Influences and Inferences

Jerry R. Hobbs[*]
Information Sciences Institute
University of Southern California
Marina del Rey, California

1. False and True Starts

I am deeply honored to receive the ACL's Lifetime Achievement Award. I'm especially honored when I look back at the list of previous winners—Chuck Fillmore, Eugene Charniak, Eva Hajičová, Fred Jelinek, Martin Kay, Aravind Joshi, and the others—they're all my heroes.

I was of course delighted to learn of this award. The most we can hope for in life is to take part in the conversation, and an award like this means that you've taken part in the conversation.

It seems to be a tradition to begin with a few formative anecdotes from childhood. For me, it all begins before I was born. My grandfather, a crusty old country lawyer in southern Indiana, told my father not to bother trying to go to law school. "You don't know English grammar," he said. "You'll flunk out." My dad accepted the challenge, bought a book entitled *English Grammar*, by Smith, Magee, and Seward (1928), and mastered it. He went on to become a very successful lawyer.

Fast-forward to when I was in junior high school. My dad was distressed that my English classes looked to him more like social studies, and barely touched on grammar. So he persuaded me—actually, he probably bribed me, but I can't remember what with—to master that same book, *English Grammar* by Smith, Magee, and Seward. This was a concession, because I was a math nerd, reading only textbooks on trigonometry and calculus, as my way of avoiding the humiliation of playing baseball. But I read the book, and I was amazed. English grammar was just like math! It had the same sorts of rules, the same kinds of abstractions, the same types of puzzles. It was actually fun!

In my junior or senior year of high school we had to take something called the Kuder Preference Test, which would help us decide what career to choose. I scored high in math and in language. So my high school counselor told me I should write math books. In fact, she got it exactly backwards. It wasn't that I should do language about math. It was that I should do math about language.

I've met any number of computational linguists with a similar story. They grew up not knowing whether they wanted to be a physicist or a poet. They just knew both sounded fascinating. Then they discovered our field.

My last near miss happened the week I was drafted into the Army. They gave us a battery of aptitude tests to see what specialties we'd be best for. One of the tests was to see if we should be sent to the Monterey Language School. Looking back on it, I realize now it was testing how well you could understand formal language theory. They'd give

---

∗ USC/ISI, 4676 Admiralty Way, Marina del Rey, CA 90292, USA. E-mail: hobbs@isi.edu.

you a bunch of rules for an artificial language, and you'd have to say whether different strings were in or not in the language. I'd never seen anything like it, but it was really fun to do. Later I met with a personnel specialist who went over my test scores. I got a 46 out of 50. He ignored that until I pointed it out to him. Then he said, "That's a mistake. Nobody ever gets more than 6 or 7 points on that test."

I said, "No, I think it might be correct."

He said, "It doesn't matter. You're not going to the Monterey Language School. You're going to South Vietnam."

Actually, I didn't go to the Monterey Language School *or* to South Vietnam. I spent two years in South Carolina, and was glad to be there. How I managed that is a story for another occasion.

So I didn't really discover computational linguistics until my third year in graduate school at New York University. In October I passed my oral exam in topics like algebraic topology and complex analysis, by one generous yes and two abstentions. In the subsequent months I discovered more and more facts about myself—for example, that I was never going to figure out a faster way of multiplying matrices, and that fascinating though recursion theory might be, I was never going to prove a theorem that Hartley Rogers would be compelled to include in his next edition. As I surveyed vaguely plausible fields, I realized I had no idea what the next problem to solve would be or even what makes a problem interesting.

Then in April, when I had nearly resigned myself to becoming a taxi driver, I discovered New York University's best-kept secret: Naomi Sager's Linguistic String Project. I think it is also computational linguistics' best kept secret as well. She was motivated by the science, not by the performance, and her very impressive work is nowhere near as well-known as it should be. I think her Linguistic String Grammar (Sager 1981) ranks, as a computational specification of English syntax, with Pollard and Sag's *Head-driven Phrase Structure Grammar* (1994), for thoroughness, insight, and elegance. So, for example, in 1992 when we developed the FASTUS system for information extraction using cascaded finite-state transducers (Hobbs et al. 1997), it was straightforward to copy the rules for Noun Groups straight from her grammar. It's no accident that in the late 1980s during the Strategic Computing Initiative and in the early 1990s in the Message Understanding Conferences, three of the most important efforts were led by Linguistic String Project alumni—Ralph Grishman's group at New York University, Lynette Hirschman's at Unisys, and my group at SRI International. I think the most important lesson I learned from Naomi Sager was to look closely at the data and to take it seriously.

My other thesis advisor was Jack Schwartz. He was a polymath, so to speak. I took a course in logic from him. I knew about his book on compilers and the classic Dunford and Schwartz on functional analysis. But when I saw his book on mathematical economics and his book on the theory of relativity, I did some research to see if there was more than one Jack Schwartz. Among his writings was an unpublished Chapter 9 of his compilers book, on parsing natural language, which I of course read.

My thesis was on Earley's algorithm applied to natural language. It quickly became apparent that the constraints on phrase structure rules had to be expressed and that one could do that with fairly simple operations on vectors of features, where among the features were what I called the "cores" of the constituents, since they bundled many of the relevant features. My "core" was what linguists came to call "head." Years later, I ran across Chapter 9 again and reread it, and realized that all the ideas in my thesis were there. So when in 1987 Schwartz told someone that I had anticipated head-driven phrase structure grammar, that was his way of saying *he* had anticipated head-driven phrase structure grammar.

