108. Semantic research in computational linguistics

Alexander Koller
University of Potsdam
Karl-Liebknecht-Str. 24-25
14476 Potsdam, Germany
koller@ling.uni-potsdam.de

Manfred Pinkal
Saarland University
Postfach 151150
66041 Saarbrücken, Germany
pinkal@coli.uni-saarland.de

March 25, 2012

108. Semantic research in computational linguistics

Computational semantics is the branch of computational linguistics that is concerned with the development of methods for processing meaning information. Because a computer system that analyzes natural language must be able to deal with arbitrary real-world sentences, computational semantics faces a number of specific challenges related to the coverage of semantic construction procedures, the efficient resolution of ambiguities, and the ability to compute inferences. After initial successes with logic-based methods, the mainstream paradigm in computational semantics today is to let the computer automatically learn from corpora. In this article, we present both approaches, compare them, and discuss some recent initiatives for combining the two.

1. Introduction

In this article, we give an overview of the state of the art in *computational semantics*, i.e. the branch of computational linguistics that deals with the processing of meaning information. The goal of computational linguistics is to develop methods for the automatic analysis and generation of natural language. Ultimately, it aims at creating computer systems that approximate the language skills of an average human speaker. But there are also more immediate and tangible real-world applications, including, for instance, information extraction systems that acquire content for a relational database from large-scale collections of business reports; spoken-language or multimodal interfaces that enable the convenient interaction of users with information systems (e.g., interfaces to healthcare websites or interactive museum guides); or machine translation systems that transfer text or speech input from a source language to a target language. All of these applications require some amount of semantic processing, although not necessarily at a very fine level of detail.

The task of semantic processing can generally be decomposed into two subproblems, namely the problem of computing a formal representation of the meaning of an expression (the *semantic construction* problem) and the task of determining the relation between such formal representations (the *inference* problem). Inference is required, for instance, when a question-answering system determines whether an answer candidate in a document collection actually answers a given question, or when an automatic summarization system must figure out to which extent two sentences describe the same event (and can therefore be compressed into one).

The classical approach to computational semantics uses some form of first-order or higher-order logic for the formal semantic representations and some form of Montague Grammar-style process for semantic construction, and solves the inference problem using programs called *theorem provers*, which can test logic formulas for entailment. This tradition of computational semantics shares its formal and conceptual framework with the mainstream of semantic research in linguistics and the philosophy of language (which we will refer to as "theoretical semantics" in this article). It strongly benefits from the wealth and detail of earlier research in these disciplines.

However, there are a number of challenges that are specific to computational semantics and call for different methods. The aim of computational semantics is to implement human language skills in computer systems – at least partially, in concrete applications. The methods that are used for this must therefore be cast into precisely formalized algorithms. One crucial aspect that drives the development of new approaches is that these algorithms must be *efficient*, even in the face of the massive *ambiguity* that arises in real-world sentences. Second, the computer systems used in computational semantics must be able to process any arbitrary sentence or discourse that can arise in the respective application scenario. The system must have *wide coverage* with respect to semantic construction, and it must also have access to the appropriate large-scale *knowledge bases* that can support the inferences that are necessary for the task at hand. It is hard to achieve all of these goals at once.

The history of computational semantics is defined by attempts to handle these problems, and we will outline some of the most prominent approaches in this article. The classical logic-based approach, which we discuss in Section 2., has made great progress in terms of processing efficiency, but still falls short of practical usability in terms of coverage and performance on the disambiguation task. As a consequence, computational semantics experienced a fundamental paradigm shift around the turn of the century; current mainstream research focuses on statistical models of word and sen-

tence meaning (Section 3.). These models have much better coverage, at the expense of the level of detail, precision, and conceptual clarity of the semantic representations. We conclude with an outlook on some novel directions of research, which are aimed at comparing and integrating the worlds of logical and statistical methods (Section 4.).

2. Computational semantics in the logical framework

Computational approaches to semantic analysis must deal with two issues. First, they must be able to determine a formal semantic representation for a given input expression; in the case of ambiguity, they also must be able to choose the contextually appropriate reading. This is called the *semantic construction* problem. Second, they must be able to relate different meaning representations to each other to detect equivalence, entailment or inconsistency between different sentences. This is the *inference* problem. Analogous problems occur in natural language generation.

Early research in artificial intelligence (AI) focused on approaches to these problems that were largely disconnected from linguistics. One influential approach was Conceptual Dependency theory (Schank 1975). Semantic representation was done without logic: Word meanings were encoded as graphs made up of a limited number of uninterpreted atomic concepts and relations (partly inspired by Fillmore's (1968) role semantics). From these, sentence representations were constructed by merging smaller graphs into larger ones using a collection of graph rewriting rules. The approach worked to some extent for sentences and texts expressing simple assertive information. However, it did not generalize easily to more complex types of information involving cardinality, quantification, negation, modality, conditional and temporal relations. These were modeled by simply attaching tags to graph edges.

Modern computational semantics started with the use of logics with well-defined model-theoretic interpretations, following the Montagovian revolution in theoretical semantics. This allowed the use of principled inference rules that were justified by soundness and completeness with respect to the model theory. Over the years, a logic-based "standard model" of computational semantics emerged: A semantic representation in first-order or higher-order logic is computed compositionally based on a syntactic analysis, and meaning relations between expressions of language are implemented using standard inference engines for logic. We refer the reader to the textbook by Blackburn & Bos (2005) for details about the standard model. Below, we sketch some of the most important methods in this paradigm.

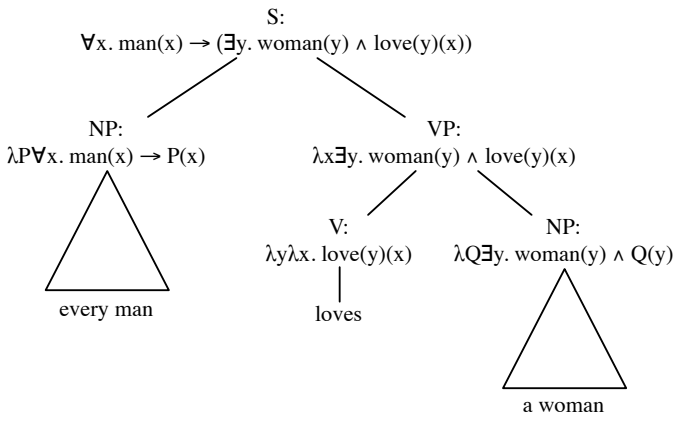

Figure 108.1: *A Montague-style derivation of a semantic representation for the sentence "Every man loves a woman."*

2.1. Semantic construction

Compositional semantics. In the early 1970s, Richard Montague presented a framework for a strictly compositional interpretation of natural-language sentences in terms of type theory, including a formal treatment of quantifier scope (Montague 1973). His work not only provided the basis for modern semantic theory, but has also had great influence on the development of computational semantics. "Standard model" computational semantics takes it as given that we can assign lambda terms to lexicon entries, combine them by traversing the parse tree bottom-up, and compute lambda terms for larger phrases compositionally out of those for smaller phrases, using functional application and beta reduction. An abbreviated example for the derivation of one reading of the sentence "every man loves a woman" is shown in Fig. 108.1.

Montague's original framework was based on an idiosyncratic version of categorial grammar. Computational linguists mostly used the formalism of *unification grammar*, i.e., phrase-structure grammar extended with feature unification, when they first started developing large-scale grammars in the 1980s. Unification grammars such as LFG (Dalrymple et al. 1995) and HPSG (Pollard & Sag 1994) offered an elegant and simple way to compute predicate-argument structures by filling the argument positions of a head with the semantic contributions of its complements using unification (see e.g. Pereira & Shieber 1987). These methods were later extended to cover more complex problems in semantic construction (Dalrymple 1999; Copestake,

Lascarides & Flickinger 2001).

Dynamic semantics. A number of "non-local" semantic phenomena turned out to be challenging for compositional semantic construction methods. For instance, anaphoric expressions establish coreferential links with antecedents at arbitrarily distant positions in the discourse; ellipsis requires us to copy parts of the antecedent's semantics into the target representation. Furthermore, structural ambiguities, e.g. of quantifier scope, undermine the tidy parallelism of syntactic and semantic structure posited by Montague Grammar.

In order to represent anaphora and, to some extent, ellipsis, the use of Discourse Representation Theory (DRT; Kamp 1981; Kamp & Reyle 1993; see article 37 *Discourse Representation Theory*) has enjoyed much attention in computational semantics. DRT conceives of meaning not in terms of truth conditions, but as context-change potential; in its standard version, it models the anaphoric potential of a text through a set of discourse referents, which are a constitutive part of the semantic representation. Dynamic Predicate Logic (Groenendijk & Stokhof 1991; see article 38 *Dynamic semantics*) is a closely related formalism that enables a compositional model-theoretic interpretation of anaphora. However, standard DRT employs a top-down, non-compositional algorithm for semantic construction. Computational applications typically combine DRS representations with higher-order logic and lambda abstraction, in order to enable a surface compositional derivation of DRSes, such as Compositional DRT (Muskens 1995) and Lambda-DRT (Kohlhase, Kuschert & Pinkal 1996).

A second issue is that computational applications for processing anaphora cannot skirt the issue of identifying the antecedent of an anaphoric expression in a given text. The possible antecedents are restricted by the hard accessibility constraints of DRT to some degree; they can be narrowed down further by modeling focusing mechanisms based on the global structure of the discourse (Grosz & Sidner 1986; Grosz, Joshi & Weinstein 1995; Asher & Lascarides 2003; see article 75 *Discourse anaphora, accessibility, and modal subordination* for more on the theoretical aspects). However, these systematic approaches to anaphoric reference leave many cases of referential ambiguity unresolved. The development of methods for *coreference resolution*, which link phrases in a given discourse that refer to the same entity, is an active field of research in computational linguistics (see e.g. Ng 2010; Stede 2011).

Quantifier storage approaches. One non-local aspect of semantic construction that has received particular attention in computational semantics is scope ambiguity. From a perspective of theoretical linguistics, the basic problem of semantic construction for sentences with scope ambiguities was essentially solved by the Quantifier Raising (QR) operation in Montague Grammar. However, QR-based approaches cannot be used effectively in computational semantics because the development of efficient parsing algorithms becomes very complicated, and it is inconvenient to develop large grammars. A second major challenge for a computational treatment of scope is that the number of readings quickly becomes very large as the sentence grows longer, and the algorithm must still remain efficient even when this happens. Algorithms for semantic construction can differ by a huge degree in this respect; recent underspecification-based methods can perform tasks that used to be completely infeasible (requiring years of computation time for one sentence) in milliseconds.

A first step towards removing the reliance on QR was *quantifier storage*, which was first proposed by Cooper (1983) and then refined by Keller (1988). The key idea in Cooper Storage was to replace Montague's treatment of scope ambiguity by a storage technique for quantifiers: Nodes in a (phrase-structure) syntax tree are assigned structured semantic representations, consisting of *content* (a λ-expression of appropriate type) and *quantifier store* (a set of λ-expressions representing noun phrase meanings). As the parse tree is traversed bottom-up, noun phrases may either be applied in situ to form new content; for the example sentence "every man loves a woman," this leads to narrow scope for the object, in essentially the same way as in the Montague-style derivation of Fig. 108.1. Alternatively, we may move the content into the quantifier store at any NP node (as shown at the node for "a woman" in Fig. 108.2) and then retrieve an item from the store and apply it to the content at the sentence node. This enables the non-deterministic derivation of different scope readings of a sentence from a surface-oriented phrase-structure grammar analysis.

