
Contents

Computational Semantics	
<i>Chris Fox</i>	3
1 Introduction	4
1.1 Outline	5
2 Background	6
2.1 A Standard Approach	6
2.2 Basic Types	10
2.3 Model Theory and Proof Theory	11
2.4 Lexical Semantics	13
3 State of the Art	14
3.1 Discourse	14
3.2 Underspecification	18
4 Research Issues	23
4.1 Type Theory	23
4.2 Intensionality	25
4.3 Non-indicatives	27
4.4 Expressiveness, Formal Power, and Computability	32
5 Corpus-based and Machine Learning Methods	35
5.1 Latent Semantic Analysis	35
5.2 Extraction of Semantic Roles	36
5.3 Word Sense Disambiguation	36
5.4 Textual Entailment	37
5.5 Relationship to Formal Semantics	39
6 Concluding Remarks	40
References	41

Computational Semantics

Chris Fox

CSEE
University of Essex
Wivenhoe Park
Colchester
Essex CO4 3SQ
United Kingdom
`foxcj@essex.ac.uk`

Cross-references (e.g. to ML methods) to be finalised when the book is compiled.

A draft chapter for the Blackwell *Computational Linguistics and Natural Language Processing Handbook*, edited by Alex Clark, Chris Fox and Shalom Lappin. This draft formatted on 30th June 2009.

1 Introduction

In this chapter we will generally use “semantics” to refer to a formal analysis of meaning, and “computational” to refer to approaches that in principle support effective implementation, following Blackburn & Bos (2005).

There are many difficulties in interpreting natural language. These difficulties can be classified into specific phenomena—such as scope ambiguity, anaphora, ellipsis and presuppositions. Historically, different phenomena have been explored within different frameworks, based upon different philosophical and methodological foundations. The nature of these frameworks, and how they are formulated, has an impact on whether a given analysis is computationally feasible. Thus the topic of computational semantics can be seen to be concerned with the analysis of semantic phenomena within computationally feasible frameworks.

Unfortunately, the range of phenomena and the number of frameworks that are of relevance to computational semantics are too vast and this chapter too short to be able to do the subject full justice in the space available. Instead, this contribution should be seen as offering merely a taste of some issues in computational semantics, focusing primarily on logic-based approaches. There are differing views on what counts as the canon of computational semantics, and what aspects of semantics are deemed to be “solved”, and which research questions are considered open, and worthy of pursuit. For these reasons, the focus of the chapter will necessarily appear biased and unbalanced, reflecting the interests and prejudices of the author.

One factor that computational semantics requires over and above formal semantics is that we take seriously the notion of a semantic representation whose behaviour can be expressed independently of any model-theoretic interpretation. This is because an effective implementation needs to be able to use and reason directly with this representation: there can be no appeal to some abstract, external model to arbitrate over what inferences are valid.

In a formal theory of semantics, the appropriate inferential behaviour of the representation should be clearly and precisely formulated. Ideally, to ensure that the behaviour corresponds with our intuitions, the relevant behaviours should be captured as transparently as possible.

For computational semantics, the entailments of the representation language should also be computationally feasible. The notions of *decidability* are relevant here. In a *decidable* system, we can determine what does and does not follow from an expression. In a *semi-decidable* system, we can only guarantee to compute things that follow from an expression. This is also called *recursive enumerability*. If something does not follow, then the decision procedure may never halt. In an *undecidable* system, we cannot even guarantee to be able to compute what follows from a statement.

If there is a choice, then typically a decidable formulation should be preferred to a semi-decidable one, which in turn should be preferred to an undecidable formulation. Even a logic that is not decidable in general might be

decidable for those inferences that are of interest—as would be the case if the domain of discourse was finite, for example—but it might be better to adopt a formalism that captures this requirement by design or nature rather than contingently.

In addition to techniques based upon formal semantics, the remit of computational semantics may be taken to include corpus-based machine learning techniques applied to aspects of interpretation, such as word-sense disambiguation, and identification of entailments and semantic roles. Some such methods are touched upon, although they are not the primary focus of this chapter.

1.1 Outline

This chapter is aimed at readers with some knowledge of syntactic theory, and predicate logic. The focus here is on the formal and logical aspects of computational semantics, rather than on linguistic data, or statistical or corpus-based techniques. It is organised as follows. In Section 2 there is a basic introduction to formal semantics, including a discussion of compositionality, elementary types, model theory and proof theory. In Section 3, the “state of the art” treatment of the formal analysis of discourse and underspecified representations of quantifier scoping are outlined. In Section 4, some relatively open formal topics are sketched, covering type theory, intensionality, and the analysis of non-indicatives. This section also includes some discussion of the issue of power versus expressiveness of formal-representation languages. This covers the idea of treating “computability” as a constraint on formal semantic theories.

Due to limitations of space, it is unavoidable that many important semantic issues will not even be mentioned, including the full range of modalities, hypotheticals, the meaning of names, mass terms and plurals, and the formal analysis of topic and focus, and tense. It is also not possible to do full justice to the many relevant corpus-based techniques, but some of the latter are briefly summarised in Section 5.

Could cross-ref to Chapters where these are discussed?

2 Background

Given that our core characterisation of computational semantics is founded on computationally tractable accounts of meaning that are rooted in formal semantics, it is appropriate to give an introductory account of what is usually meant by formal semantics.

Language is used to convey information. This can be directly, in terms of the literal “content” of an expression, or indirectly, either through accommodating the presuppositions of an expression (van der Sandt, 1992), or through some other forms of implicature (Grice, 1975, 1981).¹

We can use the following examples to illustrate the different kinds of information that can be conveyed.

- (1) a) “*The sun is rising*”
- b) “*Pick the other one!*”
- c) “*Can you pass the salt?*”

The literal content of the first sentence is the claim that sun is rising. In the case of second example, information is conveyed indirectly that there is more than one thing to pick, in addition to the more direct interpretation that something has to be picked. In the final case, we normally conclude that this is a request to pass the salt, not a mere enquiry about an ability.

Some of the more pragmatic notions of meaning may appeal to abilities outside the linguistic realm. In some contexts, the statement “*Wool is horrible when it is wet.*” might actually be a request not to wear a particular garment. Such non-literal meaning may be described as being part of pragmatics (Kadmon, 2001). The boundary between pragmatics and semantics is somewhat difficult to define (see Kamp (1979) for example). As a first approximation, one could claim that semantics is the meaning that can be deduced directly from an expression, with no extra-linguistic information, but ideally in a way that can accommodate any such information.

If we were to include in semantics that which has to be assumed in order to make any sense of what has been uttered, then that would include certain kinds of presuppositions. Indeed, there are claims that all semantic meaning may be characterised as some variety of accommodation (Kamp, 2007). In this chapter we will explore the more “traditional” view of semantics.

In the case of *computational* semantics, we are interested not just in abstract accounts of meaning, but also in their concrete formalisation in ways that, at least in principle, are able to support implementation.

2.1 A Standard Approach

In general it is difficult to reason directly in terms of sentences of natural language. There have been attempts to produce proof-theoretic accounts of

¹ For criticisms of Grice see for example Davis (1998).

sentential reasoning (Zamansky *et al.*, 2006; Francez & Dyckhoff, 2007), but it is more usual to adopt a formal language, either a logic or some form of set-theory, and then translate natural language expressions into that formal language. In the context of computational semantics, that means a precise description of an algorithmic translation rather than some intuitive reformulation of natural language.

Such translations usually appeal to a local principle of *compositionality*. This can be characterised by saying that the meaning of an expression is a function of the meaning of its parts. This view is often attributed to Frege (although see Janssen (2001) for a different view).

In computational semantics there are two common approaches to specifying compositional functions. Essentially all that is required in most cases is some mechanism for substituting the meaning of a constituent into a larger expression. Both unification (Moore, 1989) and λ -calculus (Montague, 1973; Blackburn & Bos, 2005) can achieve this end. In the case of unification based formalisms, syntactic expressions are typically in the form of feature-value structures, and the grammar gives rules of composition indicating how the features are to be unified (combined) and whether any additional constraints are to be imposed. Semantic interpretations can just be viewed as another feature, with variables that are also constrained by feature value constraints in the grammar and within the constituents.

When using the λ -calculus, the composition of semantic forms is expressed in a language that supports substitutions of arguments for variables in a term. Subject to some side-conditions on variable names, an expression of the form $\lambda x.t$ when given an argument t' will be identical to t , but with all occurrences of x in t replaced by t' . To a first approximation, $(\lambda y.\dots man'(y)\dots)(John')$ will be identical to $\dots man'(John')\dots$ ²

The choice of λ -calculus versus unification need not be exclusive, for example the instantiations of the arguments in a λ -calculus approach might itself be accomplished by way of unification. Also, unification-based formalisms might appeal to λ -calculus abstractions for certain phenomena. Indeed, the λ -calculus itself can be implemented within a unification-based framework (Blackburn & Bos, 2005; Covington, 1994; Pereira & Shieber, 1987). Some have argued that λ -calculus expressions are complex in comparison to unification-based constraint formalisms (Moore, 1989). This might be more a matter of taste: the unification approaches generally speaking adopt the machinery of constraint based grammar formalisms (such as HPSG (Pollard & Sag, 1994)), whereas λ -calculus approaches adopt the machinery of higher-order logic (or similar formalisms) and categorial grammar (Steedman (1993) for example).

To be sure that we can translate every sentence covered by a grammar into a formal representation language, we need to associate each word with some semantic representation, and each rule with a piece of information that can

² This assumes that the y does not occur within the scope of another “ λy ”.

be used to derive a representation for each possible category. Adopting the compositional approach means that the meaning of a sentence then depends upon the meaning of its parts, as analysed by the grammar.

In the case of the treatment proposed by Montague (1973), a categorial grammar (supplemented by transformational operations) was combined with higher order intensional logic (See Section 4.2, 4.4) to produce the semantic analysis. Here we follow Blackburn & Bos (2005) and others in using a context-free grammar for the syntax, and a first-order representation language combined with the λ -calculus for the semantic representations.³

With the grammar

- $$(2) \begin{array}{llll} s & \longrightarrow & np \ vp & \det \longrightarrow \text{'every'} \\ np & \longrightarrow & det \ noun & n \longrightarrow \text{'man'} \\ vp & \longrightarrow & v & n \longrightarrow \text{'woman'} \\ det & \longrightarrow & \text{'a'} & v \longrightarrow \text{'laughed'} \end{array}$$

we can parse the following sentences.

- $$(3) \begin{array}{ll} a) & \text{"A man laughed."} \\ b) & \text{"Every woman laughed."} \end{array}$$

In first order predicate calculus, we want to give these sentences translation of the form:

- $$(4) \begin{array}{ll} a) & \exists x(man'(x) \wedge laughed'(x)) \\ b) & \forall x(woman'(x) \rightarrow laughed'(x)) \end{array}$$

To this end, we can associate the words “*man*”, “*woman*” and “*laughed*” with the predicates man' , $woman'$ and $laughed'$ respectively. The determiners will have to contribute the following quantified expressions:

- $$(5) \begin{array}{ll} a) & \exists x(\langle noun \rangle(x) \wedge \langle verb \rangle(x)) \\ b) & \forall x(\langle noun \rangle(x) \rightarrow \langle verb \rangle(x)) \end{array}$$

To perform compositional semantics we need some general way of for composing the meanings of constituent categories (e.g. the noun and the verb in this case) so that they ‘fill’ the correct ‘slots’ in the quantified expression. When we combine the determiner with the noun, we want the meaning of the noun to be substituted for $\langle noun \rangle$ to give the meanings of the noun-phrases:

- $$(6) \begin{array}{ll} a) & \exists x(man'(x) \wedge \langle verb \rangle(x)) \\ b) & \forall x(woman'(x) \rightarrow \langle verb \rangle(x)) \end{array}$$

When we subsequently combine a noun-phrase with a verb-phrase we want to substitute the meaning of the verb-phrase ($laughed'$ in this case) for $\langle verb \rangle$ in the meaning of the noun-phrase. As mentioned above, this substitution could be performed if we could use some mechanism like unification, which

³ In general there may be issues to resolve when combining a logic with a λ -calculus, which we put to one side at this point (see Section 2.2).

is readily available in logic programming languages such as **Prolog**. Here we will use the λ -calculus. Typically, semantic annotations on the grammar will tell us which λ -calculus expressions to use at each stage, and the rules of the calculus will tell us how to produce the final representation.

To perform compositional semantics with a context free grammar, then for each rule in the grammar (and each word in the lexicon) we need to state how to compose the semantics of the category that is being defined. This will be defined in terms of the semantics of the constituent categories (those categories to the right of the arrow). We can use the notation: $\llbracket \langle category \rangle \rrbracket$ to indicate that we are referring to the semantics of $\langle category \rangle$

(7) An example of a grammar with semantic annotations

sentence	→ np vp	$\llbracket np \rrbracket (\llbracket vp \rrbracket)$
np	→ det noun	$\llbracket det \rrbracket (\llbracket noun \rrbracket)$
vp	→ verb	$\llbracket verb \rrbracket$
det	→ “a”	$\lambda P. \lambda Q \exists x (P(x) \wedge Q(x))$
det	→ “every”	$\lambda P. \lambda Q \forall x (P(x) \rightarrow Q(x))$
noun	→ “man”	man'
noun	→ “woman”	$woman'$
verb	→ “laughed”	$laughed'$

In an attribute-value grammar, we can represent such semantic annotations as one of the attributes of the categories (Johnson, 1988).