My first job was at Yale University as a very temporary instructor—I think the position is now called "post-doc." Over the course of the year I became convinced that syntax was a solved problem—something I still believe. But that left me adrift for problems to work on. I became discouraged, and found myself thinking again about driving that taxi. Then late one afternoon, just as I was about to go home, a graduate student named Fred Howard came into my office to ask a couple of questions. That triggered a discussion that lasted until 11 o'clock that evening. One of the wheels we reinvented was a recognition of the pervasiveness of spatial metaphor in discourse. (This was before Lakoff and Johnson [1980], but after similar observations by the 18th-century Italian philosopher Giambattista Vico (1968 [1744]) and the 20th-century English literary critic I. A. Richards [1936].) But within a year, everything else of value that remained of the content of that discussion could be compressed into a long footnote in a technical report. In any case, this conversation lit a fire that fueled my research for the next 15 or 20 years.

In particular, I began looking at texts, trying to understand how we understand them. No doubt influenced by Chuck Rieger's thesis (Rieger 1974), I asked what inferences we draw in the course of comprehension, and, an issue Rieger did not address, what inferences we do not draw. This culminated in 1976 in an unreadable (and unread) technical report (Hobbs 1976), microanalyzing one paragraph from *Newsweek*, trying to specify every bit of knowledge required for understanding the text and describing how every linguistic problem in the text invokes that knowledge to arrive at solutions. One could say that the rest of my career has been a matter of cleaning up and extending that technical report, in terms of representation, the process of inference and interpretation, and the specification of common-sense knowledge.

2. Representation

In 1977 I moved to SRI, where I fell under the influence of Nils Nilsson and Bob Moore, and of John McCarthy at nearby Stanford. They were campaigning to replace the ad hoc styles of representation of early AI with representations based on first-order logic. But the problem in a nutshell is this: When we are trying to represent an English sentence like *Pat believes Chris is tall*, we really want to write

(1)   *believe(Pat, tall(Chris))*

The difficulty is that *tall* is a predicate and *tall(Chris)* evaluates to true or false, so we are left with Pat believes a truth value, with not a hint of Chris's tallness to be found. A common solution to this is to treat *believe* not as a predicate but as an opaque operator that blocks evaluation of its operands.

Many special logics have been developed for such operators. For example, knowing about modal and temporal logics, Russell's iota operator, functionals, lambda expressions, and so on, we might represent the sentence

(2)   Maybe the boy wanted to build a boat quickly.

by the expression

(3) $(\iota x : BOY)[\Diamond PAST[WANT(x, \lambda z[(\exists y : BOAT)Quick(build)(z, y)]]]]$

This bothered me because it seemed like we were introducing a new operator with its own special logic every time we encountered a new word to define or characterize. For

20,000 words would we have to introduce 20,000 new operators? It seemed to me that we should rather stay within first-order logic, abiding by two principles:

1. All morphemes are created equal.

2. Every morpheme conveys a predication.

We could achieve this kind of representation by means of reification. Thus, if $tall'(e, Chris)$ says that eventuality e is the state or eventuality of Chris being tall, then we can represent *Pat believes Chris is tall* by

(4) $believe(Pat, e) \wedge tall'(e, Chris)$

Sentence (2) is then represented

(5) $maybe(e_5) \wedge the(x, e_3) \wedge boy'(e_3, x) \wedge want'(e_4, x, e_6) \wedge Past'(e_5, e_4)$
$\wedge build'(e_6, x, y) \wedge a(y, e_8) \wedge boat'(e_8, y) \wedge slow(e_6)$

There's nothing exotic here (other than reification). It's all first-order logic, predicates applied to arguments where the arguments are existentially quantified variables with widest possible scope, ranging over a universe of possible individuals.

The extremes to which we go in identifying morphemes with predications can be seen in the predication $the(x, e_3)$. What could that possibly mean? Well, ask what information is being conveyed by the word *the*. It is a relation between an entity x and a description e_3, and it says the entity is uniquely mutually identifiable in context by means of the description. We can give this relation a name. We could call it something like *uniquely-mutually-identifiable-in-context*. But why not keep it simple, and name the predicate after the morpheme that conveys it – *the*?

Knowledge representation schemes that use extensive reification are often called "Davidsonian," after the philosopher Donald Davidson (1967), who proposed reifying events. But he balked at reifying states, let alone negations of states and events. He would not have treated Chris's tallness as a thing. By contrast, I adopted a position that, because I was young and wild, I called "ontological promiscuity." Now that I'm older and more domesticated, I would probably call it something like "ontological prosperity" or "ontological comfortable circumstances" or maybe "ontological glut."

Many balk at such abandonment of ontological scruples. No doubt I was influenced by the near solipsism that infected many researchers in the early days of AI. Our brains could be fooling us, just as we often fool computers to test our programs. Yes, there is probably a world out there that occasionally bites back. But the world is benevolent—after all, we evolved in it. When we breathe, there is almost always oxygen there. That's no accident. So it doesn't matter very much what we believe. We can believe all sorts of crazy things and be completely ignorant of apparently real and pervasive phenomena. Until the recent past we believed in the spirits of the dead, and we were entirely ignorant of 98% of the electromagnetic spectrum. If you are willing to admit the existence of physical objects, sets, numbers, and possible worlds, what ontological scruples do you have anyway? So why should we give any credence at all to our intuitions about what exists and what doesn't? Why not simply stipulate that everything that can be talked about exists in a Platonic universe of possible individuals, since that makes it so much easier to represent and reason about the content of natural language discourse?

The result of this move and similar reifications to eliminate quantifier scopings is that the logical form of a sentence is a flat conjunction of existentially quantified propositions, with one predication per morpheme.

But there is a problem. The sentence

(6)   John is tall.

would be represented

(7) $John'(e_1, x) \wedge tall'(e_3, x)$

whereas the sentence

(8)   John is not tall.

would be represented

(9) $John'(e_1, x) \wedge not'(e_2, e3) \wedge tall'(e_3, x)$

But $P \wedge Q \wedge R$ implies $P \wedge R$, so it would seem that *John is not tall* implies *John is tall*.

The wrinkle is that $tall'(e_3, x)$ does not say that x is tall. It says that e_3 is a possible eventuality of x's being tall. The eventuality e_3 may or may not exist in the real world, and if it does, that is one of its properties – *Rexist*(e_3).

