Semantics and Computational Semantics*

Matthew Stone
Rutgers University

Abstract Computational semanticists typically devise applied computer systems simply to work as well as possible, but their work enriches the study of linguistic meaning with important tools, resources and methods which I survey here. To get useful results requires systematizing models from semantics in computational terms. Key computational challenges involve the compositional analysis of sentence meaning; the formalization of lexical meaning and related world knowledge; and modeling context and conversation to connect semantics and pragmatics. Underpinning computational semanticists' successes in all these areas, we find the distinctive roles of representation, problem solving and learning in computation.

1 Introduction

Interdisciplinary investigations marry the methods and concerns of different fields. Computer science is the study of precise descriptions of finite processes; semantics is the study of meaning in language. Thus, computational semantics embraces any project that approaches the phenomenon of meaning by way of tasks that can be performed by following definite sets of mechanical instructions. So understood, computational semantics revels in applying semantics, by creating intelligent devices whose broader behavior fits the meanings of utterances, and not just their form. IBM's Watson (Ferrucci, Brown, Chu-Carroll, Fan, Gondek, Kalyanpur, Lally, Murdock, Nyberg, Prager, Schlaefer & Welty 2010) is a harbinger of the excitement and potential of this technology.

In applications, the key questions of meaning are the questions engineers must answer to make things work. How can we build a system that copes robustly with the richness and variety of meaning in language, and with its ambiguities and underspecification? The focus is on methods that give us clear problems we can state and solve.

* Preparation of this chapter was supported in part by NSF IIS-1017811. I appreciate feedback from the editors and reviewers of the series, Brian McMahan and Una Stojnic.

Language is large, and it's often a messy but routine business to create program constructs that account for linguistic data as well as possible given the background constraints otherwise imposed on system design. So the bread-and-butter of computational semantics is the development of machine-learning methods to induce and disambiguate symbolic structures that capture aspects of meaning in language. Linguistic issues are often secondary. I will not attempt a survey of machine learning in computational linguistics, or even computational semantics, here. Useful starting points into the literature are Manning & Schütze (1999), Jurafsky & Martin (2008), Màrquez, Carreras, Litkowski & Stevenson (2008), Agirre, Màrquez & Wicentowski (2009).

Instead, in keeping with the emphasis of this volume, I will explore the scientific overlap of computer science and semantics. My experience has always confirmed a deep affinity between the perspectives of the two fields. Recent developments in computational semantics bring an exciting new suite of tools, resources and methods to the scene. These results promise not only to enliven the capabilities of the robots it seems we must inevitably talk to; they promise to profoundly enrich our understanding of meaning in language as a unique bridge between the physical, psychological and social worlds.

In Section 2, I explore the natural connection between the lingistics and computer science, through the parallels between the logical metalanguage of formal semantics and the instantiated representations of computational semantics. Section 3 reviews computational approaches to compositionality, and shows how the challenges involved, particularly in disambiguation, have led to insightful new formalisms for the derivation of sentence meanings, as well as powerful tools for understanding existing grammars and formalisms. Section 4 turns to the computational lexicon, where the problems of carrying out inferences from word meanings has shaped computational theories of the organization of the lexicon and its relationship to common-sense inference, while the sheer need to scale systems up have led to fantastic databases of words and their occurrences in discourse. Finally, Section 5 considers the explanations we give of meaning in the broader context of language use. It's particularly important to show how to enrich our representations of meaning so we can interpret utterances as making contributions to interlocutors' ongoing activity. This requires us to link computational semantics with computational models of agency—characterization of interlocutors' rational action, as prompted by represented beliefs, desires and reasons. The potential of computational semantics thus goes beyond mere operationalizing or short-circuiting of linguistic theory. It is computational semantics that transforms our discovery that the meanings of utterances are governed by rules into an explanation of how language works.

2 Representation and Semantics

The key technical idea bridging computer science and semantics is that of REPRESENTATION. A representation is a symbolic structure in a system that can be understood to encode specified information—to carry meaning—because of mechanical processes that produce the representation as a function of appropriate states of the world and use the representation to guide the system's behavior in appropriate ways. See Newell (1982), Fodor (1987), Gallistel & King (2009), Smith (to appear). Computer scientists' interests in the meanings of representations largely parallel semanticists' interests in the meanings of linguistic expressions.

Meaning in language, of course, brings distinctive problems. An instructive way of foregrounding these problems is through the development of an explicit theory that characterizes the truth conditions of sentences in a fragment of natural language. See Davidson (1967), Larson & Segal (1995). We assume that we have an agreed metalanguage for which meaning is unproblematic (or outside the scope of our investigation). That might just be the natural language we use as theorists, English perhaps. We use the metalanguage to describe the meanings of sentences (or syntactic structures as analyzed by a syntactic grammar, or uses of the relevant expressions) in some object language of interest, German say. Our theory must pair each expression G of German with an English sentence E that says when G is true:

(1) G is true if and only if E.

In (2) I give a concrete example.

(2) 'Schnee ist weiss' is true if and only if snow is white.

Since any interesting fragment of a natural language will be infinite, our theory must specify generative mechanisms for deriving these statements from a finite set of primitives.

Linguists generally adopt a formal metalanguage, for example the language of higher-order intensional logic (Montague 1974, Dowty, Wall & Peters 1981), set theory (Barwise & Cooper 1981), discourse representation theory (Kamp & Reyle 1993), dynamic semantics (Groenendijk & Stokhof 1990, Muskens 1996), and so forth. These formalisms let us specify truth conditions precisely and without ambiguity. The use of a formal metalanguage also lets us understand the generativity of semantics as a straightforwardly computational requirement. Our goal is no more and no less than an algorithm—a finite specification of a mechanical procedure for pairing input expressions (of German, say) with output expressions in a suitable formal system (a logic of meanings). In doing semantics, we can naturally interface with techniques from computational logic to implement these algorithms, and thereby to help build, refine, test or apply new ideas in semantics. I survey work along these lines in Section 3.

The linguistic problems of meaning are not limited to judgments of truth conditions, however. Native speakers intuitions about meaning capture a diverse array of further relationships. For example, they can say when one sentence is more general than another or more specific, when one is incompatible with another or even its opposite, when one seems to suggest another or take it for granted. These judgments depend in part on compositional regularities about syntactic combination and logical vocabulary, but other information is crucial: These judgments also reflect the subtle interplay between a linguistic lexicon, organized by grammatical knowledge, and the rich body of world knowledge that invariably informs our descriptions and understandings of the world.

The line between grammar and common sense is blurry. It might be a botanical discovery that no oak is an elm. But it seems to be part of the meanings of the words that nothing can be big and small. We didn't discover the distinction; language invites us to make it. We might need chemists to tell us that iron oxidizes when it rusts. But native speakers already know that something breaks when somebody breaks it; it is a single verb *break* in these different realizations. If we associate doughnuts with holes, it's only because we've learned that crullers are unpopular. But it's language that packages the door you slam (a slab) together with the door you go through (an opening).

The overlap in these sources of information is both theoretically and empirically problematic. Computational techniques offer a way to make sense of the phenomena. They invite us to focus on representation—on the content of our knowledge and judgments, on the structures that can encode that content, and on algorithmic processes that can operationalize the relevant inferences. This focus enables a clearer understanding of knowledge of meaning, and ties that understanding directly to explanations of our semantic intuitions. This focus also allows us more precisely to distinguish grammatical information from common sense background knowledge, in terms of its form and content, or in terms of its status in learning and reasoning—without denying that key judgments require the synthesis of knowledge of both kinds. This perspective informs my review of computational lexical semantics in Section 4. Computational semanticists face urgent practical needs to bridge linguistic knowledge and real-world inference, so the frameworks, corpora and databases they have developed for systematizing lexical inferences are particularly valuable.

In the end, our semantic competence reveals itself perhaps most directly and most profoundly in our ability to communicate. We can express our thoughts creatively in words; we can recover precise new ideas from others' creative utterances; we can ask questions, give answers, provide instructions; we can agree on or dispute the way things are. These skills—and the understanding of meaning they reveal—seem far removed from the abstractions of semantic theory. The gap has lead to extreme rejections of those abstractions, such as the doctrine that the meaning of a word simply is its use in the language (Wittgenstein 1953: I,43). There are good arguments against a naive identification of semantics with anything people know or do (Putnam 1975, Burge 1979). Still a purely abstract theory of meaning, disconnected from use, cannot satisfy us.

This is another case where techniques of computation and representation offer useful theoretical insights. The representational theory of mind comes into its own in explaining complex adaptive behavior by appeal to represented knowledge (Newell & Simon 1976, Newell 1982). The perspective is that of problem solving—the flexible, general but mechanistic synthesis of creative strategies to pursue a system's interests. Problem solving starts from a representation that specifies possible steps of reasoning and determines the form that a solution must take. It proceeds by searching systematically through the possibilities. Reasoning steps may involve reducing or transforming the problem to a simpler one, but they may also involve enriching the representation of the problem by taking into account represented background knowledge. Problem solving amounts to using represented information. When the system derives a result, it's because the system knows, in virtue of the information that it represents, that the result solves its problem.

The phenomena of communication thus invite us to develop a computational semantics embedded within a broader account of the represented contexts and goals of interlocutors in communicative exchanges. This is an important theme of research in computational semantics and pragmatics, which I survey in Section 5. Indeed, it is an important theme of research in semantics and pragmatics generally, and Section 5 charts a longstanding and dynamic intellectual interchange, spanning philosophy, linguistics and computer science, and bringing together common-sense intuitions, empirical data, formal representations and computational models. The vast array of open problems that remain in the area offer many exciting opportunities for semanticists to continue to explore computational ideas and tools.

For computer scientists, there is a special methodological place for a computational theory of represented meaning and its role in communication. Such a theory promises to shape our understanding of intelligence in fundamental ways. The ability to engage in human-like conversation has defined the basic criterion for attributing intelligence to machines since Turing (1950). See Shieber (2004) for a history of the intellectual influence of the Turing's ideas—and an updated defense of the Turing Test. Turing's proposal is a solid one. Even if the Turing Test wouldn't detect everything we might consider intelligence, passing it remains a distant goal.

Meaning is at the heart of the challenge of such a theory, and of the artifacts we could build with it. So far, too much engineering is needed to implement representations for us to see their content as intrinsic. As designers, we interpret the representations as meaningful, because this helps us to specify the operations the representations must support or to analyze the role of representations in the system. Nothing in the system itself privileges a description in terms of meanings. Indeed, our systems do not yet have anything like the kinds of meanings we have. See Searle (1980), Dreyfus (1972), Winograd & Flores (1986) for arguments for this skeptical position.

Computers that talk as we talk will force us to sharpen our views. Perhaps, with Searle, we will come to think of meaning—the information our thoughts carry about the world—fundamentally as a sector of the subjective economy, on a par with qualia and the other mysteries of the conscious mind. On this view, questions about the possibility of meaning in systems are just questions about the place of consciousness in nature. If that's right, we might have to abandon meaningfulness as a goal for systems, and retreat to "Weak Artificial Intelligence (AI)", the doctrine that our systems merely do what we do. However, like Turing (1950), I personally think that meaning is vital to our understanding of the action of intelligent agents. When we attribute meaning to somebody, we aren't just commenting on their inaccessible private experiences; we're explaining their behavior in terms of the kind of engagement they have with the world. Weak AI sells meaning short.