The annotated grammar (7) is sufficient for the simple sentences of (3). The semantic annotations becomes more complicated if we consider more syntactic constructions such as transitive verbs, auxiliary verbs, adjectives and adverbs. We would also need a richer semantic representation language if we were to take account of other aspects of meaning, such as tense, context dependent meaning, knowledge and belief.

To account for transitive verbs, we would need to add a rule of the form:

(8) $vp \rightarrow verb-trans\ np\ \llbracket verb-trans \rrbracket (\llbracket np \rrbracket)$

together with transitive verbs in the lexicon, such as the following:

(9) $verb-trans \rightarrow \text{“loves” } \lambda R (\lambda y (R (\lambda x loves'(x, y))))$

We can then derive the semantics of some sentences with transitive verbs.

(10) a) “A man loves a woman.”
 b) $\exists x (man'(x) \wedge \exists y (woman'(y) \wedge loves'(x, y)))$

As it turns out, this is not always an appropriate representation for transitive verbs (Section 4.2).

There are cases of ambiguity in the semantic analysis that cannot be accounted for at other levels of analysis. A prime example is that of *quantifier scope ambiguity*. The sentence

(11) “Every man loves a woman.”

could have either of the following representations:

- (12) a) $\forall x(\text{man}'(x) \rightarrow \exists y(\text{woman}'(y) \wedge \text{loves}'(x, y)))$
 b) $\exists y(\text{woman}'(y) \wedge \forall x(\text{man}'(x) \rightarrow \text{loves}'(x, y)))$

The analysis given so far just produces the first reading.

To a first approximation, there can be as many interpretations as there are permutations of the orders of the quantifiers. A strictly compositional analysis will only find one quantifier scoping. Extra machinery is required to obtain the additional readings, and to use the context to rule-out inappropriate interpretations. There are other scoping ambiguities, some, such as prepositional attachment, have a syntactic characterisation. We will look at solutions to the problem of quantifier scoping ambiguity in Section 3.2. Some proposals treat all of these ambiguities by way of *underspecification*.

Another issue concerns the representation of anaphora and ellipsis. Additional work is required to resolve anaphora such as pronouns (Section 3.1). Indeed there are general question about the most appropriate representation language and its features (Section 4). In the next section, we will say a few things about *types* in representational languages.

2.2 Basic Types

When considering the representations of nouns, verbs and sentences as properties, relations and propositions, respectively, we may have to pay attention to the nature of the permitted arguments. For example, we may have properties of individuals, relationships that hold between individuals and propositions (such as statements of belief and knowledge), and in the case of certain modifiers, relations that take properties as arguments to give a new property of individuals. Depending upon the choice of permitted arguments, and how they are characterised, there can be an impact on the formal power of the underlying theory. This is of particular concern for a *computational* theory of meaning: if the theory is more powerful than first-order logic, then some valid conclusions will not be derivable by computational means; such a logic is said to be *incomplete*,⁴ which corresponds with the notion of *decidability* (Section 1).

A critical reason for considering this issue arises if the λ -expressions used in the compositional interpretation of meaning are part of the representation language itself. There are good reasons for assuming that this is appropriate (see Section 4.2). Unfortunately, if we impose no constraints on how expressions may be combined, it is then possible to construct a logical paradox. Consider the property R of not being self-applicable. R can be defined by (13).

⁴ It is worth noting, however, that a first-order *theory* (a theory defined in a first-order logic) may be incomplete, as Gödel demonstrated for first-order arithmetic, for example.

$$(13) R(p) =_{def} \neg p(p)$$

If $R(p)$ is a proposition for an property p , then applying R to itself leads to a paradox.

$$(14) R(R) \leftrightarrow \neg R(R)$$

The conventional way of avoiding this problem is to ban self-application. This can be achieved by adopting a typed representation language

The usual approach for expressing such constraints is to implement them as well-formedness criteria for the language of representation itself, by way of *typing constraints* on the well-formedness of the logic. Typically the types are expressed as e for entity, t for a proposition, and $\langle a, b \rangle$ for an expression that takes an argument of type a and returns one of type b . The idea is that every well-formed expression has exactly one type. When interpreting this theory, it is usual to assume a set-theoretic model, where expressions of type $\langle e, t \rangle$, for example, are viewed as sets of elements e (the values for functions from entities to truth-values). This gives rise to *Simple Type Theory* (STT) (Church, 1940). In such a system, it is not possible to define a term R (13) that can be given exactly one type. Such terms are thus not permitted in the representation language, and the paradox of (14) does not arise.

Conventional *Higher Order Logic* (HOL) adopts Simple Type Theory and allows quantifiers to range over expressions of any type. The propositions of higher order logic are expressions that have the type t . In effect, Montague's *Intensional Logic* is based on a variant of this type theory, except an additional (pseudo) type is added to account for intensionality (Section 4.2).

There is some further discussion of types in Section 4.1.

2.3 Model Theory and Proof Theory

There are two ways in which traditional formal semantic accounts of indicatives have been characterised. First, we may be interested in evaluating the truth of indicatives (or at least their semantic representation) by evaluating their truth conditions with respect to the the world (or more precisely, some formal representation or model of a world). This can be described as *model-theoretic* semantics. Model-theoretic accounts are typically formulated in set-theory. Set-theory is a very powerful formalism that does not lend itself to computational implementation. In practise, the full power of set theory may not be exploited. Indeed, if the problem domain itself is finite in character, then an effective implementation should be possible regardless of the general computational properties of the formal framework (see Klein (2006) for example).⁵

On the second characterisation of formal semantic accounts, the goal is to formalise some notion of inference or entailment in natural language. If

⁵ See Section 4.4 for a little more discussion on the issue of power versus expressiveness.

one expression in language entails another, then we would like that relation to be captured by any formalisation that purports to capture the meaning of language. This can be described as *proof-theoretic* semantics.⁶ Such rules may lend themselves to fairly direct implementation (see for example van Eijck & Unger (2004); Ramsay (1995); Bos & Oka (2002) the last of which supplements theorem proving with model building).

Although a proof-theoretic approach may seem more appropriate for computational semantics, the practical feasibility of general theorem proving is open to question. Depending on the nature of the theory, the formalisation may be undecidable. Even with a decidable or semi-decidable theory, there may be problems of computational complexity (COMPLEXITY CHAPTER), especially given the levels of ambiguity that may be present (Monz & de Rijke, 2001).

These two different approaches may be considered, broadly speaking, to follow those of Tarski (interpretation) and Gentzen (proof) respectively (Tarski, 1983; Gentzen, 1969). With both the model-theoretic and proof-theoretic approach, radically different assumptions may be made about the nature of the semantic framework, its ontology, the appropriate way of encoding information in the theory, and the underlying philosophical principles that are adopted. In practice, such choices may depend upon methodology, taste and precedent rather than general, universal principles.

At an abstract level, the model-theoretic and proof-theoretic views of indicatives might not appear radically different from each other. Assuming our models of the world have some coherent notion of the relationships between the truth and falsity of various expression that exactly mimics our understanding of language, then any entailment patterns in language can be captured by considering the patterns of truth for the interpretations of the sentences in all models. An indicative expression A entails B exactly when all those models in which A is interpreted as being true also interpret B as being true.

The issue for computational semantics is one of computational tractability of some semantic representation. We could have a representation of a set-theoretic model-theory, although we might question whether in general that is computationally tractable. If possible, we would like to avoid representations that are so powerful that we cannot enumerate their theorems (let alone those for which we cannot even write down all the rules that govern their behaviour). In general, set-theoretic interpretations are among those that are problematic when it comes to computational feasibility. An easier starting point is a relatively weak proof-theoretic representation, but with appropriate expressiveness for the phenomena in question.

⁶ Aristotelean syllogisms can be viewed as a form of proof-theoretic semantics, although where the entailment patterns are captured directly in terms of natural language sentences.

Add cross-reference.

2.4 Lexical Semantics

The meaning of language is more than the ability to compose representations based on the form of sentences and construct formal proofs. Other issues include pragmatic issues of how language is used, and of course the meanings of words themselves (Pustejovsky, 1995).

A lexicon may include lexical features that indicate salient information about the syntactic and semantic arguments of lexical items which are needed to obtain a formal semantic representation. But in general we may also be interested in the concept that is represented by a given word.

For Natural Language Processing this may be difficult to capture. But there may be some aspects of meaning that can be captured and represented. These include ontological classifications of words, such *cause-of*, *agent* and relationships between words. Such relationships might be semantic in character (such as hyponym and meronym relationships etc.), or founded on co-location information, where a word is assumed to be related in meaning to other words that are used in a similar context, which might be described as “distributional” lexical semantics. Many corpus-based techniques (Section 5) assume that at least some aspects of meaning are implicitly embodied in co-location data, and further more, that word classifications can be learnt (Andy Chiu & DiMarco, 2007).

3 State of the Art

There are a range of analyses of natural language phenomena that may be said to constitute the state of the art of computational semantics. Here we pick two issues that have received a significant amount of attention over the years, namely the treatment of anaphora and of quantifier scoping. These are discussed in the sections on discourse (Section 3.1) and underspecification (Section 3.2). This is not to say that the analyses proposed are beyond question, or that all the relevant issues have been resolved, but there is certainly a relatively stable core of ideas and analyses that can be considered state of the art.

3.1 Discourse

Here “discourse” is taken to refer to a sequence of sentences where each sentence is interpreted in the context of the preceding sentences. This context provides potential antecedents for anaphoric expressions such as inter-sentential pronouns, as in the very simple example given in (15), where the antecedent to which “*She*” refers is intended to be ‘Mary’.

(15) “*Mary is a woman. She loves John.*”

where the antecedent might be inferred but is not overtly mentioned in the text. The issue is how can the discourse be represented in a way that allows anaphoric relations to be represented in a manner that is both sympathetic to concerns with quantification and scoping, and which also captures intuitions about felicitous and infelicitous anaphoric reference.⁷

Montague’s treatment of scope (Section 3.2) makes use of anaphora, but it cannot be generalised easily to other cases. One obvious solution would be to consider pronouns as variables, and define some mechanism for these variables to be bound appropriately by the quantifiers (nouns) to which they refer.

Given the sentence

(16) “*Mary is a woman. She loves John.*”

we can try to represent the pronoun “*she*” using a variable.

(17) $woman'(mary') \wedge loves'(x, john')$

Here, “*She*” is an anaphoric pronoun that needs to be resolved so that it is associated somehow with an appropriate *antecedent*. In this case, it would be legitimate to consider replacing the variable by *mary'*. Unfortunately, this solution does not generalise.

If we consider the sentences

⁷ We do not consider other issues concerning the analysis of discourse, such as topic and focus (Eva Hajičová, 1998; Rooth, 1993), or discourse segmentation. Note that there are non-logical, quantitative methods that have been applied to the latter problem (Hearst, 1997).

- (18) a) “A man drank. He fell asleep.”
 b) $\exists x(\text{man}'(x) \wedge \text{drank}'(x) \wedge \text{fell_asleep}'(y))$

the pronoun, represented by y , cannot be resolved by just replacing it with a constant. Renaming y to be x also does not work, because it lies outside the syntactic scope of the existential quantifier, and so is not bound by it.⁸

Some particularly problematic examples are given by Geach (1972), including the following so-called “donkey” examples.

- (19) a) “If a farmer owns a donkey, he beats it.”
 b) “Every farmer who owns a donkey beats it.”

The issue of concern here is that it is not clear that we have the correct analysis of quantifiers or conditionals. If pronouns are to be represented by variables, we need to ensure not only that they are bound correctly, but also that indefinites have universal force in the second example, which a naive analysis would interpret incorrectly as something like

- (20) $(\exists x \text{farmer}'(x) \wedge (\exists y \text{donkey}'(y) \wedge \text{own}'(x, y))) \rightarrow \text{beat}'(x, y)$

where both x and y in the consequent of (20) are outside the scope of the relevant quantifiers, and the sense of universality is not captured. These issues, amongst others, have led people to consider alternative ways of representing meaning, including *Discourse Representation Theory* (DRT). In addition to putting emphasis on the representation itself, rather than focusing on the model theory, DRT also provided an algorithmic account of how to generate these representations from natural language input sentences. Both features are characteristic of computational semantics.

Discourse Representation Theory

Discourse Representation Theory (DRT) and related paradigms intend to capture the notions of discourse that are relevant for resolving anaphoric pronouns by reconsidering the representation of quantifiers and some of the other logical connectives. The idea is to have a representation of the individuals that are introduced into a discourse, and allow them to be referred to in subsequent discourse where appropriate.