This means that we have to distinguish between the *content* of a sentence and its *claim*. Sentences (6) and (8) have highly overlapping content. But the claim of sentence (6) is e_3, the tall-ness, while the claim of sentence (8) is e_2, the negation of the tall-ness.

The general procedure for deciding on whether or not an eventuality really exists is as follows:

Step 1: Identify the claim.

Step 2: Propagate truth and falsity through implicatives.

Step 3: As a courtesy to the speaker, assume the other propositions are true. (But note that in modal contexts there is an ambiguity in whether the grammatically subordinated material holds in the real world [*de re*] or in the modal context [*de dicto*].)

For example, in

(10)   The lazy man did not manage to avoid attending the meeting.

Step 1 says the claim is the "not." Step 2 says that therefore "manage" is false, "avoid" is false, and "attend" is true. Step 3 says that "lazy," "man," and "meeting" are all true.

This kind of representation has the advantage of yielding a very elegant view of compositional semantics. In traditional approaches to compositional semantics, the meanings of constituents are lambda expressions, and composition happens by function application. With a flat logical form, the only role function application plays is identifying variables with each other. This gives us a two-part account of compositional semantics.

1.   The lexicon provides predicate–argument relations.

2.   Syntax identifies variables.

For the sentence

(11)   The man attended the meeting.

ignoring *the* and tense, we get from the individual words the propositions

(12) $man'(e_1, x_1), attend'(e_2, x_2, y_2), meeting'(e_3, y_3)$

When we recognize that *attended the meeting* is a verb phrase, this amounts to recognizing that $y_2 = y_3$. When we recognize *the man attended the meeting* as a clause, we have recognized that $x_1 = x_2$.

3. Interpretation

In 1979 and 1980, I had the huge good fortune to participate in a biweekly discussion group on discourse, alternating between Stanford and Berkeley, consisting of some of the most illustrious scholars of language in the world, including Mike Agar, Dwight Bolinger, Eve and Herb Clark, Chuck Fillmore, Paul Kay, George Lakoff, Geoff Nunberg, Ivan Sag, Dan Slobin, Elizabeth Traugott, and Tom Wasow. For me personally, the high point in these meetings, and one of the high points in my entire career, was when the sociologist Irving Goffman, visiting Berkeley at the time, used my paper "Conversation as Planned Behavior" (Hobbs and Evans 1980) as a club to beat the sociolinguist John Gumperz over the head with. Metaphorically speaking. We read and discussed members' papers on interpreting nominal compounds, metonymy or deferred reference, de-nominalized nouns, metaphor, and other phenomena that came to be clustered by linguists under the name of "Radical Pragmatics" (Cole 1981). (I thought a better name would be "Run-of-the-mill AI".)

Around this time, I was concerned with the problem of how we delimit the set of inferences we draw as we understand a text. The answer that seemed most promising was that we need to draw those inferences required to resolve interpretation problems of the sort we were examining in the discussion group. But what systematicity was there to this set of problems? How would you know if your list was complete?

The scheme that made the most sense to me goes like this. A text conveys predications, that is, a predicate applied to one or more arguments – $p(x)$. This gives rise to three sorts of problems:

1.   What is the predicate? What is p? This question subsumes the problems of lexical ambiguity, the interpretation of vague predicates like prepositions and *have*, and the interpretation of the implicit relation in nominal compounds.

2.   What is the argument? What is x? This question subsumes the problems of coreference and syntactic ambiguity. (Recall that syntactic structure is a matter of identifying variables in the right way.)

3.   In what way are the predicate and argument congruent? What about p and x would allow p to be true of x? This question subsumes the problems of metaphor and metonymy.

This collection of problems I called "local pragmatics." They are problems that are presented within the scope of single sentences, but they often require for their solution the entire discourse, the external context, and world knowledge. (My term never caught on probably because no one else saw this class of problems as a natural kind.)

Another issue I was thinking about during these years was the structure of discourse, in particular, that structure arising out of coherence relations between discourse segments. In this I was very much influenced by the work of the linguists Joseph Grimes (1975) and Robert Longacre (1976). I began collaborating with the anthropologist Mike Agar around this time, and we called this level of structure "local coherence" (Agar and Hobbs 1982).

In the mid-1970s Ray Perrault and Phil Cohen (Cohen and Perrault 1979) at the University of Toronto, later to be my colleagues at SRI, and Chip Bruce (Bruce and Newman 1978) at BBN were doing very exciting work analyzing the structure of discourse as arising out of the speaker's or writer's plan, employing formalizations of planning from artificial intelligence. In work with David Evans and work with Mike Agar I tried to apply these insights to the complexities of ordinary conversation and to ethnographic interviews. Agar and I called this level of structure "global coherence."

All along in investigating all three of these problems—local pragmatics, local coherence, and global coherence—it was clear that a key role was played by the notions of implicature (Grice 1975), accommodation (Lewis 1979; Thomason 1985), and abduction (Peirce 1955). To solve even elementary problems like pronoun coreference, one had to make assumptions to get a good interpretation of the text, where the only justification for the assumptions was that they led to a good interpretation.

In the fall of 1987 at SRI we organized a discussion group on abduction, reading the classic papers by Peirce, recent attempts in AI to use abduction in, for example, medical diagnosis (Pople 1973; Cox and Pietrzykowski 1986), and contemporary philosophers like Paul Thagard (1978), as well as work by Wilensky and Norvig at Berkeley (Wilensky 1983; Norvig 1987) and Charniak and Goldman at Brown (Charniak and Goldman 1988) that seemed to be taking an approach similar to ours. Among the people in our group were Mark Stickel, Doug Edwards, and the pragmatics scholar Steve Levinson, who was visiting Stanford at the time. We argued about what we were calling identity implicatures and referential implicatures, and about how to distinguish new from given information in discourse, and how to choose the best interpretation of a text.