The alternative is "Strong AI", the view that these computers will actually have semantics, in the sense that the architecture, function and causal interaction of these systems with the world gives the referents of symbols an indispensable explanatory role in describing the systems. Dennett has suggested that this might be cheap; on his view, meaning is just a matter of perspective, taking an "intentional stance" (Dennett 1987). However, most regard the realization of intrinsically meaningful representations as another distant goal. We do have some precise frameworks for thinking about when meaning can be explanatory (Newell 1982, Pylyshyn 1984), and when it is not.

Deficiencies of existing representations point to things meaningful systems would be able to do that current systems cannot. All are active areas of computational research. Grounding symbols in perceptual experience (Harnad 1990) calls for linking linguistic abilities to perception and learning (Cohen, Oates, Beal & Adams 2002, Roy & Pentland 2002, Yu & Ballard 2004). Deferring to experts (Kripke 1972, Putnam 1975) calls for tracking causal chains that connect meanings to one's linguistic community, and working actively through perception, deliberation and action to support and resolve the status of symbols as meaningful (Steels & Belpaeme 2005, DeVault, Oved & Stone 2006, Oved & Fasel 2011). Though symbol grounding is central to a certain kind of computational semantics—inquiry into the semantics we can realize through computational techniques—the issues involved are primarily philosophical and psychological, rather than linguistic. I won't discuss them further here.

3 Computation and Compositional Semantics

Computational semantics is predicated on mechanisms for constructing representations of the meanings of sentences. We shall see that it's straightforward to adopt established techniques from linguistic semantics, particularly the λ-calculus familiar from work in the Montagovian tradition. But that's only one of many possibilities. And even implementations of standard compositional fragments using λ-calculus can draw distinctively on computational insights and perspectives. The key is again the distinctive role of representation in computational work.

Linguistic semanticists use terms in the λ-calculus principally as a way of naming functions. It's the functions that play a fundamental theoretical role as meanings; the terms just specify them. For example, semanticists generally understand the principle of compositionality to apply at the level of meanings. The meaning of a larger expression depends on the meanings of the parts and the way they are put together. See Szabó (2008). The combinatorial logic of the λ-calculus provides a natural algorithmic realization of this metatheory—we use λ-terms to map out the mathematical operations by which we compose our mathematical abstractions of meanings.

The λ-calculus is also an essential computational formalism. It goes back to early work by Church as a formalization of the notion of an effective procedure (Church 1941), and served as the inspiration for LISP (McCarthy 1960), the first high-level programming language. Normalizing λ-expressions is a fundamental model of computation, and the underlying data structures of functional programming—environments, closures, continuations—offer streamlined and well-studied methods to implement and normalize λ-terms; see Appel (1992) or Abelson, Sussman & Sussman (1996). More generally, explorations of the λ-calculus in computer science have led to powerful concepts and tools for modeling composition of functions as an algorithmic process. Shan & Barker (2006), Barker & Shan (2008) showcase the linguistic applications of one such tool, DELIMITED CONTINUATIONS, which model the local context within which a scope-taking expression is evaluated. In their textbook on computational semantics, van Eijck & Unger (2010) use the built-in functions and types of the programming language Haskell as the basis for a systematic exploration of λ-calculus techniques; they take the reader from the fundamentals up to the limits of current research.

A complementary perspective comes from computational logic, where we are interested not only in normalizing λ-terms but in solving algebraic equations assuming the equalities of the λ-calculus. Huet's higher-order unification procedure enumerates these solutions (Huet 1975). An accessible inference engine is λ-prolog (Felty, Gunter, Hannan, Miller, Nadathur & Scedrov 1988, Miller & Nadathur 2012), which can be used straightforwardly to specify compositional semantics (Pareschi & Miller 1990). Another notable linguistic application of higher-order unification constructs representations of the interpretations of linguistic constituents under ellipsis (Dalrymple, Shieber & Pereira 1991).

However, the λ-calculus is only one of a diverse set of formal systems that computational semanticists use to compose meaning representations. Computational semantics is concerned with the concrete representations, not with the abstract meanings. The goal is to construct symbol structures that we can interpret as representing desired meanings. This project invites us to build representations compositionally. Our REPRESENTATIONS of the meaning of a larger expression must be calculated from REPRESENTATIONS of the meanings of its parts and the way they are put together. This syntactic perspective brings profound differences to the typical understanding of compositionality, because complex representations retain their constituents, where mathematical operations transform them. Much more flexible algorithms for specifying complex meanings are now possible. The possibilities sometimes seem to offer explanatory insights and theoretical elegance as well as practical engineering benefits.

Let's start with a simple case: UNIFICATION GRAMMAR (Shieber 1986, Moore 1989). This formalism derives semantic representations by accumulating constraints in tandem with a syntactic derivation. The constraints take the form of equations between terms which may contain variables. The formalism gets its name from the process, UNIFICATION, which determines values for variables to solve the equations.

In unification grammar, we can abstract meanings by using variables as placeholders; then we can derive concrete instances through equations that set the variables to appropriate values. For example, we might represent an unsaturated predication in two parts, an expression such as $walks(X)$ together with a specification X that indicates the argument position in this expression. Saturating the predicate is then realized by an equation unifying the argument position with a particular instance, as $X = chris$. The equation determines the corresponding saturated instance $walks(chris)$ as the overall meaning of the predication. Here is a general rule accomplishing this compositionally in tandem with syntactic derivation:

(3) $s[sem : E] \rightarrow np[sem : A] \;\; vp[sem : E, arg : A]$

In (3) syntactic productions are annotated with FEATURE STRUCTURES, records that pair attributes with corresponding values (in this case semantic representations). Using (3) thus imposes the constraint that the *sem* feature of the (subject) noun phrase must be equal to the *arg* feature of the verb phrase predicate. Thus if we have derived $np[sem : chris]$ as an analysis of *Chris* and $vp[sem : walks(X), arg : X]$ as an analysis of *walks*, then rule (3) lets us derive $s[sem : walks(chris)]$ as an analysis of *Chris walks*, by imposing the requirement that $X = chris$.

The treatment of quantification in unification grammar offers an instructive contrast with λ-calculus techniques. To model a quantifier like *everyone* in subject position, we need to instantiate the main predication to a bound variable, then construct an overall meaning representation for the sentence by universally quantifying over this predication. So *everyone* needs to know about a predication (*pred*) and the argument position of the predication (*parg*) in order to construct its semantics. That means we should analyze *everyone* through a complex representation like $qnp[sem : \forall x.G, pred : G, parg : x]$. To use this meaning compositionally, we have a rule such as (4).

(4) $s[sem : E] \rightarrow qnp[sem : E, pred : P, parg : A] \;\; vp[sem : P, arg : A]$

In applying (4) to analyze *everyone walks*, we now derive equations $X = x$ unifying the unsaturated position of *walks* with the bound variable of the quantifier and another equation $G = walks(X)$ unifying the predication itself with the nuclear scope of the quantifier. The overall meaning representation $\forall x.walks(x)$ is thus assembled as the solution to a set of constraints—not through the function–argument discipline familiar from the λ-calculus.

These examples suggest the difference in perspective that comes from adopting constraint-satisfaction formalisms for meaning construction. If we model sentence meaning as the solution to a set of equations, it's natural to expect to find elements of meaning that are simultaneously determined by multiple equations, and, conversely, elements of meaning that are left unconstrained and must be resolved by context. By contrast, λ-calculus techniques expect each element in a meaning representation to be specified exactly once. The flexibility of constraint techniques is sometimes instructive for theories of linguistic semantics. For instance, constraint techniques offer new ways to analyze idioms and multi-word expressions. Each constituent can not only specify its own meaning but can impose a constraint on the meaning of the whole (Sag, Baldwin, Bond, Copestake & Flickinger 2002). The approach offers an analogous treatment of concord phenomena in negation, tense, modality and other semantic fields. For example, negative elements can constrain a clause to have a negative meaning without any one element being privileged as contributing the negation compositionally (Richter & Sailer 2006). Conversely, constraint techniques offer a useful way to think about arguments that are optionally specified, for example in verb-frame alternations. Providing these arguments imposes a constraint on interpretation that is simply absent when these arguments are omitted (Palmer 1990). Regardless of how these analyses are ultimately settled, familiarity with constraint semantics can help linguists distinguish between the fundamental empirical challenges of semantics and the technical obstacles that arise in formalizing insights with specific tools, including λ-calculus.

In fact, computational semanticists typically adopt constraint-satisfaction techniques because of their computational advantages. More expressive constraint languages are particularly useful for representing, reasoning about and resolving ambiguity. A prototypical case is the ability to underspecify scope relationships during semantic composition. The essential idea is to augment the set of constraints imposed during meaning construction so that the meaning representations that satisfy the constraints correspond to all and only the possible readings of a sentence.

We can illustrate the idea with hole semantics (Bos 1995). In this constraint language, we write logical formulas that describe what we know about meaning representations at a particular point in syntactic analysis and semantic disambiguation. HOLES are variables in the logic that range over meaning representations; holes have to be plugged by assigning them appropriate values. LABELS are unique names for substructures of semantic representations; labels allow us to describe constraints on particular occurrences of expressions. (Unification grammar, by contrast, only allows us to constrain the expression types that appear in meaning representations.) The constraint language allows us to specify the representational structure at a specific expression occurrence, to force a specific expression occurrence to plug a specific hole, and to constrain one expression occurrence l to occur as a subexpression of another l', written $l \lhd^* l'$. Solutions are formulas obtained by finding a suitable mapping from holes to labels in such a way as to satisfy the constraints.

To illustrate, consider the two readings of *every student solved a problem* given in (5).

(5) $\forall x.student(x) \supset \exists y.(problem(y) \wedge solve(x, y))$
$\exists y.problem(y) \wedge \forall x.(student(x) \supset solve(x, y))$

Hole semantics allows us to compactly describe the two formulas in terms of their substantial shared structure and an underspecified scope relationship, as in (6).

(6) $l_1 : \forall x.student(x) \supset h_1, l_2 : \exists y.problem(y) \wedge h_2, l_3 : solve(x, y),$
$l_3 \lhd^* l_1, l_3 \lhd^* l_2, l_1 \lhd^* h_0, l_2 \lhd^* h_0$

To satisfy (6), we need to plug the holes h_0, h_1 and h_2—corresponding respectively to the full sentence semantics, the nuclear scope of the universal, and the nuclear scope of the existential—with labels l_1, l_2 and l_3—corresponding to the universal quantification, the existential quantification, and the main predication. Dominance constraints ensure that the main predication is a subformula of the nuclear scope of both quantifiers, and that both quantifications occur as subformulas of the full sentence semantics. There are two solutions: plugging $h_0 = l_1, h_1 = l_2, h_2 = l_3$ leads to left-to-right scope while plugging $h_0 = l_2, h_2 = l_1, h_1 = l_3$ leads to inverse scope.

