Let's not argue about semantics

Johan Bos

Università di Roma "La Sapienza"
Dipartimento di Informatica
Via Salaria 113 – 00198 Rome (Italy)
bos@di.uniroma1.it

Abstract

What's the best way to assess the performance of a semantic component in an NLP system? Tradition in NLP evaluation tells us that comparing output against a gold standard is a good idea. To define a gold standard, one first needs to decide on the representation language, and in many cases a first-order language seems a good compromise between expressive power and efficiency. Secondly, one needs to decide how to represent the various semantic phenomena, in particular the depth of analysis of quantification, plurals, eventualities, thematic roles, scope, anaphora, presupposition, ellipsis, comparatives, superlatives, tense, aspect, and time-expressions. Hence it will be hard to come up with an annotation scheme unless one permits different level of semantic granularity. The alternative is a theory-neutral black-box type evaluation where we just look at how systems react on various inputs. For this approach, we can consider the well-known task of recognising textual entailment, or the lesser-known task of textual model checking. The disadvantage of black-box methods is that it is difficult to come up with natural data that cover specific semantic phenomena.

1. Evaluating Meaning

Formal methods for the analysis of the meaning of natural language expressions have long been restricted to the ivory tower built by semanticists, logicians, and philosophers of language. It was only in exceptional cases that they made their way directly into open domain NLP tools. Recently, this situation has changed. Thanks to the development of treebanks (large collections of texts annotated with syntactic structures), robust statistical parsers trained on such treebanks, and the development of large-scale semantic lexica, we now have at our disposal systems that are able to produce formal semantic representations achieving high coverage (Schiehlen, 1999; Bos et al., 2004; Bos, 2005; Copestake et al., 2005; Delmonte, 2006; Sato et al., 2006; Moldovan et al., 2007).

Now, suppose we want to evaluate the semantic component of such NLP systems. How shall we go about it? Probably the most obvious way is to look at the semantic representations that the system produces and compare that with a gold standard annotation. After all, that's what we do when evaluating part-of-speech tagging, chunking, named entity recognition, and syntactic parsing. But what exactly should such a gold standard for meaning representations look like? What exactly constitutes an adequate semantic representation? Should it follow a particular (formal) theory of semantics, or rather take an independent stance? What semantic phenomena should it aim to cover?

Posing these questions is, moreover, a timely matter. As pointed out above, wide-coverage systems that claim to have genuine semantic components are now emerging and we need an unambiguous way of evaluating these systems for the sake of measuring progess and benchmarking. The key question is whether comparing to a gold standard (the so-called "Glass-Box" method) is an effective methodology for assessing semantic adequacy. Annotating text with semantic representations, is an immense task, with many choices to make, as I will show in this paper.

Alternatively, one could take "Black-Box" approaches to semantic evaluation. I will discuss two such methods in this paper: the task of recognising textual entailment, and the task of textual model checking.

2. Glass-Box Evaluation

For the glass-box evaluation we need to decide on two issues. The first is a global choice and concerns the nature of the representation language. The second concerns the depth of analysis of the various semantic phenomena that one needs to consider.

2.1 Which Representation Language?

In the scope of this paper, what I mean by a semantic representation is an interpretable structure, in other words, a logical form with a model-theoretic semantics. Such a representation has a logical foundation. There are many choices we can make here, among them, representation languages based on:

- propositional logic;
- some description logic (many choices here);
- some modal logic (many choices here, too);
- first-order logic (i.e. predicate logic);
- higher-order logic (i.e. lambda calculus).

The list above is, by and large, ordered on expressive power. An expressive language is nice to have, but often the price to pay is high. Assuming that, in the context of scalable language technology, we take "semantic analysis" not just as the task of representing meaning, but also as the task of automatic reasoning with produced meaning representations, a compromise between expressive power and practical reasoning capabilities is unavoidable.