Then late one afternoon in October 1987 Mark Stickel came into my office to say that he thought he had the answer to all our problems. He described his algorithm for weighted abduction. It struck me immediately as the double helix of computational linguistics, a feeling that has not entirely abandoned me today. First of all, it gave us a characterization of what constituted the interpretation of a stream of discourse. It gave us a clear criterion for what inferences to draw and not draw. The interpretation was the most economical explanation for what would make the text true, and an inference was appropriate if and only if it contributed to that explanation.

On my way home that night, I began driving a little more carefully. In the next few days, I saw how one would approach all the local pragmatics and local and global coherence problems in this framework. In discussions with Stu Shieber in the next few days it became apparent how one could integrate syntax smoothly into the framework. A big picture emerged (Hobbs et al. 1993).

In the early 1990s I saw an advertisement in a magazine for Polaroid cameras (quite obsolete now). It showed a man standing by the ocean, holding a camera, and looking at a scene in which the branch of a tree is on the ground and a small boat is stuck in the top of another tree. When we see this, we immediately interpret it by coming up with the best explanation for the observables (abduction). There was a storm that blew the branch down and blew the boat into the tree. There are other

possible explanations. Maybe someone chopped the branch down, and maybe the boat was lifted into the tree with a crane. But this is not as good an interpretation because we have to assume two things (the chopping and the crane) rather than just one (the storm). The first interpretation is better because it is more economical. Less explains more.

But this isn't the end of the story. There is another observable to be explained. Why is this picture in the magazine? The explanation is that it is an advertisement. That means there was an ad agency involved in posing the picture, and they very well could have done the chopping and used the crane, rather than wait for the rare event of a storm to arrange the picture for them.

We could call the first explanation the "informational" one. It explains the content of the picture, thereby explicating the information conveyed by the picture. We could call the second explanation the "intentional" one. It explains why the message occurs at all. Note that both interpretations need to be discovered if the advertisement is to be fully appreciated.

The big picture that emerges is this (see Figure 1). The brain is an abduction machine, continuously trying to prove abductively (i.e., by making necessary assumptions) that the observables in its environment constitute a coherent situation. (We can encompass action as well as perception by adding to what is proved the proposition that the owner of the brain will thrive in that situation.)

Sometimes among the observables is another agent's utterance. What is to be explained is the proposition $utter(i, u, w)$—that is, a speaker i utters to a hearer u a string
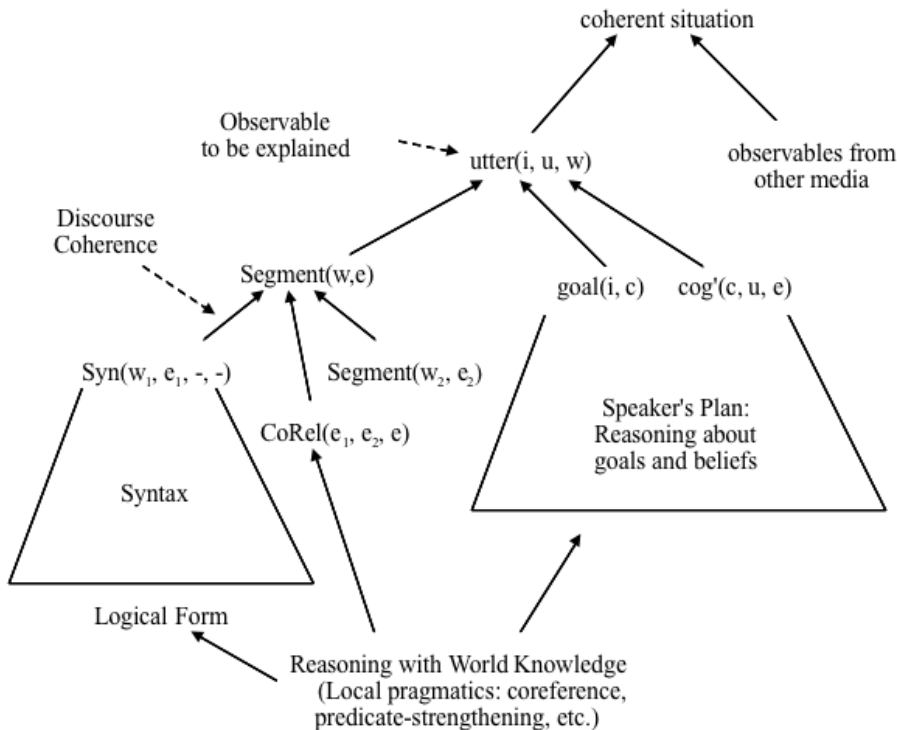
Figure 1
Interpretation as abduction, the big picture.

of words w. Generally the best explanation for an utterance is that it is an intentional act aimed at conveying information. We can capture this with the axiom

(13) $Segment(w, e) \land goal(i, c) \land cog'(c, u, e) \supset utter(i, u, w)$

That is, if w is an interpretable segment of discourse describing a situation e, and a speaker i has the goal c that a hearer u adopt some cognitive stance toward e, then (defeasibly) i will utter to u the string of words w. The first conjunct in the antecedent is the entry point into the informational side of an interpretation: What is the content of the message? The second two conjuncts are the entry point into the intentional side: Why is the speaker conveying this content?

The reason that the speaker has this particular goal is usually that it plays some role in, or is a subgoal of, a larger plan the speaker is executing in the world. This is where that reasoning occurs. It encompasses what Agar and I called "global coherence"—how does the utterance fit in with what else is going on in the world?

