Semantic annotations as complementary to underspecified semantic representations

Harry Bunt
Department of Communication and Information Sciences
Tilburg University, Netherlands
harry.bunt@uvt.nl

Abstract

This paper presents a new perspective on the use of semantic annotations. We argue that semantic annotations should (1) capture semantic information that is complimentary to the information that is expressed in the source text; (2) have a formal interpretation. If these conditions are fullfilled, then the information in semantic annotations can be effectively combined with the information in the source text by interpreting a semantic annotation language through the translation of annotations into the same formalism as underspecified semantic representations obtained through compositional semantic analysis.

1 Introduction

Annotations add information to a source text. In the pre-digital age, *ad-notations* characteristically took the form of an editor's bibliographical or historical comments, presented in notes that are added to the source text. In the digital age, annotations take on a different form, but their function is essentially the same: they add information to a source text. The following example illustrates this.

In (1a) an annotation in the form of a note adds certain historical information to the text; it is indeed additive in the sense that it contains information which is not in the text itself. In (1b), by contrast, the information in the note is in fact already contained in the text itself, and the annotation therefore does not make any sense.

(1) a. In 1634 he proposed to distinguish sixty-four elements.[132])
Note 132. A proposal to this effect had in fact been made before

Proceedings of the 8th International Conference on Computational Semantics, pages 33–44,
Tilburg, January 2009.

by Larochelle in 1544 in his epistle *"Plus d'aspects fondamentals de la materia"*)

b. In 1634 He proposed to distinguish sixty-four elements. (A proposal to this effect had in fact been made before by Larochelle in 1544 in his epistle *"Plus d'aspects fondamentals de la materia"*).[132])
Note 132. Also proposed by Larochelle in 1544.

It may seem obvious that annotations do not make sense if they do not add any information, but consider the following example of annotating text with temporal information using TimeML (Pustejovsky et al., 2003):

(2) <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>

The annotation in this case does not contain any information which is not already in the text itself; it only casts the description of a date *the first of December 2008* in the alternative format "2008-12-01.

By contrast, a case where semantic annotation would really add something, is the following.

(3) John called today.

In the absence of further temporal information (when was this sentence uttered/written/published/..?) we don't know what day is referred to by 'today'. In this case it would help to have the semantic annotation (4), added by someone (or by a computer program) who possesses that information or is able to find it, for instance by having access to relevant metadata.

(4) <timeml>
John
<EVENT called id="e1"/>
<TIMEX3 tid=t1 type="date" value="2008-10-13"/>
today
</TIMEX3>
<TLINK event="e1" relatedToTime="t1" relType="DURING"/>
</timeml>

If the point of annotations is to add certain information to a given source text, then the point of semantic annotations can only be to add semantic information that is not already in the text. We suggest that this additional information can be precisely the information whose absence causes the interpretation of sentences to be underspecified, as illustrated in (3), or that causes ambiguities such as the one in (5).

(5) John saw Peter when he left the house.

The semantic analysis of this sentence tells us that someone called 'John' saw someone called 'Peter', and that this happened at the moment that one of them left the house. If it is known that 'he' actually refers to Peter, this could be captured by the semantic annotation in (6):

(6) <refml>
<REFENTITY id=r1 John/ >
saw
<REFENTITY id=r2 Peter / >
when
<REFENTITY id=r3 he / >
left the house
<REFLINK anaphor=r3 antecedent=r2 relType=IDENTITY/ >
</refml>

Other types of ambiguity which could benefit from additional information in semantic annotations for example concern relative scoping, semantic roles, discourse relations, and dialogue acts, as illustrated in (7) - (10).

(7) Angry men and women demonstrated against the proposal.

(8) a. The movie scared Jane.
b. John scared Jane.
(intentionally: Agent role; unintentionally: Cause role)

(9) a. John called Mary; he missed her.
(Effect - Cause relation)
b. John called Mary; she was delighted to hear from him.
(Cause - Effect relation)

(10) You're not going to the cinema tonight.
(Statement/verification/prohibition)

These examples all have in common that they contain an ambiguity which cannot be resolved on the basis of the text alone. The additional information that is needed to deal with such ambiguities has to come from elsewhere, such as from donain knowledge, from knowledge about the situation of utterance, or from metadata.