On the one end of the spectrum we have got propositional logic, a logic with very attractive complexity properties, but with very limited means to model any interesting semantic phenomena. On the other end of the spectrum we got higher-order logic, a language as powerful as one needs for dealing with semantics, but for which no efficient theorem prover exist. In the middle we have first-order logic, which is often preferred to description or modal logics, as the latter lack expressive power to deal with natural language semantics.

First-order logic seems a good choice if one's aim is to provide semantic representations for sentences (i.e. statements, abstract entities that are either true or false in a given situation). If one is interested in associating meaning representations with fragments, i.e. non-sentences, such as noun phrases, verb phrases, prepositional phrases, and so on, there is almost no choice but higher-order logic. A compromise might be found in using a higher-order language for phrases that are non-sentential, and a first-order language for sentences.

2.2 Semantic Ingredients

What follows here is an inventory of ingredients that ought to be part of deep semantic representations. I am not claiming to present a complete list here, and deliberately left out some complex phenomena such as discourse relations, information structure, focus particles, and metonymy, to keep things both comprehensible and doable. For each of the phenomena, I list the basic choices one is forced to make for its inclusion in a semantic representation.

Quantification Most determiners, several noun phrases, and some adverbs introduce quantification over individuals, times and other abstract concepts. If one sticks to a pure first-order language, partitional quantifiers such as "most" cannot be adequately represented. The alternative is to follow a representation introduced by generalised quantifier theory, distinguishing a restriction and nuclear scope.

Plurals It is notoriously hard to give a simple yet accurate semantic representation for plural noun phrases. The minimal requirement is to be able to account for distributive and collective readings of noun phrases.

Events and States Following Davidson and others, it has been common practice to introduce variables denoting events or states for verb phrases, because it allows one to keep a first-order language while dealing with modification in a rather straightforward way (Parsons, 1980; Dowty, 1989). The choices here are (a) a Davidsonion representation, at the expense of a richer signature of events, or (b) a neo-Davidsonian approach, with a simple signature but with the need to use an inventory of thematic roles.

Thematic Roles There are several proposals here, such as PropBank (Kingsbury and Palmer, 2002) and FrameNet (Baker et al., 1998), with which a semantic representation could synchronise. The alternative is to come up with a small set of pseudo-semantic roles, comprising a basic set of relations such as agent, patient, theme, location, and time. It is however not easy to define a small set of roles that capture all possible relations (Dowty, 1989; Bunt, 2007) and several semantic formalisms have adopted an abstract way of coding thematic relations (Copestake et al., 2005; Kingsbury and Palmer, 2002).

Scope Quantifying noun phrases, modal adverbs, disjunction, negation, etc., all introduce scope. There are two basic choices here: (a) leaving scope underspecified; (b) resolve scope. Option (a) should come with an algorithm to resolve scope. Option (b) faces the difficulty that scope orderings are not always clear (especially in isolation) and therefore additional annotation guidelines are required to deal with such cases — one could pick the strongest or weakest reading, or follow the surface order of the scope bearing elements.

Anaphora and Reference Anaphora, a term that I use here to include personal pronouns, possessive pronouns, reflexive and emphasising pronouns, definite descriptions, proper names, can be either represented resolved to their antecedents or co-referring expressions, or left underspecified in the semantic representation. Either way, we need to decide on the descriptive content contributed by pronouns (for instance, the pronoun "she" introduces the property of being of female gender), and whether to lexically decompose possessive pronouns or not.

Presupposition Many words or phrases trigger presuppositions, and there has been a fair amount of consensus among semanticists what counts as presupposition triggers and what doesn't. Presuppositions clearly have semantic content. The choices here are: do we explicitly represent the presuppositions, and if so, how do we represent them, what triggers will be dealt with (only noun phrases, or also verb phrases, particles, and so on), and finally, do we resolve them or not.

Ellipsis and Gapping In English, ellipsis occurs frequently in the form of elided verb phrases, one-anaphora, or in comparative expressions. The obvious dilemma here is to resolve them or not. If one opts for the latter, the question is whether ellipsis should be encoded in a certain recognisable way or not.

