

FROM LEXICAL ITEM TO DISCOURSE MEANING: COMPUTATIONAL AND REPRESENTATIONAL TOOLS

1. THE SPAN OF COMPUTATIONAL SEMANTICS

Computing meanings is something that we do when we read, when we write, when we listen and when we speak, and to a certain extent also when we think and when we dream. We seem to do it virtually all the time, and yet it isn't even entirely clear what we *mean* by saying that we compute *meanings*; the very concept of meaning is not beyond discussion.

Computers are superior computing devices; if we could get them to compute meanings in the sense that they associate similar meanings with natural language utterances and texts as people usually do, then that would open fascinating possibilities such as machines that understand what we mean when we talk to them, or machines that can express in ordinary language the regularities that they find in huge amounts of data, or machines that can find exactly those documents and document passages on the internet that are relevant for a particular purpose, based on an understanding of the texts and of the purpose.

In order to realize such possibilities, it would clearly help to have a good understanding of what we mean by 'meaning', and of the kinds of processes and information that are involved in the computations that people carry out when they associate meanings with linguistic objects. Even though we do not know much about the way people compute meanings, there are certain things that we can say on the basis of a conceptual and logical analysis of what it is that we are trying to compute, and of what must necessarily go into the computation.

First, when we compute the meaning of an utterance, two of the main sources of information that we have are the words that make up the utterance and the particular way in which they form the sentence. The intuitively obvious facts that (1) words carry meaning, and (2) the way in which we put words together to form phrases, clauses and sentences (and texts and dialogues) also carries meaning, is the background of the *compositionality* assumption, which says, roughly, that the meaning of a sentence is a function of the meanings of the constituent words and the syntactic structure of the sentence. These two

sources, lexical meanings and syntactic composition, make up the *linguistic* information that we have at our disposal for computing meanings, together with prosodic information in the case of spoken utterances, and punctuation and layout for textual utterances. The compositionality principle plays a guiding role in formal semantics, and is also useful in computational semantics precisely because it says something (although at a rather abstract level) about the way meanings of complex linguistic objects may be computed from constituent meanings; at the same time it is dangerously restrictive in that it precludes nonlinguistic information to go into the computation of meaning. (See further Bunt and Muskens, 1999 and Janssen, 1997, for detailed discussions of the principle of compositionality.)

Second, formal semantics has traditionally been preoccupied with studying the meanings of *phrases* and *sentences*, but when we read, speak, listen, or write, we are computing meanings not of words or sentences, i.e. of abstract linguistic constructs, but of concrete objects such as spoken dialogue utterances, movie subtitles, web pages, lines of text on a monitor, and textual elements in an advertisement. The meanings that we compute for such objects are closely tied to the *context* in which they occur, and they are tied to that in a double sense. First, the meaning of, for example, a spoken dialogue utterance has to be something that operates within that context. This is not something like a truth condition or a function from possible worlds to truth conditions, but rather something that relates to the current intentions and beliefs of the dialogue participants. Similarly for web pages and billboard advertisements, where the roles of the designer and the reader may be compared to those of the speaker and hearer in a spoken dialogue. Second, the contextually relevant meaning of an utterance can obviously not be computed solely from the linguistic information in the utterance. There is also linguistic information *outside* the utterance, since a word or a sentence in reality never comes in isolation, but is always part of a larger text or conversation; the meanings that we compute therefore depend strongly on the discourse context. And the information that goes into the computation of utterance meanings is of course not restricted to linguistic information; to understand an utterance in the context in which it occurs one should consider it in relation to other visual and auditory information, such as the graphical elements of a web page, the pictures in an advertisement, the action in the movie that is subtitled, and the gestures and mimics of a speaker in a conversation, and one should also take into account the social and institutional positions of discourse participants and the nature of the discourse that the utterance forms part of. More generally, we always seem to compute context-dependent meanings (or ‘interpretations’) of linguistic objects by combining linguistic information with nonlinguistic

‘world knowledge’ of various kinds. In that way we *exploit* the ambiguity and vagueness that natural language expressions have *per se*, and can use natural language as an extremely flexible means of communication. Computing such context-dependent meanings thus means that we have to deal with the entire span from lexical items to context and discourse meanings, which is conceptually enormously wide, and which gives rise to many complex technical issues in computational semantics.

This is a pervasive phenomenon, that emerges virtually everywhere in the computation of meaning, notably in dealing with ambiguity and vagueness. One particular manifestation of this is the phenomenon of *metonymy*, where a speaker says one thing while clearly meaning something else, as illustrated in the following examples:

- (1) a. Mary finished the book.
- b. Mary enjoyed the book.