If we instrument a constraint-based grammar so that semantic composition delivers descriptions of underspecified scope relationships, we can derive compact summaries of possible interpretations, rather than enumerating different possibilities separately. The computational savings is formidable. Blackburn & Bos (2005) offer a comprehensive tutorial on hole semantics. And other formalisms for computational semantics adopt a similar approach to underspecifying scope by constraining structural relationships among expression occurrences, notably including underspecified minimal recursion semantics (UMRS) (Copestake, Lascarides & Flickinger 2001), the constraint language for lambda structures (Egg, Koller & Niehren 2001, Koller, Niehren & Thater 2003), and hybrid logic constraint semantics (Baldridge & Kruijff 2002).

Different constraint languages embody different insights into the patterns of ambiguity that tend to arise together in wide-coverage grammars. We need the right set of constraints, in harmony with a perspicuous organization of the grammar, to be able to fold alternative analyses together into a single representation. In UMRS, for example, we can constrain the functor associated with a particular label to a small set of values; this way we can write a single semantic constraint that captures word-sense ambiguity. We can also separately constrain the argument positions present at a particular label, without committing to the arity of the functor; this helps with the variation in argument structure seen across different syntactic frames. Finally, UMRS privileges "flat" semantics—conjunctions of predications characterizing the same generalized individuals, as found in Davidsonian event semantics (Davidson 2002), Hobbs's ontologically promiscuous semantics (Hobbs 1985), and the original formalism of minimal recursion semantics (Copestake, Flickinger, Pollard & Sag 2005). A special set of constraints makes it easy to conjoin semantic constraints in a common scope. Copestake et al. (2001) show how to use these constraints effectively in conjunction with HPSG syntax to derive useful and compact underspecified semantic representations in practical wide-coverage grammars.

In reasoning about ambiguity, it's often helpful to abstract away from concrete syntactic structure as well as concrete semantic representations. One technique is to take syntactic dependency relationships as the input to semantics, rather than a full syntax tree. This makes it easier to describe systematically the ambiguities that arise in binding and scoping complements and modifiers, since all complements and modifiers are represented as dependents of a single lexical item. An influential approach is the glue logic presented by Dalrymple, Lamping & Saraswat (1993) and Dalrymple, Lamping, Pereira & Saraswat (1997), originally developed for LFG. The basic prediction of glue logic is that any scoping of complements and modifiers is possible, as long as the constructed meaning representation type-checks and contains exactly one copy of the material contributed by the head and each of its dependents. This can be modeled as an inference problem in a fragment of linear logic. Lev (2007) summarizes the resulting ambiguities in an underspecified formalism. Similar insights can be used to assemble semantic representations for other dependency formalisms, including lexicalized TAG (Kallmeyer & Joshi 2003).

Linguists interested in scaling up compositional semantic theories to larger fragments can now draw on a range of systems that are capable of delivering semantic representations with wide coverage. For English, examples include the English Resource Grammar in HPSG (Copestake & Flickinger 2000), with UMRS semantics; the C&C tools and Boxer for CCG parsing and DRT semantic representations (optionally including anaphor resolution) (Curran, Clark & Bos 2007); and the XLE tools and ParGram project grammars for LFG and glue logic (Butt, Dyvik, King, Masuichi & Rohrer 2002). These systems model a large fraction of the words and syntactic constructions found in written texts and can associate these analyses with semantic representations that specify interpretive dependencies and scope relationships. Unfortunately, these grammars have rather impoverished analyses of much of the fine structure of linguistic meaning, because of the scarcity of semantic resources and the limits of disambiguation techniques.

For what they cover, computational techniques offer an attractive way to explore the predictions of formal analyses, because they can systematically explore the possibilities for interactions among grammatical specifications. To systematically describe the structure and meaning of general texts, we need diverse specifications, which leads to proliferating analyses even for simple sentences. Alexander Koller (p.c.) reports that the English Resource Grammar yields about 65,000 readings for (7).

(7) But that would give us all day Tuesday to be there.

Such statistics illustrate how dangerous it can be to rely on pencil and paper alone to assess the predictions of theories.

More generally, the semantic representations embodied in wide-coverage systems go a long way to explain how knowledge of meaning can actually be used in real time in language processing. The constraint-based approach allows for monotonic accumulation of information about meaning during incremental processing. With UMRS, for example, it's possible to give semantic representations that summarize the information available about meaning after each of the successive steps of analysis in a language processing pipeline—starting with part-of-speech tagging, continuing through syntactic attachment, word-sense disambiguation, recognition of predicate–argument structure, and so forth. These techniques thus allow for a wide range of reasoning methods to be applied to resolve ambiguities, and moreover enables them to proceed by inferring additional semantic constraints rather than by enumerating and comparing individual readings. These techniques come with little cost, because it's often surprisingly elegant to instrument syntactic specifications to deliver a flexible and precise constraint semantics.

The success perhaps offers lessons for semantics. We aim to account for links between words and the world. Whatever does this must supervene on human practices, and so must ultimately depend on individual speakers' cognitive representations of meaning. The success of computational techniques suggest that semantic theory too might best be served by describing the algorithmic construction of cognitive representations of meaning, rather than the compositional relationships among meanings themselves. There is now a sizable contingent of researchers in linguistics and cognitive science who adopt just this view (Jackendoff 1990, Hamm, Kamp & van Lambalgen 2006, Pietroski 2008: among others). The jury is out; representationalism is not without its theoretical challenges (see Steedman & Stone 2006). So readers interested in following up the linguistic consequences of distinctively computational formalisms are entering a lively and important theoretical debate.

4 Computation and Lexical Semantics

Compositional semantics gives us representations to capture aspects of the logical structure of meaning. But it typically abstracts away from the rich particulars of linguistic content—what specific entities and concepts does a sentence describe, and what specific claim does it make about them? This utterance content depends on a wide range of substantive knowledge about the world, not just the logical structure of meaning. Thus, to develop precise representations of linguistic content we need a corresponding formalization of the conceptual and inferential relationships among word meanings. The project is complex because it ties the organization of grammar together with speakers' knowledge of the common-sense world and with the patterns of use which reveal our ordinary ways of thinking and talking about what matters to us.

Computational research, with its ability to manage large databases of linguistic structures and examine large corpora of linguistic evidence, brings unique insights into the complexity of lexical semantics. This section reviews research into three different aspects of word meaning: WORD SENSES, conceptual distinctions that are needed to capture lexical inferences; SEMANTIC FRAMES, linguistic and conceptual structures that connect lexical and compositional inferences; and SEMANTIC FIELDS, associations among concepts that explain the plausibility or naturalness of the ensembles of meaning we characteristically use.

To make my presentation more concrete, I will refer throughout to a specific example, the English verb *charge*. The word originates, via Norman French, in a Latin root referring to the activity of loading a wagon. This sense is now rare, if not entirely forgotten. Instead, *charge* is now the word we use for payment by credit card, for the supply of energy to electronic devices, and for the formalities of proceedings in criminal courts. There are lots of new things to talk about, and we have pressed our old words into service! The examples in (8) illustrate these (and other) uses.

(8)
a. The brigade charged the enemy positions.
b. The cavalry charged.
c. Kim charged my cellphone for me.
d. The battery charges in around 3 hours.
e. They charged us fees for their services.
f. The bank charged interest on the loan.
g. They charged us a commission to sell the armoire.
h. I charged $200 on my credit card.
i. The board charged the committee to study the proposal.
j. The DA charged them with murder.
k. Prosecutors charge that Quimby took $10000 in bribes.

An inventory of WORD SENSES attempts to specify, in formal terms, a list of the qualitatively different relationships that words in a language can express. The examples in (8) show that *charge* has senses meaning "to bear down on in attack", "to supply with electrical power", and "to demand payment", among others. These particular labels come from the electronic lexical database WordNet (Miller 1995, Fellbaum 1998), a standard resource in computational lexical semantics. We return to WordNet and its inventory of word senses and lexical relationships below. Computational techniques have proved helpful not only in systematizing word senses for broad coverage, as WordNet does, but also in clarifying and streamlining the methodology for positing word senses in response to corpus evidence and native speakers' linguistic judgments.

An inventory of SEMANTIC FRAMES, meanwhile, attempts to specify the qualitatively different syntactic realizations that a language can allow for the same word sense. Consider (8a) and (8b). Evidently, in using *charge* describing an attack, English speakers have the option of spelling out the quarry as a direct object noun phrase, but they also have the option to leave the quarry implicit. In either case, the attacker is the subject. Contrast this with (8c) and (8d). In describing the provision of energy to a device, *charge* can also be used either transitively or intransitively. But it's the device that's always specified, either as the direct object in the transitive frame, as in (8c), or as the subject in the intransitive frame, as in (8d). Different syntactic realizations depend both on the combination of arguments that are expressed overtly, as in the previous cases, and also on the syntactic category through which those arguments are realized. Examples (8e–8g) all describe goods and services requiring payment, but use *for* plus a noun phrase, *on* plus a noun phrase, and *to* plus a verb phrase.

Computational semantics approaches these differences through databases of semantic frames, such as VerbNet (Kipper, Korhonen, Ryant & Palmer 2008) and FrameNet (Baker, Fillmore & Lowe 1998). Each frame identifies the roles or participants implicitly related to a target concept and describes the different constructions that allow these participants to be specified in construction with a word used to express the concept. As is the case with word senses, computational techniques for analyzing semantic frames support linguistics by helping to organize corpus data, speakers' judgments, and their connections to specifications of grammar.

Finally, the data in (8) illustrates the importance of lexical information in describing and resolving semantic ambiguities. Look at the arguments of *charge* in (8e–8g). The data reflects an obvious constraint: what can be charged, in the sense of demanded as payment, is quite limited in scope. It include not just *fees*, *interest* or *a commission*, as here, but a family of associated items—*expenses*, *fines*, *prices*, *sums*, *amounts*, *a fortune*. Similar items recur as what can be paid, deducted, earned, and so forth. By identifying these clusters, we learn more about how to identify what sense of *charge* is at play in a particular example. We also get new insights into what words have similar meanings, including not only the similarity of payment words like *fees* and *interests* that we find in a common cluster, but also the indirect similarity we find among verbs like *charge*, *pay* and *deduct* that take similar clusters as arguments. We find analogous clusters at other argument positions: the military units that typically get orders to charge, as in (8a); the devices that typically require a charge, as in (8c); the crimes one can be charged with, as in (8j). In these cases, where clusters clearly showcase expressions that people tend to use, we can see that they will be statistical constructs giving information about frequent events and and the common perspectives speakers take on them, as well as encoding restrictions about what makes sense. Computational semanticists have combined lexical resources, corpus data, language processing techniques and statistical inference to find many ways of clustering meanings for applications. See Turney & Pantel (2010) for a review. These studies provide important tools for understanding the evidence that speakers must use in learning, disambiguating and reasoning with the meanings of words and expressions.

Armed with this perspective, let us tour the landscape of computational lexical semantics in more detail.

A linguist looking to study word senses can build on a range of research results. First, there are large-coverage lexical databases. In English, the prime example is WordNet (Miller 1995, Fellbaum 1998). WordNet is not just a dictionary. It provides an inventory of concepts, specifies key relationships among concepts, and maps out the many-to-many correspondences between concepts and words.