The next level of analysis happens when we decompose the segment of discourse into smaller segments, using the axiom

(14) $Segment(w_1, e_1) \land Segment(w_2, e_2) \land CoRel(e_1, e_2, e)$
         $\supset Segment(w_1 w_2, e)$

This axiom says that if w_1 is a segment describing situation e_1, and w_2 is a segment describing situation e_2, and there is a relation between e_1 and e_2, then the concatenation is a segment describing a situation e somehow derivable from the relation. When we backchain on this axiom, we are explaining an interpretable segment of discourse by breaking it into parts, explaining the parts, and explaining the relation between them.

The possible coherence relations are just the sort of relations that frequently obtain between two states or events: causality, similarity, identity, a strong sort of temporal succession I have called "occasion," the figure–ground relation, and predicate–argument relations. These are similar to other catalogues of discourse relations that others have come up with. However, the intent is to capture the information that can be conveyed by adjacency. By contrast, the relations of Rhetorical Structure Theory (Mann and Thompson 1986) are a mixture of informational relations like similarity and intentional relations like justification. The first is what is conveyed by adjacency; the second is what the speaker is using adjacency to do. Often the coherence relation conveyed by adjacency is expressed redundantly (and with less ambiguity) in a conjunction (*so*), an adverb (*consequently*), or a referential expression (*That made ...*). This does not pose a problem, assuming the two do not conflict; discourse is rife with redundancy.

Decomposition of a discourse in this fashion yields a tree or tree-like structure. It bottoms out in individual clauses, and this is where syntax takes over. Adjacency in larger stretches of discourse can convey a variety of possible relations. As we saw at the end of Section 2, adjacency within clauses conveys predicate–argument relations. Syntax is a set of rules that enable us to convey and interpret complex predicate–argument relations with the rather crude device of concatenation. The best explanation of a clause is the decomposition given to us by compositional semantics. The best explanation for an individual morpheme is that it is intended to convey its corresponding predication. Thus, the syntactic analysis of a clause bottoms out in its logical form.

Now all that remains to be explained is the logical form. It was the original insight of the "Interpretation as Abduction" framework that the best abductive proof (i.e., the best explanation) of the logical form solved the local pragmatics problems as a side effect.

I won't make an extended argument for that here, but one example should convey the basic idea.

The sentence, due to Hirst (1987),

(15)   The plane taxied to the terminal.

has three lexical ambiguities. A plane could be an airplane or a wood-smoother, a terminal could be an airport terminal or a computer terminal, and taxiing could be a plane moving on the ground or a person riding in a cab.

We assume we have axioms expressing these possibilities

airplane(*x*) ⊃ *plane*(*x*)
wood-smoother(*x*) ⊃ *plane*(*x*)
airport-terminal(*y*) ⊃ *terminal*(*y*)
computer-terminal(*y*) ⊃ *terminal*(*y*)
move-on-ground(*x*, *y*) ∧ *airplane*(*x*) ⊃ *taxi*(*x*, *y*)
ride-in-cab(*x*, *y*) ∧ *person*(*x*) ⊃ *taxi*(*x*, *y*)

together with a rule that says airports have airplanes and airport terminals.

airport(*z*) ⊃ *airplane*(*x*) ∧ *airport-terminal*(*y*)

Then the most economical explanation (Figure 2) is constructed by assuming there is an airport and that an airplane we expect to find there is moving on the ground to the airport terminal we expect to find there. Note that the ambiguous words are disambiguated as a by-product by virtue of the axioms that are used in the explanation. The predicate *airport-terminal* plays a role; the predicate *computer-terminal* doesn't.

All of this raises a question. If the framework is so elegant and so all-encompassing, why isn't it more widely adopted?

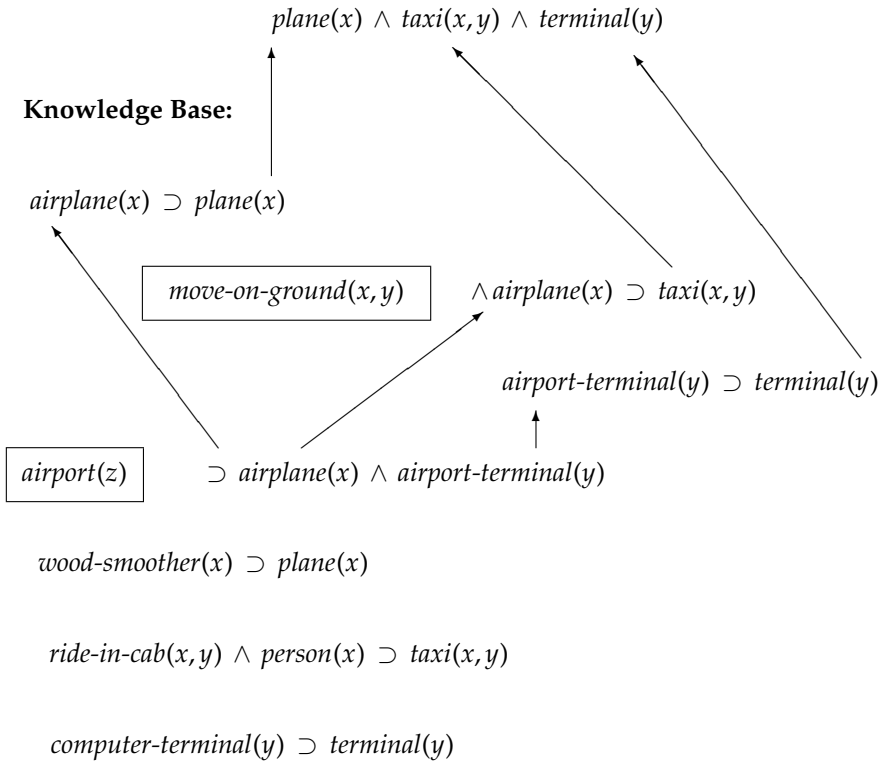I think there are three reasons for this, historically.