The verb ‘finish’ semantically takes an eventuality as its complement. Depending on whether Mary is known to be a writer, (1a) will be interpreted as *Mary finished writing the book* or as *Mary finished reading the book*. (Other readings are also possible, for instance if Mary is an illustrator, a text editor, a proof reader,...) Similarly, (1b) has as its most obvious interpretation the one where Mary enjoyed reading the book. Such interpretations can be obtained by adding to the lexical item for *book* (or to a supercategory of it) the information that books are meant for reading, and that books are written (and illustrated, edited, proof read, printed,...). If such information is lexically available, then compositional rules can produce the intended readings. However, as Lascarides and Copestake (19987), have argued, if Mary is a goat, then (1a) means that Mary ate the last of the book, and (1b) that Mary enjoyed eating the book - meanings which cannot be obtained from the lexical meanings of the words in the utterances, since the lexicon will not characterize books as being intended to be eaten. This illustrates that real-world knowledge (‘goats eat anything, even books’) can override purely linguistic knowledge in the computation of meanings. (Lascarides and Copestake 1998 go on to outline a way in which this can be organized, using ‘persistent default unification’ for the percolation of lexical knowledge and the pragmatic theory DICE for taking world knowledge into account.)

Third, when we say that computing meanings means ‘taking world knowledge into account’, we are in fact saying that the computation of meaning involves *reasoning*. Reasoning, or ‘making inferences’, is a process that can take many forms, such as logical deduction, statistically-based pattern recognition, or activity in a neural network. Blackburn *et al.* in their chapter in

this book discuss the role of inference in computational semantics, but they forego a definition of ‘inference’; it seems clear, though, that the application of grammar rules is typically *not* regarded as a form of inference (although it can be cast in the form of applying an inference system to a set of axioms, as in the ‘interpretation-as-abduction’ approach of Hobbs et al.; see Hobbs et al., 1983; Hobbs, 2000). The difference between the application of grammatical and lexical rules in a parser/interpreter on the one hand, and the use of general reasoning and world knowledge on the other, seems mainly a matter of the degree in which processes are constrained, general reasoning being less constrained in which pieces of information may be combined to form a correct inference than the process of parsing and interpreting a sentence.

In models of the computation of meaning, intended to apply in meaning computation by computers, *representations* of the input, outputs, and intermediate results in inference processes play a crucial role. This is due to the fact that inference systems based on logical formalisms are tied to particular representations. In computational semantics, representations are important at four levels:

1. at the lexical level, in the representation of the meaning aspects encoded in lexical items;
2. at the grammatical level, to represent the meanings of the combination of words into phrases and clauses;
3. at the discourse level, in representing the contextual meanings of utterances, texts, dialogue contributions, and other concrete linguistic objects;
4. at the level of context, to represent the context information that goes into computing meanings (linguistic, discourse information, as well as nonlinguistic situational information and general world knowledge).

Traditional logical representation formalisms, such as first-order predicate calculus, modal logics, higher-order logics, or typed λ -calculus, can all be said to have *merciless precision*: in contrast to natural language, where ambiguity and vagueness are essential features, the languages of formal logic are traditionally designed to be fully precise and unambiguous. This often creates problems in the application of these formalisms to natural language interpretation, for example because they tend to lead to architectures where disambiguation is forced with a finer ‘granularity’ than sensible, and at a stage where the information that would be needed for disambiguation is not available, with the effect that spelling out all ambiguity leads to an explosion of possibilities, most of which have to be ruled out at a later stage (see the discussion in Bunt and Muskens, 1999). One of the interesting discoveries of computational semanticists in the last decade is the use of *underspecification* in all kinds of representation (see Bos, 1999; Pinkal, 1999; Bunt and

Muskens, 1999). The use of underspecification allows meaning computation processes to avoid being overly precise and to avoid premature and unnecessary disambiguation. Underspecified representations are currently considered not only at the phrasal and clausal level, but also at the other three levels mentioned above.

2. ABOUT THIS BOOK

The contributions in this book cluster around four themes, relating to the span of computational semantics as outlined above, and to the computational and representational tools involved in dealing with that span:

1. Lexical semantics
2. Inference
3. Underspecified semantic representation
4. Context and contextual meaning

The first chapter, by **Patrick Blackburn, Johan Bos, Michael Kohlhase and Hans De Nivelle**, discusses inference in computational semantics. The authors argue that state-of-the-art methods in first-order *theorem proving* and *model generation* are of direct relevance to inference for natural language processing. This claim is supported by discussing an implementation by Johan Bos of Rob van der Sandt's presupposition projection algorithm in Discourse Representation Theory, an approach which demands sustained use of powerful inference mechanisms.

The next four chapters are concerned both with inference and with lexical semantics. In their contribution *Building a Semantic Lexicon: Structuring and Generating Concepts*, **Federica Busa, Nicoletta Calzolari, Alessandro Lenci and James Pustejovsky** address the representation of lexical semantic information. One of the main challenges for computational lexical semantics is to bridge the gap between on the one hand theoretical research on the organization of the lexicon and on the formal representation of word meaning, and on the other hand the increasing need of natural language processing systems to access large repositories of lexical knowledge. Starting from some recent extensions of Pustejovsky's *Generative Lexicon* theory (Pustejovsky 1995; Pustejovsky 1998), the authors present a general model for the development of a set of large-scale lexical resources developed in the context of the SIMPLE project.

They argue that the principles of the Generative Lexicon provide a framework for structuring word meaning which allows for important synergies



<http://www.springer.com/978-1-4020-0175-8>

Computing Meaning

Volume 2

Bunt, H.; Muskens, R.; Thijsse, E. (Eds.)

2001, VI, 306 p., Hardcover

ISBN: 978-1-4020-0175-8