A related approach was proposed by Hobbs & Shieber (1987) first, and later generalized to *Quasi-Logical Form* (QLF; Alshawi & Crouch 1992), which became a central part of SRI's Core Language Engine (CLE; Alshawi 1990): During parsing, preliminary semantic representations (QLFs) are built up, which contain the quantifier representations in the argument positions of their main predicate. In a second step, rewrite rules on the QLFs move quantifiers to their appropriate position, leaving a variable behind to bring about proper binding. For the above example, this system would first derive the QLF term $\mathsf{love}(\langle \mathsf{every}, x, \mathsf{man}\rangle, \langle \mathsf{some}, y, \mathsf{woman}\rangle)$, from which it

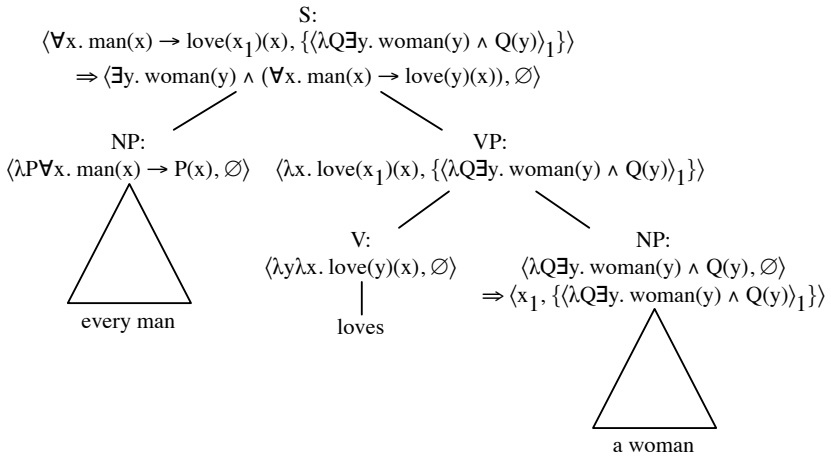

Figure 108.2: *A Cooper Storage derivation for the second reading of the sentence "Every man loves a woman."*

would derive the two readings in Fig. 108.1 and Fig. 108.2 by either scoping $\langle \mathsf{every}, x, \mathsf{man} \rangle$ over love first and then $\langle \mathsf{some}, y, \mathsf{woman} \rangle$ over the result, or vice versa.

Underspecification. As grammars grew larger in order to extend their coverage of free text, a further problem emerged. In a sentence with multiple scope ambiguities, the number of readings can grow exponentially with the number of quantifiers or other scope-bearing operators (such as negations or modal operators) in the sentence. The following sentence from Poesio (1994), which has $(5!)^2 = 14400$ readings in which each quantifier and modal operator takes scope in its own clause alone, illustrates this problem. In practice, the problem is even worse because large-scale grammars tend to make generalizing assumptions (e.g., that all noun phrases take scope) that can cause innocent-looking sentences to be assigned millions of readings.

(1) A politician can fool most voters on most issues most of the time, but no politician can fool every voter on every single issue all of the time.

The standard approach to handling massive ambiguity like this in large-scale grammars today is *underspecification*. Underspecification approaches derive a compact representation of all readings from the syntactic analysis, and proceed to single specific readings only by need, and after irrelevant

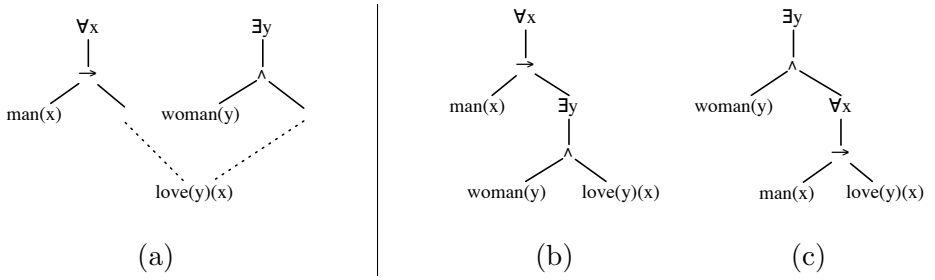

Figure 108.3: *A dominance graph for "every man loves a woman" (a), along with the two trees it describes (b,c).*

readings have been filtered out by inferences. Most underspecification approaches that are used in practice specify the parts from which a semantic representation is supposed to be built, plus constraints that govern how the parts may be combined. For instance, the *dominance graph* (Egg, Koller & Niehren 2001; Althaus et al. 2003) for the earlier example sentence "every man loves a woman" is shown in Fig. 108.3a. The parts of this graph may be combined in all possible ways that respect the dotted dominance edges, yielding the two trees in Fig. 108.3b,c. These trees represent the semantic representations that we also derived in Fig. 108.2.

Most modern large-scale grammars use underspecification in one form or another. HPSG grammars use Minimal Recursion Semantics (MRS, Copestake et al. 2005). The Glue Logic system used by many LFG grammars (Dalrymple 1999) can be seen as an underspecification approach as well; note that some recent LFG grammars also use a simpler rewriting mechanism for semantic construction (Crouch & King 2006). Underspecification-based semantic construction algorithms have also been defined for Tree Adjoining Grammars (Kallmeyer & Romero 2008; Gardent 2003). Hole Semantics (Blackburn & Bos 2005) is a particularly easy-to-understand underspecification formalism. The algorithmic foundations of underspecification have been worked out particularly well for dominance graphs, into which MRS and Hole Semantics can be translated. Dominance graphs also support powerful inference algorithms for efficiently reducing the set of possible readings without even computing them (Koller & Thater 2010). For more information about underspecification, we refer to article 24 *Semantic underspecification* in this handbook.

One popular grammar formalism in computational linguistics that follows the original Montagovian program more directly is Combinatory Categorial Grammar (Steedman 2000; Bos et al. 2004). CCG is a variant of

categorial grammar, with which it shares a very elegant and direct mapping of syntactic to semantic representations. Although this forces CCG into modeling semantic ambiguities as syntactic ambiguities, CCG can still be parsed efficiently by representing both kinds of ambiguity together in a parse chart.

2.2. Inference

The major added value of logic as a representational framework in computational linguistics is its suitability for the development of provably correct *inference procedures*. Because logical deduction is backed by the truth-conditional concept of logical entailment, it is possible to define under what conditions a deduction system is sound and complete, and to develop such systems. This is crucial when we model the processes which people perform when interpreting or producing an utterance – e.g., deriving relevant implicit information from the utterance's semantic interpretation, integrating meaning information into their knowledge, or reducing ambiguity by the exclusion of inconsistent interpretations.

For first-order predicate logic, *theorem provers* – that is, computer programs that test formulas for validity or unsatisfiability – have become efficient enough to support the practical application of deduction systems. Theoretically, first-order logic is undecidable; but theorem provers, which were originally designed for mathematical applications, have nonetheless achieved an impressive average performance on standard tasks. Currently, a variety of highly efficient off-the-shelf theorem provers are available which can be used as general purpose inference engines for natural language processing (Riazanov & Voronkov 2002; Hillenbrand 2003); there are also tools called *model builders* which can test a formula for satisfiability and build satisfying models for them (McCune 1998; Claessen & Sörensson 2003). There has been some research on theorem provers for dynamic logics, such as DRT (van Eijck, Hegueiabehere & O Nuallain 2001; Kohlhase 2000), but these provers have not been engineered as thoroughly as standard first-order provers, and it is more efficient in practice to translate dynamic logic into static logic and use the standard tools (Bos 2001). One example for an end-to-end system of the "standard model", involving semantic construction and the use of first-order theorem provers, is Bos & Markert (2005).

It is known that first-order logic is not expressive *enough* to represent genuinely higher-order or intensional phenomena in natural language, such as embedding under propositional attitudes. Some researchers have directly applied theorem provers for higher-order logic (e.g., Andrews & Brown 2006)

to natural-language inference tasks; see e.g. Gardent & Konrad (2000). However, higher-order theorem provers are much less efficient in practice than first-order provers. To compensate for this restriction, computational semantics has a strong tendency towards avoiding higher-order constructs, choosing first-order analyses in the case that semantic theory offers them as an option, and sometimes even using first-order representations to approximate phenomena that would be modeled appropriately with higher-order logic only (e.g. in the "ontological promiscuity" approach (Hobbs 1985); see also Pulman (2007) for a more recent case study).

Conversely, one can explore the use of logics that are *less* expressive than first-order logic in order to maximize efficiency, for restricted tasks and applications. *Description logics* (Baader et al. 2003) are a family of fragments of first-order logic designed to model terminological knowledge and reasoning about the membership of objects in the denotation of concepts, of which the KL-ONE system is an early representative (Brachman & Schmolze 1985). They are supported by very fast reasoning systems (Haarslev & Möller 2001; Tsarkov, Horrocks & Patel-Schneider 2007). Because they offer only restricted types of quantification, however, they have mostly been used for small domains or for specific problem, such as the resolution (Koller et al. 2004) and generation (Areces, Koller & Striegnitz 2008) of referring expressions.

Historically, another fragment of first-order logic that experienced widespread use in computational semantics is *Horn Clause Logic*, which underlies the programming language Prolog. Horn Clause Logic is limited by its inability to express true logical negation, which in Prolog must be approximated as "negation by failure": A negation $\neg A$ is considered as true iff A cannot be proved from the database. Prolog has been widely used in computational linguistics (Pereira & Shieber 1987; Blackburn & Bos 2005) – among other reasons, because it can model the full process of natural-language understanding including parsing, semantic construction, and inference uniformly, by using logical deduction. However, its use has declined due to the availability of fast theorem provers and of NLP software libraries for mainstream programming languages, as well as the growing importance of numeric processing for statistical methods (see Section 3. below).

A final challenge is the modeling of *common-sense reasoning*. Inference steps needed in the process of natural-language understanding may be valid only in the typical case, and thus their results can be overwritten, if more specific contradicting information is added. Knowing that Tweety is a bird allows us to infer that Tweety can fly; adding the information that Tweety is a penguin forces us to revise the derived information. This raises the infer-

ence task to another level of difficulty. Standard predicate-logic deduction just adds information, extending the knowledge base in a monotonic way, and has no mechanism for knowledge revision. Several alternative logic frameworks supporting *non-monotonic deduction* have been proposed, most importantly default logic (Reiter 1980), abductive reasoning (Lipton 2001), and auto-epistemic logic (Moore 1985). Of these, default logic (particularly in the context of SDRT, Asher & Lascarides 2003) and abductive reasoning (i.e., reasoning from observations to the best explanation, particularly in the text understanding framework of Hobbs et al. 1993) have become influential in computational semantics.

2.3. Knowledge resources for computational semantics

So far, we have sketched how logic-based semantic representations can be automatically built, and how inferences with these representations can be efficiently computed using theorem provers. To make real use of these systems, we need wide-coverage knowledge bases, which provide us with facts about the meaning of predicates and constants. Consider the following examples:

(2) a. Socrates is a man.
All men are mortal.

b. Socrates is mortal.

(3) a. Bill bought a convertible.

b. Bill bought a car.

(4) a. John went shopping.

b. Did he bring enough money?

(5) a. Which genetically caused connective tissue disorder has severe symptoms and complications regarding the aorta and skeletal features, and, very characteristically, ophthalmologic subluxation?

b. Marfan's is created by a defect of the gene that determines the structure of Fibrillin-11. One of the symptoms is displacement of one or both of the eyes' lenses. The most serious complications affect the cardiovascular system, especially heart valves and the aorta.

The range of inferences that we can draw from semantic representations alone without any additional knowledge is very limited. We may be able to

do simple syllogistic reasoning as in (2); but the vast majority of intuitively plausible inferences require additional background knowledge. The inference in (3) requires the lexical-semantic information that convertibles are cars; to make sense of the dialogue sequence (4), we must have common-sense knowledge about what happens when people go shopping. The example (5) gives an impression of the complex inferences that a natural-language interface to a medical information system must be able to draw, and of the kind and amount of domain knowledge which is required for this.

Theorem provers support such inferences if they have access to logical knowledge bases which contain this information. Unfortunately, the amount of knowledge which may in principle be relevant for inference is huge, and so hand-crafting comprehensive knowledge bases is a very expensive and cumbersome task. In general, coverage is at present a much harder problem for logic-based inference than efficiency.