1.   Parsers were not accurate enough to produce good logical forms from which inference could start.

2.   Algorithms for abduction were too inefficient.

3.   There was a lack of an adequate knowledge base.

Each of these problems has been alleviated somewhat in the past few years. There are now highly accurate statistical parsers, and for several of these (e.g., Boxer; Bos 2008) a component for translating into a flat logical form has been implemented.

Recent work by Naoya Inoue and Kentaro Inui (2011) implements weighted abduction as a problem in integer linear programming, building on earlier work by Charniak and Santos (Santos 1996). Our experience with this is that when we switched from a naive backchaining implementation to the ILP implementation, we got a speed-up of two orders of magnitude.

Finally, there have been ongoing efforts to build large knowledge bases, manually and automatically, from a number of different perspectives. Efforts to use Cyc for natural language processing applications have had mixed success at best. But Schubert's efforts (2002) to build a knowledge base by analyzing language use looks very promising. Some applications have attempted to use OpenMind. WordNet hierarchies are used very widely and Harabagiu and Moldovan (2002) developed XWN, a conversion of WordNet glosses into logical axioms, and reported success with its use in question-

Logical Form:

$$plane(x) \wedge taxi(x,y) \wedge terminal(y)$$

Knowledge Base:

$airplane(x) \supset plane(x)$

$\boxed{move\text{-}on\text{-}ground(x,y)} \qquad \wedge airplane(x) \supset taxi(x,y)$

$airport\text{-}terminal(y) \supset terminal(y)$

$\boxed{airport(z)} \qquad \supset airplane(x) \wedge airport\text{-}terminal(y)$

$wood\text{-}smoother(x) \supset plane(x)$

$ride\text{-}in\text{-}cab(x,y) \wedge person(x) \supset taxi(x,y)$

$computer\text{-}terminal(y) \supset terminal(y)$

Figure 2
Interpretation of *The plane taxied to the terminal*.

answering. FrameNet has been converted into logical axioms by Ovchinnikova et al. (2013), and she and her colleagues have shown that an abduction engine using a knowledge base derived from these sources is competitive with the best of the statistical systems in textual entailment and semantic role labeling.

My own particular take on building a knowledge base for inferential NLP is described in the next section.

4. Knowledge

We understand discourse so well because we know so much. Thus, one of the central problems in the study of language is how we use our knowledge of language and the world to interpret discourse. This breaks into two subproblems:

1.   How do we encode the common-sense knowledge required for understanding discourse?

2.   How do we use this knowledge in the processing of discourse?

I had a conversation with Eugene Charniak in the early 1990s in which I said I thought the second of these is a solved problem. The answer is abduction. He agreed

with me. We both agreed that the first problem was now the most important focus of research. But he said that he despaired of encoding that knowledge manually, and that's why he had reoriented his research toward statistical methods. I disagreed for two reasons. I think the kind of knowledge that we want at the very core of a knowledge base for NLP can only be done manually by thoughtful people and cannot be done by any automatic methods currently imaginable. And I think there is systematicity that will make the task more tractable than we might believe at the outset.

As to the first point, suppose we want to define or characterize the word *range*, as in

(16)   The scores on the test ranged from 38 to 96.

From the definition we would want to be able to answer the questions

Did someone get a 96 on the test? Yes.
Did someone get a 54 on the test? Maybe.
Did someone get a 25 on the test? No.

In addition, we want to capture generalizations and we don't want to multiply word senses needlessly. So we would like to have the same definition work for the sentences

(17)   The timber wolf ranges from northern Mexico to southern Alaska.

(18)   His behavior ranges from sullen to downright hostile.

(19)   The hepatitis cases range from moderate to severe.

We don't want a "species" sense of *range*, and a "behavioral" sense and an "epidemiological" sense.

The sort of axiom we need for this is as follows:

(20) $(\forall x, y, z) range(x, y, z) \equiv$
          $(\exists s, s_1, u_1, u_2) scale(s) \wedge subscale(s_1, s) \wedge bottom(y, s_1)$
              $\wedge\, top(z, s_1) \wedge u_1 \in x \wedge at(u_1, y) \wedge u_2 \in x \wedge at(u_2, z)$
              $\wedge\, (\forall u \in x)(\exists v \in s_1) at(u, v)$

That is, x ranges from y to z if and only if there is a scale s with a subscale s_1 whose bottom is y and whose top is z, such that some member u_1 of x is at y, some member u_2 of x is at z, and every member u of x is at some point v in s_1. (I'll discuss the predicate *at* subsequently.)

It is difficult for me to believe we will any time soon be able to discover automatically rules of this complexity and at the same time rules of this level of abstractness. I'm sure we'll be able to discover automatically facts such as "One has to be married before getting divorced," and "Houses normally have thermostats." But facts like the definition of "range" require human brains.

My formative experience in encoding common-sense knowledge came when I was at Yale in the early 1970s and immersed myself in the linguistics literature. Among the papers that struck a chord the most were the Generative Semanticists, like Jeffrey Gruber, George Lakoff, Haj Ross, James McCawley, and others. They were analyzing the verb *kill* into *cause to become not alive* and the verb *move*, as in *x moves y from z to w*, into *x causes a change from y being at z to y being at w*. They also speculated on the abstract nature of the *at* relation as a source for many of the frozen spatial metaphors that pervade language.

Generative semantics dropped out of favor rather soon, but I think their fundamental insights were exactly right. To my mind, they failed for two reasons. First, they were doing in tree transformations what they should have been doing in logic,

a mistake being repeated today by those working on so-called "natural language inference." Second, they lacked a notion of defeasibility, so that when they found examples of when *X killed Y* and Y didn't end up not alive, they thought their theory was refuted.

These interests in lexical semantics got put on the back burner for several years. I returned to it in the mid-1980s when Bill Croft, Doug Edwards, Ken Laws, and I worked on building up a knowledge base. Our goal, which we almost achieved, was to be able to prove as a theorem that wear on a component of an artifact can cause the artifact to fail, because wear is a loss of material and this causes a change of shape, and shape in artifacts is normally functional. At the time we were working on U.S. Navy texts dealing with worn-out air compressors.