Ambiguities whose resolution requires information from outside the text are a problem for compositional semantics approaches. Compositional generation of all the possible alternative semantic representations (and subsequent filtering) leads to a combinatorial explosion in the interpretation process (see Bunt & Muskens, 1999). Underspecified semantic representations (USRs) have been proposed as way to get around this, but suffer from the limitation that reasoning directly with USRs is problematic; in most applications, it is necessary to resolve the underspecifications at some stage and to create a fully disambiguated representation.

In this paper we argue that semantic annotations can be helpful for effectively dealing with ambiguities if they have a formal semantics, and in particular if their interpretation makes use of the same representational formalism as that of underspecified semantic representations.

Since digital texts and their annotations are machine-readable and electronically exchangeable, an issue for annotation in the digital age is that it would be beneficial if different researchers use the same concepts for expressing the same information and put their annotations in a suitable interchange format, thus allowing the effective re-use of each other's annotated resources. This ideal has in recent years been taken up by an expert group of the international organization for standardization ISO, concerned with the interoperability of language resources.

The inspiration for this paper comes from participating in the 'Semantic Annotation Framework' initiative of the ISO organization and the European eContent project LIRICS (Linguistic Infrastructure for Interoperable Resources and Systems, `http://lirics.loria.fr`), that was set up and carried out by ISO expert group members. Building on studies on in the LIRICS project, two ISO projects have started in 2007 and 2008, respectively, that aim at proposing standards for the annotation of temporal information and for annotating the communicative functions of dialogue utterances. Both projects include the design of sets of well-defined and well-documented concepts for semantic annotation which are made publicly available in an online registry (following ISO standard 12620 - see `http://www.isocat.org`).

Modern annotations typically take the form of XML tags, as illustrated in (2), (4), and (6), where the kind of attributes and values in the tags depend on the purpose of the annotation: morphosyntactic, part-of-speech,

syntactic, etc. Following the Linguistic Annotation Framework (Ide and Romary, 2004; ISO, 2008b), ISO projects insist on using *standoff annotation*, where the annotations are contained in a separate file with pointers to the source text file, rather than using in-line annotation as in (2), (4), and (6). We will return to this in section 3, where this will turn out to be important for the correct combination of semantic annotations and USRs.

The rest of this paper is organized as follows. In Section 2 we very briefly consider recent work aiming at the definition of semantic annotation languages that have a formal semantics. Sections 3 and 4 deal with two 'alignment' problems that arise in the combination of semantic annotations with underspecified semantic representations. First, the two should preferably be 'aligned' in using the same representation formalism. This is the subject of Section 3. Second, the components of semantic annotation structures and underspecified semantic representation structures should be aligned in that they relate to the same stretches of source text. This is the subject of Section 4. Section 5 closes with some concluding remarks.

2 The semantics of semantic annotations

Like other forms of annotation, such as POS annotation, semantic annotation has mostly been viewed as a form of text *labelling*. This may for example be useful in corpus-linguistic research, supporting the search of certain linguistic patterns, or for finding certain types of information, such as temporal information. On the other hand when we look at the (simplified) TimeML annotations shown in (2) and (4), we note that there is in fact an effort to use XML attributes and values to not just put a flag in a text, signalling that there is temporal information there, but also to describe the content that information. What is lacking 'only' is a semantics of this language.

Recent attempts to provide a semantics for semantic annotations include the Interval Temporal Logic semantics for TimeML by Pratt-Hartman (2005); 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 (2007) and Bunt & Overbeeke (2008b), where a semantic annotation language is presented with a formal semantics, that integrates temporal information, semantic roles, and coreference relations. This semantics translates annotations in a systematic, compositional manner into first-order or second-order logic.[1]

[1]First-order logic suffices in most cases, but second-order logic is needed for cases of

Since first-order logic is formally equivalent with Discourse Representation Structures (and second-order logic to DRSs with second-order discourse referents), this semantics can be recast in the form of a translation into DRSs. Rather than spelling out how this can be done, we refer to Bunt & Overbeeke (2008b) and exploit the well-established equivalence with DRSs. For example, the annotation representation in (6) translates into the DRS $< \{x, y, z\}, \{\mathsf{john}(x), \mathsf{peter}(y), \mathsf{saw}(x, y), \mathsf{male}(z), \mathsf{leftthehouse}(z), z = y\} >$

3 Combining USRs and semantic annotations

Reasoning is the combination of pieces of information – so that's what needs to be done when the information in semantic annotations is combined with that in USRs. If we are able to interpret semantic annotations by translating them into the same representational format as USRs, then the reasoning process can take on a very simple form: unification.