Certain types of lexical-semantic knowledge are provided by WordNet (Fellbaum 1998), with impressively wide coverage for English and a variety of other languages (Vossen 2004; Hamp & Feldweg 1997). WordNet distinguishes various *senses* of each word in the lexicon, groups them into *synsets* of synonymous senses, and specifies different semantic relations between these synsets, such as hyponymy (subsumption) and meronymy (part-of). Other resources, such as FrameNet (Baker, Fillmore & Cronin 2003) and VerbNet (Kipper-Schuler 2006) contribute information about described situation type, thematic roles, and alternative syntactic realization patterns for lexical expressions, in particular verbs. For a more detailed discussion of lexical-semantic resources and methods for acquiring lexical-semantic knowledge, see article 110 *Semantics in computational lexicons* in this handbook.

However, there are many kinds of knowledge which are not formalized in WordNet and related resources. Examples are script-like information as in the supermarket example above, or stereotypical properties of concepts such as the ability of birds to fly. While it can be debated whether such knowledge should be packaged into the lexicon as components of word meaning or whether it is non-linguistic common-sense knowledge about the world, there is no doubt that such knowledge is necessary for full text understanding; see also article 32 *Word meaning and world knowledge*. Because of the magnitude of the task, few attempts have been made to comprehensively axiomatize world knowledge by hand. One notable exception is the Cyc project (Lenat 1995); its aim is to hand-axiomatize enough knowledge that an automated system could then learn more knowledge from natural language text. At the time of writing, Cyc contains five million assertions about several hundreds of thousands of concepts, and has recently become freely available

for research purposes as ResearchCyc (Matuszek et al. 2006). Because it aims at massive coverage, Cyc is a rather heavyweight system. It is also optimized for fine-grained reasoning on the conceptual level, rather than for natural-language processing and inference. For instance, Cyc distinguishes between 23 different senses of spatial "in", all of which have different axioms. This degree of ambiguity causes substantial problems for ambiguity resolution, and therefore Cyc can be of only limited use for language-related semantic processing tasks.

3. Statistical methods in computational semantics

The "standard model" we have presented so far enables us to compute logic-based meaning representations, which can be used by theorem provers to draw inferences. This works efficiently and with impressive accuracy, if hand-crafted grammars and knowledge resources are available that cover all information that is required for the interpretation. However, logic-based semantic methods run into a number of fundamental problems:

- Natural language is extremely ambiguous, and understanding of utterances implies *ambiguity resolution*: the determination of a contextually appropriate reading. Underspecification methods enable an efficient representation of semantic ambiguity, but they make no attempt to resolve it. A particular challenge is word-sense disambiguation, because lexical ambiguity comprises a large and extremely heterogenous class of individual phenomena.

- Modeling *inference* for open-domain text understanding with logic requires us to encode a huge amount of *world knowledge* in logic-based knowledge bases, as we have discussed. Such knowledge bases are not available; even large-scale efforts at manual resource creation like WordNet and Cyc have coverage problems.

- Despite the progress in hand-crafting large grammars with semantic information, many free-text sentences cannot be completely analyzed by these grammars: Knowledge-based grammar processing still faces *coverage* problems. Because traditional algorithms for semantic construction can only work on complete parses, no semantic representations can be computed for these sentences. That is, semantic construction procedures are not *robust* to coverage problems.

As a consequence, logic-based methods for computational semantics have not been very successful as part of applications in language technology. In

retrospect, this is not entirely surprising. As we know from psycholinguistics, human language use and language learning are not purely categorical processes, but are strongly influenced by statistical expectations. This awareness of preferences speeds up the interpretation process, and in particular enables people to disambiguate expressions effortlessly and in real time. In the nineties, computational linguistics as a whole experienced a "statistical turn". The basic idea behind *statistical* (or, more generally: *data-intensive*) methods is to let a computer system discover statistical regularities in language use in large text corpora (or even the entire Internet), and then exploit them to analyze previously unseen texts or discourses. Because the system learns from data, this approach is also called *machine learning*. The idea was first worked out in the area of automatic speech recognition, and was later applied successfully to syntactic parsing. Today, it is the dominant paradigm in semantic research in computational linguistics as well.

Logic-based and data-intensive approaches are complementary in their strengths and weaknesses. Data-intensive approaches typically take a very shallow view on language from a linguistic point of view. The models they build of natural-language expressions have little to say about issues such as the logical structure of a sentence. They are typically not related to logic, perhaps not even based on a full syntactic parse of the sentence, and the inferences they support are judged to a standard of practical usefulness rather than logical correctness. However, these models can automatically learn information that is implicit in large text corpora, achieving wide coverage with comparatively little human effort. This gives us tools for addressing the coverage problems listed above. Furthermore, the knowledge provided by statistical methods is soft preferential knowledge, in terms of frequencies or probability estimates, which support disambiguation tasks well, and may even be appropriate for modeling defeasible common-sense knowledge.

We assume that a reader of this handbook is less familiar with machine learning techniques than with logic-based approaches. Therefore, the presentation in this section will be more basic than in the rest of the article. We try to give a flavor of statistical methodology, and at the same time provide a short overview of three prominent areas of research in computational semantics: *word-sense disambiguation*, *semantic role labeling*, and the modeling of *semantic relatedness*. These topics and other research in statistical computational linguistics are discussed at greater length in the standard textbooks by Jurafsky & Martin (2008) and Manning & Schütze (1999).

3.1. Word-sense disambiguation: Basics in statistical semantics

Word-sense disambiguation. Lexical ambiguity is pervasive in natural languages, and the determination of the contextually appropriate word meaning, known as *word-sense disambiguation* (WSD), has long been recognized as a hard problem in computational linguistics. Over fifty years ago, Yehoshua Bar-Hillel argued in his famous report on automatic translation (Bar-Hillel 1960) that "a translation machine should not only be supplied with a dictionary but also with a universal encyclopedia". For example, to appropriately translate "the box was in the pen" into another language, a computer program must know about typical sizes and shapes of boxes and pens to conclude that "pen" is used in the "enclosure" sense rather than the "writing implement" sense. Bar-Hillel commented that any attempt to solve this problem with knowledge-based methods was "utterly chimerical and hardly deserves any further discussion".

We can get a first grasp on the problem of WSD from lexical-semantic resources that define an inventory of possible word senses for each word of a language. Two such resources for English are WordNet (Fellbaum 1998) and Roget's Thesaurus (Chapman 1977). WordNet lists Bar-Hillel's two senses for the noun "pen", along with the senses "correctional institution" and "female swan". English WordNet contains about 29,000 polysemous words, each of these with 3 different senses on average. Neither of these resources contains the information (e.g., box and pen sizes) that is necessary to reliably determine the sense in which a word was used in a given sentence.

Machine learning and WSD. WSD in early large-scale NLP systems was typically done by hand-written rules that were developed specifically for the application and the relevant domain (see e.g. Toma 1977; Hobbs et al. 1992; Koch, Küssner & Stede 2000). Early attempts at defining generic rule-based methods for WSD are (Wilks 1975; Hirst & Charniak 1982). The weighted abduction approach by Hobbs et al. (1993) supported a generic, logic-based mechanism for disambiguation, but suffered from efficiency issues and required a large hand-coded knowledge base to work.

By contrast, statistical approaches attempt to solve the WSD problem by automatically learning the choice of the appropriate word sense from text corpora. The fundamental idea of such a machine learning approach is to build a *classifier*, which for each occurrence of a word w in some context c determines the sense s of this occurrence of w. This classifier is automatically learned from observations in a text corpus, in which each occurrence of each word has been manually *annotated* with its sense; one corpus that has been

annotated with WordNet senses is the SemCor corpus (Landes, Leacock & Tengi 1998).

Machine learning approaches in which the training data is assumed to be annotated in this way are called *supervised*. The context c is usually approximated by a collection f of *features* that can be automatically extracted from the text. The machine learning system is trained on the annotated training corpus, i.e., it observes the pairs of sense annotations and extracted feature instantiations, for all instances of w, and derives from these data a *statistical model* of the correlation between feature patterns and word senses. The system can then be executed on unseen, unlabeled documents to label each word token automatically with its most plausible word sense, given the feature information extracted from the token's context.

Different approaches to statistical WSD are distinguished by the features they use and the machine learning method. The simplest choice for the features is to use *context words*. For instance, Yarowsky's (1995) system automatically identified the context words *life*, *animal*, and *species* as strong statistical indicators of the biological sense of the target word *plant*, and *manufacturing*, *equipment*, and *employee* as strong indicators of its "factory" sense. To address the disambiguation problem in a systematic way, we might determine the 2000 most frequent content words $w_1, \ldots, w_{2000}$ in the corpus. For any occurrence of a target word w, we could then assign the feature f_i the value 1 if the context word w_i occurs within a window of n words (for $n = 5, 10, 30, \ldots$) before or after w, and 0 otherwise. Approaches to machine learning differ substantially in the exact way in which they make use of the feature information to solve their classification task. For an overview of different approaches to machine learning, see Mitchell (1997), Russell & Norvig (2010), or Witten, Frank & Hall (2011).

Modeling context. The choice of features is a crucial part of designing a successful machine-learning-based WSD system: Since only the information encoded in features is visible to the machine learning system, the design of the feature space entails a decision about the information made available to the disambiguation process. The simplistic view of context as a set of co-occurring content words can be refined by adding more features representing different kinds of information. We can, e.g., include precedence information (does the context word occur to the left or to the right of the target?) or use positional information (does the context word occur as the immediate left and right neighbor of the target instance?). We may enrich the context information with linguistic information provided by available, reasonably

efficient and reliable analysis tools: Using lemma and part-of-speech information is standard; adding syntactic information through shallow syntactic parsing is another frequently chosen option.

In principle, it would be desirable to use deeper and more informative context features than this. However, extracting such features tends to be expensive (it may again require large hand-crafted grammar and knowledge resources) or extremely noisy, if it can be done at all. Nevertheless, even the simple context-word approach can capture a remarkable amount of information on different levels of contextual knowledge and their interaction, however. Consider the following example; the common noun *dish* is ambiguous between a "plate" and a "food" sense.

(6) Yesterday night we went to a restaurant; I ordered an expensive dish.

The verb *order* contributes selectional preference information for its object position, and *restaurant* provides relevant topical or situational information. The two pieces of contextual evidence interact in a way that supports a strong prediction of the "food" sense of *dish*. Explicit modeling of the inference process leading to the correct reading would require very specific common-sense knowledge. A simple statistical model is able to predict the effects of this interaction with good results, based on the simple co-occurrence counts of these context words.

Measuring system performance. A machine learning system generalizes from observations without human intervention, and typically only has access to shallow features. The goal in designing such a system is therefore never that it is infallible. Instead, the aim is to balance maximum coverage with making relatively few mistakes. In order to examine the quality of such a system, one *evaluates* it on data for which the correct responses are known. To this end, one splits the manually annotated corpus into two separate portions for training and testing. The machine learning system is trained on the training corpus, and then used to classify every single word in the test corpus. One can, e.g., compute the *accuracy*, i.e., the percentage of word tokens in the test corpus for which the system computed the annotated word sense. This makes it possible to compare the performance of different systems using well-defined measures.