Comparison Phrases In order to have an adequate semantic representation for expressions comparing measurements, such as comparative or superlative modifiers, one might need some form of semantic decomposition. A simple representation for comparatives could be a two-place relation between the two entities that are compared, but in English the entity compared to can be implicit, and hence the question arises whether to resolve it or not in such cases. Superlative expressions could be modelled in terms of the analysis given to comparatives. Independent of this, one has a choice of incorporating the dimension of comparison or not (e.g. "taller" is a comparison of sizes, "older" a comparison of ages).

Tense and Aspect For a language such as English, the least a semantic representation should encode is whether an event takes place in the past, present or future. Such an analysis can be extended to deal with more complex cases covering the progressive or the perfect. In addition, there is a choice to introduce entities for time points and relations between them, such as overlap, culmination, and so on, for instance following Reichenbach (Kamp and Reyle, 1993) or Allen (Allen, 1995).

Time Expressions Expressions indicating dates and clocktimes can have complicated syntactic structures. On the semantic level there is a choice to represent time expressions closely resembling the surface structure, or opt for normalisation of time expressions, as suggested in TimeML (Pustejovsky et al., 2003). The former is easier but lacks generalisation; the latter is probably preferable from an inference point of view but harder to realise in the compositional semantics. Example: the expression "at 7:40pm on May 23, 2007" could be represented as λe.$\exists$x[at(e,x) & 7:40(x)&pm(x) & $\exists$y[at(e,y) & may(y) & 23(y) & 2007(y)]] or normalised as λe.time(e,1940) & date(e,23052007).

Implicit Relations Many semantic relations are implicit and need to be inferred from the context. A case in point, for English, are noun-noun compounds, but the same applies to possessives and relational nouns. One could choose to leave these relations unspecified, or instead resolve them over a set of relations dealing with the most common instances.

2.3 Logical and Non-Logical Conventions

A semantic representation consists of both logical and non-logical symbols, the logical symbols usually being the boolean operators and the quantifiers, and the non-logical symbols those that introduce properties and relations between entities. There is no standard convention how to represent the non-logical symbols. We could take the morphological root or the original token, add a part-of-speech tag or not, and apply word sense disambiguation (for instance against WordNet) or leave the sense underspecified.

The logical symbols could add further complication caused by the fact that two semantic representations could have different syntactic structure, but be logically equivalent. It seems counter-productive to annotate all possible logical equivalent structures for a sentence, if not impossible. To deal with this problem, a normal form will be essential for the representation of quantifiers and other scope-bearing operators.

2.4 Summing up

It should be clear that, for a meaningful semantic annotation, many choices need to be made. It will be a right kerfuffle to design an integrated annotation scheme for semantic representation. I am not necessarily arguing against doing so — but I don't see much point in arguing too much about it. Whatever representation one goes for, it will always be an approximation, and no doubt one day there will be a semanticist kindly pointing out the shortcomings of the analysis for the odd semantic phenomenon. In any case, I expect it to be hard to come up with one level of representation that is both feasible, reflecting the state-of-the-art in the field, and reasonably semantically adequate.

The example in Figure 1 is meant to illustrate this point, showing difference in semantic representation with respect to Davidsonian vs. neo-Davidsonian analysis of event structure, underspecified vs. resolved noun-noun compounds, scopeless vs. explicit scope for negation, and surface vs. deep analysis of tense. An annotated corpus with various coding levels of granularity, reflecting the depth of semantic analysis for each phenomenon, would partially solve the semantic annotation problem.

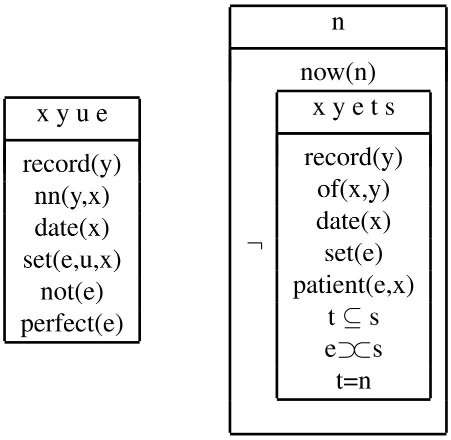

Figure 1: Similar, yet different. Two examples of semantic representations cast in DRT's Discourse Representation Structures (Kamp and Reyle, 1993) for the sentence **A record date hasn't been set**, an example taken from the Penn Treebank (Marcus et al., 1993). Spot the differences.