A range of representational and processing techniques have been proposed for underspecified semantic representation; in the overview in Bunt (2007), it is argued that the use of labels (as in UDRT, Reyle,1993) and hole variables (as in Hole Semantics, Bos, 1996) or 'handles' (as in MRS, Copestake et al., 1997), in combination with the use of metavariables (as proposed e.g. by Pinkal, 1999) allows the underspecified representation of a wide range of semantic phenomena. Labels etc. are particularly useful for the underspecified respresentation of *structural* ambiguities like relative scoping and PP attachment, while metavariables are suitable for *local* ambiguities like anaphora, metonymy, and sense ambiguities. We will therefore cast the formal semantics of semantic annotations in the form of UDRSs with labels and hole variables, extended by allowing metavariables to occur in conditions.[2]

3.1 Unifying USRs and annotation interpretation

We first illustrate the combination of USRs and semantic annotations for simple cases of (a) relative scope resolution; (b) coreference resolution; (c) the interpretation of temporal deixis. In the next subsection we show that more complex cases may involve a technical complication for ensuring that the underspecified parts of USRs and the information in semantic annota-

collective coreference.

[2]Another extension, which we will not consider here, is that of allowing second-order discourse referents; cf. previous footnote.

tions refer to the same segments of the source text, and we indicate how this 'alignment' problem can be solved.

In (11b) we see on the left the underspecified representation of the quantifier scopes in (sentence 11a), and on the right the AIR of the annotation, indicating in its bottom part that the universal quantifier has wider scope. The bottom part of the USR contains the scope constraints on the possible ways of combining the various conditions and sub-DRS's contained in the upper part into a complete DRS. The operator '⊗' constructs a DRS from the labeled structures that it operates on.

(11) a. Every man loves a woman.

b.

USR	AIR
L4: x, L8: y	T1: a, T4: b
L1: L3 → h1 L2: ⊗{h2,h3}, L3: ⊗{L4,L5}, L5: man(x), L6: love(x,y), L7: ⊗{L8,L9}, L9: woman(y)	T2: man(a), T3: ⊗{T1,T2}, T5: woman(b), T6: ⊗{T4,T5}
L3 > L6, L7 > L6	T3 > T6

Unification of the two representations includes the label unifications T3=L3 and T6=L7, which has the effect that the AIR scope constraint adds the constraint L3 > L7 to the USR. This has the result that of the two possible 'pluggings' of the hole variables in the USR (h0=L1, h1=L2, h2=L6, h3=L7; and h0=L2, h1=L6, h2=L1, h3=L7) the second one is ruled out. This reflects that the semantic annotation resolves the ambiguity.

In (12b), the part on the left shows the USR of the sentence (12a), while the part on the right shows the Annotation Interpretation Representation (AIR) in the same UDRS-based formalism. Combination of the two takes the form of a simple unification, where label variables are unified as well as discourse markers. The unification(with T1=L1, a=x; T3=L3, b=y; L2=T2; L4=T4; L8=T6), results in (13), which is an ordinary DRS (of which the unified labels have been suppressed).

(12) a. John saw Bill – he was happy

b.

USR	AIR
L1: x, L3: y, L5: e, L7: z	T1: a, T3: b, T5: c
L2: john(x), L4: bill (y), L6: saw (e,x,y), L8: he(z), L9: happy(z),	T2: john(a), T4: bill (b), T6: he(c), T7: c=a,

(13)

x, y, e, z
john(x), bill (y), saw (e,x,y), he(z), happy(z), z=x

Example (14) shows the use of metavariables in the USR for representing underspecified deictic information. The predicates representing 'me' and 'today' have an asterisk to indicate their status as metavariables.

(14) a. John called me today

b.