The basic entries in WordNet, corresponding to word senses or concepts, are SETS OF WORDS that can be used synonymously—in the right context—to convey the same concept. For example, WordNet lists *bill* as a synonym for *charge* in defining the sense of demanding payment. These entries are called SYNSETS for short. When a word is ambiguous, WordNet represents each sense with corresponding synset which includes that word. In keeping with the ambiguities seen in (8), *charge* is associated in WordNet with many synsets.

WordNet also describes a number of semantic relationships among word senses. TROPONYMS of a verb are more specific verbs that describe the specific manner in which something unfolds. Troponyms are a special case of

Wordnet's HYPONYM links that connect terms with more specific instances. The WordNet troponyms of *charge* (in the synset of demanding payment) include *levy* and *invoice*. WordNet's HYPERNYM links, meanwhile, connect words to more general terms. WordNet's hypernyms for *charge* (demand payment) include *account* and *calculate*.

WordNet has been an inspiration for similar efforts across a wide range of languages. Because it links concepts to semantic relationships, across a wide range of vocabulary, it can be thought of as a coarse semantic network. Researchers have used it to measure the distance among concepts (Budanitsky & Hirst 2006) and, perhaps more interestingly for semantics, to acquire meaning postulates that connect the meanings of different words and help to automate inferences such as entailment (Bos & Markert 2005).

The specific inventory of senses that WordNet offers, however, is open to criticism. The conceptual distinctions among senses are not always clear, not only because of overlaps among the senses themselves but also because of interactions between lexical semantics, compositional semantics and argument structure. It is striking, for example, that WordNet attempts to distinguish three senses of *charge*, one for demanding payment from somebody in a specific case, another for pricing a service in general at a certain amount, and another for updating an account by entering a certain amount as a charge. It will be difficult to tease these cases apart in the interpretation of a specific sentence, since in the normal case, the way businesses demand payment is to prepare an invoice claiming the current general price against the customer's account, effectively undertaking all three kinds of charging. Indeed, perhaps we can say everything we need to say about demanding payment if we approach the different cases syntactically, and give a compositional semantics that supplements a single word sense with appropriate generic or presupposed interpretations for arguments that are not explicitly realized. At the same time, WordNet's coverage of world knowledge is limited. For example, none of these senses of *charge* is related in WordNet to the obvious common sense implication of both (8e) and (8f) that someone has bought something, someone has sold something, someone has paid someone, and so forth.

A different thread of research in computational semantics is prompted by these limitations. This research focuses explicitly on evidence and methodologies for inventorying word senses in a language and identifying the word senses at play in specific uses. A key principle is Yarowsky's one sense per collocation heuristic, which says that statistically reliable associations of words—collocations—are good indications of a distinct underlying concept (Yarowsky 1993). We've seen some examples for this in the case of *charge*. Kilgarriff and colleagues (Kilgarriff & Rundell 2002, Kilgarriff, Rychly, Smrz & Tugwell 2004) describe WordSketches, a way of finding key collocations in large corpora and presenting the results interactively to support lexicographers in identifying and distinguishing prominent word senses. These are the basis for wide coverage efforts to build better dictionaries—especially educational dictionaries for (human) language learners.

Given an array of word senses, we can now ask native speakers to indicate which word sense is at issue in particular utterances. We can use the resulting annotated corpora in linguistics to understand patterns of use, and in computer science to train and evaluate methods for disambiguating word senses. Over the last couple decades, the Senseval and Semeval projects have carried out a wide range of annotation projects in support of the research community and made the data available. See Kilgarriff & Palmer (2000), Edmonds & Kilgarriff (2002), Wilcock, Ide & Romary (2004), Agirre et al. (2009). Senseval has focused on providing comprehensive data about a limited range of phenomena of theoretical and practical interest; the corpora are not broad coverage.

A number of resources now offer a more comprehensive picture of aspects of lexical semantics. FrameNet (Baker et al. 1998) is a pioneering example. FrameNet combines a regimented analysis of a wide range of verbs with annotated data classifying attested uses. It focuses specifically on the alternative patterns of meaning and structure exhibited by verbs as in (8). FrameNet consists of an electronic database that describes several thousand English verbs. Each verb sense is associated with a frame, which identifies a broad class of eventualities and specifies the roles that different participants may play in these eventualities. For example, FrameNet captures the payment sense of *charge* by linking *charge* to the *commerce_collect* frame, a subframe of the *commerce_money-transfer* frame that explicitly indicates how the seller comes to have money as a result of an economic transaction. The frame maps out five privileged roles: the buyer, seller, goods, and money, as well as a rate role when money is specified per unit of goods. (The frame also suggests common modifiers for this type of eventuality, including manner, means, place, purpose, reason, time and the unit of measure for goods in the transaction.) In addition, FrameNet describes the alternative patterns of syntactic realization through which roles can be realized. These patterns are documented by attested textual examples, which illustrate both the different possibilities for realizing specific role fillers as well as the possibilities for leaving some role fillers implicit. You can get a sense of this data from the fragments presented in (9).

(9)
 a. Firms [seller] have a right to charge what the market will bear [money]...
 b. Nimslo [seller] will charge £9.99 [money] to produce 18 prints [goods]...
 c. The new lender [seller] will charge its legal expenses [money] to you [buyer].

VerbNet (Kipper et al. 2008) offers a similar electronic database of verbs. In fact, the VerbNet project distributes a mapping between VerbNet entries and corresponding entries in WordNet and FrameNet. VerbNet offers more fine-grained verb senses than FrameNet, but coarser-grained specifications of the semantic roles associated with those word senses. However, what makes VerbNet particularly useful is that it is linked with PropBank (Palmer, Gildea & Kingsbury 2005), a large corpus of text that has been annotated not only with syntactic parse trees but also with predicate–argument relationships and semantic roles. With VerbNet, it is therefore possible to get quantitative information about the distribution of lexical semantic patterns for verbs. In computational linguistics, these resources provide the starting point for machine learning methods to disambiguate the senses of verbs and the roles of their arguments. But the data is also a natural starting point for empirical tests of linguistic theories of argument structure and word meaning.

Of course, there's more to lexical semantics than verbs. Pustejovsky (1991), for example, argues that we also need to assign frames to nouns and adjectives to account for the participants in lexical semantic relationships, implicit and explicit, and to formalize the syntactic patterns for filling these roles. Resources beyond verbs remain rather limited, however. A notable exception is the Penn Discourse Treebank (Prasad, Dinesh, Lee, Miltsakaki, Robaldo, Joshi & Webber 2008), which extends corpus-based models of argument structure to discourse connectives. These are words like *but*, *however* and *instead* that serve to express rhetorical relationships in discourse (Webber, Stone, Joshi & Knott 2003). The Penn Discourse Treebank marks up the textual source for the explicit and implicit arguments of these relationships.

Example (10) illustrates.

(10) As an indicator of the tight grain supply situation in the U.S., market analysts said that **late Tuesday the Chinese government**, *which often buys U.S. grains in quantity*, **turned** <u>instead</u> **to Britain to buy 500,000 metric tons of wheat**.

The underlined *instead* relates the boldfaced clause describing China's purchase of wheat from Britain to the non-restrictive relative clause which describes the understood alternative: China's normal purchases of U.S. grain.

As difficult as they are to build with wide coverage, even semantic frames are quite shallow in the kinds of inferences they support. The difficulties are exacerbated by interactions between lexical semantics, compositional semantics and common-sense inference. Consider the examples in (11).

(11) a. They charged me twice.
b. If they charged $25 a day, how many days would you stay?
c. It was excellent work for what they charged.

Dictionaries of semantic frames do indicate that *charging* as in (11a) is a kind of commercial transaction involving a buyer and a seller, but there's no real model of what happens in such situations. Deeper knowledge would be required to reach the conclusion—obvious to any human reader, and crucial to inferences we'd like to automate about the information, attitude and coherence of the text—that (11a) describes a problem or error. Conversely, understanding whether a *charge* sentence describes a generic offer, as considered (hypothetically) in (11b) or a specific completed transaction, as seems to be the case in (11c), is ultimately a matter of recognizing a hidden generic interpretation. Argument structure provides an important clue, since it's natural to leave the buyer unspecified in a generic offer, as in (11b), provided that's a grammatical realization of a verb's semantic frame (as it is for *charge*). But we also need to know the grammatical and discourse context, because the buyer may be omitted because it's presupposed in context, as in (11c).

Traditionally, systems and resources have been very limited in tracking these kinds of interactions among compositional semantics, lexical semantics and context. This is one motivation for recent workshops on RECOGNIZING TEXTUAL ENTAILMENT (RTE). RTE is the problem of deciding whether one text can be inferred from another, given plausible resolution of ambiguities and common-sense background knowledge. The RTE competitions have developed a set of standardized tasks, data sets and evaluation protocols to assess the performance of systems on this inference problem. RTE was spearheaded by the PASCAL Network of Excellence and has since become part of the Text Analysis Conference managed by the National Institute of Standards in the US. The resources distributed for past competitions remains a natural starting point for new research projects.

While much research on RTE focuses on shallow and machine learning techniques, RTE has in fact prompted a number of projects with substantial interest for formal semantics. Bos & Markert (2005), for example, integrate a formal semantic framework based on discourse representation theory with automated theorem proving and common-sense inference rules derived from WordNet. Nairn, Condoravdi & Karttunen (2006), meanwhile, identify the factuality of embedded clauses by modeling the lexical meaning of implicative verbs and its interaction with compositional semantics, including complementation and negation.

In general in computational linguistics, richer linguistic resources—such as the lexical meanings encoded in WordNet, FrameNet or VerbNet—only exacerbate the already urgent problem of disambiguation. With syntax and compositional semantics, we have some hope that the combinatorial structure of grammatical derivations can zero in on plausible readings, perhaps with the help of coarse surface patterns. If we must also disambiguate lexical alternatives like the different senses of *charge* at issue in (8), it seems we need a qualitatively richer and deeper understanding of what makes semantic relationships plausible to native speakers. But we presently have no way to derive such understandings from available representations of grammatical or world knowledge.

What we do have in abundance, however, are patterns of word use from corpora. Linguists have argued that these patterns offer robust, statistical evidence about word meaning (Harris 1954, Firth 1957). Concretely, heuristics like one sense per discourse or one sense per collocation, which help to motivate word senses in the first place, also provide leverage for resolving the ambiguities. When we find common combinations of words, we can assume that they are to be interpreted in their specific senses that normally occur together. This heuristic provides a corpus-based metric for semantic plausibility that can inform a wide range of larger projects, from engineering approaches to disambiguation to the elaboration of psycholinguistic models.

Most analyses begin by creating a measure of similarity between word distributions. Similarity is a trigger for inferring relationships between words and generalizing from one word to another. The key difference is the knowledge sources for computing similarity. The simplest are techniques such as the influential model of latent semantic analysis (Landauer, Foltz & Laham 1998: LSA), which simply starts from a matrix describing which words occur together in the same document in a text collection. LSA uses a statistical technique called principal component analysis (Jolliffe 2002) to infer weighted clusters of words that explain most of the statistical variation in the matrix. These patterns intuitively correspond to document topics or semantic fields. Other clustering methods start from recognized syntactic relationships in large corpora. For example, Pereira, Tishby & Lee (1993) and Lee (1999) use the joint distribution of English verbs and their object nouns to infer clusters and measure similarity. Background knowledge can also be added when it is available. Resnik (1999) creates a measure of semantic similarity by enriching the WordNet hierarchy with probability information derived from the frequency of occurrence of words in text.