Then back to the back burner until 1999 or 2000, since which time knowledge encoding has been the principal focus of my research. It is not easy research to get funding for, because its payoff in comparison to building special-purpose applications is very long-term. One has to find short-term applications that would be helped by general knowledge in the next logical domain to attempt. For example, I was able to work with people like George Ferguson, Pat Hayes, and Drew McDermott on developing the so-called "OWL-Time," a comprehensive ontology of time (Hobbs and Pan 2004), for DARPA's DAML program on the Semantic Web, and ARDA's AQUAINT program on question-answering provided the resources for my work with Feng Pan and Rutu Mulkar-Mehta on vague durations of events. Ram Nevatia's ARDA-sponsored MOVER project provided the opportunity to develop an ontology of event structure called VERL (Video Event Representation Language, Alexandre et al. 2005), and this led to work with Chris Welty, Mike Gruninger, and people at Cycorp on the ARDA-sponsored IKRIS project for developing an interlingua among several event and process ontologies. DARPA's Machine Reading Program supported my student Rutu Mulkar-Mehta's work on granular or "how-to" causality (Mulkar-Mehta, Hobbs, and Hovy 2011) and Niloofar Montazeri's work defining or characterizing several hundred common event-related words (Montazeri and Hobbs 2011). My work with Andrew Gordon on encoding common-sense psychology (Gordon and Hobbs 2004) has been funded by various agencies over the years, most recently by ONR. But some of the research has been "stealth" research—work you don't tell anyone about until it's finished for fear your boss will find out and make you work on other stuff. My papers on causality and modality (Hobbs 2005) and on scales and half orders of magnitude (Hobbs 2000) were like this.

The goal is to develop what I have come to call "Deep Lexical Semantics" (Hobbs 2008). It is not enough to decompose "move" into "cause - change - at." It is not good enough to simply stipulate these as primitives. We need to explicate these concepts in core theories, a theory of causality, a theory of change of state, and a theory of composite entities and the figure–ground relation. Lexical decompositions have to be anchored in such theories so we can not only decompose meanings but also be able to reason with the decomposed meanings.

The structure of the effort is this: We have the predicates corresponding to the morphemes of the language. We have the underlying core theories. And we have axioms defining or characterizing the former in terms of the latter. Thus, in the "range" example, *range* is the predicate corresponding to the morpheme. There is a core theory of scales that provides the predicates *scale*, *lessThan*, *subscale*, *top*, *bottom*, and *at*. Axiom (20) is the rule that links the lexical predicate with the core theory.

Next I will sketch several very basic core theories and show their utility in defining words for the textual entailment task.

Composite Entities and the Figure–Ground Relation: A composite entity is a thing made of other things. This is intended to cover physical objects like a telephone, mixed objects like a book, abstract objects like a theory, and events like a concert. It is characterized by a set of components, a set of properties of the components, a set of relations among its components (the structure), and relations between the entity as a whole and its environment (including its function). The predicate *at* relates an external entity, the figure, to a component in a composite entity, the ground. Different figures and different grounds give us different meanings for *at*.

(21)   Spatial location: Pat is at the back of the store.

(22)   Location on a scale: Nuance closed at 58.

(23)   Membership in an organization: Pat is now at Google.

(24)   Location in a text: The table is at the end of the article.

(25)   Time of an event: At that moment, Pat stood up.

(26)   Event at event: Let's discuss that at lunch.

(27)   At a predication: She was at ease in his company.

When *at* is specialized in this way, we tap into a whole vocabulary for talking about the domain, including concepts like *move* and *range*.

Change of State: The predication $change(e_1, e_2)$ says that state e_1 changes into state e_2. Its principal properties are that e_1 and e_2 should have an entity in common—a change of state is a change of state of *something*. States e_1 and e_2 are not the same unless there is an intermediate state. The predicate *change* is defeasibly transitive; in fact, backchaining on the transitivity axiom is one way to refine the granularity on processes.

Causality: We distinguish between the "causal complex" for an effect and the concept "cause." A causal complex includes all the states and events that have to happen or hold in order for the effect to occur. We say that flipping a switch causes the light to go on. But many other conditions must be in the causal complex—the light bulb can't be burnt out, the wiring has to be intact, the power has to be on in the city, and so on. The two key properties of a causal complex are that when everything in the causal complex happens or holds, so will the effect, and that everything that is in the causal complex is relevant in a sense that can be made precise. "Causal complex" is a rigorous or monotonic notion, but its utility in everyday life is limited because we almost never can specify everything in it.

"Cause" by contrast is a defeasible or nonmonotonic notion. It selects out of a causal complex a particular eventuality that in a sense is the "active" part of the causal complex, the thing that isn't necessarily normally true. Flipping the switch, in most contexts, is the action that *causes* the light to come on. Causes are the focus of planning, prediction, explanation, and interpreting discourse, but not diagnosis, because in diagnosis, something that normally happens or holds, doesn't.

Let us now define a few words in terms of these predicates. The verb *let* as in *x lets e happen* means x does not cause e not to happen.

(28) $let(x, e_1) \equiv not(e_8) \land cause'(e_8, x, e_9) \land not'(e_9, e_1)$

In its most abstract sense *go* just means a change of state, as in *Sometimes I go crazy*.

(29) $go'(e_1, y, e_2, e_3) \equiv change'(e_1, e_2, e_3)$

Something y is *free* of a set of constraints c to do e_5 means that it is not the case that the constraints cause e_5 not to happen.

(30) $free'(e_3, y, c, e_5) \equiv not'(e_3, e_2) \wedge cause'(e_2, c, e_4) \wedge not'(e_4, e_5)$

Something x holds y at z means that x causes y not to change from being at z.