USR	AIR.
L1: x, L3: y, L5: e, L7: t1	T1: a, T3: b, T5: t2
L2: john(x), L4: *ME(y), L6: call(e,x,y, t1), L8: *TODAY(t1)	T2: john(a), T4: harrybunt(b), T6: 20080923(t2)

The use of metavariables assumes an interpretation process where these variables are at some stage instantiated by ordinary expressions of the representation language. By treating metavariables indeed as variables in the unification process, they are instantiated by the corresponding terms in the semantic annotations.

The above examples all suggest that the information contained in semantic annotations can be combined with that in underspecified semantic representations in a straightforward way, using unification. There is a complication, however, which does not turn up in these simple examples, namely that the correct combination of the two pieces of information requires the

components of the two representation sructures to be 'aligned' in the sense of being related to the same parts of the source text. This issue of 'textual alignment' is addressed in the next subsection.

3.2 Textual alignment

Consider the text fragment (15), in which the anaphoric pronoun 'he' has three occurrences, which are ambiguous between having John or Bill as their antecedent.

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

b.

USR	AIR
L1: x, L3: y, L5: e1, L6: t1, L8: z, L11: t2, L12: e2, L15: u, L19: v	T1: a, T3: b, T5: c, T8: d, T11: f, T14: g, T16: h
L2: john(x), L4: bill (y), L7: saw (e1,x,y,t1), L9: he(z), L10: z=x ∨ z=y, L13: leftthehouse(e2,z,t2), L14: t1=t2, L16: he(u), L17: happy(u), L18 bill(v), L20: etc.	T2: john(a), T4: bill (b), T6: he(c), T7: c=b, T9: he(d), T10: d=a, T12: bill(f), T13: f=b, T1: him(g), T15: g=a, T17: he(h), etc.

The AIR in (15) makes perfect sense if we interpret the discourse referent 'c' as corresponding to the first occurrence of 'he'; 'd' to the second; and 'h' to the third. There is however nothing in the AIR that enforces this interpretation; the AIR is not in any way 'aligned' with the source text or with the USR, and allows e.g. the components {T5: c, T6: he(c), T7: c=b} to be unified with the USR components {L1: x, L2: john(x)}.

This problem can be resolved by taking into account that, as mentioned above, according to the ISO Linguistic Annotation Framework annotations should be represented in a stand-off format, in a separate file with pointers to source text segments. This means that, instead of an in-line representation like (6), we should consider annotations in a format like (16), where the referential information is 'anchored' to source text segments:

(16) <refml>
<SOURCE m1="John" m2="saw" m3="Bill" m4="when" m5="he"
m6="left the house"/ >
<REFENTITY id="r1" anchor="m1" / >
<REFENTITY id="r2" anchor="m3" / >
<REFENTITY id="r3" anchor="m5" gender="male"/ >
<REFLINK anaphor="r3" antecedent="r2" relType="IDENTITY"/ >
<refml/ >

The interpretation of semantic annotations should not throw this textual anchoring away. This information can subsequently be exploited when combining the AIR with the USR, if the USR components are likewise anchored to the source text segments that they interpret – see (17).

(17) a. John saw Bill when he left the house. He was happy.

b. m1="John" m2="saw" m3="Bill" m4="when" m5="he" m6="left" m7="the" m8="house" m9="he" m10="was" m11="happy"

USR	AIR
<m1, L1: x>, <m3, L3: y>, <m2, L5: e>, <m1..m3, L6: t1>, <m5, L8: z>, <m9, L11:e2>, <[m5..m8], L12:t2>, <m10, L15: u>	<m1, T1: a>, <m3, T3: b>, <m5, T5: c>, <m9, T8: d>
<m1, L2: john(x)>, <m3, L4: bill (y)>, <[m1,m2,m3]: L7: saw (e,x,y,t1)>, <m5, L9: he(z)>, <m5, L10: z=x ∧ z=y>, <[m5..m8], L13:leftthehouse(e2,z,t2)>, <m4, L14: t2=t1>, <m9, L16: he(u)>, <m9, L17: u=x ∧ u=y> <[m9..m11], L18: happy(u)>, <[m1,..,m7], L19: ⊗{L1,...,L18} >	<m1, T2: john(a)>, <m3, T4: bill (b)>, <m5: T6: he(c)>, <m5: T7: c=a>, <m9: T9: he(d)>, <m9: T10: d=a>, <[m1,m3,m4], T11: ⊗{T1,...,T10}>

By unifying pairs $< m, L : \alpha >$ of the USR and $< m', T : \beta >$ of the AIR rather than elements $< L : \alpha >$ and $< T : \beta >$, we enforce that the unifications consider only AIR and USR components that apply to the same source text segments.