Using a *construction algorithm*, DRT systematically builds a representation of the individuals described in a discourse, and the properties and relationships that hold between them. The basic notion in DRT is that of a *Discourse Representation Structure* (DRS), which has the following form:

$$(21) \begin{array}{|l} \langle \text{referents} \rangle \\ \langle \text{conditions} \rangle \end{array}$$

⁸ In this case, we would want x to be evaluated in the same way as the other x 's in the representation. This cannot happen if it is not bound by the same quantifier.

The top part of the box contains individuals described in the discourse. The bottom part contains conditions on those individuals. The conditions may include DRSs.

Essentially, existentially quantified noun phrases introduce a new individual into the current DRS with appropriate conditions.

(22) a) “A woman cried.”

x
b) $woman'x$ $cried'x$

Universally quantified noun phrases introduce a conditional DRS as a condition of the DRS representing the current discourse.

(23) a) “Every man laughed.”

...			
b) <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px;">y</td> </tr> <tr> <td style="padding: 2px;">$man'y$</td> </tr> </table> → <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px;">$laughed'y$</td> </tr> </table>	y	$man'y$	$laughed'y$
y			
$man'y$			
$laughed'y$			

There are rules that govern from where a discourse referent may be referred to, and the construction algorithm indicates where analysis of subsequent discourse should appear in the DRS. Resolution of anaphora can be expressed as equations over discourse referents.

(24) a) “Mary is a woman.”

m
b) $m = mary'$ $woman'(m)$

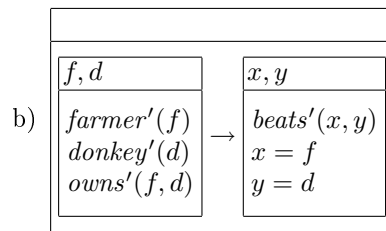
c) “Mary is a woman. She loves John.”

m, j, x
d) $m = mary'$ $woman'(m)$ $j = john'$ $loves'(j, x)$ $x = m$

Here, the pronoun “she” is represented by x , and resolved by the condition $x = m$.

A typical “donkey sentence” where the conditional is interpreted with universal force is exemplified next.

(25) a) “If a farmer owns a donkey, he beats it.”



Accessibility of referents is defined in such a way that the farmer and donkey (f, d) are not accessible from any subsequent discourse (at least, not as singular antecedents).

If DRT is combined with a notion of abstraction and application, then it is possible to produce a more conventional compositional presentation of the construction process (Blackburn & Bos, 1999).

DRT has been exploited for more things than just pronominal anaphora. Examples include underspecification (Reyle (1993) and Asher (1993) for example) presuppositions (van der Sandt, 1988, 1992; Beaver, 2002; Krahmer & Piwek, 1999) and discourse relations (Asher & Lascarides, 2003).

There are many issues that require a more sophisticated analysis, such as plural anaphora,

(26) “*John_i and Mary_j went to Paris. They_{i+j} met at the Eiffel tower.*”

conditional examples where universal quantification is not the most natural interpretation (Pelletier & Schubert, 1989), as in

(27) “*If you have a penny, put it in the box.*”

and examples where it is difficult to see how the appropriate representation might be obtained (Heim, 1990; Kadmon, 1990), such as

(28) “*Most farmers who own a donkey beat it.*”

where the most natural reading is that most donkey owning farmers beat donkeys that they own, rather than the unnatural quantification over farmer-donkey pairs that would be obtained by an unmodified DRT-style analysis.

Dynamic Accounts

There are many other approaches to dealing with pronominal anaphora. The accounts using *dynamic logic* effectively redefine the meaning of quantification and conditionality (Groenendijk & Stokhof, 1990a, 1991). The aim is to allow variables to be bound outside the syntactic scope of existential quantifiers and to give existentials a universal interpretation when appearing as the antecedent of a conditional. This is an example of where the need to deal with a particular phenomena leads to a re-appraisal of the formalism and techniques of conventional classical logic.

Syntactically, the net result is a logic that has the appearance of a classically quantified logic, but where examples such as (20) have the appropriate

semantics by way of a modified interpretation of the logical operators and quantifiers.

DRT and logic are equivalent in their ability to analyse simple discourse with singular pronouns.

Type Theoretic Approaches

We finish this section on discourse by briefly mentioning some type-theoretic approaches. As Sundholm (1989) observed, there are certain aspects of constructive type theory that appear to capture the appropriate behaviour for interpreting discourse involving singular anaphora. In particular, the dependent types that feature in constructive type theory can be used to capture contextual effects. This idea was developed by Ranta (1994) and Ahn & Kolb (1990).

There are alternative approaches that use types for dealing with discourse problems. For example, it is possible to exploit dependent types within a classical framework (Smith, 1984; Turner, 1992; Fox, 2000). Perhaps a more radical approach is due to Lappin & Francez (1994) and Lappin (1989). Rather than characterising the problem of resolving anaphora as one of finding an element with which to equate a pronoun, these proposals suggest that the problem can be construed as one of finding the appropriate *type* for the variable representing the pronoun. This idea is developed in Fox & Lappin (2005) in the context of Property Theory with Curry Typing (PTCT).

Additional relevant information may also be found in the CHAPTERS on DIALOGUE (Ruslan) and DISCOURSE (Fernandez and Ginsburg). We briefly mention constructive type theory and dependent types again in Section 4.1.

Add cross-references.

3.2 Underspecification

One problem for a compositional analysis is semantic ambiguity. This is typically exemplified by the issue of quantifier scoping, but also arises with other scope taking elements, such as modifier expressions, prepositional phrases, negation and other logical operators, as well as anaphoric reference (Poesio & Reyle, 2001). In the case of scoping, the issue is that a sentence with more than one scope taking element is ambiguous in a way that is not usually evident in any syntactic analysis. For example, in an ambiguous sentence such as

(29) *“Every student took a course.”*

it is unclear whether there was one particular course taken by every student, or whether every student took at least one course, but not necessarily the same one in each case.

Montague (1974) offered an approach to the quantifier scoping problem that used additional rules which effectively reordered the quantifiers in the

syntactic analysis, and hence changed the scope in the semantic representation. The current consensus is that it is better to have a systematic account that does not require changes to the syntactic analysis, and which provides an intermediate representation that is unspecified, or underspecified with respect to scope orderings, but which permits all appropriate scope orderings to be generated when required.

Cooper Storage

The prime example of a system intended to allow the generation of scoped readings is Cooper Storage (Cooper, 1983). Although there are other proposals, they can be construed as variations and refinements of this proposal. Cooper Storage builds semantic representations using a data-structure known as a *store*. This can be thought of as providing an underspecified representation of the meaning of a sentence.

The store contains a “core” representation (typically representing the main verb) together with the representations of the generalised quantifiers (typically representing the noun-phrases). The argument positions in the core representation are associated with indices identifying which generalised quantifier (noun phrase) binds that position.

The approach can be illustrated by analysing the following sentence using Cooper Storage.

(30) “*Every man loves a woman.*”

The stored representation will be something like⁹

(31) $\langle \text{love}'(z_6, z_7),$
 $(\lambda p(\forall x(\text{man}'(x) \rightarrow p(x))), 6),$
 $(\lambda p(\exists y(\text{woman}'(y) \wedge p(y))), 7) \rangle$

The derivation of this is sketched in Figure 1.

Given an unscoped representation in the store, *retrieval* operations can be used to generate fully scoped representations. The generalised quantifiers can be applied to the core representation in any order, thus giving rise to the different quantifier scopings. The index is used to ensure that each generalised quantifier binds the correct argument position, so that the meaning of the sentence is not corrupted by the reordering of the quantifiers. Blackburn & Bos (2005, p108) give a worked example of this.

With our example, if we retrieve 6 (“*Every man*”) first, then the store is

(32) $\langle \lambda p(\forall x(\text{man}'(x) \rightarrow p(x)))(\lambda z_6(\text{love}'(z_6, z_7))),$
 $(\lambda p(\exists y(\text{woman}'(y) \wedge p(y))), 7) \rangle$

Applying β -reduction gives us

⁹ The precise subscripts (in this case 6 and 7) and place holder variable names (z_6 , z_7) may vary.

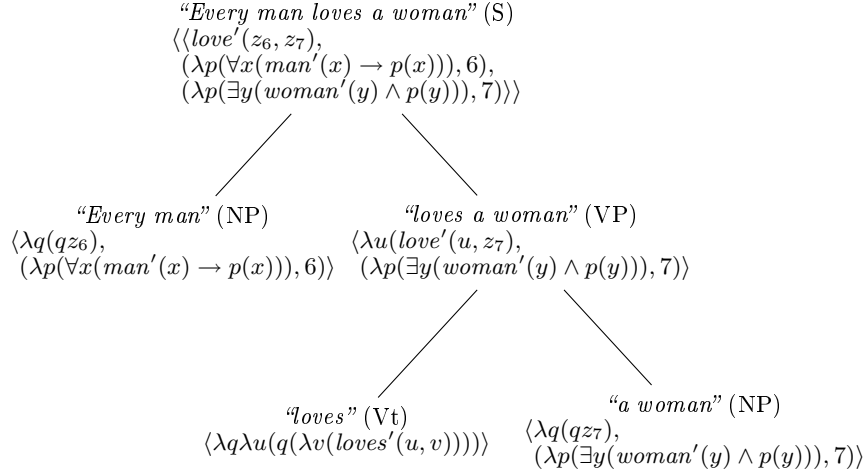


Figure 1. Derivation of semantic representation with storage.

$$(33) \langle \forall x(\text{man}'(x) \rightarrow \text{loves}'(x, z_7)), \\ (\lambda p(\exists y(\text{woman}'(y) \wedge p(y))), 7) \rangle$$

The second, and final, retrieval operation gives us

$$(34) \langle (\lambda p(\exists y(\text{woman}'(y) \wedge p(y)))(\lambda z_7 \forall x(\text{man}'(x) \rightarrow \text{loves}'(x, z_7))) \rangle$$

which after β -reduction is

$$(35) \langle \exists y(\text{woman}'(y) \wedge \forall x(\text{man}'(x) \rightarrow \text{loves}'(x, y))) \rangle$$

Retrieving the items in the opposite order would give us the alternative scope reading for this example.¹⁰

In the account as given, there are some problems in handling relative clauses and complex noun phrases with prepositions. Such phrases can give rise to nested, or hierarchical noun-phrases. If the storage and retrieval operations are not sensitive to such structures, then ill-formed representations may be generated. Consider the following sentence.

$$(36) \text{"Mary knows every owner of a pub"}$$

We should not be able to retrieve the representation of "a pub" until we have retrieved the core part of the noun phrase "every owner". The need for constraints on retrieval is addressed by *nested* or *Keller* storage, where nested stores are used to 'lock up' constituent parts of a noun phrase which can only be accessed once we have retrieved the core noun phrase that contains those parts (Keller, 1988).

A comprehensive account of underspecification needs to handle scoping of negation, conjunction, modification, modalities and propositional attitudes.

¹⁰ Blackburn & Bos (2005, p108) provide more details of this approach.

Furthermore, we might consider approaches that allow *partially* specified representations that can accommodate incremental constraints on acceptable scopings, as in the following example (taken from Fox & Lappin (2007))

- (37) a) Speaker 1: “*Every student wrote a program for some professor.*”
 b) Speaker 2: “*Yes, I know the professor. She taught the Haskell course.*”
 c) Speaker 3: “*I saw the programs, and they were all list-sorting procedures.*”

We can assume the following

- (38) a) “*some professor*” in the first sentence (37a) is the antecedent for both “*the professor*” and “*she*” in the second sentence (37b).
 b) “*a program*” in the first sentence (37a) is the antecedent for both “*the programs*” and “*they*” in the third sentence (37c).

The first assumption (38a) gives “*some professor*” scope over “*every student*” in the first sentence (37a). The second assumption (38a) leads to “*a program*” taking narrow scope with respect to “*every student*” in the first sentence (37a). From this it can be seen that as the discourse proceeds, (37b) and (37c) force on the first sentence (37a) a fully resolved scope order, namely

- (39) “*some professor*”, “*every student*”, “*a program*”

Most treatments of quantifier scoping based on storage do not by themselves provide an efficient analysis of such incremental constraints, nor do they necessarily support direct reasoning with such partially specified scopings.

Other Treatments of Scope Ambiguity

Bos (1995), and Blackburn & Bos (1999) develop a constraint-based system for underspecified representation for first-order logic that they refer to as *Predicate Logic Unplugged* (PLU). This system is a generalisation of the *hole semantics* approach to underspecification which Reyle (1993) first developed within the framework of *Underspecified Discourse Representation Theory* (UDRT).

Minimal Recursion Semantics (Copestake *et al.*, 2006) is an application of hole semantics within a typed feature structure grammar (HPSG). Normal Dominance Conditions (Koller *et al.*, 2003) can be seen as a refinement and development of the central ideas of hole semantics. See Ebert (2005) for detailed discussion and results concerning the formal relations among these theories with respect to their expressive power.