WSD has been an active field of research in computational semantics for the last two decades. An early successful WSD system was presented by Yarowsky (1992). One can get a sense of the current state of the art from the results of the "Coarse-grained English All Words Task" (Navigli, Litkowski

& Hargraves 2007), a competition advertised for the SemEval 2007 workshop. This task consists in annotating the words in a given corpus with a coarse-grained sense inventory derived from WordNet. The random baseline, which assigns each word a random sense, achieved an accuracy of about 52% on this task. Because one sense of a word is often strongly predominant, the simple policy of assigning the instances of each word always its globally most frequent sense achieves 79% accuracy on the dataset, which is a much more demanding baseline for WSD systems. On the other hand, the *inter-annotator agreement*, i.e. the percentage of tokens for which the human annotators agreed when creating the SemEval 2007 test data was 94%. This is usually taken to indicate the upper bound for automatic processing. The best-performing WSD system in the 2007 competition reached an accuracy of about 88%, beating the most-frequent-sense baseline significantly. Although the WSD system does not reach human performance yet, it does come rather close. Recent overview articles about WSD are McCarthy (2009) and Navigli (2009).

3.2. Semantic role labeling: The issue of feature design

Semantic roles. WSD algorithms predict atomic meaning representations for lexical items in a text. In order to compute a semantic representation for an entire sentence, we must compose these lexical meaning representations into larger structures. Recent research has focused on the computation of *predicate-argument structures* as the first step in the semantic composition process. This is not a trivial problem, because the syntactic realization of semantic argument positions is subject to considerable variation. The central theoretical concept relating syntactic complements and semantic arguments is that of a *semantic role*. The practical task of computing predicate-argument structures is called *semantic role labeling (SRL)*.

The first issue that one needs to address in SRL is what inventory of semantic roles to use. Fillmore (1968) originally proposed a small universal set of *thematic roles*, such as "agent", "patient", "recipient", etc.; see also article 18 *Thematic roles*. This assumption has turned out to be impractical for wide-coverage lexicons, because it is impossible to map the variation and conceptual wealth of natural-language semantics cleanly to such a small role inventory. For example, in the description of a commercial transaction in (7) does the subject "China Southern" fill the *agent* role (since it pays money to Airbus), or the *recipient* role (since it receives planes from Airbus)?

(7) China Southern buys five A380 planes from Airbus.

FrameNet and PropBank. Research on SRL in computational linguistics therefore tends to use semantic role inventories which do not assume universal semantic roles, either in FrameNet (Fillmore & Baker 2010) or in PropBank style (Palmer, Gildea & Kingsbury 2005).

FrameNet organizes the lexicon into *frames*, which correspond to situation types. The FrameNet database currently contains about 12,000 lexical units, organized into 1,100 frames. Semantic roles (called *frame elements*) are then assumed to be specific to frames. For example, the verbs "replace" and "substitute" (as "exchange" and "switch", and the nouns "replacement" and "substitution") evoke the REPLACING frame; core roles of this frame are *Agent*, *Old*, and *New*. The names of these roles are meaningful only within a given frame. This makes the role concept of FrameNet rather specific and concrete, and makes it possible to annotate role information with high intuitive confidence. Two major corpora that have been annotated with FrameNet data are the Berkeley FrameNet Corpus (Baker, Fillmore & Cronin 2003) and the SALSA Corpus for German (Burchardt et al. 2006). An example that illustrates how different verbs can induce the same predicate-argument structure in FrameNet is shown in (8).

(8) a. [$_{Agent}$ Lufthansa] is replacing$_{REPLACING}$ [$_{Old}$ its 737s] [$_{New}$ with Airbus A320s].

b. [$_{Agent}$ Lufthansa] is substituting$_{REPLACING}$ [$_{New}$ Airbus A320s] [$_{Old}$ for its 737s].

The PropBank approach proposes an even more restricted notion of a semantic role. PropBank assumes specific roles called *arg0, arg1, arg2, ...* for the senses of each verb separately, and thus only relates syntactic alternations of the same predicate to each other. Role label identity between complements of different verbs is not informative, as the examples in (9) illustrate:

(9) a. [$_{Arg0}$ Lufthansa] is *replacing* [$_{Arg1}$ its 737s] [$_{Arg2}$ with Airbus A320s].

b. [$_{Arg0}$ Lufthansa] is *substituting* [$_{Arg1}$ Airbus A320s] [$_{Arg3}$ for its 737s].

Of the two approaches, FrameNet is the more ambitious one, in that it supports a more informative encoding of predicate-argument structure than PropBank role labeling. However, annotating a corpus with PropBank roles

is easier and can be done much more quickly than for FrameNet. As a consequence, exhaustively annotated corpora are available for several languages; the English PropBank corpus is a version of the Penn Treebank (Marcus, Santorini & Marcinkiewicz 1993) in which the arguments of all verb tokens are annotated with semantic roles.

Semantic role labeling systems. The SRL task for FrameNet or PropBank can be split into two steps. First, because roles are specific to FrameNet frames or PropBank verb senses, we must determine the frame or sense in which a given verb token is being used. This is a WSD task, and is usually handled with WSD methods.

Assuming that each predicate in the sentence has been assigned a frame, the second step is to identify the arguments and determine the semantic roles they fill. The first system that did this successfully was presented by Gildea & Jurafsky (2002) – originally for FrameNet, but the approach has also been adapted for PropBank (see Palmer, Gildea & Kingsbury 2005). It uses a set of features providing information about the target verb, the candidate role-filler phrase, and their mutual relation. Most of the features refer to some kind of syntactic information, which is typically provided by a statistical parser. Features used include the phrase type (e.g., NP, PP, S); the head word of the candidate phrase; the voice of the head verb; the position of the candidate phrase relative to the head verb (left or right); and the path between candidate phrase and head verb, described as a string of non-terminals. Based on this information, the system estimates the probability that the candidate phrase stands in certain role relations to the target predicate, and selects the most probable one for labeling.

Feature design and the sparse data problem. The Gildea & Jurafsky system (as well as more recent approaches to WSD) uses syntactic information, but only looks at a handful of specific features of a syntax tree; much of the available information that the syntax tree contains is hidden from the machine learning system. Even a human annotator would sometimes have difficulties in predicting the correct semantic roles given just this information. If the SRL system assumes that it has full syntactic information anyway, why does it ignore most of it? Couldn't its performance be improved by adding additional features that represent more detailed syntactic information?

This question touches upon a fundamental challenge in using statistical methods, the *sparse data problem*. Every statistical model is trained from

a limited set of observations in the corpus, and is expected to make accurate predictions on unseen data. The reliability of these predictions depends greatly on the size of the training corpus and the number of features. If we add features, we increase the number of possible combinations of feature-value pairs, i.e., the size of the *feature space*. For a given size of the training data, this means that certain feature-value combinations will be seen only once or not at all in training, which implies that the estimate of the statistical model becomes too inaccurate to make good predictions. *Smoothing* and *back-off* techniques can improve the performance of systems by assigning some kind of positive probability to combinations that have never or rarely been seen in training. But even these methods ultimately reduce the system's predictions on rare events to educated guesses.

The trade-off between informativity and occurrence frequency is one of the major challenges to statistical NLP. Sensible *feature design*, i.e. selecting a feature set which provides maximal information while keeping the feature space manageable, is a task where combined technical and linguistic expertise is required.

Further reading. For a more detailed introduction to standard SRL, we refer the reader to Jurafsky & Martin (2008). Just as for WSD, a good starting point to get a sense of the state of the art is to look at recent SRL competitions (Carreras & Marquez 2004; Carreras & Marquez 2005; Hajic et al. 2009).

3.3. Semantic relatedness: Minimizing supervision

All data-intensive methods we have described so far are supervised methods: They require manually annotated corpora for training. The sparse data problem we just mentioned arises because annotating a corpus is costly and time-intensive, which limits the size of available corpora (Ng 1997). Conversely, this means that supervised methods can only be used with relatively inexpressive features.

Data expansion methods attempt to work around this problem by partially automating the annotation process. These methods train an initial model on a small amount of manually annotated *seed data*; use this model to identify instances in a large un-annotated corpus whose correct annotation can be predicted with high confidence; add the automatically annotated instances to the corpus; use the extended corpus to retrain the model; and then repeat the entire process in a "bootstrapping cycle". Such *semi-supervised* methods have been quite successful in early WSD systems (Yarowsky 1995),

and more recently also for SRL (Fürstenau & Lapata 2009). Another strategy of reducing annotation effort is known as *active learning*: A model is trained on a seed corpus, but it is then used for the identification of low confidence instances. Specifically annotating these low-confidence cases will usually add more relevant information than annotating large numbers of cases that the learning system already "is certain about" (Settles 2009).

Learning from unannotated text. A class of popular approaches take this idea one step further, by requiring no manual annotation of training corpora at all. They are in particular attractive for the acquisition of world knowledge and lexical knowledge, because these tasks require large amounts of training data to achieve thematic coverage. An early representative of this tradition is Hearst (1992), who learned hyponym relations between words by considering occurrences of patterns like "an X such as Y". If this string occurs significantly more frequently than would be expected from the frequencies of X and Y alone, the system infers that Y is a hyponym of X. The approach was later generalized to other semantic relations, e.g. to meronymy (Girju, Badulescu & Moldovan 2006) and certain semantic relations between verbs (Chklovski & Pantel 2004).

Although such pattern-matching approaches sometimes find incorrect pairs (the top Google hit for the above pattern at the time of writing was "a fool such as I"), their great advantage is that they can operate on raw text and require no annotation effort. They can even be used on the entire Web, with certain caveats that are discussed e.g. by Keller, Lapata & Ourioupina (2002), and therefore achieve huge lexical coverage. However, these approaches still require human intervention in the specification of the patterns for which the corpus should be searched. To alleviate the problem, Ravichandran & Hovy (2002) present a bootstrapping approach that can simultaneously learn patterns and instances of the relation.

Distributional models. A more radical approach to the problem of learning knowledge from unannotated corpora is offered by methods which automatically learn from co-occurrence frequencies what expressions are *semantically similar* and do not even require the specification of search patterns. The basic idea, known as the *Distributional Hypothesis*, is that words with similar meaning tend to occur together with the same words. The basic insight can be traced back to the 1950s (Harris 1951). The catchy phrase "You shall know a word by the company it keeps" is due to Firth (1957).

In its basic version, distributional semantics approximates word meaning

	factory	flower	tree	plant	water	fork
grow	15	147	330	517	106	3
garden	5	200	198	316	118	17
worker	279	0	5	84	18	0
production	102	6	9	130	28	0
wild	3	216	35	96	30	0

Figure 108.4: *Some co-occurrence vectors from the British National Corpus.*

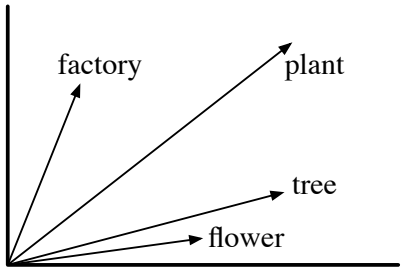

Figure 108.5: *Graphical illustration of co-occurrence vectors.*

through counts of context words occurring in the neighborhood of target word instances. Take, as in the WSD example above, the n (e.g., 2000) most frequent content words in a corpus as the set of relevant context words; then count, for each word w, how often each of these context words occurred in a context window of n before or after each occurrence of w. Fig. 108.4 shows the co-occurrence counts for a number of target words (columns), and a selection of context words (rows) obtained from a 10% portion of the British National Corpus (Clear 1993).

The resulting frequency pattern encodes information about the meaning of w. According to the Distributional Hypothesis, we can model the semantic similarity between two words by computing the similarity between their co-occurrences with the context words. In the example of Fig. 108.4, the target *flower* co-occurs frequently with the context words *grow* and *garden*, and infrequently with *production* and *worker*. The target word *tree* has a similar distribution, but the target *factory* shows the opposite co-occurrence pattern with these four context words. This is evidence that trees and flowers are more similar to each other than to factories.