3. Black-Box Evaluation

There are ways of evaluating semantic components of implemented systems that do not require a look under the hood. I will discuss two of such black-box evaluation methods: recognising textual entailment, and textual model checking.

3.1 Recognising Textual Entailment

The task of automatically recognising entailment relations for pairs of text was, as far as I am aware, first proposed in the FRACAS project (Cooper et al., 1996), comprising a test suite with ca 350 linguistically interesting examples. The FRACAS test suite is grouped into linguistic and semantic phenomena and presented in a list of textual premisses followed by a yes/no-question and the correct answer (Yes, No, or Don't know). The test suite is not balanced: the majority of the examples are positive entailment pairs. Several example pairs of the FRACAS test suite are shown in Figure 2.

Recognising textual entailment (RTE, in short) became popular when it was organised as a shared task in the context of the PASCAL challenges (Dagan et al., 2006; Bar-Haim et al., 2006; Sekine et al., 2007). The PASCAL RTE data is similar to the FRACAS test suite examples, but there are important differences. The PASCAL examples are based on real data (rather than artificially constructed examples), and are not grouped or marked for any specific semantic phenomenon. The PASCAL data set consists of text-hypothesis pairs, and are marked as either true (the hypothesis follows from the text) or false (hypothesis doesn't follow from the text). It is a balanced test suite: the number of positive examples is the same as the number of negative ones. An example is shown in Figure 3.

The RTE task is an attractive method for evaluating semantics because it is completely independent of the type of semantic formalism ("theory neutral") and it works on

3.18: *Every European has the right to live in Europe. Every European is a person. Every person who has the right to live in Europe can travel freely within Europe.*

Can every European travel freely within Europe? (Yes)

3.33: *An Irishman won a Nobel prize.*

Did an Irishman win the Nobel prize for literature? (Don't know)

3.38: *No delegate finished the report.*

Did any delegate finish the report on time? (No)

Figure 2: FRACAS test suite examples for generalised quantifiers (Cooper et al., 1996)

T: *Jader Barbalho, once president of the country's largest political party, was arrested on Saturday in the Amazonian city of Belem, after officials in the neighbouring state of Tocantins issued a warrant against him.*

H: Jader Barbalho is the president of Amazon.

Figure 3: Example 7 of the second PASCAL RTE challenge (Bar-Haim et al., 2006)

open domain data. An issue with the open domain PASCAL RTE examples is the difficulty of isolating the semantics task from the task of acquiring the relevant background knowledge (Zaenen et al., 2005). In contrast, the FRACAS test suite was deliberately constructed to limit the amount of additional background knowledge to make the required inferences.

A second criticism is that the PASCAL RTE text-hypothesis pairs are not designed to measure performance on specific phenonema. As a consequence, it is difficult to detect why a system fails or succeeds in certain cases, and it is often hard to explain why certain systems are good and others are bad. The FRACAS examples are grouped according to various linguistic phenomena (and are meant to test only one phenomenon per example), but a practical obstacle with the FRACAS style of examples is the difficulty to produce a natural test suite of training, development and test data.

Finally, there is often debate as to what constitutes "textual entailment". This is surprising, as there is a long tradition to use textual entailment examples in formal semantic textbooks to illustrate the concept of meaning. (Indeed, the first examples of textual entailment are the syllogisms of Aristotle.) Perhaps it helps to use more commonplace terminology. Here is what I propose, given a pair of two texts A and B: B is *informative* wrt A (not entailed), B is *not informative* wrt A (entailed), or A and B are *inconsistent*. Hence, the example in Figure 3 is classified as informative — the H text contains new information with respect to the T text.