Dalrymple *et al.* (1999) and Crouch & van Genabith (1999) suggest a theory in which representations of generalised quantifiers and core relations are expressed as premises in an underspecified semantic *glue language*. The premises are combined using the natural deduction rules of linear logic (Girard, 1987) to yield a formula that represents the scope reading of a sentence.

Packed Representations (Crouch, 2005) ‘compress’ the scoped interpretations derived using glue language. Components of meaning shared by several

readings are expressed as a single common clause. This uses an approach that is applied in chart parsing to construct a graph for non-redundant representation of the full set of possible syntactic structures for a parsed phrase.

van Eijck & Unger (2004) develop an approach to underspecified representations, in the functional programming language Haskell, which uses *relation reduction* and arbitrary arity relations. This is based on a proposal due to Keenan (1992). This work inspired a proposal by Fox & Lappin (2005) which represents underspecified representations in a data-structure that can be formalised within the representation language PTCT itself. This means that there is no appeal to meta-semantic machinery as such, and also allows the full power of the representation language to be used to express constraints governing the legitimate readings, including incremental constraints. This addresses the concerns of Ebert (2005) with regard to expressive completeness, although it still leaves outstanding the problem of dealing with the significant combinatorial complexity of computing the desired readings.

4 Research Issues

There are many open research questions in computational semantics. Some are concerned with how to analyse particular aspects of meaning, including phenomena that are not easily analysed by way of a direct truth-conditional interpretation. Others are concerned with representations that provide the most appropriate machinery to express and reason with the meaning of natural language in a computationally tractable manner. Here there is only space to consider a small selection of such issues.

4.1 Type Theory

Typically, the types used for natural language semantics are based on Simple Type Theory (Section 2.2). But there are other kinds of types, and other ways of imposing typing constraints. Indeed, it is not entirely clear that Simple Type Theory is the most appropriate type system for natural language semantics. Here we consider some other options.

Polymorphism

The simply typed Higher Order Logic (Section 2.2) might be considered somewhat restrictive. One area in which it appears excessively rigid and inexpressive concerns certain type-general phenomena that are apparent in natural language, such as the apparent polymorphism of conjunction.

- (40) a) *“John and Mary saw Peter”.*
 b) *“John saw and heard Peter”.*
 c) *“The book was red and white”.*

In these examples, the conjuncts and the conjunctive phrase itself are all of the same category. Partee & Rooth (1983) deal with this phenomena by introducing generalised quantification that “raises” the type of the basic conjunction and disjunction operators to t -ending types. An alternative is to adopt a more flexible type system that permits polymorphic types of the form

$$(41) \forall' X.T$$

where \forall' is a type quantifier that allows X in type T to range over all types.¹¹ As an example, an expression of the type $\forall' X.\langle X, t \rangle$ will form a proposition (type t) given an argument of any type. The type of a coordinating expression can then be given as

$$(42) \forall' X.\langle X, \langle X, t \rangle \rangle$$

¹¹ We might restrict the quantification so that it only ranges over non-polymorphic types. See Section 4.4 and Fox & Lappin (2005).

This can also be used to capture other type-general phenomena—such as verbs like “*fun*” that can take nominal expressions, infinitives and gerunds as arguments—without resorting to a universal type for example (Chierchia, 1982).¹²

In addition to looking at ways in which the type system could be made more expressive to match the needs of natural language, as with polymorphic types, there may also be some merit in considering what *constraints* there are on the type system required for natural language.

First-Order Sorts

Rather than adopting a typed higher-order logic, an alternative approach to constraining the way in which entities of the theory may felicitously be combined is to have sortal predicates that classify terms as representing individuals, relations and properties. Logical rules can then be given that express analogues of type inference rules.

For example, we might have predicates *entity'*, *property'*, *proposition'* and a rule that says

$$(43) \quad (\textit{entity}'(i) \wedge \textit{property}'(p)) \rightarrow \textit{proposition}'(p(i))$$

Here the notion of *proposition'* and *property'* must be seen as distinct from those notions in the language in which these statements are being expressed.

Characterising *properties* and *propositions* by way of sortal predicates means that properties and propositions are being treated as first-order terms of the theory. We then need to find some way of asserting that $p(i)$ is true. One option would be to introduce a predicate *holds'* (Kowalski & Sergot, 1986; Miller & Shanahan, 1999) that relates a *proposition'* to an event ($\textit{holds}'(p(i), e)$), or situation (giving rise to a form of event calculus), or else introduce a truth predicate *true'*, as in Property Theory (Turner (1992) for example). Care needs to be taken about what terms count as propositions in order to avoid paradoxes of the kind illustrated by (14).

Sortal constraints can mimic expressive types such as dependent types (Turner, 1992; Smith, 1984), which can provide a treatment for analysing discourse anaphora (Sundholm, 1989; Ranta, 1994) (Section 3.1). There are alternative ways of mimicking higher-order type theory within a first-order logic using Curry typing with polymorphic types without expressing types (Fox & Lappin, 2005), as we shall see below.

Property Theory with Curry Typing

It is possible to avoid some of the strictures of Church typing, by separating out the typing system from the λ -calculus presentation. That is, we can adopt the untyped λ -calculus, and then have typing rules that allow us to infer the

¹² An alternative approach would be to use *schematic* polymorphism (Pollard, 2004).

types of the λ -expressions (Curry & Feys, 1958). This is the approach adopted by Property Theory with Curry Typing (Fox & Lappin, 2005). This approach allows additionally flexibility in developing a type system that is focused on the specific requirements of natural language semantics, including separation types (a form of subtype) and polymorphic types. This is formulated in an essentially first-order language.

Constructive Theories and Dependent Types

Constructive type theory was mentioned before in relation to analysing anaphora (Section 3.1). The constructive approach offers an alternative to classical logic. In constructive systems, propositions are only considered to be true if there is an appropriate proof or *witness*. Using the Curry-Howard isomorphism, propositions can then be viewed as types whose members are their proofs. Such logical systems are slightly weaker than corresponding classical formulations in that they do not support proof-by-contradiction.

As already noted, the intrinsically dynamic nature of the dependent types can be exploited in the analysis anaphoric phenomena, although such types are not exclusive to constructive theories (Section 3.1).

There are other kinds of dependent types, including *record types*, which generalise the notion of dependency over a collection of type expressions. The use of such types has been proposed both for the analysis of discourse and as a language in which attribute-value grammars can be formulated (Cooper, 2005; Cooper & Ginzburg, 2002).

4.2 Intensionality

We need to be able to represent sentences where the verb expresses a relationship involving not just individuals but also propositions and predicates, as in the following examples.

- (44) a) “*John believes that every cat is furry.*”
 b) “*Mary likes red.*”

With some verbs, we also need to be able to distinguish between *de re* and *de dicto* interpretations of the arguments of some verbs. For example, in the sentences

- (45) a) “*John seeks a football.*”
 b) “*John seeks a unicorn.*”

it is clear that in the first example, John may be seeking a real entity that exists and is a football. This is a *de re* interpretation. In the latter case, he is seeking something that does not exist, but he can still be said to be *intending* to find a unicorn. This is a *de dicto* interpretation. In the former case, with a *de re* interpretation we cannot be certain that John knows that he is seeking

a football; he might know it by some other description, such as “*the object lying in the yard*”.

The conventional view is that this requires predicates that can take things other than individuals as their arguments.

A semantic interpretation along the following lines might seem appropriate.

- (46) a) “*John believes that every cat is furry.*”
 $believe'(John', \exists x(cat'(x) \wedge furry'(x)))$
 b) “*Mary likes red.*”
 $likes'(Mary', red')$
 c) “*John seeks a football.*”
 $\exists x(football'(x) \wedge seeks'(John, x))$ *de re*
 $seeks'(John', \lambda P \exists x(football'(x) \wedge P(x)))$ *de dicto*
 d) “*John seeks a unicorn.*”
 $seeks'(John', \lambda P \exists x(unicorn'(x) \wedge P(x)))$

The felicity of this approach depends in part upon the nature of propositions and predicates. If propositions are identified with truth values, then there are only two propositions. Further, any truth-conditionally equivalent propositions may be substituted for each other. This is a particular problem for mathematical truths which are necessarily true together but not identical. This gives some incorrect predictions about equivalence in the meaning of distinct sentences. Similarly, if predicates are just sets, then distinct predicates may be accidentally equated.

The Montagovian Analysis

The classical approach to intensionality is attributed to Montague (1973), although it has its roots in earlier work (for example, (Kripke, 1963; Carnap, 1947)). In the representation, a *function*, or *operator* \cap is introduced that takes a single proposition or predicate as its argument. The result is an intensional expression that can appear as an argument to predicates that have an appropriate type. A second *function/operator* \cup can undo the operation of \cap , so that $\cup \cap p = p$.

For the so-called transparent verbs, such as “*find*”, *meaning postulates* can be introduced that allow the *de re* interpretation to be derived from the *de dicto* one.

Following Montague (1973) we can interpret this intensional theory using *possible worlds* semantics. Possible worlds are commonly used to model modality, such as *possibility*, and *necessity*, *permission* and *obligation* for example (Kripke, 1963; Carnap, 1947; von Wright, 1967). Propositions can be treated as sets of possible worlds; or (equivalently) functions from possible worlds to truth values. Properties can be modelled as functions from individuals to sets of possible worlds (propositions). Propositions that are true together in the

current world may be distinguished from each other provided there are worlds in which their truth values differ.

If p is of type A , then $\bigwedge p$ will be of type $\langle s, A \rangle$. Following Gallin (1975), the type s can be thought of as corresponding to a possible world index. Types of the form $\langle s, A \rangle$ are then functions from world indices to expressions of type A .

Other Approaches

Montague's possible-worlds approach dominates linguistic work in intensionality, but it does have problems. The type system is inflexible and the notions of modality and intensionality are conflated. As a result, the analysis is not sufficiently fine-grained in its treatment of intensionality: for example, propositions that are necessarily true together cannot be distinguished from each other. Such propositions are exemplified by mathematical truths.

An alternative is to take what Montague writes as $\bigwedge p$ to be some kind of *representation* or *encoding* of p that does not conflate propositions merely because they are necessarily true (or false) together. We could take $\bigwedge p$ to be an individual (a term). The identity criteria for propositions would then be syntactic in nature, rather than truth conditional. Such prefixes \bigwedge and \bigcup then serve as functions from propositions (and predicates) to terms, and terms to propositions (predicates). In practice, we may prefer that the default interpretation of p be a term rather than a truth-conditional proposition. This avoids conceptual problems in having a function \bigwedge that increases the intensionality of its argument. Of course, some interpretation of these expressions is then required in order to find an appropriate model theory. This requires a relatively expressive language of terms.

There are potential risks with this strategy. If we are not careful about what can be represented as an individual (and hence appear as an argument to a predicate), then we may introduce paradoxes. Theories that take this approach (or variations of it) include *Property Theory* (Turner, 1992; Bealer, 1982; Cocchiarella, 1985), Property Theory with Curry Typing (PTCT) (Fox & Lappin, 2005), and *Situation Theory* (Barwise & Perry, 1983; Barwise & Etchemendy, 1990), the latter of which comes with a particular philosophical perspective on the nature of "situated" meaning. Another alternative is to find suitably intensional models for theories that are syntactically not far-removed from Montague's IL (Thomason, 1980; Gilmore, 2001; Fox *et al.*, 2002; Pollard, to appear), or use theory that combines different notions of intensionality (Materna *et al.*, To appear; Tichý, 1988). An alternative is to interpret our representation language using some form of intensional set theory (Jubien, 1989).

4.3 Non-indicatives

So far we have only considered indicative sentences. Indeed this is the focus of perhaps the majority of work in computational semantics. But if we are

interested in computing the meaning of language in general, then it is vital to consider non-indicatives.

As before, what follows is not intended to be a comprehensive survey of the work in the respective fields. We merely present a taste of some of the general methodological and practical issues that can arise in computational semantics. To this end, we briefly sample some proposals for the analysis of two significant non-indicative categories, namely questions and imperatives. The fundamental issue that lies behind all of these examples results from the fact there are “entailment” patterns which we might like to capture, but which are not overtly truth-conditional in nature: we do not usually think of questions or imperatives being “true” or “false”.

In the case of imperatives there is some debate about the appropriate nature of any entailment patterns, and even whether a logical approach is possible. In the case of questions, and their answers, it is generally accepted now that an appropriate notion which should be captured by “entailments” between questions can be viewed in terms of answerhood criteria, although there is debate about how these are best expressed. The point of particular interest here lies in the difficulty of specifying what constitutes an answer in a computationally tractable fashion.

This is taken to illustrate the point that in general, many of the core aspects of semantics, such as the truth of a proposition, answerhood for a question, etc., may not themselves be characterised completely within a computationally tractable theory. That is not to say this is a critical flaw for the computational semantics programme, merely that some aspects of a computational theory in effect will include properties of implementations, rather than implemented properties.

Questions and Answers

For questions and answers, we might consider the notion of *answerhood conditions* in place of *truth conditions*. Such an idea was proposed by Belnap (1982) among others. This requires consideration of what might constitute a legitimate answer to a question, and when an answer to one question is also an answer to another.