Technically, we represent each word w as a vector in a high-dimensional

vector space, with one dimension for each context word; the value of the vector at a certain dimension v is the co-occurrence frequency of w with v. We define a similarity measure between words based on their respective vector representations. A commonly used measure is the *cosine* of the angle between the two vectors, which can be computed easily from the co-occurrence counts. It assumes the value 1 if the vectors' directions coincide (i.e., the proportions of their context-word frequencies are identical), and 0 if the vectors are orthogonal (i.e., the distributions are maximally dissimilar). In the 5-dimensional word-space of our example, we obtain a high distributional similarity between the targets *tree* and *flower* (cosine of 0.752, representing an angle of about 40°), and a low similarity (cosines of 0.045 and 0.073, respectively, representing angles of about 85°) between either of the two and the target *factory*, as illustrated in Fig. 108.5.

Discussion. Standard distributional models offer only a rough approximation to lexical meaning. Strictly speaking, they do not model semantic similarity in terms of the "likeness" of lexical meaning, but a rather vague notion of "semantic relatedness", which includes synonymy, topical relatedness, and even antonymy (Budanitsky & Hirst 2006). This is in part because the notion of context is rather crude. A deeper problem is that textual co-occurrence patterns provide essentially incomplete and indirect information about natural-language meaning, whose primary function is to connect language to the world. We will come back to the issue in Section 4.4..

Nevertheless, distributional approaches to semantics are attractive because they are *fully unsupervised:* They do not require any annotation or other preparatory manual work, in contrast to the supervised and semi-supervised methods sketched above. Therefore, one gets wide-coverage models almost for free; the only prerequisite is a text corpus of sufficient size. In particular, distributional models can be easily obtained for languages for which no lexicon resources exist, and adapted to arbitrary genre-specific or domain-specific sub-languages. They have proven practically useful for several language-technology tasks. Examples are word-sense disambiguation (McCarthy & Carroll 2003; Li, Roth & Sporleder 2010; Thater, Fürstenau & Pinkal 2011), word-sense induction (Schütze 1998), information retrieval (Manning, Raghavan & Schütze 2008), and question answering (Dinu 2011).

Contextualization. An obvious flaw of the basic distributional approach is that it counts *words* rather than *word senses*. Because of lexical ambiguity, the distributional pattern of a word is therefore a mixture of the distribu-

tional patterns of its individual senses. While ideally each occurrence of *plant* should be either highly similar to *factory* or to *tree*, the model will uniformly assign them a value that is somewhere in between, as indicated by the *plant* arrow in Fig. 108.5.

Dealing with this problem is tricky; adding word-sense information to the corpus is not a real option, since this would throw us back to supervised methods, requiring expensive manual annotation. An approach that has received recent attention is to *contextualize* a target instance, by modifying its meaning with information provided by its actual context words (using algebraic operations on the respective vector representations, such as addition or component-wise multiplication). The effect is that the vector of of an occurrence of *plant* in the context of *water* is "pulled" towards the vector of *tree*, thus modeling a preference for the botanical word sense (Schütze 1998; Mitchell & Lapata 2008; Erk & Padó 2008; Thater, Fürstenau & Pinkal 2010).

Refining distributional similarity measures. The basic approach of distributional similarity modeling has been refined in various ways. Different alternative measures for the association of a target word with the context and for computing similarity between a pair of target words have been proposed. Recent work makes frequent use of "hidden variable" techniques (Dinu & Lapata 2010), which were originally developed for Information Retrieval (Landauer, Foltz & Laham 1998; Schütze 1998). Syntactic information has been added to the model in different ways in order to achieve a finer-grained analysis of distributional similarity, e.g. in the contextualization approaches of Erk & Padó (2008) and Thater, Fürstenau & Pinkal (2010). Lin & Pantel (2001) present an interesting syntax-enriched variant of distributional semantics, which generalizes to *multiword relational patterns*. Their system can discover, for example, that "X solves Y" and "X finds a solution to Y" are paraphrases, based on the fact that the frequency distributions of fillers for the X and Y slots are similar. Work on contextualization and syntactic refinement has initiated a discussion about compositionality in distributional semantics – that is, methods for computing distributional representations for complex expressions from distributional information about individual words (Mitchell & Lapata 2008; Grefenstette & Sadrzadeh 2011).

Unsupervised methods for semantic relatedness are currently a very active field of research, and it will be interesting to see how the area will develop in the future. For a recent detailed overview over the state of the art, see Turney & Pantel (2010).

4. Current developments

We conclude our overview with a discussion of some recent developments in computational semantics. We will look at a general evaluation scheme for computational semantics systems (*textual entailment*, Section 4.1.), an approach to shallow logic-based inference that may be a starting point for bringing logic back into broad-coverage computational semantics (*natural logic*, Section 4.2.), approaches to the automated learning of wide-coverage semantic construction resources (Section 4.3.), and approaches to learning data-intensive models that ground word meaning directly in the real world (Section 4.4.). Common to all of these approaches is that they are in their early stages, and there is no telling whether they will be successful in the long run; but they are all promising, active research areas, which may contribute to bringing knowledge-based and data-intensive semantics closer together in the future.

4.1. Textual entailment

As we have argued above, *inference* is the touchstone for computational semantics. It is the capability of supporting inferences that makes semantic processing potentially useful in applications. The performance of a semantic processing method is therefore strongly dependent on its performance in modeling inference. While the evaluation of WSD or SRL systems is straightforward, the question of how to assess a system's performance on the more global task of modeling inference appropriately has long been an open issue in the computational semantics community.

FraCaS. A first step in this direction was the creation of a test suite of inference problems by the FraCaS project in the 1990s (Cooper et al. 1996). Each problem consisted of a premise and a candidate conclusion (phrased as a yes/no question), plus information about their logical relation; systems could then be evaluated by making them decide the logical relation between the sentences and comparing the result against the gold standard. Two of the about 350 examples are shown below:

(10) *P:* ITEL won more orders than APCOM
Q: Did ITEL win some orders?
→ *YES*

(11) *P:* Smith believed that ITEL had won the contract in 1992
H: Had ITEL won the contract in 1992?

→ *UNKNOWN*

The FraCaS testsuite was hand-crafted to cover challenging semantic phenomena (such as quantifiers, plural, anaphora, temporal reference, and attitudes), while minimizing the impact of problems like syntactic complexity and word-sense ambiguity. This made it a valuable diagnostic tool for semanticists, but it also limited its usefulness for the performance evaluation of semantic processing systems on real-world language data, in which syntactic complexity is uncontrolled and word-sense ambiguity is prevalent.

RTE. A milestone in the development of an organized and realistic evaluation framework for natural-language inference was the *Recognizing Textual Entailment (RTE)* challenge initiated by Ido Dagan and his colleagues in the PASCAL network (Dagan, Glickman & Magnini 2006). The RTE dataset consists of pairs of sentences (a *text* T and a *hypothesis* H) derived from text that naturally occurred in applications such as question answering, information retrieval, and machine translation, plus an annotation specifying whether each sentence pair stands in an "entailment" relation.

In RTE, "entailment" is defined as follows:

> "We say that T entails H if the meaning of H can be inferred from the meaning of T, as would typically be interpreted by people. This somewhat informal definition is based on (and assumes) common human understanding of language as well as common background knowledge." (Dagan, Glickman & Magnini 2006)

For instance, the following sentence pair from the second RTE challenge (Bar-Haim et al. 2006) is in the entailment relation.

(12) *T:* In 1954, in a gesture of friendship to mark the 300th anniversary of Ukrainian union with Russia, Soviet Premier Nikita Khrushchev gave Crimea to Ukraine.
H: Crimea became part of Ukraine in 1954.
→ *YES*

Crucially, "textual entailment" is not a logical notion; it is a relation between textual objects. The above definition has been criticized for its vagueness and for its insufficient theoretical grounding, in that it blurs the distinction between logical entailment, common-sense inference, presupposition, and conversational implicature (Zaenen, Karttunen & Crouch 2005). However, it was deliberately intended as a specification of a pre-theoretic

concept, which is neutral with respect to any particular semantic theory. Determining textual entailment seems to be a quite natural task for people, and is motivated from applications (Manning 2006); one effect of this is that annotators agree quite well on RTE-style entailment judgments (Bos & Markert 2005), whereas agreement on the precise and theoretically well-motivated distinctions tends to be difficult. For instance, it is doubtful whether the following logical reformulation of (12) is logically or analytically sound, given the semantics of the predicates and the sortal information about the argument fillers.

(13) give-to(Khrushchev, Crimea, Ukraine)
$\models$ become-part-of(Crimea, Ukraine)

However, (12) is still a clear case of entailment in the sense of the above definition.

For the RTE challenge, two datasets were created, intended as training and evaluation corpus, respectively. They contained 800 sentence pairs each, annotated with respect to entailment. Participating systems could be tuned on the training corpus, which was made available several weeks in advance. For evaluation, they had to automatically determine for the unseen sentence pairs in the test corpus whether they stand in the entailment relation or not. Performance was measured in terms of accuracy, i.e. the percentage of sentence pairs on which the system's judgment agreed with the annotation in the test corpus. The RTE challenge has established itself as a yearly event, with new datasets every year, and some variation in dataset and evaluation design.

RTE systems. The simplest reasonable baseline system for textual entailment recognition is one which checks for word overlap between T and H: It takes the percentage of words in the second sentence that occur in the first sentence as well as an indicator for entailment, and returns "yes" if this percentage exceeds a certain threshold. Such a system might classify (12) as a positive entailment case because "Crimea", "Ukraine", "in", and "1954" occur both in H and T. A word-overlap system typically gets about 60% of the sentence pairs right, depending on the particular instance of RTE. The accuracy can be increased by combining word overlap with semantic similarity measures (Jijkoun & de Rijke 2005; Glickman, Dagan & Koppel 2005), but the potential for such purely shallow and knowledge-lean improvements seems to be limited.

Pure logic-based systems, located at the other end of the spectrum, have completely failed at the RTE task, which was shown impressively by Bos & Markert (2005). They applied a state-of-the-art logic-based system along the lines of Section 2.. Where this system claims entailment for a given sentence pair, its judgment is quite reliable; but because it only claimed entailment for less than 6% of the pairs, it gave far fewer correct answers overall than a simple word-overlap model. This demonstrates the severity of the knowledge bottleneck in logic-based semantics, which we mentioned above.

A standard system architecture that emerged from the experiences in RTE combines syntactic and semantic knowledge with machine learning technology. A typical inventory of knowledge types includes syntactic dependency information contributed by knowledge-based or statistical parsers plus lexical semantic information taken from WordNet or distributional models, potentially complemented by semantic role information (FrameNet, PropBank) and lexical semantic and world knowledge from other sources (e.g., DIRT (Lin & Pantel 2001), VerbOcean (Chklovski & Pantel 2004), or the YAGO knowledge base (Suchanek, Kasneci & Weikum 2008)). This information is used as input to a supervised machine-learning system, which learns to predict the entailment status of a sentence pair from features indicating structural and semantic similarity. Systems enhanced with linguistic knowledge in such ways typically outperform the purely overlap-based systems, but only by a rather modest margin, with an accuracy around 65% (see e.g. Giampiccolo et al. (2007) for an overview).

A notable exception is Hickl & Bensley (2007), a system submitted by an industrial company (LCC) in the RTE-3 Challenge, which achieved 80% accuracy, using a variety of rich resources in a machine learning approach. A second LCC system (Tatu & Moldovan 2007) used a special-purpose theorem prover (Moldovan et al. 2007) and reached a high accuracy as well. Although neither the knowledge repositories nor the details about the method are available to the public, it is likely that the success of these systems stems from language and knowledge resources of various kinds that have been built over years with enormous manpower, accompanied by a consistent optimization of methods based on repeated task-oriented evaluations. This suggests that at the end of the day, the decisive factor in building high-performing systems for entailment checking is not a single theoretical insight or design decision, but rather the availability of huge amounts of information about language and the world. The key difference between the logic-based and machine-learning paradigms is that the latter degrades more gracefully when this information is not sufficiently available.