3.2 Textual Model Checking

An alternative black-box evaluation method that I would like to present here is to perform the evaluation of a semantic component by model checking. Simply put, an NLP system is given a sentence (or text) and an abstract description of a situation (a model) and is asked whether the sentence is true or false (or unknown) in the given situation. Systems are then evaluated on the accuracy of correctly evaluated statements. As far as I know this proposal hasn't been made before, although there is an obvious link with evaluating natural language queries to relational databases (Tang and Mooney, 2000; Minock, 2005).

M	=	<D,F>
D	=	{d1,d2,d3,d4,d5,d6}
F(bill)	=	d1
F(hillary)	=	d2
F(harvard)	=	d5
F(stanford)	=	d6
F(graduate)	=	{d3,d4}
F(patient)	=	{(d3,d1),(d4,d2)}
F(from)	=	{(d3,d5),(d4,d5)}

true	Bill graduated.
true	Hillary graduated.
true	Bill and Hillary graduated.
true	Bill or Hillary graduated.
false	Either Bill or Hillary graduated.
false	Bill never graduated.
false	Hillary graduated from Stanford.
false	Hillary did not graduate from Harvard.

Figure 4: Example of a (simplified) model and true and false statements.

What I mean by a model is the usual concept of a first-order model as employed in formal semantics. Formally, a model M consists of a domain (D) of entities and an interpretation function (F). The interpretation function maps properties and relations symbols to elements of D, to sets of elements of D, or to set of tuples of elements of D. Such models only represent all positive information (all information not represented is implicitly negative). Figure 4 shows an example model and a set of true and false statements. An example application of textual model checking is the CURT system (Blackburn and Bos, 2005).

Like the RTE task, this method has the advantage that it is reasonably independent of the semantic representation language, even though it presupposes a common signature of non-logical symbols and is not completely theory-neutral (the example in Figure 4 assumes a neo-Davidsonian approach to events). It remains unclear how hard it is to construct large test suites of natural, open domain examples. However, it seems that, once there is an initial set of examples of texts and model pairs, the amount of training data can be rather straightforwardly extended by inducing (small) variants of the existing models. An additional advantage is that it is easy to construct both positive and negative examples which facilitates the creation of balanced test suites.

4. Conclusion

Evaluating a semantic component of an NLP system is hard. A first idea is to annotate texts with semantic representations, and compare those with what systems produce — the glass-box method. Even though the design of annotation schemes has been initiated for single semantic phenomena (Pustejovsky et al., 2003; Bunt, 2007), there exists no annotation scheme (as far as I know) that aims to integrate a wide range of semantic phenomena all at once. It would be welcome to have such a resource at one's disposal, and ideally a semantic annotation scheme should be multi-layered, where certain semantic phenomena can be properly analysed or left simply unanalysed. This will decrease the risk of running into endless discussion on what an adequate level of analysis or which theory should be adapted to tackle a certain semantic phenomenon.

Alternatively, systems can be evaluated independently of their semantic formalism or theory used — the black-box method. Well-known is the task of recognising textual entailment, another method is textual model checking. The disadvantage of the black-box method that it is often impossible to distinguish the task of semantic interpretation with that of acquiring the relevant background knowledge, a problem that has been around since the story comprehension tasks pioneered in the early 1970s (Winograd, 1971; Charniak, 1972). Trying to separate background knowledge from the task usually leads to artificial test examples, because, as we all know, natural language is inherently intertwined with knowledge of the world.

Acknowledgements

I am supported by a "Rientro dei Cervelli" grant awarded by the Italian Ministry for Research. I would like to thank Malvina Nissim, Michael Schiehlen, and Josef van Genabith for comments and discussion that helped improve this paper.

5. References

James Allen. 1995. *Natural Language Understanding*. Benjamin/Cummings, 2nd edition.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics. Proceedings of the Conference*, Université de Montréal, Montreal, Quebec, Canada.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, and Danilo Giampiccolo. 2006. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, Venice, Italy.

Patrick Blackburn and Johan Bos. 2005. *Representation and Inference for Natural Language. A First Course in Computational Semantics*. CSLI.