One influential and comprehensive account of the semantics of questions is due to Groenendijk & Stokhof (1984, 1990b, 1997a). In their model, a question partitions the set of all possible worlds, where each partition corresponds to a different possible answer. A yes/no question would give rise to two partitions, one corresponding to the underlying proposition being true, the other to it being false. A wh-question would give rise to a more complex set of partitions corresponding to the underlying property being applicable to different individuals. A true answer is then considered to be anything that provides the information needed for the questioner to determine in which partition the actual world lies. A partial, true answer would indicate a collection of partitions in which the actual world may lie. In general an answer (whether true or

false) will provide a means of “eliminating” certain worlds, and hence certain possibilities, from consideration.

Due to the model-theoretic approach, the theory is not presented directly in terms of inference rules concerning the nature of answers and answerhood conditions; although it might provide a useful model, it does not necessarily lend itself to direct implementation. There is also the issue of combinatorial explosion when it comes to the evaluation of wh-questions. If the size of the domain is n , then checking the consistency of every field of a wh-question requires 2^n inferences (Bos & Gabsdil, 2000).

One key question concerns the nature of the answerhood relationship itself, and in what way the rules governing answerhood may be implemented. We could try to build on the view advocated by Groenendijk & Stokhof and others, that yes/no questions are really questions of “whether” something is true or false, and an answer to such a question allows you to determine that the proposition in question is true, or that it is false. We could seek to model this explicitly in terms of knowledge.

$$(47) \frac{\text{“Know whether } p\text{” } p \text{ True}}{\text{“Know that } p\text{”}} \quad \frac{\text{“Know whether } p\text{” } \neg p \text{ True}}{\text{“Know that } \neg p\text{”}}$$

Alternatively, we might seek to express this internally as some state of “knowledge” Γ .

- (48) A proposition p answers a question q ? in a context Γ if Γ and p together allow us to either infer q or infer $\neg q$ (and Γ by itself allows us to infer neither).¹³

However they are expressed, these are essentially constraints over reasoning systems involving answerhood, but in general they may not be directly expressible *within* the representation language itself, or any implementation of such a theory.

This issue is apparent even in other attempts to capture the notion of answerhood within a first-order framework. Bos & Gabsdil (2000) adopt answerhood conditions for wh-questions that are expressed in a first-order language. Essentially they translate wh-questions into a formula with domain D and body B . Putting to one side the DRT aspects of their notation, essentially wh-questions can then be given in the form $D?B$. An answer A is defined as proper for the question if at least one of the following propositions is *consistent*, and at least one is *inconsistent*:

- (49) a) $\forall x(Dx \rightarrow Bx) \wedge A$;
 b) $\exists x(Dx \wedge Bx) \wedge A$;
 c) $\exists x(Dx \wedge \neg Bx) \wedge A$;
 d) $\neg \exists x(Dx \wedge Bx) \wedge A$.

So if we had the question

¹³ This mirrors Groenendijk & Stokhof (1997a, Fact 4.3).

(50) “*Who loves John?*”

this might be represented as something like

(51) $person'(x)?loves'(j', x)$

A proper answer is one that is consistent with at least one, but not all of the following possibilities.

- (52) a) “*Everybody loves John*”;
 b) “*Somebody loves John*”;
 c) “*Somebody does not love John*”;
 d) “*Nobody loves John*”;

This characterisation loses some of the fine-grained distinctions that Groenendijk & Stokhof (1984, 1990b) make concerning answers and exhaustive answers, but reduces the number of permutations that have to be considered when determining whether *A* is a proper answer. Unfortunately it cannot avoid the fundamental problem that the property of answerhood for a given question is not necessarily tractable for arbitrary domains, and that it cannot be internalised into the representation language.

The extensional, model-theoretic interpretation of questions of Groenendijk & Stokhof (1984, 1990b, 1997a) is not universally accepted. Ginzburg & Sag (2001) argue that it is incorrect to interpret questions by their exhaustive answerhood criteria. There may be contextual effects that change what constitutes an exhaustive answer, and different questions may have the same exhaustive answers. These arguments echo those concerned with propositions and truth conditions (Section 4.2); just as it can be argued that propositions are more than their truth conditions, perhaps questions are more than their answerhood conditions.

An alternative is to treat questions as something more basic, perhaps represented by propositions with abstracted variables. To do it justice, such an account needs to be formulated in a theory that does not automatically conflate any such propositional abstracts with “mere” properties and relations. Ginzburg & Sag (2001) develop such an approach to questions and answers in the context of situation-theoretic semantics (Barwise & Perry, 1983; Barwise & Etchemendy, 1990). It should be possible to adapt the key aspects of this approach to other semantic frameworks.

Imperatives

It seems appropriate to consider some kind of theory of entailment for imperatives that can determine when one imperative “implies” another. As with propositional connectives, we may wish to consider notions of entailment between simple and complex imperatives, including conjoined imperatives, disjoined imperatives, and imperatives containing negation. Certainly it seems appealing to assume that there is a form of entailment relationship that can say something about the following pairs of examples.

- (53) a) “*Go to work!*”
 b) “*Go to work and write a paper!*”
 (54) a) “*Go to the beach or watch a film!*”
 b) “*Go to the beach!*”
 (55) a) “*Eat the apple!*”
 b) “*Don’t eat the apple!*”

The intuitions behind even these cases are not always straight-forward. For example, in the case of disjunction there are two potential readings, the so called free choice and weak readings (Kamp, 1973). Imperatives may also combine with propositions.

- (56) “*If you see John, say hello!*”

In such a case, we might want to “infer” the imperative to say hello in the event that John is seen (or possibly that the subject might want to avoid seeking John).

There are also the so-called pseudo-imperative constructions (Franke, 2006) whose formal analysis appears non-trivial, as with the following examples.

- (57) a) “*Have another drink and you will die!*”
 b) “*Have another drink and you will be happy!*”
 c) “*Have another drink or you will die!*”

A fundamental question is what counts as a relevant notion of entailment for imperatives. There are similarities with questions, in that it does not seem appropriate to assign imperatives a direct truth-conditional interpretation. Unlike interrogatives, imperatives are not so easily embedded inside other expressions. Nor is there an overtly *linguistic* counter-part to an “answer”. The question about what kinds of behaviour should be modelled by a semantic analysis of imperatives revolves around notions of what are sometimes referred to as *satisfaction* and *validity* (Ross, 1945). In the case of the former, there is a notion of an imperative being satisfied by some response. Entailments may then be expressed in terms that describe which *other* imperatives are satisfied by such a response.

As elsewhere, different frameworks have been adopted and adapted to capture appropriate patterns of behaviour associated with imperatives. Many accounts assume a possible-worlds perspective, with actions (or possible actions) that update the state of the world so that it satisfies some propositional analogue of the imperative. The question arises as to whether an imperative is satisfied by a propositional description of the desired state, or by a particular agent engaging in an appropriate action. Given the following imperative

- (58) “*Shut the window!*”

any natural utterance of this will typically be directed at an individual (or group of individuals), with the expectation that it is *that* individual who

will cause the particular desired outcome, or that the particular individual engaging in the associated activity *is* the desired outcome, so that

(59) “*John shuts the window.*”

is a propositional description of the satisfaction criteria of (58). A more elaborate view might additionally contemplate a counter-factual element to satisfaction, so that John’s shutting of the window only genuinely satisfies the imperative if John would not otherwise have shut the window.

Many accounts of imperatives (including those of Segerberg (1990), Lascarides & Asher (2004) and many others) have sought to avoid what has come to be known as *Ross’ Paradox*. This is the view expressed by Ross (1945) that a logic of imperatives is impossible to achieve because it appears impossible to discern a coherent collection of inferences that encapsulate the notions of satisfaction and validity. That is, inferences cannot both allow us to conclude (a) which other imperatives are satisfied given that the imperative in the premise has been satisfied (satisfaction) and (b) that the requirement to comply with a particular command in the premise entails that we should comply with a command in the conclusion (validity). The example often cited in favour of this view concerns disjunction introduction. Consider the following two imperatives.

- (60) a) “*Post the letter!*”
 b) “*Post the letter or burn the letter!*”

To many, the most natural inference is from (60b) to (60a) (Kamp, 1973). This corresponds to an inference concerning validity. However, if a logic of imperatives follows the usual rule of disjunction introduction, the inference should go the other way around. This can only correspond to an inference concerning satisfaction: if we have satisfied the requirement to post the letter, we would also have satisfied a requirement to post or burn the letter.

Even if both notions (satisfaction and validity) cannot be encapsulated by a single rule, that does not mean there can be no meaningful logics of satisfaction and validity. We might instead consider a logic of satisfaction independently of a logic of validity. The former might in many cases parallel inferences of indicative reasoning, whereas the latter may be more akin to a notion of *refinement* from computer science (Wirth, 1971). This latter view may also correspond to a more pragmatic analysis of imperatives which characterises the problem of “what should be done” in terms of lists of obligations that need to be fulfilled (Piwiek, 2000; Portner, 2005).

4.4 Expressiveness, Formal Power, and Computability

In computational semantics there is a tension. We want a theory that is computationally tractable but also sufficiently expressive to handle the natural language phenomena of interest. In many cases the most convenient way of obtaining expressiveness is by adopting a more powerful representation language.

Yet more formal power is typically accompanied by computational intractability. In some cases, however, it is possible to find a virtuous combination of appropriate expressiveness without an undesirable increase in formal power beyond what is computationally tractable.

The issue of computability arises in many guises. For example, theoremhood is, in general, intractable in higher-order formalisms. This is because such formalisms do not have a decidable proof-theory: the theorems of higher-order theories are not recursively enumerable. This suggests that systems with the power of first-order logic should be preferred to higher-order systems. There are other cases where a sacrifice in expressiveness may be appropriate. For example, in the case of arithmetic and quantifiers of number, we may prefer weaker more tractable theories such as Presburger arithmetic (Presburger, 1929) over Peano arithmetic. In general, these trade-offs in power may mean that certain pertinent notions are not expressible (such as the quantifier “*infinitely many*” in the case of a genuinely first-order theory, and the notion of multiplication in the case of Presburger arithmetic).

Related to this is the specific problem of *impredicativity*. This can arise with type quantification—as used in (41) of Section 4.1. If we allow such type quantification to range over all types, including polymorphic types, then the evaluation of polymorphic types can be deeply problematic in a computational system (the evaluation of such a type requires us to quantify over the very type that we are attempting to evaluate). Fortunately it appears that natural language does not require such a powerful typing system; we can compromise by having the expressivity of polymorphic types, but restricted so that there is no problematic quantification over polymorphic types themselves. This is a case where it is possible to have a more *expressive* theory without increasing the *power* of the system beyond what is computationally tractable.

There are other notions that cannot be formalised within any computable theory besides impredicative types, such as the notion of truth, and answerhood conditions (Section 4.3). We may define constraints on how truth and answerhood should behave, but that does not mean the notions themselves are intrinsically amenable to definition *within* a tractable theory. Here we might begin to see how notions relevant to computational semantics might not be directly expressed *by* an implementation, but they may be properties *of* such an implementation.

This suggests an alternative characterisation of computational semantics, where the idea of computability itself is considered as a *constraint* on appropriate formalisations and models (Turner, 2007). As a methodological constraint, this may be relevant even in the event that the behaviours being described by a formal theory are not directly relevant to any conceivable practical implementation. Rather, the claim might be made that a computable theory potentially has more explanatory power than a theory expressed in an intrinsically intractable framework.

Indeed, we can contemplate using the constraint of computability not just in the context of the formal representations of meaning, but also in the process

of translating natural language into those representations. It is conventional only to require that the translation is compositional (Section 2.1). Unfortunately,

it turns out that if there are no restrictions on the nature of the functions used to combine the meaning of the parts, then compositionality does not impose any effective restriction on the nature of the interpretation (Zadrozny, 1994). In effect, compositionality is a constraint only on the form of the translation rules, and their coverage, not their function. That is, as usually defined, compositionality is a restriction on *the way in which a translation is produced*, rather than necessarily being a restriction on *the end result of that translation*.

If we ignore the evaluation of the functions that are applied in a compositional translation, then the constraint of compositionality ensures that the translation process is recursive on the structure of the expression. For every syntactic constituency rule there should be a corresponding rule for determining the semantic representation to be associated with the head of the expression as a function of the semantic representation of its constituent parts. This guarantees that every syntactic analysis has a corresponding semantic interpretation.

Of course, the evaluation of the functions used in a compositional interpretation are important. If the functions themselves are meta-theoretic, and not part of the semantic theory as such, then they need to be applied to produce a well-formed representation. Even if they are part of the semantic theory, we may need to apply the functions in order to derive a representation in some “normal form”. In either case, it would be appropriate to consider constraints on the nature of the functions themselves. At the very least, we would expect them to be computable.