Discussion. Between 2005 and 2010 a total of about 300 different systems in total were evaluated. This has helped a lot in providing a clear picture of the potential of different methods and resources on the task. However, the RTE Challenges reveal a current state of the art that is not entirely satisfactory. Statistical systems appear to hit a ceiling in modeling inference. This is not just a technical problem: the fundamental shortcoming of purely text-based approaches is that they do not model the truth conditions of the sentences involved, and therefore cannot ground entailment in truth. It is difficult to imagine how a notion of inference for semantically complex sentences can be approximated by a model that does not in some way or another subsume the conceptual framework of logic-based semantics. On the other hand, direct implementations of the logic-based framework do not solve the problem either, because such systems are rendered practically unusable by the lack of formalized knowledge. Resolving this tension remains the central challenge for computational semantics today.

4.2. Natural logic inference

One promising direction of research that might help solve the dilemma is to model truth-based entailment directly in natural language, without resorting to explicit logical representations. The idea is old – indeed, before the introduction of formal logic, it was the only way of analyzing inference –, but was revived and formalized in the 1980s by Johan von Benthem under the heading of *natural logic* (van Benthem 1986; Sanchez-Valencia 1991). Consider the following examples:

(14) a. Last year, John bought a German convertible.

b. Last year, John bought a German car.

To determine the entailment relation between (14a) and (14b), we need not compute the respective logical representations and employ a deduction system. We just need to know that "convertible" is a hyponym of "car". The argument does not apply in general. Replacing "convertible" with "car" in "John didn't buy a convertible" or "John bought two convertibles" has different semantic effects: In the former case, entailment holds in the inverse direction, in the second, the two sentences are logically independent. The differences are due to the different monotonicity properties (in the sense of Barwise & Cooper 1981) of the contexts in which the respective substitutions take place. In addition to knowledge about lexical inclusion relations, we need syntactic information, a mechanism for monotonicity marking, and

monotonicity or polarity information for the functor expressions (in the sense of categorial grammar or type theory).

Natural logic and RTE. MacCartney & Manning (2008) and MacCartney (2009) propose a model for textual entailment recognition which is based on natural logic and extends and complements the framework in several aspects. Compared to the original approach of Sanchez-Valencia, they use a refined inventory of semantic relations. Wide-coverage knowledge about lexical semantic relations is obtained from WordNet, with distributional similarity as a fallback. Monotonicity handling includes the polarity analysis of implicative and factive verbs (Nairn, Condoravdi & Karttunen 2006), in addition to the standard operators (negation, determiners, conjunctions, modal expressions) and constructions. Their full model also processes sentence pairs that require multiple substitutions, deletions, or insertions; the global entailment relation between the sentences is computed as the joint entailment effect of the individual edit steps.

Because the preposition "without" introduces a downward monotonic context, the system can thus make the correct, but nontrivial judgment that (15a) and (15b) do not entail each other, based on the edits shown in (16).

(15) a. Some people are happy without a car.

b. Some professors are happy without an expensive convertible.

(16) Some SUBST(people, professors) are happy without an INSERT(expensive) SUBST(car, convertible).

The global entailment relation between the sentences is computed as the joint entailment effect of the single edit steps. Because the preposition "without" is downward monotonic in its internal argument, the system can thus make the correct, but nontrivial judgment that (15a) and (15b) do not entail each other, based on the edits shown in (16).

MacCartney's NATLOG system has been shown to achieve an accuracy of 70% on the FraCaS test suite. This demonstrates that the system can handle logically non-trivial inference problems, although some phenomena, like ellipsis, are outside the system's coverage. On the RTE-3 test set, the system has an accuracy of 59%, which does not exceed the performance achieved by simple word-overlap systems. However, the positive message is that that the natural-logic-based approach is able to avoid the robustness issues that make semantic construction for standard logic-based systems so

difficult. Combining NATLOG with the the shallow Stanford RTE system (de Marneffe et al. 2006) increases the accuracy of the shallow system from 60.5% by 4%, which proves that the "deep" inferences captured by the natural-logic-based system are able to complement shallow RTE methods in a substantial way.

Discussion. The natural logic approach does not capture all inferences that a predicate logic approach would. It does not deal with inferences that require multiple premises, and can only relate sentence pairs in which the lexical material is exchanged while the global structure stays the same (e.g., de Morgan's Law is outside its reach). However, the approach does cover many inference patterns that are relevant in natural language, and the overhead for semantic construction and the disambiguation of irrelevant parts of sentences is eliminated, because no translation to logical representation is required.

4.3. Statistical methods in semantic construction

One reason for the low performance of logic-based inference systems in the standard framework of computational semantics is the lack of wide-coverage semantic construction procedures. Natural logic gets around the problem by dispensing with semantic construction altogether. An alternative that has recently been explored is the use of machine learning techniques for the automatic assignment of rich semantic representations.

To get a better idea of the task, it is helpful to consider its relationship to systems for syntactic parsing. The two problems are similar from a high-level perspective, in that both compute structured linguistic representations for natural language expressions. The dominant approach in syntactic parsing is to apply supervised statistical approaches to syntactically annotated corpora, in order to learn grammars and estimate the parameters of a syntactic probability model. For semantic construction, statistical approaches have been much less successful. Even for Semantic Role Labeling, the results are noisier than for syntax. The assignment of complex logical structures as representations for full sentences is harder, due to the fine granularity of the target representations and the difficulty of finding surface features that are indicative of deep semantic phenomena. This makes the specification of annotation guidelines that would allow non-experts to reliably annotate a corpus challenging.

Nevertheless, a considerable amount of research in the past few years has investigated the use of supervised learning in semantic parsers, trained on

small domain-specific corpora. Logical annotations are typically obtained by converting the annotations from existing corpora, e.g., the Geo880 corpus (Zelle & Mooney 1996; Tang & Mooney 2000) of 880 geographical queries and the ATIS corpus (Dahl et al. 1994), a corpus of about 5000 spoken queries to a travel planning system. Both of these corpora were originally annotated with database queries that correspond to the natural-language query. When these are converted into lambda terms, examples look as follows:

(17) What states border Texas?
$\lambda x.\mathsf{state}(x) \wedge \mathsf{borders}(x, \mathsf{texas})$

(18) on may four atlanta to denver delta flight 257
$\lambda x.\mathsf{month}(x, \mathsf{may}) \wedge \mathsf{day\_number}(x, \mathsf{fourth}) \wedge \mathsf{from}(x, \mathsf{atlanta}) \wedge$
$\mathsf{to}(x, \mathsf{denver}) \wedge \mathsf{airline}(x, \mathsf{delta}) \wedge \mathsf{flight}(x) \wedge \mathsf{flight\_number}(x, 257)$

Current approaches for training semantic parsers typically employ methods from statistical machine translation, such as probabilistic synchronous grammars (Chiang 2007). These grammars simultaneously describe a tree for the syntactic representation of the natural-language string and a tree for the semantic representation, i.e. the lambda term. Because the syntactic parses are not explicitly given in the corpora mentioned above, these approaches assume a very permissive syntactic grammar, which allows many ungrammatical analyses of the input expression in addition to the grammatical ones. They then estimate parameters for a probability model that makes the ungrammatical analyses improbable, and maps the grammatical analyses to the correct semantic representations.

One key challenge that research in this area must overcome compared to pure syntactic parsing is that the annotated structures are not syntax trees, but lambda terms, which can be rewritten by $\alpha\beta\eta$-equality. The exact way in which this problem is addressed depends on the grammar formalism that a particular system uses. Wong & Mooney (2007) use a synchronous context-free grammar with an extra mechanism for representing variable binding. Zettlemoyer & Collins (2005) and Kwiatkowski et al. (2010) instead use probabilistic CCG grammars (Steedman 2000), which model the combination of lambda terms directly. The best-performing systems today achieve an accuracy of about 89% exact matches on the Geo880 corpus and still about 82% on the ATIS speech corpus (see Kwiatkowski et al. (2011) for an overview), which demonstrates that the method is feasible in principle. These are very promising numbers, but it is important to keep in mind that

these methods have so far been applied only to relatively small corpora from limited domains, and it remains to be seen how well they will scale up.

4.4. Grounded models of meaning

Standard systems of distributional semantics learn meaning information purely from text; but semantics, unlike syntax or morphology, is essentially concerned with the relationship of language with the outside world. Children do not learn what "chair" means by hearing people talk about chairs, but by observing chairs in connection with hearing the word "chair". Certain regularities in the real world are reflected in statistical patterns in texts (chairs are used for sitting, so the word "chair" frequently co-occurs with the word "sit"). But ultimately it is unsurprising that computer systems cannot learn the full semantics of words and sentences, when they are exposed to a much poorer and fundamentally incomplete stimulus.

While the simulation of human meaning acquisition in a full-fledged realistic environment is not feasible, a number of alternative methods have been explored to integrate restricted layers or pieces of extralinguistic information into the learning process. One option is the creation of multimodal corpora consisting of visual material – e.g., pictures or videos – labeled with linguistic descriptions. Large-scale data collections of this kind can be obtained through Internet-based experiments or games; examples are the Google Image Labeler (von Ahn & Dabbish 2004), which lets people annotate pictures with textual descriptions, and the Microsoft Research Video Description Corpus (Chen & Dolan 2011), which was collected by asking people to describe the activities shown in short YouTube videos.

Data of this kind can be used in two ways. First, one may completely disregard the nonlinguistic information, and use picture and video IDs just as indices of the natural-language expressions. This tells the system that the different descriptions of the same picture refer to the same scene: they are proper paraphrase candidates and definitely will not contain contradictory information. A similar effect is obtained by corpora containing parallel texts, which are known to describe the same event. For instance, Titov & Kozhevnikov (2010) use collections of alternative weather forecasts for the same day and region. Their system learns that "cloudy" and "sunny" stand in a different semantic relationship than "cloudy" and "overcast": while both pairs occur in similar linguistic contexts, the former but not the latter are identified as describing two different states of sky cover, because they do not co-occur as descriptions of one world state.

Other approaches have taken the further step of analyzing the contents

of the picture or video, typically using methods from computer vision, in order to let the computer system learn an actual mapping of language to extralinguistic objects. For example, Marszalek, Laptev & Schmid (2009) train a machine-learning system to identify instances of activities such as "drinking" in movies. Their training data is the movie itself together with textual descriptions of the current scene collected from subtitles and movie scripts. Learning a mapping between words and the external world is a problem that is frequently considered in cognitive robotics (Gold & Scassellati 2007; Kruijff et al. 2007), where a human user may explicitly teach the robot how to interpret spoken utterances in its environment. This also adds an *interactive* dimension to the process of automated language learning.

The core problem of mapping language to the extralinguistic environment can also be studied in more abstract settings. This has the advantage that the learning system can access the environment more directly. For instance, a system can learn the meaning of expressions referring to actions in a simulated robot soccer game (Chen, Kim & Mooney 2010), and the interpretation of help texts as actions in the Windows GUI, such as clicking buttons or entering text into certain input fields (Branavan et al. 2009). A middle ground is struck by approaches trained on virtual 3D environments (Orkin & Roy 2007; Fleischman & Roy 2005). An instructive account of alternative methods to connect language to real world or virtual reality is given in (Roy & Reiter 2005).

All these approaches to learning meaning representations are necessarily constrained in that they consider only some modalities and some aspects of non-linguistic information. Nevertheless, they form an exciting avenue of future research. From the perspective of semantic theory, they are perhaps most interesting because they open up a new direction in which the use of computers can support research on natural language meaning: as an instrument which connects natural-language expressions with large quantities of data about objects, properties, and events in the real world in a meaningful way.

5. Conclusion

Determining the meaning of a natural-language expression is crucial for many applications in computational linguistics, and computational semantics has long been a very active field of research. An approach to computational semantics that is to be useful for such applications must balance the depth of the linguistic analysis with the ability to compute such analyses reliably with wide coverage, i.e. for arbitrary sentences. Research in compu-

tational semantics is characterized by navigating this tension between depth and coverage.