These measures of semantic similarity give shallow but robust techniques for reasoning about meaning on a large scale. Clustering the arguments of verbs gives a computational reconstruction of the linguistic notion of a selectional restriction (Resnik 1996, Light & Greiff 2002). More generally, models of semantic distance make it possible to measure whether a possible analysis of a sentence is semantically similar to examples we have seen in training data, and assess how likely it is to be correct (Dagan, Lee & Pereira 1999). In fact, these distributional predictions correlate closely with human judgments of semantic plausibility (Padó, Crocker & Keller 2009).

Meanwhile, the presence of specific patterns in large collections of text is often a good proxy for specific real-world relationships. Computational semanticists have used distributional patterns to establish logical relationships such as whether operators are downward entailing (Danescu-Niculescu-Mizil & Lee 2010). They have discovered semantic relationships at the level of the lexicon, including hyponymy and part–whole relationships (Hearst 1998, Pantel & Pennacchiotti 2006, Snow, Jurafsky & Ng 2006). And they have captured real-world inferences, including inferences about the typical duration or order of common events (Chambers & Jurafsky 2008, 2009, Gusev, Chambers, Khaitan, Khilnani, Bethard & Jurafsky 2010).

As a result of these successes, many engineers are looking to replace logical models with distributional models that provide a useful proxy for linguistic semantics across a wide range of tasks. See for example Mitchell & Lapata (2010), Baroni & Lenci (2010). That has clear limits—among other things in handling context-dependent expressions such as pronouns or ellipsis, and in connecting language to other representations of the world. But linguists would do well to understand the basis for the models and appreciate why they are powerful and necessary in computational investigations.

5 Computing Meaning in Context

The techniques of Sections 3 and 4 allow us to derive meaning representations which feature precise real-world predicates in grammatically-specified logical relationships. Such representations do not, as yet, make explicit the understood links between linguistic expressions and the discourse contexts. Accordingly, computational semanticists typically subject these representations to a further stage of resolution and inference to recover these links. The models computational semanticists use extend a tradition that extends to seminal early projects from the AI literature, such as (Charniak 1973, Hobbs 1978, Webber 1983). In many respects, the work offers similar insights and techniques to work on discourse in formal semantics (Karttunen 1976, Kamp 1981, Heim 1982). However, because of its emphasis on common-sense inference and naturally-occurring data, computational work tends to adopt modeling assumptions and mathematical techniques whose relationship to linguistic ideas may not be immediately apparent. This section sketches the most important computational techniques specifically in relation to comparable formal proposals.

At the outset, note that the goal of these techniques is actually to recover the intended interpretation of context-dependent elements in specific utterances. Imagine the instruction in (12) addressed to a robotic kitchen assistant, for example.

(12) Put the dough hook back onto your mixer.

We expect the robot to comply with this instruction. To settle on a specific action to perform, the robot must figure out who *your* refers to, what is meant by *the mixer* and *the dough hook*, and where the dough hook goes *back onto*. Computational semanticists hope to to deliver answers to such questions by combining models of meaning with models of context.

Linguistic theory is an important ingredient in computational models. Following Kaplan (1989), we can represent the contributions of context-sensitive elements with variables in the semantic representation. For (12), for example, we'll want to find a person p for *your*, objects m and h for *the mixer* and *the dough hook*, and an action type a that will represent the content of the instruction. As in DRT (Kamp & Reyle 1993), we can use the discourse context to establish a list of possible values for variables. This will involve documenting specific available referents; for example, we might wind up with a symbol m_1 that identifies the specific KitchenAid Pro 6-quart stand mixer that the user is equipped with. But it will also require us to spell out referents that can be inferred indirectly—it's a fact that each mixer of this general type comes with its own dough hook, which we might therefore represent with a complex term as $h_1(m_1)$ to illustrate the functional dependency. The relationship matters: Even if we know about many dough hooks in the kitchen, we'll take *the dough hook* to mean the one the mixer comes with.

Given this information, the remaining challenge is to predict likely interpretations, taking into account the recent history of the conversation, the situation in which the conversation is playing out, and the broad commonsense background that interlocutors take for granted. A common perspective is that this context can be modeled as a knowledge base or information state (Poesio & Traum 1997, Larsson & Traum 2001). Linguists should think of the information state as a realization of Stalnaker's common ground (Stalnaker 1973, 1979, 1998). Stalnaker's proposal links the problem of communication to the inherent dynamics of coordinating agents in a way that brings out the natural commonalities between human–human conversation and the engineering enterprise of supporting human–computer interaction. Agents that coordinate need to track the joint information that each can naturally expect the other to rely on in understanding one another's actions (Lewis 1969). Tracking the common ground is thus as much a feature of collaborative agent designs in AI (Power 1977, Grosz & Sidner 1990, Lochbaum 1998, Carberry & Lambert 1999) as it is in models of human inquiry in linguistics and the philosophy of language.

Computational semanticists are, however, typically more explicit about the rich structure of information states in conversation. This goes beyond the content interlocutors have agreed to in the conversation, as emphasized by Stalnaker (1979). For example, the information state also records the organization of the ongoing conversation into hierarchically structured discourse segments (Grosz & Sidner 1986). Each of these segments is associated with a distinct conversational purpose that it addresses, which is achieved in part through the purposes of its constituent segments and contributes in turn to meeting the goals of later and larger segments. And the information state must model interlocutors' attention in conversation, which establishes some potential resolutions of meaning as salient and others as more remote. See Ginzburg (2012) for a recent formal model of the organization of dialogue that explores the consequences of rich models of conversation for linguistic grammar. Clark (1996) and Roberts (2012) offer more programmatic arguments for bridging pragmatic models of context with the models of joint activity often used to build collaborative agents in AI. Concretely, in the context of (12), we can imagine appealing to a diverse set of factors in our information state, including the broad outlines of the ongoing recipe, which frames the purposes and organization of the conversation; general functional knowledge about the available items, including when, how and why they are typically used; as well as the specific history of utterances and real-world actions that have taken place thus far in the cooking project. Note that while it's currently prohibitively time-consuming to build knowledge bases of this kind at a large scale, AI research offers a powerful repertoire of tools for representing the relevant information and drawing the relevant inferences. See Brachman & Levesque (2004) or Koller & Friedman (2009) for different approaches.

We can now model the resolution of underspecified aspects of meaning as a process of default inference that shows how the interpretive requirements of an utterance are most plausibly met in the context. Concretely, we assume that any utterance with context-dependent elements will be associated with a specification of the information in the context that we use to resolve the unknown values. For (12), something like (13) is in order:

(13) $$\mathit{dough}(d) \wedge \mathit{hook}(h) \wedge \mathit{nn}(h,d) \wedge \mathit{addressee}(u,p) \wedge \mathit{poss}(p,m) \wedge \mathit{mixer}(m) \wedge \mathit{result}(a,\mathit{on}(h,m)) \wedge \mathit{past}(\mathit{on}(h,m)) \wedge \mathit{contributory}(a)$$

This information is quite heterogeneous. (13) includes the constraint that the referent of *your* must be the addressee of the current utterance u, which falls under Kaplan's (1989) concept of the CHARACTER of an indexical item. (13) includes anaphoric presuppositions (Kripke 2009) triggered by particular lexical items, such as the constraint $\mathit{past}(\mathit{on}(h,m))$ that encodes the requirement, from the word *back*, that the end-state of action a must already have been realized at an earlier stage of the conversation. Reference resolution is modeled by constraints contributed by the semantics of the definite noun phrases: something we can see either as an additional presupposition, as is usual in DRT, or just as a representation of the speaker's inferred referential intentions (Kripke 1977). Meanwhile, the requirement that the action described make a natural contribution to the ongoing task, represented as the constraint *contributory*(a), is almost certainly a speaker presupposition that is more or less independent of grammar and encodes general background knowledge about what kinds of discourse contributions make sense in the current context. Computational semanticists tend to be rather sanguine about running such diverse constraints together. Hobbs, Stickel, Appelt & Martin (1988, 1993), for example, refer to the full set of interpretive constraints derived in the course of utterance understanding simply as logical form.

Resolution proceeds by posing the interpretive constraints, such as those in (13) as a query against the information state, represented as a knowledge base as sketched above. The query yields a "top proof"—one that uses maximally salient information, most probable premises, or makes the fewest assumptions. As a side effect, the proof specifies a substitution replacing variables in the interpretive constraints with contextually-defined values. This substitution then determines the contextually-resolved meaning of the utterance.

Techniques of this kind first come to prominence in the middle 1980s. Mellish (1985) and Haddock (1987) used constraint satisfaction to resolve reference; Dale & Haddock (1991) used it to generate references. These techniques made minimal use of logical inference in processing queries. Hobbs et al. (1988, 1993) used a model with a full set of inferential mechanisms, including the ability to make abductive assumptions, which in effect allows for partial and probabilistic matches against the context. They show how the inference resolves a wide range of "local pragmatic" effects. Their examples included reference resolution, bridging, and noun-noun compounds—all featured in (12)—as well as metonymy and coercion of predicates.

A major challenge to this approach is the need to model the dynamic local contexts which we must use to reason about nested contextually-dependent elements. To make the problem concrete, consider a variant of (12).

(14) Put the dough hook on each of your mixers.

The correct interpretation here resolves *the dough hook* to a functional expression $h_1(x)$ interpreted within the scope of *each of your mixers* and represented in terms of the corresponding bound variable x. To start, it's hard to get nested interpretive constraints right in the first place: the relevant components of meaning sometimes seems to take scope in complex ways (Kamp & Rossdeutscher 1994) and sometimes seems relatively orthogonal to the rest of sentence meaning (Potts 2005). Given nested interpretive constraints, Bos's (2003) influential implementation of the anaphoric presupposition theory of van der Sandt (1992) formalizes nested scope correctly in linguistic terms, but offers limited facilities for bridging and other inferential relationships in nested dependencies. A more sophisticated technique is due to Krahmer and Piwek (Krahmer & Piwek 1999, Piwek & Krahmer 2000). They model local contexts using hypothetical reasoning in constructive logic. A constructive proof of $A \supset B$ is a proof of B that uses A as an extra assumption; constructive proofs also guarantee, in ways classical proofs do not, that existential claims are backed up by specific instances that satisfy them. The two properties make it possible to characterize the information available for bridging reference in a nested scope and to extract a suitable representation of a dependent discourse referent like $h_1(x)$ from a demonstration that the interpretive constraints that apply in the nested scope are satisfied. Because of the close affinities of constructive and modal logic, similar techniques carry over to inference mechanisms that use modal logic to characterize the changing information of interlocutors in conversation (Stone 2000).