5 Corpus-based and Machine Learning Methods

Although the focus of this chapter, and indeed much work in computational semantics, has largely been on the application of techniques for computationally tractable semantic analysis based upon representations in formal logic, there are other computational approaches that involve less traditional forms of semantic analysis which do not rely upon strictly logical theories of meaning. These include approaches that exploit corpus-based techniques and machine learning. We will briefly survey a small sample of these techniques and their applications, and speculate on the role that a more formal analysis may play in their application, particularly in the case of textual entailment.

5.1 Latent Semantic Analysis

Latent Semantic Analysis (LSA) (Landauer *et al.*, 2007, 1998) is a technique that aims to determine a “conceptual” or “semantic” space for words and the documents in which they occur. The number of “concepts” used is invariably smaller than the number of different words in the documents. The idea is that words denoting similar concepts will be mapped on to each other in this reduced space. The technique is able to determine when word-meanings—and documents—are related, even when the words never occur in the same context and the documents have few words in common. Two words may be deemed to be conceptually related because the words that they appear with occur together in other documents. This allows us to compare and process words and documents in concept space.

The technique takes as input a word-document matrix where each entry indicates the number of times a given word appears in a given document. It then uses *Singular Value Decomposition* (SVD) (Golub & van Loan, 1989) effectively to “rotate” the word-document space to a different set of dimensions. These dimensions (the “latent space”) are such that they give the axes of greatest variation for the original word-document matrix. Dimension reduction can then be applied by pruning those dimensions with the smallest contribution. The dimensions that are left are considered to correspond to some notion of a “concept”. In the matrix of reduced dimensionality, words which make a similar contribution are effectively merged together. The intuitive argument is that different words will have similar vector representations in this reduced space if they denote a similar concept.

This technique has a number of applications (Landauer *et al.*, 2007, 1998) including document indexing and search (*Latent Semantic Indexing* (LSI)) (Deerwester *et al.*, 1990) and automatic essay marking (Thomas K. Landauer & Laham, 1998). It can also be used to cluster documents according to their conceptual similarity. In the case of LSI, the terms occurring in a query expression can also be mapped to the corresponding concepts, which are then used to retrieve the documents in which those concepts occur. This allows docu-

ments to be retrieved that do not necessarily contain the terms in question, but which do include terms that correspond to the same “concepts”.

Terms that are combined by the dimension reduction into a single concept may be indicative of an underlying synonym, although the notion of a concept here is a mathematical abstraction that need not correspond to any natural category.

It has been argued that the dimension reduction employed by LSA has certain problems including the fact that the reduced matrix can contain negative values, which is counter-intuitive if the values are interpreted as counts of concept occurrences. (Quesada, 2003; Hofmann, 2001). An alternative approach is *Probabilistic Latent Semantic Analysis* (PLSA) which employs a dimension reduction strategy that is claimed to have a more solid statistical foundation. In this technique, the number of concepts is decided, and words are fitted to those concepts using *Expectation Maximisation* (Hofmann, 1999, 2001).

5.2 Extraction of Semantic Roles

Identification of semantic roles is useful for a range of problems such as question answering (Narayanan & Harabagiu, 2004; Sun *et al.*, 2005; Kaisser, 2006; Shen & Lapata, 2007), dialogue systems (Liu, 1995), and information extraction (Riloff, 1993).

The notion of semantic role is connected with the notions of subcategorisation and selection preferences, which may determine the syntactic function and “thematic role” of an entity. (In some cases, coercion by way of metaphor or some other semantic relation may be needed to obtain a natural interpretation.) The syntactic role of a verb’s complement can give an indication of the semantic role of nominal expressions, such as *agent*, *patient*, *theme* etc. (Fillmore, 1968; Dowty, 1991). More specific roles may also be defined, as in Frame semantics (Fillmore, 1976), and FrameNet languages (Baker *et al.*, 1998). Resources such as PropBank (Palmer *et al.*, 2005), provide a hand-corrected body of predicate-argument annotations of the Penn Treebank.

There are machine learning methods (both supervised and unsupervised) for automatically determining semantic roles. Such methods can be used to learn to label constituents of a sentence with the semantic roles of a target frame (Gildea & Jurafsky, 2002). One problem is that the correspondence between syntactic categories and semantic roles is not always direct or easy to predict. Machine learning techniques that have been applied to this problem include maximum entropy, rule-based, memory-based, and kernel methods (see CHAPTERS ???, ???, ???).

Add specific references to chapters (and other works?)

5.3 Word Sense Disambiguation

Word sense disambiguation (Ide & Véronis, 1998) is a useful step when dealing with various essentially semantic issues, such as question answering and

intelligent document retrieval. The objective is to be able to distinguish between various senses of a word. Machine learning techniques can be used in a variety of ways to achieve this. One common feature is to identify word senses from the different contexts in which a given word is used. For many tasks, a fine discrimination between senses might not be required (Ide & Wilks, 2006).

Knowledge-based approaches may use dictionaries and thesauri to provide examples of different word senses, and the other words associated with a given sense (Lesk, 1986), as well as ontological relationships (Roberto Navigli, 2005). In general such approaches may be limited by the quality and relevance of the information sources used.

Data-driven approaches seek to determine different senses of a word by identifying patterns, or clusters, of co-occurrences and contexts, both local and global (McCarthy *et al.*, 2004). They may involve supervised or unsupervised learning. In the former case, sense-tagged corpora may be used to train a sense disambiguation algorithm. In the latter case, clustering techniques may be used to identify different collocation contexts, which are assumed to correspond to different word senses. Various assumptions may be made to aid training. One such assumption is that generally speaking a word appearing more than once in a given document is likely to share the same word sense (Gale *et al.*, 1992a).

Bilingual corpora may also be used to help identify the different senses of a word by identifying systematic differences in translation (Gale *et al.*, 1992b; Kaji & Morimoto, 2005).

Sense disambiguation is discussed in CHAPTER ANNOTATION (Palmer and Xue).

Add Chapter cross-reference.

5.4 Textual Entailment

One of the purposes of a computationally feasible formal semantic analysis of language is to determine what is entailed by a given text. This is called *textual entailment*. It can be thought of as capturing relationships of the form $t \Rightarrow h$, where t is some natural language text, and h is some hypothesis, also expressed in natural language. The usual notion of entailment is one where h would not follow without t ; that is, h cannot be obtained from any of the background information that is being used to capture entailment relations. Textual entailment can be applied to the problems of information extraction, question answering, translation, summarisation and other NLP tasks (Glickman *et al.*, 2005). In some cases, it is possible to capture a notion of textual entailment using statistical and probabilistic techniques, rather than a purely logic-based analysis of meaning. Indeed, the term “textual entailment” is often used in a context that does not presuppose a rigorous, logic-based analysis of meaning.

The mechanisms for obtaining appropriate entailment patterns include hand-coded rules, acquired knowledge and machine learning (see CHAPTER, XXX, XXX, XXX). For example, a range of machine learning techniques can

Add Chapter references.

be applied to find approximations to human judgements concerning entailment patterns, or rules. Another possibility is to exploit patterns of words that indicate some intended inference or relation. For example, hyponyms and meronyms may be identified by discovering ontological relationships from corpora that are indicated by particular patterns (Hearst, 1992; Berland & Charniak, 1999). Various other semantic relationships may also be discovered, including causal relations (Girju, 2003; Cole *et al.*, 2006) and temporal ordering and other relationships between verb meanings (Chklovski & Pantel, 2004). This may not be entirely robust. There may be problems to overcome with patterns that are over-general, negative polarity contexts, and anaphoric expressions (see Sanchez-Graillet & Poesio (2007); Sanchez-Graillet *et al.* (2006) for example).

The method may be made more robust if there is a notion of the semantic class of a word (Girju *et al.*, 2006), although this requires additional work in identifying the relevant semantic classes.

Such methods may help to identify particular kinds of entailments. There are however other more general corpus-based approaches to inference, some of which rely upon a traditional formal semantic analysis, where a semantic analysis of the documents in question is produced, along with the hypothesis that is to be checked. In general this requires a broad coverage deep syntactic analysis, comprehensive semantic analysis and a robust theorem prover. For some problem domains, such as question-answering, it is possible that techniques based on pattern-matching of the semantic representations may be adequate (Ahn *et al.*, 2005). Additional sources of information may have to be analysed to determine relevant relationships between information in a given document (or document collection) and a hypothesis that is being tested, or a question that is being asked. In addition to finding evidence of such relationships from supplementary sources of information, there have been proposals to improve the robustness of theorem proving by allowing costed abductive assumptions (Raina *et al.*, 2005) which allow some degree of flexibility in unifying the terms that appear in a proof. Cost functions can be used to minimise the contribution of abductive reasoning that is permitted within a proof to avoid perverse results.

It is sometimes argued that contemporary formal techniques (which have been the focus of this chapter) are too fragile and incomplete to be used for such applications. Alternative approaches seek to represent knowledge, and capture textual entailments, using shallower, less abstract representations of the text. Such methods include hierarchical representations based upon description logics (de Salvo Braz *et al.*, 2005). These seek to capture structural, relational and other semantic properties. Other approaches use representations that are closer to the surface form of language, including lexically-based parse-tree representations (Dagan *et al.*, 2008), perhaps augmented with annotations (for negation and modality, for example). A relevant work on this topic is Dagan *et al.* (2009). A comprehensive analysis of textual entailment almost certainly needs to address questions of resolving anaphora. But corpus-

based methods have also been applied to this problem (Niyu Ge & Charniak, 1998; Paul *et al.*, 1999; Poesio & Alexandrov-Kabadjov, 2004).

5.5 Relationship to Formal Semantics

We may wonder about the nature of the relationship between corpus based methods and formal, logic-based approaches. The relevance of this issue is perhaps most obvious in the case of textual entailment, which aims to address one of the objectives of formal methods—that of capturing legitimate entailment relationships.

One question is whether it is realistic to assume that formal approaches will ever be able to model the full range of textual entailments, or whether entailments captured by corpus-based methods will ever be as trustworthy as logic-based inference. We offer no view on this matter here, but observe that some aspects of textual entailment may need to be informed by something resembling a formal analysis for us to know what counts as a legitimate or illegitimate entailment, and why—even if only to ensure an element of consistency, and confidence, in the conclusions drawn. Regardless of the underlying mechanism used for capturing textual entailment relationships, it also seems appropriate to formalise normative rules concerning how a coherent notion of textual entailment should behave; that is, we should consider formulating a logic of textual entailment to characterise the properties that the relationship $t \Rightarrow h$ should support.

Another topic that may merit further exploration is a better understanding of the relationships, if any, between a logic-based conception of semantics, and the notion of semantics as used in work that builds on word and phrase co-occurrence data and its generalisations, such as LSA. At the time of writing, it appears there have been few if any attempts to reconcile these different views on the nature of natural language semantics.

6 Concluding Remarks

This chapter has presented some of the basic ideas behind computational semantics, with some sample topics and research questions. Some corpus-based techniques that embody a notion of semantics have been sketched, but the primary focus has been on logic-based approaches. One idea that arises in the presentation is not merely to think of computational semantics as describing theories of semantics that lend themselves to implementation, but to consider computability itself as a constraint on theories of meaning and semantic analysis. We can also distinguish between those aspects of a theory of meaning that lend themselves to direct implementation, and those that describe the properties of an implementation, without themselves necessarily being implementable.

References

- Ahn, Kisuh, Johan Bos, James R. Curran, Dave Kor, Malvina Nissim, & Bonnie Webber (2005), Question answering with QED at TREC-2005, in Voorhees & Buckland (eds.), *The Fourteenth Text REtrieval Conference, TREC 2005*.
- Ahn, René M. C. & Hans-Peter Kolb (1990), Discourse representation meets constructive mathematics, in L. Kálmán & L. Pólos (eds.), *Papers from the Second Symposium on Logic and Language*, Akadémiai Kiadó, Budapest, (105–124).
- Andy Chiu, Pascal Poupart & Chrysanne DiMarco (2007), Generating lexical analogies using dependency relations, in *Proceedings of the International Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Prague, Czech Republic.
- Asher, Nicholas (1993), *Reference to Abstract Objects in English: a Philosophical Semantics for Natural Language Metaphysics*, Studies in Linguistics and Philosophy, Kluwer, Dordrecht.
- Asher, Nicholas & Alex Lascarides (2003), *Logics of Conversation*, Cambridge University Press, Cambridge.
- Baker, Collin F., Charles J. Fillmore, & John B. Lowe (1998), The Berkeley Framenet project, in *Proceedings of the COLING-ACL*, Montreal, Canada.
- Barwise, Jon & John Etchemendy (1990), Information, inferences, and inference, in Robin Cooper, Kuniaki Mukai, & John Perry (eds.), *Situation Theory and Its Applications, I*, CSLI Publications, Stanford, CSLI Lecture Notes Number 22, (33–78).
- Barwise, Jon & John Perry (1983), *Situations and Attitudes*, MIT Press (Bradford Books), Cambridge, MA.
- Bealer, G. (1982), *Quality and Concept*, Clarendon Press, Oxford.
- Beaver, David I. (2002), Presupposition projection in drt: a critical assessment, in David Beaver, Stefan Kaufmann, Brady Clark, & Luis Casillas (eds.), *Stanford Papers on Semantics*, CSLI Publications.
- Belnap, Nuel D. (1982), Questions and answers in Montague grammar, in S. Peters & E. Saarinen (eds.), *Processes, Beliefs, and Questions*, D. Reidel, Dordrecht.
- Berland, Matthew & Eugene Charniak (1999), Finding parts in very large corpora, in *ACL*.
- Blackburn, Patrick & Johan Bos (1999), Working with discourse representation theory: An advanced course in computational semantics, draft Book.
- Blackburn, Patrick & Johan Bos (2005), *Representation and Inference for Natural Language*, CSLI Publications, Stanford, ISBN 1575864967 (Paperback) 1575864959 (Cloth).
- Bos, Johan (1995), Predicate logic unplugged, in *Proceedings of the Tenth Amsterdam Colloquium*, Amsterdam, Holland.
- Bos, Johan & Malte Gabsdil (2000), First-order inference and the interpretation of questions and answers, in M. Poesio & D. Traum (eds.), *Fourth Workshop on the Semantics and Pragmatics of Dialogue Proceedings of Goetalog 2000. Fourth Workshop on the Semantics and Pragmatics of Dialogue*, volume 00-5 of *Gothenburg Papers in Computational Linguistics*, (43–50).
- Bos, Johan & Tetsush Oka (2002), An inference-based approach to dialogue system design, in *Proceedings of the Nineteenth COLING*, (113–119).
- Carnap, Rudolf (1947), *Meaning and Necessity*, University of Chicago Press, Chicago.