Johan Bos, Stephen Clark, Mark Steedman, James R. Curran, and Julia Hockenmaier. 2004. Wide-Coverage Semantic Representations from a CCG Parser. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING '04)*, Geneva, Switzerland.

Johan Bos. 2005. Towards wide-coverage semantic interpretation. In *Proceedings of Sixth International Workshop on Computational Semantics IWCS-6*, pages 42–53.

Harry Bunt. 2007. The semantics of semantic annotation. In *Proceedings of PACLIC-21, the 21st Pacific Asia Conference on Language, Information and Computation*, pages 13–28, Seoul, Korea.

Eugene Charniak. 1972. *Towards a Model of Children's Story Comprehension*. Ph.D. thesis, Massachusetts Institute of Technology.

Robin Cooper, Dick Crouch, Jan Van Eijck, Chris Fox, Josef Van Genabith, Jan Jaspars, Hans Kamp, Manfed Pinkal, David Milward, Massimo Poesio, and Steve Pulman. 1996. Using the Framework. Technical report, FraCaS: A Framework for Computational Semantics. FraCaS deliverable D16.

Ann Copestake, Dan Flickinger, Ivan Sag, and Carl Pollard. 2005. Minimal recursion semantics: An introduction. *Journal of Research on Language and Computation*, 3(2–3):281–332.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Lecture Notes in Computer Science*, volume 3944, pages 177–190.

Rodolfo Delmonte. 2006. Venses–a linguistically-based system for semantic evaluation. In *Lecture Notes in Computer Science*, volume 3944, pages 344–371.

David Dowty. 1989. On the semantic content of the notion thematic role. In *Properties, Types, and Meanings*, volume 2. Kluwer.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT*. Kluwer, Dordrecht.

Paul Kingsbury and Martha Palmer. 2002. From treebank to propbank. In *Proceedings of the 3rd LREC*, Las Palmas, Canary Islands, Spain.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

Michael Minock. 2005. A phrasal approach to natural language access over relational databases. In *Proc. of Applications of Natural Language to Data Bases (NLDB)*, pages 333–336, Alicante, Spain.

Dan I. Moldovan, Christine Clark, Sanda M. Harabagiu, and Daniel Hodges. 2007. Cogex: A semantically and contextually enriched logic prover for question answering. *Journal of Applied Logic*, 5(1):49–69.

Terence Parsons. 1980. Modifiers and quantifiers in natural language. *Canadian Journal of Philosophy*, 6:29–60.

James Pustejovsky, José Castano, Robert Ingria, Roser Saurí, Rob Gaizauskas, Andrea Setzer, and Graham Katz. 2003. TimeML: Robust specification of event and temporal expressions in text. In *Proceedings of the Fifth International Workshop on Computational Semantics (IWCS-5)*, Tilburg, January.

Manabu Sato, Daisuke Bekki, Yusuke Miyao, and Jun'ichi Tsujii. 2006. Translating hpsg-style outputs of a robust parser into typed dynamic logic. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 707–714, Morristown, NJ, USA. Association for Computational Linguistics.

Michael Schiehlen. 1999. *Semantikkonstruktion*. Ph.D. thesis, Universität Stuttgart.

Satoshi Sekine, Kentaro Inui, Ido Dagan, Bill Dolan, Danilo Giampiccolo, and Bernardo Magnini, editors. 2007. *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. Association for Computational Linguistics, Prague, June.

Lappoon R. Tang and Raymond J. Mooney. 2000. Automated construction of database interfaces: Integrating statistical and relational learning for semantic parsing. In *Joint Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 133–141, Hong Kong.

Terry Winograd. 1971. *Procedures as a Representation for Data in a Computer Program for Understanding Natural Language*. Ph.D. thesis, Massachusetts Institute of Technology.

Annie Zaenen, Lauri Karttunen, and Richard Crouch. 2005. Local textual inference: Can it be defined or circumscribed? In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 31–36, Ann Arbor, Michigan, June. Association for Computational Linguistics.