Another challenge to the approach is the problem of ranking interpretations. The formal mechanisms proposed by work such as Hobbs et al. (1993) lack a comprehensive treatment of the dynamics of attention, and more flexible formalisms, as explored for example by Stone & Thomason (2002) or Asher & Lascarides (2003), are not robust enough to apply broadly to interpretation. In fact, most work on preferences in interpretation has focused on special cases of resolving pronoun references, and abstracted away from the interactions that are generally possible in constraint-satisfaction models by processing single references at a time. An effective heuristic is to assigning referents dynamic prominence based on their syntactic realization. The idea goes back to Hobbs (1978); Strube (1998) offers a recent empirical assessment and defense of simple strategies against more complex models. However, it's clear that the prominence of a referent is a function not only of its recent syntactic realization but also of the discourse relation that connects the current sentence to the ongoing conversation (Grosz, Joshi & Weinstein 1995, Kehler 2001, Asher & Lascarides 2003). For example, the preference for continued reference to the subject across a whole subsequent utterance is a feature of extended descriptions and narratives; it does not apply when an utterance presents a parallel or a contrast to previous discourse (Kehler 2001). Meanwhile, when cause–effect relations are at issue, whatever preferences are in play are easily overridden by common-sense background knowledge. In practice, of course, the added discourse and world knowledge that's necessary to model and test interpretive preferences at a large scale is generally unavailable—though semantic generalizations discovered from corpora, of the sort discussed in Section 4, may soon change this. Instead, computational work uses machine learning methods to find shallow cues and patterns that provide good guides to reference; see Ng (2010).

6 Conclusion

My aim in this chapter has been to offer an overview and introduction to computational semantics that lets linguists see the overlap between engineering questions and scientific questions. I have specifically aimed to highlight the potential of computational tools and resources to support scientific investigations. In offering this survey, I hope to have suggested how closely interwoven computational techniques already are with the perspectives and results of formal semantics. I cannot imagine serious theory building in semantics that does not engage with computational infrastructure for exploring broad annotated corpora, formal grammars, lexical databases, and contextual inference. These tools can be strikingly easy to use. Where they seem obscure, it may be because they reflect insights not yet fully assimilated into the perspectives of linguistic semantics.

This is not to say that formal semantics and computational semantics are the same thing. There is a big disjuncture between engineering projects—the challenges of disambiguation, or the needs of applications—and questions linguists are interested in. Linguistic science inevitably focuses attention on telling analyses of rare events that highlight empirical distinctions among comparatively esoteric proposals. In most cases, however, computational semanticists still struggle to discover simple and robust techniques that get the frequent cases right. The gap can be extreme.[1] The divergent goals of linguistics and computer science do limit the possibilities for collaboration. But this is no obstacle for linguists to take up computational techniques on linguistic terms, and adapt those techniques to ask and answer questions of linguistic interest.

That said, many of the divergences we see when we consider superficial analyses of well-studied language like English disappear in the broader purview of meaning. For example, both formal semantics and computational semantics make headway by regimenting and formalizing speakers' intu-

1 The IBM speech group in the 1970s was famous for an aphorism from its director (possibly apocryphal): Every time I fire a linguist, my performance goes up.

tions. They are the raw data for semantic theorizing, but they are also the raw data for machine learning and evaluation experiments. Thus, linguists and computer scientists can find common cause in working to annotate deeper aspects of utterance structure and interpretation. This remains profoundly challenging for both fields. Likewise, both formal semantics and computational semantics bring increasing concern for underdocumented languages. These languages can be the most useful testbed for new methodologies and applications. But they also turn out in many cases to instantiate theoretically-significant typological properties. Here computer scientists and linguists find common cause in exploiting and deepening our understanding of linguistic diversity while empowering speakers and making information more broadly accessible. Let's close, then, on with optimistic prospect that formal semantics and computational semantics continue to join together in the exploration of such synergies.

References

Abelson, Harold, Gerald J. Sussman & Julie Sussman. 1996. *Structure and interpretation of computer programs, second edition*. MIT Press. http://mitpress.mit.edu/sicp/.

Agirre, Eneko, Lluís Màrquez & Richard Wicentowski. 2009. Computational semantic analysis of language: SemEval-2007 and beyond. *Language Resources and Evaluation* 43(2). 97–104. http://dx.doi.org/10.1007/s10579-009-9091-2.

Appel, Andrew. 1992. *Compiling with continuations*. Cambridge, UK: Cambridge University Press.

Asher, Nicholas & Alex Lascarides. 2003. *Logics of conversation*. Cambridge, UK: Cambridge University Press.

Baker, Collin F., Charles J. Fillmore & John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the 36th annual meeting of the Association for Computational Linguistics and 17th international conference on computational linguistics, volume 1*, 86–90. Montreal, Canada. http://www.aclweb.org/anthology/P98-1013.

Baldridge, Jason & Geert-Jan M. Kruijff. 2002. Coupling CCG and hybrid logic dependency semantics. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 319–326. http://www.aclweb.org/anthology/P02-1041.pdf.

Barker, Chris & Chung-chieh Shan. 2008. Donkey anaphora is in-scope binding. *Semantics and Pragmatics* 1(1). 1–46. http://dx.doi.org/10.3765/sp.1.1.

Baroni, Marco & Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics* 36(4). 673–721. http://aclweb.org/anthology-new/J/J10/J10-4006.pdf.

Barwise, Jon & Robin Cooper. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophy* 4. 159–219. http://www.jstor.org/pss/25001052.

Blackburn, Patrick & Johan Bos. 2005. *Representation and inference for natural language. a first course in Computational semantics*. Stanford, CA: CSLI Publications.

Bos, Johan. 1995. Predicate logic unplugged. In *Proceedings of the 10th Amsterdam colloquium*, 133–143.

Bos, Johan. 2003. Implementing the binding and accommodation theory for anaphora resolution and presupposition projection. *Computational Linguistics* 29(2). 179–210. http://dx.doi.org/10.1162/089120103322145306.

Bos, Johan & Katja Markert. 2005. Recognising textual entailment with logical inference. In *Proceedings of the 2005 conference on empirical methods in natural language processing (EMNLP)*, 628–635. http://www.aclweb.org/anthology/H/H05/H05-1079.

Brachman, Ronald & Hector J. Levesque. 2004. *Knowledge representation and reasoning*. Morgan Kaufmann.

Budanitsky, Alexander & Graeme Hirst. 2006. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics* 32(1). 13–47. http://dx.doi.org/10.1162/coli.2006.32.1.13.

Burge, Tyler. 1979. Individualism and the mental. In P. French, T. Uehling & H. Wettstein (eds.), *Midwest studies in philosophy: Studies in metaphysics*, 73–122. Minneapolis: University of Minnesota Press.

Butt, Miriam, Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi & Christian Rohrer. 2002. The parallel grammar project. In *COLING-02 workshop on grammar engineering and evaluation*, http://aclweb.org/anthology-new/W/W02/W02-1503.pdf.

Carberry, Sandra & Lynn Lambert. 1999. A process model for recognizing communicative acts and modeling negotiation subdialogues. *Computational Linguistics* 25. 1–53. http://aclweb.org/anthology-new/J/J99/J99-1001.pdf.

Chambers, Nathanael & Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, 789–797. http://www.aclweb.org/anthology/P/P08/P08-1090.

Chambers, Nathanael & Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the joint conference of the 47th annual meeting of the ACL and the 4th international joint conference on natural language processing of the AFNLP*, 602–610. http://www.aclweb.org/anthology/P/P09/P09-1068.

Charniak, Eugene. 1973. Jack and Janet in search of a theory of knowledge. In *Proceedings of the 3rd international joint conference on artificial intelligence*, 337–343. http://ijcai.org/Past%20Proceedings/IJCAI-73/PDF/035.pdf.

Church, Alonzo. 1941. *The calculi of lambda conversion*. Princeton, NJ: Princeton University Press.

Clark, Herbert H. 1996. *Using language*. Cambridge, UK: Cambridge University Press.

Cohen, Paul R., Tim Oates, Carole R. Beal & Niall Adams. 2002. Contentful mental states for robot baby. In *Proceedings of the eighteenth national conference on artificial intelligence (AAAI)*, 126–131.

Copestake, Ann & Dan Flickinger. 2000. An open-source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the second conference on language resources and evaluation (LREC-2000)*, http://www.delph-in.net/erg/.

Copestake, Ann, Dan Flickinger, Carl Pollard & Ivan Sag. 2005. Minimal recursion semantics: An introduction. *Research on Language and Computation* 3. 281–332. http://dx.doi.org/10.1007/s11168-006-6327-9.

Copestake, Ann, Alex Lascarides & Dan Flickinger. 2001. An algebra for semantic construction in constraint-based grammars. In *Proceedings of the 39th annual meeting of the Association for Computational Linguistics*, 140–147. http://www.aclweb.org/anthology/P01-1019.

Curran, James, Stephen Clark & Johan Bos. 2007. Linguistically motivated large-scale NLP with C&C and Boxer. In *Proceedings of the 45th annual meeting of the Association for Computational Linguistics (companion volume)*, 33–36. http://www.aclweb.org/anthology/P07-2009.

Dagan, Ido, Lillian Lee & Fernando Pereira. 1999. Similarity-based models of cooccurrence probabilities. *Machine Learning* 34(1-3). 43–69. http://dx.doi.org/10.1023/A:1007537716579.

Dale, Robert & Nicholas Haddock. 1991. Content determination in the generation of referring expressions. *Computational Intelligence* 7(4). 252–265. http://dx.doi.org/10.1111/j.1467-8640.1991.tb00399.x.

Dalrymple, Mary, John Lamping, Fernando C. N. Pereira & Vijay A. Saraswat.

1997. Quantifiers, anaphora, and intensionality. *Journal of Logic, Language and Information* 6(3). 219–273. http://dx.doi.org/10.1023/A:1008224124336.

Dalrymple, Mary, John Lamping & Vijay A. Saraswat. 1993. LFG semantics via constraints. In *Sixth conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 97–105. http://aclweb.org/anthology-new/E/E93/E93-1013.pdf.

Dalrymple, Mary, Stuart M. Shieber & Fernando C. N. Pereira. 1991. Ellipsis and higher-order unification. *Linguistics and Philosophy* 14. 399–452. http://dx.doi.org/10.1007/BF00632780.

Danescu-Niculescu-Mizil, Cristian & Lillian Lee. 2010. Don't 'have a clue'? Unsupervised co-learning of downward-entailing operators. In *Proceedings of the ACL short papers*, 247–252.

Davidson, Donald. 1967. Truth and meaning. *Synthese* 17(1). 304–23. http://www.jstor.org/stable/20114563.

Davidson, Donald. 2002. The logical form of action sentences. In *Essays on actions and events*, 105–121. Oxford: Clarendon Press.

Dennett, Daniel C. 1987. *The intentional stance*. Cambridge, MA: MIT Press.

DeVault, David, Iris Oved & Matthew Stone. 2006. Societal grounding is essential to meaningful language use. In *Twenty-first national conference on artificial intelligence (AAAI-06)*, 747–754.

Dowty, David R., Robert E. Wall & Stanley Peters. 1981. *Introduction to Montague semantics*. Boston: Reidel.

Dreyfus, Hubert L. 1972. *What computers can't do: A critique of artificial reason*. New York: Harper and Row.

Edmonds, Philip & Adam Kilgarriff. 2002. Introduction to the special issue on evaluating word sense disambiguation systems. *Natural Language Engineering* 8. 279–291. http://dx.doi.org/10.1017/S1351324902002966.

