maximally specific and contextually appropriate interpretations.

nociteReyle93

References

Asher, N. (1994). *Reference to Abstract Objects in Discourse*. Dordrecht: Kluwer.

Bonial, C., W. Corvey, V. Petukhova, M. Palmer, and H. Bunt (2011). A Hierarchical Unification of LIRICS and VerbNet Thematic Roles. In *Proceedings Workshop on Semantic Annotation for Computational Linguistic Resources*, Stanford.

Bos, J. (1996). Predicate logic unplugged. In *Proceedings 10th Amsterdam Colloquium*, pp. 133-143.

Bunt, H. (2007a). The Semantics of Semantic Annotation. In *Proceedings of the 21st Pacific Asia Conference on Language, Information and Computation (PACLIC21)*, Korean Society for Language and Information, pp. 13–29.

Bunt, H. (2007b). Underspecified Semantic Representation: Which Technique for What Purpose? In H. Bunt, R. Muskens, and E. Thijsse (Eds.), *Computing Meaning, Vol. 3*, pp. 115–140. Dordrecht: Springer.

Bunt, H. (2010). A methodology for designing semantic annotation languages exploiting semantic-syntactic ISO-morphisms. In *Proceedings of the Second International Conference on Global Interoperability for Language Resources (ICGL 2010)*, Hong Kong.

Bunt, H. (2011). Introducing abstract syntax + semanticss in semantic annotations, and its consequences for the annotation of time and events. In E. Lee and A. Yoong (Eds.), *Recent Trends in Language and Knowledge Processing*, Volume 2. Seoul: Hankukmunhwasa.

Bunt, H. (2013a). A methodology for designing semantic annotations. TiCC Technical Report TR 2013-001, Tilburg University.

Bunt, H. (2013b). A Context-Change Semantics for Dialogue Acts. *This volume.*

Bunt, H., J. Alexandersson, J.-W. Choe, A. Fang, K. Hasida, K. Lee, V. Petukhova, A. Popescu-Belis, L. Romary, C. Soria, and D. Traum (2010). Towards an ISO standard for dialogue act annotation. In *Proceedings of LREC 2010, Malta*, Paris. ELDA.

Bunt, H., J. Alexandersson, J.-W. Choe, A. Fang, K. Hasida, V. Petukhova, A. Popescu-Belis, and D. Traum (2012). A semantically-basedstandard for dialogue annotation. In *Proceedings of LREC 2012, Istanbul*, Paris. ELRA.

Bunt, H. and C. Overbeeke (2008a). An extensible, compositional semantics of temporal annotation. In *Proceedings of LAW-II: the Second Linguistic Annotation Workshop*, Marrakech.

Bunt, H. and C. Overbeeke (2008b). Towards formal interpretation of semantic annotation. In *Proceedings 6th International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech. Paris: ELRA.

Bunt, H., R. Prasad. and A. Joshi (2010). First steps toward an ISO standard for the annotation of discourse relations. In *Proceedings of ISA-7, Seventh Joint ISO - ACL SIGSEM Workshop on Interoperable Semantic Annotation*, Workshop at LREC 2012, Istanbul. Paris: ELRA.

Bunt, H. and J. Pustejovsky (2012). Annotating temporal and event quantification. In *Proceedings of ISA-5, 5th International Workshop on Interoperable Semantic Annotation*, Workshop at the Second International COnference on Global Interoperability for Language resources (ICGL-2), Hong Kong. City University of Hong Kong.

Bunt, H. and L. Romary (2002, May). Towards Multimodal Content Representation. In K. S. Choi (Ed.), *Proceedings of LREC 2002, Workshop on International Standards of Terminology and Language Resources Management*, Las Palmas, pp. 54–60. Paris: ELRA.

Copestake, A., D. Fllickinger, and I. Sag (1996). *Minimal Recursion Semantics: An Introduction*. Cambridge, UK: Cambridge University Press.

ISO (2012a). *Language Resource Management - Semantic Annotation Framework (SemAF) - Part 1: Time and Events*. International Organisation for Standardisation ISO. ISO International Standard 24617-1:2012(E)).

ISO (2012b, September). *Language Resource Management - Semantic Annotation Framework (SemAF) - Part 2: Dialogue acts*. International Organisation for Standardisation ISO. ISO International Standard 24617-2:2012(E)).

ISO (2012c). *Linguistic Annotation Framework (LAF)*. International Organisation for Standardisation ISO. ISO International Standard 24612-2:2012(E)).

ISO (2013). *Language Resource Management - Semantic Annotation Framework (SemAF) - Part 5: Semantic roles*. International Organisation for Standardisation ISO. ISO Committee Draft, CD 24612-5:2013).

Kamp, H. and U. Reyle (1993). *From Discourse to Logic*. Dordrecht: Kluwer Academic Publishers.

Katz, G. (2007). Towards a Denotatial Semantics for TimeML. In F. Schilder, G. Katz, and J. Pustejovsky (Eds.), *Annotation, Extraction, and Reasoning about Time and Events*. Dordrecht: Springer.

Lee, K. (2008). Formal Semantics for Interpreting Temporal Annotation. In *Unity and Diversity of Languages: Special Lectures for the 18th International Conference of Linguists*. Amsterdam: Benjamins.

LIRICS (2006). *Project LIRICS Deliverable D4.3, Documented compilation of semantic data categories*. Available at http://lirics.loria.fr.

Pinkal, M. (1999). On semantic underspecification. In H. Bunt and R. Muskens (Eds.), *Computing Meaning, vol. 1.*, pp. 33–56. Dordrecht: Kluwer.

Prasad, R., N. Dinesh, E. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, and B. Webber (2008). The Penn Discourse Treebank 2.0. In *Proceedings of LREC 2008, 6th International Conference on Language Resources and Evaluation*, Marrakech, Morocco.

Pratt-Hartmann, I. (2007). From TimeML to Interval Temporal Logic. In *Proceedings of the Seventh International Workshop on Computational Semantics (IWCS-7)*, Tilburg, Netherlands, pp. 166–180.

Pustejovsky, J., H. Bunt, and K. Lee (2010). ISO-TimeML. In *Proceedings of LREC 2010*, Malta. ELDA, Paris.

Pustejovsky, J., H. Bunt, K. Lee, and L. Romary (2010). ISO-TimeML: An International Standard for Semantic Annotation. In *Proceedings 7th International Conference on Language Resources and Evaluation (LREC 2010)*, Malta. ELDA, Paris.

Pustejovsky, J., J. Castano, R. Ingria, R. Gaizauskas, G. Katz, R. Saurí, and A. Setzer (2003, January). TimeML: Robust Specification of Event and Temporal Expressions in Text. In *Proceedings of the Fifth International Workshop on Computational Semantics (IWCS-5)*, Tilburg, Netherlands, pp. 337–353.

Pustejfsky. J., J. Moszkowics and M. Verhagen (2012). The current status of ISO-Space. In *Proceedings of ISA-7, Seventh Joint ISO - ACL SIGSEM Workshop on Interoperable Semantic Annotation*, Workshop at LREC 2012, Istanbul. Paris: ELRA.

Reyle, U. (1993) Dealing with Ambiguities by Underspecification. *Journal of Semantics* 10 (2), 123–179.

TEI (2009) *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. Text Encoding Initiative Consortium, Oxford.