In this article, we have sketched a number of prominent approaches in our field. Direct implementations of logic-based theories of semantics managed to overcome initial efficiency problems and, to some extent, deal with the massive amount of ambiguity that such approaches face in practice. However, making wide-coverage semantic construction robust and acquiring wide-coverage knowledge resources for inferences remain open problems. By contrast, data-intensive approaches have had very impressive successes in extracting useful semantic information from text corpora. But they tend to work with shallower meaning information than logic-based approaches; deeper representations still require a modeling effort by humans. The most promising recent research brings these two paradigms together, and combines them with novel ideas for models of meaning that are grounded in the environment. In our view, this makes the present a very exciting time for research in computational semantics indeed.

Acknowledgments. We gratefully acknowledge Ingo Reich, Caroline Sporleder, Stefan Thater, and Margaret Delap for valuable comments on this article. Several examples are due to Collin Baker, Josef Ruppenhofer, and Gerhard Weikum. Finally, we thank Claudia Maienborn for the infinite patience and cheerfulness with which she handled the perpetually almost-finished state of our manuscript.

6. References

von Ahn, Luis & Laura Dabbish 2004. Labeling images with a computer game. In: *Proceedings of the ACM CHI Conference*.

Alshawi, Hiyan (ed.) 1990. *The Core Language Engine*. MIT Press.

Alshawi, Hiyan & Richard Crouch 1992. Monotonic semantic interpretation. In: *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*.

Althaus, Ernst, Denys Duchier, Alexander Koller, Kurt Mehlhorn, Joachim Niehren & Sven Thiel 2003. An efficient graph algorithm for dominance constraints. *Journal of Algorithms* 48, 194–219.

Andrews, Peter B. & Chad E. Brown 2006. TPS: A hybrid automatic-interactive system for developing proofs. *Journal of Applied Logic* 4, 367–395.

Areces, Carlos, Alexander Koller & Kristina Striegnitz 2008. Referring expressions as formulas of description logic. In: *Proceedings of the 5th International Natural Language Generation Conference*. Salt Fork.

Asher, Nicholas & Alex Lascarides 2003. *Logics of Conversation*. Cambridge University Press.

Baader, Franz, Diego Calvanese, Deborah McGuiness, Daniele Nardi & Peter Patel-Schneider (eds.) 2003. *The Description Logic Handbook: Theory, implementation and applications*. Cambridge University Press.

Baker, Collin, Charles Fillmore & Beau Cronin 2003. The structure of the FrameNet database. *International Journal of Lexicography* 16, 281–296.

Bar-Haim, Roy, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini & Idan Szpektor 2006. The second PASCAL Recognising Textual Entailment challenge. In: *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Bar-Hillel, Yehoshua 1960. The present status of automatic translation of languages. *Advances in Computers* 1, 91–163.

Barwise, Jon & Robin Cooper 1981. Generalized quantifiers and natural language. *Linguistics & Philosophy* 4, 159–219.

van Benthem, Johan 1986. *Essays in Logical Semantics*. Dordrecht: Reidel.

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

Bos, Johan 2001. DORIS 2001: Underspecification, Resolution and Inference for Discourse Representation Structures. In: *Proceedings of the Third International Workshop on Inference in Computational Semantics*.

Bos, Johan, Stephen Clark, Mark Steedman, James Curran & Julia Hockenmaier 2004. Wide-coverage semantic representations from a CCG parser. In: *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*.

Bos, Johan & Katja Markert 2005. Recognising textual entailment with logical inference. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 628–635.

Brachman, Ronald & James Schmolze 1985. An overview of the KL-ONE knowledge representation system. *Cognitive Science* 9, 171–216.

Branavan, S.R.K., Harr Chen, Luke S. Zettlemoyer & Regina Barzilay 2009. Reinforcement learning for mapping instructions to actions. In: *Proceedings of the Joint Conference of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing (ACL-IJCNLP)*.

Budanitsky, Alexander & Graeme Hirst 2006. Evaluating WordNet-based measures of semantic distance. *Computational Linguistics* 32(1), 13–47.

Burchardt, Aljoscha, Katrin Erk, Anette Frank, Andrea Kowalski, Sebastian Padó & Manfred Pinkal 2006. The SALSA Corpus: a German corpus resource for lexical semantics. In: *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*. 969–974.

Carreras, Xavier & Lluis Marquez 2004. Introduction to the CoNLL-2004 shared task: Semantic role labeling. In: *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL)*.

Carreras, Xavier & Lluis Marquez 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In: *Proceedings of the Ninth Conference on Computational Language Learning (CoNLL)*. 152–164.

Chapman, R. 1977. *Roget's International Thesaurus*. New York: Harper & Row.

Chen, David & Bill Dolan 2011. Building a persistent workforce on Mechanical Turk for multilingual data collection. In: *Proceedings of the 25th Conference on Artificial Intelligence (AAAI-11)*.

Chen, David L., Joohyun Kim & Raymond J. Mooney 2010. Training a Multilingual Sportscaster: Using Perceptual Context to Learn Language. *Journal of Artificial Intelligence Research* 37, 397–435.

Chiang, David 2007. Hierarchical phrase-based translation. *Computational Linguistics* 33, 201–228.

Chklovski, Timothy & Patrick Pantel 2004. VerbOcean: Mining the Web for fine-grained semantic verb relations. In: *Proceedings of EMNLP*.

Claessen, Koen & Niklas Sörensson 2003. New techniques that improve MACE-style model finding. In: *Proceedings of the CADE-19 Workshop on Model computation – Principles, Algorithms, Applications*. 11–27.

Clear, Jeremy 1993. *The British National Corpus*. MIT Press.

Cooper, Robin 1983. *Quantification and Syntactic Theory*. Dordrecht: Reidel.

Cooper, Robin, Richard Crouch, Jan van Eijck, Chris Fox, Johan van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio & Steve Pulman 1996. Using the framework. FraCas project deliverable D-16, Technical Report LRE 62-051; *ftp://ftp.cogsci.ed.ac.uk/pub/FRACAS/del16.ps.gz*.

Copestake, Ann, Dan Flickinger, Carl Pollard & Ivan Sag 2005. Minimal Recursion Semantics: An Introduction. *Research on Language and Computation* 3, 281–332.

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*. Toulouse, 132–139.

Crouch, Dick & Tracy Holloway King 2006. Semantics via f-structure rewriting. In: *Proceedings of the LFG06 Conference*.

Dagan, Ido, Oren Glickman & Bernardo Magnini 2006. The PASCAL Recognising Textual Entailment challenge. In: J. Quiñonero-Candela, I. Dagan, B. Magnini & F. d'Alché Buc (eds.) *Machine Learning Challenges*, Springer. 177–190.

Dahl, Deborah A., Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky & Elizabeth Shriberg 1994. Expanding the scope of the ATIS task: the ATIS-3 corpus. In: *Proceedings of the ARPA Human Language Technology Workshop*.

Dalrymple, Mary (ed.) 1999. *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach*. MIT Press.

Dalrymple, Mary, Ronald M. Kaplan, John T. Maxwell & Annie Zaenen (eds.) 1995. *Formal Issues in Lexical-Functional Grammar*. CSLI Publications.

Dinu, Georgiana 2011. *Word Meaning in Context: A Probabilistic Model and its Application to Question Answering*. Ph.D. thesis, Saarland University.

Dinu, Georgiana & Mirella Lapata 2010. Topic models for meaning similarity in context. In: *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*.

Egg, Markus, Alexander Koller & Joachim Niehren 2001. The constraint language for lambda structures. *Logic, Language, and Information* 10, 457–485.

van Eijck, Jan, Juan Hegueiabehere & Breanndan O Nuallain 2001. Tableau reasoning and programming with dynamic first order logic. *Logic Journal of the IGPL* .

Erk, Katrin & Sebastian Padó 2008. A structured vector space model for word meaning in context. In: *Proceedings of EMNLP*.

Fellbaum, Christiane (ed.) 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Fillmore, Charles 1968. Lexical entries for verbs. *Foundations of Language* 4, 373–393.

Fillmore, Charles J. & Collin F. Baker 2010. A frame approach to semantic analysis. In: B. Heine & H. Narrog (eds.) *Oxford Handbook of Linguistic Analysis*, Oxford: Oxford University Press.

Firth, John 1957. *Papers in Linguistics 1934–1951*. Oxford University Press.

Fleischman, Michael & Deb Roy 2005. Intentional context in situated language learning. In: *Proceedings of the Ninth Conference on Natural Language Learning (CoNLL)*.

Fürstenau, Hagen & Mirella Lapata 2009. Semi-supervised semantic role labeling. In: *Proceedings of the 12th EACL*.

Gardent, Claire 2003. Semantic construction in feature-based TAG. In: *Proceedings of the 10th Meeting of the European Chapter of the Association for Computational Linguistics (EACL)*. 123–130.

Gardent, Claire & Karsten Konrad 2000. Understanding "Each Other". In: *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP)*. 319–326.

Giampiccolo, Danilo, Bernardo Magnini, Ido Dagan & Bill Dolan 2007. The third PASCAL Recognizing Textual Entailment challenge. In: *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.

Gildea, Daniel & Daniel Jurafsky 2002. Automatic labeling of semantic roles. *Computational Linguistics* 28, 245–288.

Girju, Roxana, Adriana Badulescu & Dan Moldovan 2006. Automatic discovery of part-whole relations. *Computational Linguistics* 32.

Glickman, Oren, Ido Dagan & Moshe Koppel 2005. A probabilistic classification approach for lexical lextual entailment. In: *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI)*.

Gold, Kevin & Brian Scassellati 2007. A robot that uses existing vocabulary to infer non-visual word meanings from observation. In: *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI)*.

Grefenstette, Edward & Mehrnoosh Sadrzadeh 2011. Experimental support for a categorical compositional distributional model of meaning. In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Groenendijk, Jeroen & Martin Stokhof 1991. Dynamic predicate logic. *Linguistics & Philosophy* 14, 39–100.

Grosz, Barbara, Aravind Joshi & Scott Weinstein 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics* 21, 203–225.

Grosz, Barbara & Candace Sidner 1986. Attention, intention, and the structure of discourse. *Computational Linguistics* 12, 175–204.

Haarslev, Volker & Ralf Möller 2001. Description of the RACER system and its applications. In: *Proceedings of the International Workshop on Description Logics (DL-2001)*. 131–141.

Hajic, Jan, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antonia Marti, Lluis Marquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Stepanek, Pavel Stranak, Mihai Surdeanu, Nianwen Xue & Yi Zhang 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In: *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL): Shared Task*.

Hamp, B. & H. Feldweg 1997. GermaNet – a lexical-semantic net for German. In: *Proceedings of the ACL Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*.

Harris, Zellig S. 1951. *Methods in structural linguistics*. University of Chicago Press.

Hearst, Marti A. 1992. Automatic acquisition of hyponyms from large text corpora. In: *Proceedings of the Fourteenth Conference on Computational Linguistics (COLING)*. Nantes, 539–545.

Hickl, Andrew & Jeremy Bensley 2007. A discourse commitment-based framework for recognizing textual entailment. In: *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.

Hillenbrand, Thomas 2003. CITIUS ALTIUS FORTIUS: Lessons learned from the theorem prover WALDMEISTER. In: I. Dahn & L. Vigneron (eds.) *Proceedings of the 4th International Workshop on First-Order Theorem Proving*. Elsevier.

Hirst, Graeme & Eugene Charniak 1982. Word sense and case slot disambiguation. In: *Proceedings of the Second National Conference on Artificial Intelligence (AAAI)*.

Hobbs, Jerry, Douglas Appelt, John Bear, Mabry Tyson & David Magerman 1992. Robust processing of real-world natural language texts. In: P. Jacobs (ed.) *Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval*, Lawrence Erlbaum. 13–33.

Hobbs, Jerry & Stuart Shieber 1987. An algorithm for generating quantifier scopings. *Computational Linguistics* 13, 47–63.