Note that, contrary to what the title of this paper suggests, the AIR and the USR parts in the above examples are in fact not entirely complimentary. In (12), for example, they both include representations of John

and Bill and of the discourse referent introduced by 'he'. The conditions 'john(a)', 'bill(b)' and 'he(c)' would seem to anchor 'a', 'b' and 'c' to their intended antecedents in the USR, but example (15) showed that this is an optical illusion. The textual anchoring of the AIR and the USR makes conditions like 'john(a)' in the AIR fully redundant, and allows it to be reduced to the introduction of the discourse referents and the conditions specifying the coreference relations. The corresponding annotation is then indeed complimentary to the USR.

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, by exploiting their semantics. We have thereby assumed that 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, and preliminary studies by Pratt-Hartmann, Katz, Lee, and the author have demonstrated the feasibility of doing so for substantial fragments of semantic annotation languages.

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 UDRS-like representation format that is also suitable for underspecified semantic representation, allowing fairly straightforward unification to combine the information from annotations with that obtained through local, compositional semantic analysis.

Is this useful? Isn't the (automatic) construction of the semantic annotations the most difficult part of the interpretation enterprise, rather than something that's waiting to be exploited? Maybe so; that depends very much on the kind of linguistic material to be interpreted and on the kinds of semantic information that annotations aim to capture. One thing is clear: semantic annotations are constructed using entirely different techniques (machine learning from corpora, exploitation of domain ontologies, searching metadata,..) than the compositional syntactic-semantic analysis techniques that make the semantic content at sentence level explicit. The approach that we have outlined here makes it possible to effectively combine such very heterogeneous processes and sources of information.

References

[1] Bos, J. (1997). Predicate Logic Unplugged. In *Proc. 10th Amsterdam Colloquium*, Amsterdam. ILLC.

[2] Bunt, H. (2007a). The Semantics of Semantic Annotation. In *Proceedings of the 21st Pacific Asia Conference on Language, Information and Computation (PACLIC21)*, pages 13–28.

[3] Bunt, H. (2007b). Underspecified semantic representation: Which technique for what purpose? In Bunt, H. and Muskens, R. (eds.) *Computing Meaning, Vol. 3*, pages 115–140. Springer, Dordrecht.

[4] Bunt, H. and Muskens, R. (1999). Computational semantics. In H.Bunt and Muskens, R., (eds.) *Computing Meaning, Vol. 1*, pages 1–15. Kluwer Academic Press, Dordrecht.

[5] Bunt, H. and Overbeeke, C. (2008a). An extensible compositional semantics for temporal annotation. In *Proceedings LAW-II: Second Linguistic Annotation Workshop*, Marrakech, Morocco. Paris: ELRA.

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

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

[8] Copestake, A., Flickinger, D., and Sag, I. (1997). *Minimal Recursion Semantics: an Introduction*. CSLI, Stanford University.

[9] Ide, N. and Romary, L. (2004). International Standard for a Linguistic Annotation Framework. *Natural Language Engineering*, 10:211–225.

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

[11] Lee, K. (2008). Formal Semantics for Interpreting Temporal Annotation. In P. van Sterkenburg (ed) *Unity and Diversity of Languages: Special Lectures for the 18th International Congress of Linguists*. Amsterdam: Benjamins.

[12] Pinkal, M. (1999). On semantic underspecification. In Bunt, H. and Muskens, R. (eds.) *Computing Meaning, vol. 1*, pages 33–56. Kluwer, Dordrecht.

[13] Pratt-Hartmann, I. (2007). From TimeML to Interval Temporal Logic. In *Proc. Seventh International Workshop on Computational Semantics (IWCS-7)*, pages 166–180, Tilburg.

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

[15] Pustejovsky, J., Knippen, R., Littman, J., and Saurí, R. (2007). Temporal and Event Information in Natural Language Text. In Bunt, H. and Muskens, R. (eds.) *Computing Meaning, vol. 3*, pages 301–346. Springer, Dordrecht.

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