- Chierchia, Gennaro (1982), Nominalisation and Montague grammar: a semantics without types for natural languages, *Linguistics and Philosophy* 5:303–354.
- Chklovski, Timothy & Patrick Pantel (2004), Verbocean: Mining the web for fine-grained semantic verb relations, in *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*, Barcelona, Spain.
- Church, Alonzo (1940), A formulation of the simple theory of types, *Journal of Symbolic Logic* 5:56–68.
- Cocchiarella, Nino (1985), Two lambda-extensions of the theory of homogeneous simple types as a second order logic, *Notre Dame Journal of Formal Logic* 26:377–407.
- Cole, Stephen V., Matthew D. Royal, Marco G. Valtorta, Michael N. Huhns, & John B. Bowles (2006), A lightweight tool for automatically extracting causal relationships from text, in *SoutheastCon, 2006. Proceedings of the IEEE*, (125–129), doi:10.1109/second.2006.1629336.
- Cooper, Robin (1983), *Quantification and Syntactic Theory*, Synthese Language Library, D. Reidel, Dordrecht.
- Cooper, Robin (2005), Records and record types in semantic theory, *Journal of Logic and Computation* 15(2), ISSN 0955-792X, doi:10.1093/logcom/exi004.
- Cooper, Robin & Jonathan Ginzburg (2002), Using dependent record types in clarification ellipsis, in *Proceedings of the 6th Workshop on the Semantics and Pragmatics of Dialogue (EDIALOG)*, (45–52).
- Copestake, Ann, Dan Flickinger, & Ivan A. Sag (2006), Minimal recursion semantics, *Research on Language and Computation* 3:281–332.
- Covington, Michael A. (1994), *Natural Language Processing for Prolog Programmers*, Prentice-Hall.
- Crouch, Richard (2005), Packed rewriting for mapping semantics to KR, in *Proceedings of the Sixth International Workshop on Computational Semantics*, Tilburg, (103–114).
- Crouch, Richard & Josef van Genabith (1999), Context change, underspecification, and structure of glue language derivations, in M. Dalrymple (ed.), *Semantics and Syntax in Lexical Functional Grammar*, MIT, Cambridge, MA, (117–189).
- Curry, Haskell B. & Robert Feys (1958), *Combinatory Logic*, volume 1 of *Studies in Logic*, North Holland.
- Dagan, Ido, Roy Bar-Haim, Idan Szpektor, Iddo Greental, & Eyal Shnarch (2008), Natural language as the basis for meaning representation and inference, in *CI-Ling*, (151–170).
- Dagan, Ido, Dan Roth, Fabio Zanzotto, & Graeme Hirst (editor) (2009), *Recognizing Textual Entailment*, Synthesis Lectures on Human Language Technologies, Morgan and Claypool.
- Dalrymple, Mary, John Lamping, Fernando Pereira, & Vijay Saraswat (1999), Quantification, anaphora, and intensionality, in M. Dalrymple (ed.), *Semantics and Syntax in Lexical Functional Grammar*, MIT, Cambridge, MA, (39–89).
- Davis, Wayne A. (1998), *Implicature: Intention, Convention, and Principle in the Failure of Gricean Theory*, Cambridge University Press, Cambridge.
- Deerwester, Scott, Susan Dumais, Thomas K. Landauer, George W. Furnas, & Richard A. Harshma (1990), Indexing by latent semantic analysis, *Journal of the American Society for Information Science* 41(6):391–407.
- Dowty, David (1991), Thematic proto-roles and argument selection, *Language* 67(3):547–619.

- Ebert, Christian (2005), *Formal Investigation of Underspecified Representations*, Ph.D. thesis, Department of Computer Science, King's College London, unpublished.
- van Eijck, Jan & Christina Unger (2004), *Computational Semantics with Functional Programming*, unpublished ms., CWI, Amsterdam.
- Eva Hajičová, Petr Sgall Eric Sgall, Barbara Hall Partee (1998), *Topic-focus articulation, tripartite structures, and semantic content*, Springer, ISBN 0792352890, 9780792352891.
- Fillmore, Charles J. (1968), The case for case, in Emmon W. Bach & Robert T. Harms (eds.), *Universals in Linguistic Theory*, Holt, Rinehart & Winston, New York, (1–88).
- Fillmore, Charles J. (1976), Frame semantics and the nature of language, *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech* 280:20–32.
- Fox, Chris (2000), *The Ontology of Language*, CSLI, Stanford.
- Fox, Chris & Shalom Lappin (2005), *Formal Foundations of Intensional Semantics*, Blackwell, Oxford.
- Fox, Chris & Shalom Lappin (2007), Expressive completeness and computational efficiency for underspecified representations, in Lars Borin & Staffan Larsson (eds.), *Festschrift for Robin Cooper*, celebrating the occasion of Robin Cooper's 60th birthday.
- Fox, Chris, Shalom Lappin, & Carl Pollard (2002), A higher-order, fine-grained logic for intensional semantics, in G. Alberti, K. Balough, & P. Dekker (eds.), *Proceedings of the Seventh Symposium for Logic and Language*, Pecs, Hungary, (37–46).
- Francez, Nissim & Roy Dyckhoff (2007), Proof-theoretic semantics for natural language, in *Tenth meeting of the Association for Mathematical Language (MOL)*, Los Angeles, to appear in a journal special issue.
- Franke, Michael (2006), Pseudo-imperatives: A case study in the ascription of discourse relations, 28th Annual Meeting DGfS, Bielefeld.
- Gale, William A., Kenneth W. Church, & David Yarowsky (1992a), One sense per discourse, in *In Proceedings of the DARPA Speech and Natural Language Workshop*.
- Gale, William A., Kenneth W. Church, & David Yarowsky (1992b), Using bilingual materials to develop word sense disambiguation methods, in *Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, Montreal, (101– 112).
- Gallin, Daniel (1975), *Intensional and Higher-Order Modal Logic*, North-Holland, Amsterdam.
- Geach, Peter T. (1972), A program for syntax, in Donald Davidson & Gilbert Harman (eds.), *Semantics of Natural Language*, D. Reidel, Dordrecht, (483–497).
- Gentzen, Gerhard (1969), *The Collected Papers of Gerhard Gentzen*, North Holland.
- Gildea, Daniel & Daniel Jurafsky (2002), Automatic labeling of semantic roles, *Computational Linguistics* 28(3):245–288.
- Gilmore, Paul Carl (2001), An intensional type theory: Motivation and cut-elimination, *Journal of Symbolic Logic* 66:383–400.
- Ginzburg, Jonathan & Ivan A. Sag (2001), *Interrogative Investigations: The Form, Meaning and Use of English Interrogatives*, CSLI Publications, ISBN 1575862786 (Paperback), 1575862778 (Cloth).

- Girard, Jean-Yves (1987), Linear logic, *Theoretical Computer Science* 50(1):1–102.
- Girju, Roxana (2003), Automatic detection of causal relations for question answering, in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003)*, workshop on "Multilingual Summarization and Question Answering - Machine Learning and Beyond".
- Girju, Roxana, Adriana Badulescu, & Dan Moldovan (2006), Automatic discovery of part-whole relations, *Computational Linguistics* 32(1):83–136.
- Glickman, Oren, Ido Dagan, & Moshe Koppel (2005), A probabilistic classification approach for lexical textual entailment, in *Twentieth National Conference on Artificial Intelligence (AAAI-05)*.
- Golub, Gene H. & Charles F. van Loan (1989), *Matrix Computations*, The Johns Hopkins University Press, Baltimore.
- Grice, H. Paul (1975), Logic and conversation, in P. Cole & J. Morgan (eds.), *Syntax and Semantics*, Academic Press, New York, volume 3: Speech Acts, reprinted in *Studies in the Way of Words*, ed. H. P. Grice, pp. 22–40. Cambridge, MA: Harvard University Press (1989).
- Grice, H. Paul (1981), Presuppositions and conversational implicatures, in P. Cole (ed.), *Radical Pragmatics*, Academic Press, New York, (183–98), reprinted in *Studies in the Ways of Words*, ed. H. P. Grice, pp. 269–282. Cambridge, MA: Harvard University Press (1989).
- Groenendijk, J. & M. Stokhof (1990a), Dynamic Montague grammar, in L. Kálmán & L. Polós (eds.), *Papers from the Second Symposium on Logic and Language*, Akadémiai Kiadó, Budapest, Hungary, (3–48), also available as ITLI Prepublication Series LP-90-02.
- Groenendijk, J. & M. Stokhof (1991), Dynamic predicate logic, *Linguistics and Philosophy* 14(1):39–100.
- Groenendijk, Jeroen & Martin Stokhof (1984), On the semantics of questions and the pragmatics of answers, in F. Landman & F. Veltman (eds.), *Varieties of Formal Semantics*, Foris, Dordrecht, (143–170).
- Groenendijk, Jeroen & Martin Stokhof (1990b), Partitioning logical space, Annotated handout, Second European Summer School on Logic, Language and Information, Leuven.
- Groenendijk, Jeroen & Martin Stokhof (1997a), Questions, in Johan van Benthem & Alice ter Meulen (eds.), *Handbook of Logic and Language*, Elsevier/MIT Press, Amsterdam/Cambridge, Mass, (1055–1124).
- Groenendijk, Jeroen & Martin Stokhof (1997b), Questions, in Johan van Benthem & Alice ter Meulen (eds.), *Handbook of Logic and Linguistics*, North Holland, Amsterdam, (1055–1124).
- Hearst, Marti A. (1992), Automatic acquisition of hyponyms from large text corpora., in *Proceedings of the Fourteenth International Conference on Computational Linguistics*, Nantes, France, (539–545).
- Hearst, Marti A. (1997), Texttiling: Segmenting text into multi-paragraph subtopic passages, *Computational Linguistics* 23(1):33–64.
- Heim, Irene Roswitha (1990), E-type pronouns and donkey anaphora,, *Linguistics and Philosophy* 13:137–177.
- Hofmann, Thomas (1999), Probabilistic latent semantic analysis, in *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, Stockholm.
- Hofmann, Thomas (2001), Unsupervised learning by probabilistic latent semantic analysis, *Machine Learning* 42(1–2):177–196.