Egg, Markus, Alexander Koller & Joachim Niehren. 2001. The constraint language for lambda structures. *Journal of Logic, Language and Information* 10(4). 457–485. http://dx.doi.org/10.1023/A:1017964622902.

van Eijck, Jan & Christina Unger. 2010. *Computational semantics with functional programming*. Cambridge, UK: Cambridge University Press. http://www.computational-semantics.eu/.

Fellbaum, Christiane (ed.). 1998. *WordNet: An electronic lexical database*. MIT.

Felty, Amy P., Elsa L. Gunter, John Hannan, Dale Miller, Gopalan Nadathur & Andre Scedrov. 1988. Lambda-prolog: An extended logic programming language. In *9th international conference on automated deduction (CADE)*, 754–755. http://dx.doi.org/10.1007/BFb0012882.

Ferrucci, David, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer & Chris Welty. 2010. Building Watson: An overview of the DeepQA project. *AI Magazine* 31(3). 59–79. http://www.aaai.org/ojs/index.php/aimagazine/article/view/2303.

Firth, John Rupert. 1957. *Papers in linguistics, 1934-1951.* Oxford, UK: Oxford University Press.

Fodor, Jerry. 1987. *Psychosemantics.* Cambridge, MA: MIT Press.

Gallistel, C. R. & Adam Philip King. 2009. *Memory and the computational brain: why cognitive science will transform neuroscience* Blackwell/Maryland lectures in language and cognition. Chichester, UK: Wiley-Blackwell.

Ginzburg, Jonathan. 2012. *The interactive stance.* Oxford, UK: Oxford University Press.

Groenendijk, Jeroen & Martin Stokhof. 1990. Dynamic predicate logic. *Linguistics and Philosophy* 14. 39–100. http://www.jstor.org/pss/25001418.

Grosz, Barbara, Aravind Joshi & Scott Weinstein. 1995. Centering: A framework for modelling the local coherence of discourse. *Computational Linguistics* 21(2). 203–226. http://www.aclweb.org/anthology-new/J/J95/J95-2003.pdf.

Grosz, Barbara J. & Candace L. Sidner. 1986. attention, intentions, and the structure of discourse. *Computational Linguistics* 12(3). 175–204. http://aclweb.org/anthology-new/J/J86/J86-3001.pdf.

Grosz, Barbara J. & Candace L. Sidner. 1990. Plans for discourse. In Philip Cohen, Jerry Morgan & Martha Pollack (eds.), *Intentions in communication*, 417–444. Cambridge, MA: MIT Press.

Gusev, Andrey, Nathanael Chambers, Pranav Khaitan, Divye Khilnani, Steven Bethard & Dan Jurafsky. 2010. Using query patterns to learn the duration of events. In *Proceedings of the ninth international conference on computational semantics*, 145–154. http://aclweb.org/anthology-new/W/W11/W11-0116.pdf.

Haddock, Nicholas J. 1987. Incremental interpretation and combinatory categorial grammar. In *Proceedings of the 10th international joint conference on artificial intelligence (IJCAI)*, 661–663. http://ijcai.org/Past%20Proceedings/IJCAI-87-VOL2/PDF/012.pdf.

Hamm, Fritz, Hans Kamp & Michiel van Lambalgen. 2006. There is no opposition between formal and cognitive semantics. *Theoretical Linguistics* 32.

1–40. http://dx.doi.org/10.1515/TL.2006.001.
Harnad, Stevan. 1990. The symbol grounding problem. *Physica D* 42. 335–346. http://dx.doi.org/10.1016/0167-2789(90)90087-6.
Harris, Zellig. 1954. Distributional structure. *Word* 10(23). 146–162.
Hearst, Marti A. 1998. Automatic discovery of WordNet relations. In Christiane Fellbaum (ed.), *Wordnet: An electronic lexical database*, 131–152. Cambridge, MA: MIT Press.
Heim, Irene. 1982. *The semantics of definite and indefinite noun phrases*: University of Massachusetts, Amherst dissertation. Published 1987 by Garland Press.
Hobbs, Jerry, Mark Stickel, Douglas Appelt & Paul Martin. 1993. Interpretation as abduction. *Artificial Intelligence* 63. 69–142. http://dx.doi.org/10.1016/0004-3702(93)90015-4.
Hobbs, Jerry R. 1978. Resolving pronoun references. *Lingua* 44(4). 311 – 338. http://dx.doi.org/10.1016/0024-3841(78)90006-2.
Hobbs, Jerry R. 1985. Ontological promiscuity. In *Proceedings of the 23rd annual meeting of the Association for Computational Linguistics*, 60–69. http://www.aclweb.org/anthology/P85-1008.
Hobbs, Jerry R., Mark Stickel, Douglas Appelt & Paul Martin. 1988. Interpretation as abduction. In *Proceedings of the 26th annual meeting of the Association for Computational Linguistics*, 95–103. http://www.aclweb.org/anthology/P88-1012.
Huet, Gérard P. 1975. A unification algorithm for typed lambda-calculus. *Theoretical Computer Science* 1(1). 27–57. http://dx.doi.org/10.1016/0304-3975(75)90011-0.
Jackendoff, Ray S. 1990. *Semantic structures*. Cambridge, MA: MIT Press.
Jolliffe, I.T. 2002. *Principal component analysis, second edition* Springer Series in Statistics. Heidelberg: Springer.
Jurafsky, Daniel & James H. Martin. 2008. *Speech and language processing (2nd edition) (Prentice Hall series in artificial intelligence)*. Upper Saddle River, NJ: Prentice Hall 2nd edn.
Kallmeyer, Laura & Aravind K. Joshi. 2003. Factoring predicate argument and scope semantics: Underspecified semantics with LTAG. *Research on Language and Computation* 1(1–2). 3–58. http://dx.doi.org/10.1023/A:1024564228892.
Kamp, Hans. 1981. A theory of truth and semantic representation. In Jeroen Groenendijk, Theo Janssen & Martin Stokhof (eds.), *Truth, interpretation and information*, Dordrecht: Foris.

Kamp, Hans & Uwe Reyle. 1993. *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*. Dordrecht: Kluwer.

Kamp, Hans & Antje Rossdeutscher. 1994. DRS-construction and lexically driven inference. *Theoretical Linguistics* 20. 97–164. http://dx.doi.org/10.1515/thli.1994.20.2-3.165.

Kaplan, David. 1989. Demonstratives. In Joseph Almog, John Perry & Howard Wettstein (eds.), *Themes from Kaplan*, 481–564. Oxford: Oxford University Press.

Karttunen, Lauri. 1976. Discourse referents. In James D. McCawley (ed.), *Syntax and semantics 7: Notes from the linguistic underground*, 363–385. New York: Academic Press.

Kehler, Andrew. 2001. *Coherence, reference and the theory of grammar*. Stanford, CA: CSLI Publications.

Kilgarriff, Adam & Martha Palmer. 2000. Introduction to the special issue on SENSEVAL. *Computers and the Humanities* 34(1–2). 1–13. http://www.jstor.org/stable/30204787.

Kilgarriff, Adam & Michael Rundell. 2002. Lexical profiling software and its lexicographic applications—a case study. In *Proceedings of the tenth EURALEX congress*, 807–818. http://www.euralex.org/elx_proceedings/Euralex2002/090_2002_V2_Adam%20Kilgarriff%20&%20Michael%20Rundell_Lexical%20Profiling%20Software%20and%20its%20Lexicographic%20Ap.pdf.

Kilgarriff, Adam, Pavel Rychly, Pavel Smrz & David Tugwell. 2004. The sketch engine. In *Proceedings of the eleventh EURALEX congress*, 105–116. http://www.euralex.org/elx_proceedings/Euralex2004/011_2004_V1_Adam%20KILGARRIFF,%20Pavel%20RYCHLY,%20Pavel%20SMRZ,%20David%20TUGWELL_The%20%20Sketch%20Engine.pdf.

Kipper, Karin, Anna Korhonen, Neville Ryant & Martha Palmer. 2008. A large-scale classification of English verbs. *Language Resources and Evaluation* 42(1). 21–40. http://dx.doi.org/10.1007/s10579-007-9048-2.

Koller, Alexander, Joachim Niehren & Stefan Thater. 2003. Bridging the gap between underspecification formalisms: Hole semantics as dominance constraints. In *10th conference of the European Chapter of the Association for Computational Linguistics*, 195–202. http://www.aclweb.org/anthology-new/E/E03/E03-1024.pdf.

Koller, Daphne & Nir Friedman. 2009. *Probabilistic graphical models: principles and techniques*. Cambridge, MA: MIT Press.

Krahmer, Emiel J. & Paul Piwek. 1999. Presupposition projection as proof construction. In Harry C. Bunt & Reinhart A. Muskens (eds.), *Computing meaning: Volume 1*, 281–300. Dordrecht: Kluwer.

Kripke, Saul. 1972. *Naming and necessity*. Cambridge, MA: Harvard University Press.

Kripke, Saul. 1977. Speaker's reference and semantic reference. In Peter A. French, Thomas Uehling, Jr. & Howard K. Wettstein (eds.), *Midwest studies in philosophy, volume II*, 255–276. Minneapolis: University of Minnesota Press. Volume 2: Studies in Semantics.

Kripke, Saul A. 2009. Presupposition and anaphora: Remarks on the formulation of the projection problem. *Linguistic Inquiry* 40(3). 367–386. http://dx.doi.org/10.1162/ling.2009.40.3.367.

Landauer, Thomas K, Peter W. Foltz & Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse Processes* 25(2-3). 259–284. http://dx.doi.org/10.1080/01638539809545028.

Larson, Richard & Gabriel Segal. 1995. *Knowledge of meaning: An introduction to semantic theory*. Cambridge, MA: MIT Press.

Larsson, Staffan & David Traum. 2001. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering* 6. 323–340. http://dx.doi.org/10.1017/S1351324900002539.

Lee, Lillian. 1999. Measures of distributional similarity. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, 25–32. http://www.aclweb.org/anthology/P99-1004.

Lev, Iddo. 2007. *Packed computation of exact meaning representations*: Stanford University dissertation. http://sites.google.com/site/pulcproject/material/suthesis-iddolev-2side.pdf.

Lewis, David K. 1969. *Convention: A philosophical study*. Cambridge, MA: Harvard University Press.

Light, Marc & Warren Greiff. 2002. Statistical models for the induction and use of selectional preferences. *Cognitive Science* 26(3). 269 – 281. http://dx.doi.org/10.1016/S0364-0213(02)00070-8.

Lochbaum, Karen E. 1998. A collaborative planning model of intentional structure. *Computational Linguistics* 24(4). 525–572. http://aclweb.org/anthology-new/J/J98/J98-4001.pdf.

Manning, Christopher D. & Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. Cambridge, MA: MIT Press.

Màrquez, Lluís, Xavier Carreras, Kenneth C. Litkowski & Suzanne Stevenson. 2008. Semantic role labeling: An introduction to the special issue. *Com-*

putational Linguistics 34(2). 145–159. http://dx.doi.org/10.1162/coli.2008.34.2.145.