Hobbs, Jerry R. 1985. Ontological promiscuity. In: *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics (ACL)*. 61–69.

Hobbs, Jerry R., Mark E. Stickel, Douglas E. Appelt & Paul A. Martin 1993. Interpretation as abduction. *Artificial Intelligence* 63, 69–142.

Jijkoun, Valentin & Maarten de Rijke 2005. Recognizing textual entailment using lexical similarity. In: *Proceedings of the PASCAL Recognising Textual Entailment Challenge*.

Jurafsky, Dan & James Martin 2008. *Speech and Language Processing*. Prentice Hall.

Kallmeyer, Laura & Maribel Romero 2008. Scope and situation binding in LTAG using semantic unification. *Research on Language and Computation* 6, 3–52.

Kamp, Hans 1981. A theory of truth and semantic representation. In: J. Groenendijk, T. Janssen & M. Stokhof (eds.) *Formal Methods in the Study of Language*. Amsterdam, 277–322.

Kamp, Hans & Uwe Reyle 1993. *From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language, Formal Logic, and Discourse Representation Theory*. Dordrecht: Kluwer.

Keller, Frank, Maria Lapata & Olga Ourioupina 2002. Using the Web to Overcome Data Sparseness. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Keller, William 1988. Nested Cooper storage: The proper treatment of quantification in ordinary noun phrases. In: U. Reyle & C. Rohrer (eds.) *Natural Language Parsing and Linguistic Theories*, Dordrecht: Reidel. 432–447.

Kipper-Schuler, Karin 2006. *VerbNet: A Broad-coverage, Comprehensive Verb Lexicon*. Ph.D. thesis, University of Pennsylvania.

Koch, Stephan, Uwe Küssner & Manfred Stede 2000. Contextual disambiguation. In: W. Wahlster (ed.) *Verbmobil: Foundations of Speech-to-speech Translation*, Heidelberg: Springer. 466–480.

Kohlhase, Michael 2000. Model generation for discourse representation theory. In: *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI)*. 441–445.

Kohlhase, Michael, Susanna Kuschert & Manfred Pinkal 1996. A type-theoretic semantics for λ-DRT. In: P. Dekker & M. Stokhof (eds.) *Proceedings of the 10th Amsterdam Colloquium*. 479–498.

Koller, Alexander, Ralph Debusmann, Malte Gabsdil & Kristina Striegnitz 2004. Put my galakmid coin into the dispenser and kick it: Computational linguistics and theorem proving in a computer game. *Journal of Logic, Language, and Information* 13, 187–206.

Koller, Alexander & Stefan Thater 2010. Computing weakest readings. In: *Proceedings of the 48th ACL*. Uppsala.

Kruijff, Geert-Jan M., Hendrik Zender, Patric Jensfelt & Henrik I. Christensen 2007. Situated dialogue and spatial organization: What, where ... and why? *International Journal of Advanced Robotic Systems* 4, 125–138.

Kwiatkowski, Tom, Luke Zettlemoyer, Sharon Goldwater & Mark Steedman 2010. Inducing probabilistic CCG grammars from logical form with higher-order unification. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Kwiatkowski, Tom, Luke Zettlemoyer, Sharon Goldwater & Mark Steedman 2011. Lexical generalization in CCG grammar induction for semantic parsing. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Landauer, Thomas, Peter Foltz & Darrell Laham 1998. An introduction to latent semantic analysis. *Discourse Processes* 25, 259–284.

Landes, Shari, Claudia Leacock & Randee I. Tengi 1998. Building semantic concordances. In: C. Fellbaum (ed.) *WordNet: An Electronic Lexical Database*, Cambridge, MA: MIT Press.

Lenat, Douglas 1995. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM* 38, 33–38.

Li, Linlin, Benjamin Roth & Caroline Sporleder 2010. Topic models for word sense disambiguation and token-based idiom detection. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Lin, Dekang & Patrick Pantel 2001. Discovery of inference rules for question answering. *Natural Language Engineering* 7, 343–360.

Lipton, Peter 2001. *Inference to the Best Explanation*. London: Routledge.

MacCartney, Bill 2009. *Natural language inference*. Ph.D. thesis, Stanford University.

MacCartney, Bill & Christopher D. Manning 2008. Modeling semantic containment and exclusion in natural language inference. In: *Proceedings of the 22nd International Conference on Computational Linguistics (COLING-08)*.

Manning, Christopher 2006. Local Textual inference: It's Hard to Circumscribe, but You Know It When You See It – and NLP Needs It. Ms., Stanford University. *http://nlp.stanford.edu/~manning/papers/LocalTextualInference.pdf*.

Manning, Christopher, Prabhakar Raghavan & Hinrich Schütze 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Manning, Christopher & Hinrich Schütze 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

Marcus, Mitchell P., Beatrice Santorini & Mary Ann Marcinkiewicz 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* 19, 313–330.

de Marneffe, Marie-Catherine, Bill MacCartney, Trond Grenager, Daniel Cer, Anna Rafferty & Christopher Manning 2006. Learning to distinguish valid textual entailments. In: *Proceedings of the Second PASCAL Workshop on Recognizing Textual Entailment*.

Marszalek, Marcin, Ivan Laptev & Cordelia Schmid 2009. Actions in context. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Matuszek, Cynthia, John Cabral, Michael Witbrock & John DeOliveira 2006. An introduction to the syntax and content of Cyc. In: *Proceedings of the AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering.*

McCarthy, Diana 2009. Word sense disambiguation: An overview. *Language and Linguistics Compass* 3, 537–558.

McCarthy, Diana & John Carroll 2003. Disambiguating nouns, verbs, and adjectives using automatically acquired selectional preferences. *Computational Linguistics* 29, 639–654.

McCune, William 1998. Automatic proofs and counterexamples for some ortholattice identities. *Information Processing Letters* 65, 285–291.

Mitchell, Jeff & Mirella Lapata 2008. Vector-based models of semantic composition. In: *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT).* 236–244.

Mitchell, Tom 1997. *Machine Learning.* McGraw Hill.

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

Montague, Richard 1973. On the proper treatment of quantification in ordinary English. In: R. Thomason (ed.) *Formal Philosophy: Selected papers of Richard Montague*, Yale University Press. 247–270.

Moore, R. C. 1985. Semantical considerations on nonmonotonic logic. *Artificial Intelligence* 25, 75–94.

Muskens, Reinhard 1995. *Meaning and Partiality.* CSLI Publications.

Nairn, Rowan, Cleo Condoravdi & Lauri Karttunen 2006. Computing relative polarity for textual inference. In: *Proceedings of the Fifth Workshop on Inference in Computational Semantics (ICoS-5).* 67–76.

Navigli, Roberto 2009. Word sense sisambiguation: A survey. *ACM Computing Surveys* 41, 1–69.

Navigli, Roberto, Kenneth C. Litkowski & Orin Hargraves 2007. SemEval-2007 Task 07: Coarse-grained English all-words task. In: E. Agirre, L. Marquez & R. Wicentowski (eds.) *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval 2007).*

Ng, Hwee Tou 1997. Getting serious about word sense disambiguation. In: *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?* 1–7.

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 (ACL)*. 1396–1411.

Orkin, Jeff & Deb Roy 2007. The Restaurant Game: Learning social behavior and language from thousands of players online. *Journal of Game Development* 3, 39–60.

Palmer, Martha, Daniel Gildea & Paul Kingsbury 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics* 31, 71–105.

Pereira, Fernando C. N. & Stuart M. Shieber 1987. *Prolog and Natural-Language Analysis*. CSLI Publications.

Poesio, Massimo 1994. Ambiguity, underspecication, and discourse interpretation. In: *Proceedings of the First International Workshop on Computational Semantics*.

Pollard, Carl & Ivan Sag 1994. *Head-driven Phrase Structure Grammar*. University of Chicago Press.

Pulman, Stephen 2007. Formal and computational semantics: A case study. In: J. Geertzen, E. Thijsse, H. Bunt & A. Schiffrin (eds.) *Proceedings of the Seventh International Workshop on Computational Semantics (IWCS)*. 181–196.

Ravichandran, Deepak & Eduard Hovy 2002. Learning surface text patterns for a question answering system. In: *Proceedings of the 40th ACL*.

Reiter, R. 1980. A logic for default reasoning. *Artificial Intelligence* 13, 81–132.

Riazanov, Alexandre & Andrei Voronkov 2002. The design and implementation of VAMPIRE. *AI Communications* 15, 91–110.

Roy, Deb & Ehud Reiter 2005. Connecting language to the world. *Artificial Intelligence* 167, 1–12.

Russell, Stuart & Peter Norvig 2010. *Artificial Intelligence: A Modern Approach*. Prentice Hall.

Sanchez-Valencia, Victor 1991. *Studies on Natural Logic and Categorial Grammar*. Ph.D. thesis, University of Amsterdam.

Schank, Roger 1975. *Conceptual Information Processing*. Elsevier.

Schütze, Hinrich 1998. Automatic word sense discrimination. *Computational Linguistics* 24, 97–123.

Settles, Burr 2009. *Active learning literature survey*. Computer Sciences Technical Report 1648, University of Wisconsin-Madison. *http://www.cs.cmu.edu/~bsettles/pub/settles.activelearning.pdf*.

Stede, Manfred 2011. *Discourse Processing*. Morgan & Claypool.

Steedman, Mark 2000. *The Syntactic Process*. MIT Press.

Suchanek, Fabian M., Gjergji Kasneci & Gerhard Weikum 2008. YAGO: A large ontology from Wikipedia and WordNet. *Journal of Web Semantics* 6, 203–217.

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

Tatu, Marta & Dan Moldovan 2007. COGEX at RTE 3. In: *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.

Thater, Stefan, Hagen Fürstenau & Manfred Pinkal 2010. Contextualizing semantic representations using syntactically enriched vector models. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Thater, Stefan, Hagen Fürstenau & Manfred Pinkal 2011. Word meaning in context: A simple and effective vector model. In: *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP)*.

Titov, Ivan & Mikhail Kozhevnikov 2010. Bootstrapping semantic analyzers from non-contradictory texts. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*. 958–967.

Toma, Peter 1977. SYSTRAN as a multilingual machine translation system. In: *Overcoming the language barrier. Third European Congress on Information Systems and Networks*. 569–581.

Tsarkov, Dmitry, Ian Horrocks & Peter F. Patel-Schneider 2007. Optimizing terminological reasoning for expressive description logics. *Journal of Automated Reasoning* 39, 277–316.

Turney, Peter & Patrick Pantel 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research* 37, 141–188.

Vossen, Piek 2004. EuroWordNet: a multilingual database of autonomous and language-specific wordnets connected via an Inter-Lingual-Index. *International Journal of Linguistics* 17.

Wilks, Yorick 1975. A preferential, pattern seeking, semantics for natural language inference. *Artificial Intelligence* 6.

Witten, Ian H., Eibe Frank & Mark A. Hall 2011. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.

Wong, Yuk Wah & Raymond J. Mooney 2007. Learning synchronous grammars for semantic parsing with lambda calculus. In: *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yarowsky, David 1992. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In: *Proceedings of COLING*.

Yarowsky, David 1995. Unsupervised word sense disambiguation rivaling supervised methods. In: *Proceedings of the 33rd ACL*.

Zaenen, Annie, Lauri Karttunen & Richard Crouch 2005. Local textual inference: Can it be defined or circumscribed? In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Zelle, John M. & Raymond J. Mooney 1996. Learning to parse database queries using Inductive Logic Programming. In: *Proceedings of the Thirteenth National Conference on Aritificial Intelligence (AAAI)*. 1050–1055.

Zettlemoyer, Luke S. & Michael Collins 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In: *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI)*.

Keywords:
computational linguistics, knowledge-based methods, corpus-based methods

Alexander Koller, Potsdam (Germany)
Manfred Pinkal, Saarbrücken (Germany)