- Ide, Nancy & Jean Véronis (1998), Word sense disambiguation: The state of the art, *Computational Linguistics* 24(1):1–41.
- Ide, Nancy & Yorick Wilks (2006), Making sense about sense, in Eneko Agirre & Philip Edmonds (eds.), *Word Sense Disambiguation: Algorithms and Applications*, chapter 3.
- Janssen, Theo M. V. (2001), Frege, contextuality and compositionality, *Journal of Logic, Language and Information* 10(1):115–136.
- Johnson, Mark (1988), *Attribute-Value Logic and the Theory of Grammar*, CSLI Publications, ISBN ISBN-10: 0-937073-36-9 (Paper); 0-937073-37-7 (Cloth); ISBN-13: 978-0-937073-36-0 (Paper); 978-0-937073-37-7 (Cloth).
- Jubien, Michael (1989), On properties and property theory, in G. Chierchia, B. Partee, & R. Turner (eds.), *Properties, Types and Meaning*, Kluwer, Dordrecht, volume 1, (159–175).
- Kadmon, Nirit (1990), Uniqueness, *Linguistics and Philosophy* 13:237–324.
- Kadmon, Nirit (2001), *Formal Pragmatics: Semantics, Pragmatics, Presupposition, and Focus*, Wiley-Blackwell, ISBN 978-0-631-20121-2, 978-0-631-20120-5.
- Kaisser, Michael (2006), Web question answering by exploiting wide-coverage lexical resources, in *Proceedings of the 11th ESSLLI Student Session*, (203–213).
- Kaji, Hiroyuki & Yasutsugu Morimoto (2005), Unsupervised word sense disambiguation using bilingual comparable corpora, *IEICE Transactions on Information and Systems* E88-D:289–301, doi:10.1093/ietisy/E88-D.2.289.
- Kamp, Hans (1973), Free choice permission, *Proceedings of the Aristotelian Society* 74:57–74.
- Kamp, Hans (1979), Semantics versus pragmatics, in Franz Guenther & Siegfried J. Schmidt (eds.), *Formal Semantics and Pragmatics for Natural Language*, D. Reidel, Synthese Language Library, (255–287).
- Kamp, Hans (2007), Henry sweet lecture, Talk at the LAGB.
- Keenan, Edward (1992), Beyond the fregean boundary, *Linguistics and Philosophy* 15:199–221.
- Keller, William R. (1988), Nested cooper storage: The proper treatment of quantification in ordinary noun phrases, in U. Reyle & C. Rohrer (eds.), *Natural Language Parsing and Linguistic Theories*, Reidel, Dordrecht.
- Klein, Ewan (2006), Computational semantics in the *Natural Language Toolkit*, in *Proceedings of the 2006 Australasian Language Technology Workshop (ALTW2006)*, (26–33).
- Koller, Alexander, Joachim Niehren, & Stefan Thater (2003), Bridging the gap between underspecified formalisms: Hole semantics as dominance constraints, in *Proceedings of 11th EACL*, Budapest.
- Kowalski, Robert & Marek Jozef Sergot (1986), A logic-based calculus of events new generation computing 4:67–95.
- Krahmer, Emiel & Paul Piwek (1999), Presupposition projection as proof construction, in H. Bunt & R. Muskens (eds.), *Computing Meanings: Current issues in Computational Semantics*, Kluwer Academic Publisher, Dordrecht, Studies in Linguistics and Philosophy Series, an earlier version appeared in the Proceedings of IWCS II, Tilburg, 1997.
- Kripke, Saul (1963), Semantical considerations on modal logic, *Acta Philosophica Fennica* 16:83–89.
- Landauer, Thomas, Peter W. Foltz, & Darrell Laham (1998), Introduction to latent semantic analysis, *Discourse Processes* 25:259–284.

- Landauer, Thomas K., Danielle S. McNamara, Simon Dennis, & Walter Kintsch (2007), *Handbook of latent semantic analysis*, Routledge, ISBN 0805854185, 9780805854183, 532 pages.
- Lappin, Shalom (1989), Donkey pronouns unbound, *Theoretical Linguistics* 15:263–286.
- Lappin, Shalom & Nissim Francez (1994), E-type pronouns, I-sums, and donkey anaphora, *Linguistics and Philosophy* 17:391–428.
- Lascarides, Alex & Nicholas Asher (2004), Imperatives in dialogue, in Peter Kühnlein, Hans Rieser, & Henk Zeevat (eds.), *The Semantics and Pragmatics of Dialogue for the New Millennium*, Benjamins, (1–24).
- Lesk, Michael (1986), Automatic sense disambiguation using machine-readable dictionaries: How to tell a pine cone from an ice cream cone, in *Proceedings of the SIGDOC Conference*, Toronto, Canada.
- Liu, Bing (1995), Intelligent air travel and tourist information systems, in *Proceedings of the 8th international conference on Industrial and engineering applications of artificial intelligence and expert systems*, Melbourne, Australia, ISBN 2-88449-198-8, (603–610).
- Materna, Pavel, Marie Duzi, & Bjørn Jespersen (To appear), *Procedural Semantics for Hyperintensional Logic (Foundations and Applications of TIL)*, Springer.
- McCarthy, Diana, Rob Koeling, Julie Weeds, & John Carroll (2004), Finding predominant senses in untagged text, in *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*.
- Miller, Rob S. & Murray P. Shanahan (1999), The event-calculus in classical logic — alternative axiomatizations, *Electronic Transactions on Artificial Intelligence* 3(1):77–105.
- Montague, Richard (1973), The proper treatment of quantification in ordinary English, in K. J. J. Hintikka, J. M. E. Moravcsik, & P. Suppes (eds.), *Approaches to Natural Language*, D. Reidel, Dordrecht, synthese Library.
- Montague, Richard (1974), The proper treatment of quantification in ordinary english, in Richmond Thomason (ed.), *Formal Philosophy*, Yale UP, New Haven.
- Monz, Christof & Maarten de Rijke (2001), Deductions with meaning, in *Logical Aspects of Computational Linguistics*, volume 2014/2001 of *Lecture Notes in Computer Science*, ISSN 978-3-540-42251-8, (1–10), doi:10.1007/3-540-45738-0_1.
- Moore, Robert C. (1989), Unification-based semantic interpretation, in *Proceedings of the 27th annual meeting on Association for Computational Linguistics*, Association for Computational Linguistics, Morristown, NJ, USA, (33–41), doi: <http://dx.doi.org/10.3115/981623.981628>.
- Narayanan, Srini & Sanda Harabagiu (2004), Question answering based on semantic structures, in *International Conference on Computational Linguistics (COLING 2004)*, Geneva, Switzerland.
- Niyu Ge, J. Hale & E. Charniak (1998), A statistical approach to anaphora resolution, in *Proceedings of the Sixth Workshop on Very Large Corpora, COLING-ACL '98*, Montreal, Canada, (161–170).
- Palmer, Martha, Paul Kingsbury, & Dan Gildea (2005), The proposition bank: An annotated corpus of semantic roles, *Computational Linguistics* 31(1):71–106, doi: 10.1162/0891201053630264.

- Partee, Barbara & Mats Rooth (1983), Generalised conjunction and type ambiguity, in R. Bäuerle, C. Schwartz, & A. von Stechow (eds.), *Meaning, Use and Interpretation of Language*, Walter de Gruyter, Berlin, (361–383).
- Paul, Michael, Kazuhide Yamamoto, & Eiichiro Sumita (1999), Corpus-based anaphora resolution towards antecedent preference, in *In Proceedings of 1999 ACL Workshop on Coreference and Its Applications*, (47–52).
- Pelletier, Francis Jeffrey & Lenhart K. Schubert (1989), Generically speaking, in G. Chierchia, B. H. Partee, & R. Turner (eds.), *Properties, Types, and Meaning*, Kluwer, Dordrecht, volume 2.
- Pereira, Fernando C. N. & Stuart M. Shieber (1987), *Prolog and Natural-Language Analysis*, volume 10 of *CSLI Lecture Notes Series*, Center for the Study of Language and Information, Italian translation: *Prolog e Analisi del Linguaggio Naturale*, Tecniche Nuove, Milan, 1992.
- Piwek, Paul (2000), Imperatives, commitment and action: Towards a constraint-based model, *LDV Forum: GLDV-Journal for Computational Linguistics and Language Technology, Special Issue on Communicating Agents* 17(1,2), iSSN 0175–1336.
- Poesio, Massimo & Mijail Alexandrov-Kabadjov (2004), A general-purpose, off-the-shelf system for anaphora resolution, in *Proceedings of LREC*, Lisbon.
- Poesio, Massimo & Uwe Reyle (2001), Underspecification in anaphoric reference, in *Proceedings of the Fourth International Workshop on Computational Semantics*, Tilburg.
- Pollard, Carl (2004), Higher-order categorial grammar, in M. Moortgat (ed.), *Proceedings of the International Conference on Categorial Grammars (CG2004)*, Montpellier, France.
- Pollard, Carl (to appear), Hyperintensions, *Journal of Logic and Computation*.
- Pollard, Carl & Ivan Sag (1994), *Head Driven Phrase Structure Grammar*, University of Chicago Press and CSLI, Chicago.
- Portner, Paul (2005), The semantics of imperatives within a theory of clause types, in Kazuha Watanabe & Robert B. Young (eds.), *Proceedings of Semantics and Linguistic Theory 14*, CLC Publications, Ithaca, NY, paper presented at SALT 14, 14th–16th March 2004.
- Presburger, Mojzesz (1929), Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt, in *Comptes Rendus du I congrès de Mathématiciens des Pays Slaves, Warszawa*, (92–101).
- Pustejovsky, James (1995), *The Generative Lexicon*, MIT Press.
- Quesada, Jose (2003), *Latent problem solving analysis (LPSA): a computational theory of representation in complex, dynamic problem solving tasks*, Ph.D. thesis, Psychology, University of Granada.
- Raina, Rajat, Andrew Y. Ng, & Christopher Manning (2005), Robust textual inference via learning and abductive reasoning, in *Twentieth National Conference on Artificial Intelligence (AAAI-05)*.
- Ramsay, Allan (1995), Theorem proving for intensional logic, *Journal of Automated Reasoning* 14:237–255.
- Ranta, Aarne (1994), *Type Theoretic Grammar*, Oxford University Press, Oxford.
- Reyle, Uwe (1993), Dealing with ambiguities by underspecification: Construction, representation and deduction, *Journal of Semantics* 10:123–179.

- Riloff, Ellen (1993), Automatically constructing a dictionary for information extraction tasks, in *National Conference on Artificial Intelligence*, (811–816).
- Roberto Navigli, Paola Velardi (2005), Structural semantic interconnections: a knowledge-based approach to word sense disambiguation, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 27(7):1–12.
- Rooth, Mats (1993), A theory of focus interpretation, *Natural Language Semantics* 1:75–116.
- Ross, Alf (1945), Imperatives and logic, *Philosophy of Science* 11:30–46.
- de Salvo Braz, Rodrigo, Roxana Girju, Vasin Punyakanok, Dan Roth, & Mark Sammons (2005), Knowledge representation for semantic entailment and question-answering, in Farah Benamara, Marie-Francine Moens, & Patrick Saint-Dizier (eds.), *IJCAI-05 Workshop on Knowledge and Reasoning for Answering Questions*, Edinburgh, (71–80).
- Sanchez-Graillet, Olivia & Massimo Poesio (2007), Negation of protein-protein interactions, *Bioinformatics* 23(13):424–432.
- Sanchez-Graillet, Olivia, Massimo Poesio, Mijail A. Kabadjov, & Roman Tesar (2006), What kind of problems do protein interactions raise for anaphora resolution? - a preliminary analysis, in *Proceedings of the Second International Symposium on Semantic Mining in Biomedicine (SMBM 2006)*, Jena, (9–12).
- van der Sandt, Rob A. (1988), *Context and Presupposition*, Croom Helm, London.
- van der Sandt, Rob A. (1992), Presupposition projection as anaphora resolution, *Journal of Semantics* 9(4):333–377, doi:10.1093/jos/9.4.333.
- Segerberg, Krister (1990), Validity and satisfaction in imperative, *Notre Dame Journal of Formal Logic* 31(2):203–211.
- Shen, Dan & Mirella Lapata (2007), Using semantic roles to improve question answering, in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Association for Computational Linguistics, Prague, (12–21).
- Smith, Jan M. (1984), An interpretation of Martin-Löf's Type Theory in a type-free theory of propositions, *Journal of Symbolic Logic* 49.
- Steedman, Mark (1993), Categorical grammar (tutorial overview), *Lingua* 90:221–258.
- Sun, Renxu, Jing Jiang, Yee Fan Tan, Hang Cui, Tat-Seng Chua, & Min-Yen Kan (2005), Using syntactic and semantic relation analysis in question answering, in *Proceedings of the 14th Text Retrieval Conference (TREC 2005)*.
- Sundholm, Göran (1989), Constructive generalized quantifiers, *Synthese* 79:1–12.
- Tarski, Alfred (1983), *Logic, Semantics, Metamathematics*, Hackett, Indianapolis, second edition.
- Thomas K. Landauer, Peter W. Foltz & Darrell Laham (1998), Introduction to latent semantic analysis, *Discourse Processes* 25:259–284.
- Thomason, R. (1980), A modeltheory for propositional attitudes, *Linguistics and Philosophy* 4:47–70.
- Tichý, Pavel (1988), *The Foundations of Frege's Logic*, De Gruyter, Berlin, New York.
- Turner, Raymond (1992), Properties, propositions and semantic theory, in M. Rosner & R. Johnson (eds.), *Computational Linguistics and Formal Semantics*, Cambridge University Press, Cambridge, Studies in Natural Language Processing, (159–180).

- Turner, Raymond (2007), Computable models, *Journal of Logic and Computation* (Advanced Access), doi:10.1093/logcom/exm002.
- Wirth, Niklaus (1971), Program development by stepwise refinement, *Communications of the ACM* 14(4):221–227.
- von Wright, Georg Henrik (1967), Deontic logics, *American Philosophical Quarterly* 4:136–143.
- Zadrozny, Wlodek (1994), From compositional to systematic semantics, *Linguistics and Philosophy* 17:329–342.
- Zamansky, Anna, Nissim Francez, & Yoad Winter (2006), A ‘natural logic’ inference system using the Lambek calculus, *Journal of Logic Language and Information (JoLLI)* 15(3):273–295, ISSN 0925–8531, doi:10.1007/s10849-006-9018-x.