McCarthy, John. 1960. Recursive functions of symbolic expressions and their computation by machine, part I. *Communications of the ACM* 3. 184–195. http://doi.acm.org/10.1145/367177.367199.

Mellish, Christopher S. 1985. *Computer interpretation of natural language descriptions*. Chichester, UK: Ellis Horwood.

Miller, Dale & Gopalan Nadathur. 2012. *Programming with higher-order logic*. Cambridge, UK: Cambridge University Press.

Miller, George A. 1995. WordNet: A lexical database for English. *Communications of the ACM* 38(11). 39–41. http://dx.doi.org/10.1145/219717.219748.

Mitchell, Jeff & Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science* 34(8). 1388–1429. http://dx.doi.org/10.1111/j.1551-6709.2010.01106.x.

Montague, Richard. 1974. The proper treatment of quantification in ordinary English. In Richmond H. Thomason (ed.), *Formal philosophy: Selected papers of Richard Montague*, New Haven: Yale University Press.

Moore, Robert C. 1989. Unification-based semantic interpretation. In *Proceedings of the 27th annual meeting of the Association for Computational Linguistics*, 33–41. http://www.aclweb.org/anthology/P89-1005.

Muskens, Reinhard. 1996. Combining Montague semantics and discourse representation. *Linguistics and Philosophy* 19(2). 143–186. http://www.jstor.org/pss/25001621.

Nairn, Rowan, Cleo Condoravdi & Lauri Karttunen. 2006. Computing relative polarity for textual inference. In *Proceedings of the fifth international workshop on inference in computational semantics (icos-5)*, 67–76. http://aclweb.org/anthology/W/W06/W06-3907.pdf.

Newell, Allen. 1982. The knowledge level. *Artificial Intelligence* 18. 87–127. http://www.aaai.org/ojs/index.php/aimagazine/article/view/99.

Newell, Allen & Herbert A. Simon. 1976. Computer science as empirical inquiry: Symbols and search. *Communications of the ACM* 19(3). 113–126. http://doi.acm.org/10.1145/360018.360022.

Ng, Vincent. 2010. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th annual meeting of the Association for Computational Linguistics*, 1396–1411. http://www.aclweb.org/anthology/P10-1142.

Oved, Iris & Ian Fasel. 2011. Philosophically inspired concept acquisition for artificial general intelligence. In Jürgen Schmidhuber, Kristinn R.

Thórisson & Moshe Looks (eds.), *Artificial general intelligence*, vol. 6830 Lecture Notes in Computer Science, 363–369. Heidelberg: Springer. http://dx.doi.org/10.1007/978-3-642-22887-2_44.

Padó, Ulrike, Matthew W. Crocker & Frank Keller. 2009. A probabilistic model of semantic plausibility in sentence processing. *Cognitive Science* 33(5). 794–838. http://dx.doi.org/10.1111/j.1551-6709.2009.01033.x.

Palmer, Martha, Daniel Gildea & Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics* 31(1). 71–105. http://aclweb.org/anthology-new/J/J05/J05-1004.pdf.

Palmer, Martha Stone. 1990. *Semantic processing for finite domains* Studies in natural language processing. Cambridge, UK: Cambridge University Press.

Pantel, Patrick & Marco Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the Association for Computational Linguistics*, 113–120. http://www.aclweb.org/anthology/P06-1015.

Pareschi, Remo & Dale Miller. 1990. Extending definite clause grammars with scoping constructs. In David H.D. Warren (ed.), *Logic programming*, 373–389. Cambridge, MA: MIT Press.

Pereira, Fernando, Naftali Tishby & Lillian Lee. 1993. Distributional clustering of English words. In *Proceedings of the 31st annual meeting of the assocation for computational linguistics*, 183–190. http://www.aclweb.org/anthology/P93-1024.

Pietroski, Paul. 2008. Minimalist meaning, internalist interpretation. *Biolinguistics* 4. 317–341. http://www.biolinguistics.eu/index.php/biolinguistics/article/view/70.

Piwek, Paul & Emiel J. Krahmer. 2000. Presuppositions in context: Constructing bridges. In Pierre Bonzon, Marcos Cavalcanti & Rolf Nossum (eds.), *Formal aspects of context*, 85–106. Dordrecht: Kluwer.

Poesio, Massimo & David R. Traum. 1997. Conversational actions and discourse situations. *Computational Intelligence* 13. 309–347. http://dx.doi.org/10.1111/0824-7935.00042.

Potts, Christopher. 2005. *The logic of conventional implicatures* Oxford Studies in Theoretical Linguistics. Oxford: Oxford University Press.

Power, Richard. 1977. The organisation of purposeful dialogues. *Linguistics* 17. 107–152. http://dx.doi.org/10.1515/ling.1979.17.1-2.107.

Prasad, Rashmi, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi & Bonnie Webber. 2008. The Penn discourse treebank 2.0.

In *Proceedings of the sixth international conference on language resources and evaluation (LREC)*, European Language Resources Association (ELRA).

Pustejovsky, James. 1991. The generative lexicon. *Computational Linguistics* 17(4). 409–441. http://aclweb.org/anthology-new/J/J91/J91-4003.pdf.

Putnam, Hilary. 1975. The meaning of 'meaning'. In *Mind, language and reality*, 215–271. Cambridge, UK: Cambridge University Press.

Pylyshyn, Zenon W. 1984. *Computation and cognition : toward a foundation for cognitive science*. Cambridge, MA: MIT Press.

Resnik, Philip. 1996. Selectional constraints: an information-theoretic model and its computational realization. *Cognition* 61(1-2). 127 – 159. http://dx.doi.org/10.1016/S0010-0277(96)00722-6.

Resnik, Philip. 1999. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research* 11. 95–130. http://dx.doi.org/doi:10.1613/jair.514.

Richter, Frank & Manfred Sailer. 2006. Modeling typological markedness in semantics: The case of negative concord. In Stefan Müller (ed.), *The proceedings of the 13th international conference on head-driven phrase structure grammar*, 305–325. Stanford: CSLI Publications. http://cslipublications.stanford.edu/HPSG/7/.

Roberts, Craige. 2012. Information structure: Towards an integrated formal theory of pragmatics. *Semantics and Pragmatics* 5(6). 1–69. http://dx.doi.org/10.3765/sp.5.6.

Roy, Deb & Alex Pentland. 2002. Learning words from sights and sounds: A computational model. *Cognitive Science* 26(1). 113–146.

Sag, Ivan A., Timothy Baldwin, Francis Bond, Ann A. Copestake & Dan Flickinger. 2002. Multiword expressions: A pain in the neck for nlp. In *Proceedings of the third international conference on computational linguistics and intelligent text processing* CICLing '02, 1–15. London, UK, UK: Springer. http://dl.acm.org/citation.cfm?id=647344.724004.

van der Sandt, Rob. 1992. Presupposition projection as anaphora resolution. *Journal of Semantics* 9(2). 333–377.

Searle, John R. 1980. Minds, brains, and programs. *Behavioral and Brain Sciences* 3(03). 417–424. doi:10.1017/S0140525X00005756. http://dx.doi.org/10.1017/S0140525X00005756.

Shan, Chung-chieh & Chris Barker. 2006. Explaining crossover and superiority as left-to-right evaluation. *Linguistics and Philosophy* 29(1). 91–134.

Shieber, Stuart M. 1986. *An introduction to unification-based approaches to* *grammar*, vol. 4 CSLI Lecture Notes Series. Stanford, CA: CSLI Publications.

Shieber, Stuart M. 2004. *The Turing test*. Cambridge, MA: MIT Press. http://www.theturingtest.com/.

Smith, Brian Cantwell. to appear. *The age of significance, volume 1: Groundwork*. MIT Press. http://www.ageofsignificance.org.

Snow, Rion, Daniel Jurafsky & Andrew Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the Association for Computational Linguistics*, 801–808. http://www.aclweb.org/anthology/P06-1101.

Stalnaker, Robert. 1973. Presuppositions. *Journal of Philosophical Logic* 2(4). 447–457.

Stalnaker, Robert. 1979. Assertion. In Peter Cole (ed.), *Syntax and semantics 9*, 315–332. New York: Academic Press.

Stalnaker, Robert. 1998. On the representation of context. *Journal of Logic, Language and Information* 7. 3–19.

Steedman, Mark & Matthew Stone. 2006. Is semantics computational? *Theoretical Linguistics* 32. 73–89. http://dx.doi.org/10.1515/TL.2006.006.

Steels, Luc & Tony Belpaeme. 2005. Coordinating perceptually grounded categories through language. A case study for colour. *Behavioral and Brain Sciences* 28(4). 469–529. http://www.isrl.uiuc.edu/~amag/langev/paper/steels_BBS_color.html.

Stone, Matthew. 2000. Towards a computational account of knowledge, action and inference in instructions. *Journal of Language and Computation* 1. 231–246.

Stone, Matthew & Richmond H. Thomason. 2002. Context in abductive interpretation. In *Proceedings of the 6th international workshop on the semantics and pragmatics of dialogue (edilog)*, 169–176. Edinburgh, UK.

Strube, Michael. 1998. Never look back: An alternative to centering. In *Proceedings of the 36th annual meeting of the Association for Computational Linguistics and 17th international conference on computational linguistics, volume 2*, 1251–1257. http://www.aclweb.org/anthology/P98-2073.

Szabó, Zoltán Gendler. 2008. Compositionality. In Edward N. Zalta (ed.), *The stanford encyclopedia of philosophy*, Winter 2008 edn. http://plato.stanford.edu/archives/win2008/entries/compositionality/.

Turing, A. M. 1950. Computing machinery and intelligence. *Mind* 59(236). 433–460. http://www.jstor.org/stable/2251299.

Turney, Peter D. & Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research* 37. 141–188. http://dx.doi.org/10.1613/jair.2934.

Webber, Bonnie L., Matthew Stone, Aravind K. Joshi & Alistair Knott. 2003. Anaphora and discourse structure. *Computational Linguistics* 29(4). 545–587. http://dx.doi.org/10.1162/089120103322753347.

Webber, Bonnie Lynn. 1983. So what can we talk about now? In Michael Brady & Robert C. Berwick (eds.), *Computational models of discourse*, 331–371. Cambridge, MA: MIT Press.

Wilcock, Graham, Nancy Ide & Laurent Romary (eds.). 2004. *Proceedings of SENSEVAL-3, the third international workshop on the evaluation of systems for the semantic analysis of text*. http://aclweb.org/anthology-new/W/W04.

Winograd, Terry & Fernando Flores. 1986. *Understanding computers and cognition: A new foundation for design*. New York: Addison-Wesley.

Wittgenstein, Ludwig. 1953. *Philosophical investigations*. Oxford, UK: Blackwell. Translated by G.E.M. Anscombe.

Yarowsky, David. 1993. One sense per collocation. In *Proceedings of the workshop on human language technology* HLT '93, 266–271. http://aclweb.org/anthology-new/H/H93/H93-1052.pdf.

Yu, Chen & Dana H. Ballard. 2004. A multimodal learning interface for grounding spoken language in sensory perceptions. *ACM Transactions on Applied Perception* 1. 57–80. http://doi.acm.org/10.1145/1008722.1008727.