(31) $hold'(e_2, x, y, z)$
 $\equiv cause'(e_2, x, e_4) \wedge not'(e_4, e_5) \wedge change'(e_5, e_6, e_7) \wedge at'(e_6, y, z)$

Something x releases y from z means that there is a change from x's holding y at z.

(32) $release'(e_1, x, y, z) \equiv change'(e_1, e_2, e_3) \wedge hold'(e_2, x, y, z)$

The Recognizing Textual Entailment task is to determine from a text whether a hypothesis follows from it or not. For example, from the text *A Filippino hostage in Iraq was released* we would like to be able conclude the hypothesis *The captors let the hostage go free*. Figure 3 illustrates the proof of this entailment relation, using the five axioms we just wrote, together with rules from the core theories saying if something exists, nothing causes it to not exist, and if there is a change from a state, that state no longer holds (Montazeri and Hobbs 2011).

The final set of examples I'll give of Deep Lexical Semantics come from work I have been doing with Andrew Gordon on axiomatizing common-sense psychology, or how we think we think. We have developed approaches to memory, belief, and mutual belief, envisioning causal chains in explanation and prediction, perception and control of the body, and goals and plans. I will focus on goals.

We adopt the strong AI position that people are in a sense planning mechanisms. We have goals, we develop plans to achieve these goals, we execute the plans, we monitor the execution, and if things go wrong, we modify our plans and execute the new plans.
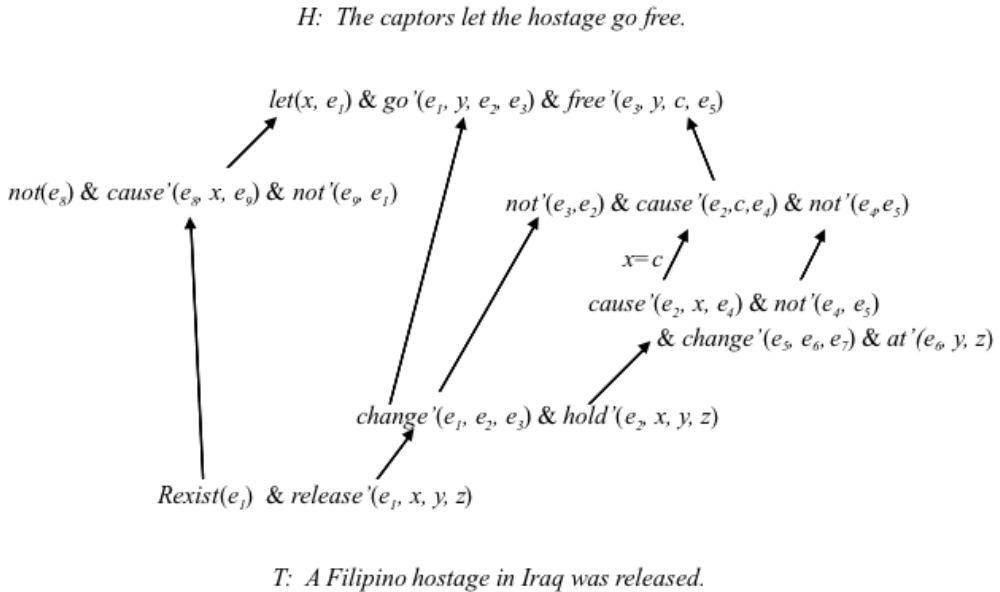
Figure 3
A textual entailment example.

There are two chief properties of goals. The first says that if an agent has a goal e_2 and believes e_1 causes e_2, then defeasibly that will cause the agent to have e_1 as a subgoal. The second property is a similar rule for enablement. These are the planning axioms; they generate hierarchical plans.

The word *help* can be explicated in terms of theories of goals and causality. We can distinguish three levels of helping. At the lowest level, inadvertent helping, you help someone when you do an action that is in a causal complex for one of their goals. In this sense, John McCain helped Barack Obama get elected by picking Sarah Palin as his running mate. A second level, intentional helping, is like the first with the addition that the helper performs the action in the service of the helpee's goal. For example, if I take away a drunk friend's car keys, I help him survive, a goal of his, but not driving is no part of his plan to survive. The third level, collaborative helping, happens when the helper and helpee engage in a shared plan together, as in helping someone carry a sofa.

We can define the word *try*, as in *X tries to do E*, as having the goal that E be accomplished, and having it be a goal cause one to accomplish one of its subgoals. To *succeed* at doing E is to try to do E, and to have that trying cause E to be accomplished. To *fail* to do E is to try to do E and have E not happen.

Often in comprehending discourse about artifacts we need to know a great deal about the structure and function of the artifact. The theory of goals and planning gives us a handle on this, because normally the structure of an artifact reflects a plan to achieve its functionality, a goal. For example, the function of a coffee cup is to move coffee. We achieve this by breaking it into two subgoals: having a cup contain the coffee and moving the cup. We achieve moving the cup by attaching a handle to the cup and moving the handle. Artifacts are plans made concrete (or in the case of a coffee cup, ceramic).

One of the most salient aspects of common-sense psychology is our emotions. What about emotions? Is it possible to formalize a theory of emotions? I think it is, and much of it involves goals. In general, we can characterize emotions in terms of what causes them and what they cause. Emotions, like cognition more generally, mediate between perception and action. Thus, for a particular emotion, we specify an abstract type of perceived eventualities that cause the emotion, and we specify an abstract class of typical responses. This is basically the knowledge we humans need to fake emotions.

Consider happiness. Happiness occurs when one's goals are being satisfied (or sometimes merely when we anticipate that). That must mean that one's beliefs in the relevant area are working, especially one's beliefs about what causes what. So one effect of happiness is a higher level of activity—we plan to do more because our planning process is in good working order. A second effect of happiness is that we are not very open to a change of beliefs. If our beliefs are working, why should we change them?

Other emotions can be defined in similar terms. Sadness is the opposite of happiness in cause and effects. Fear, anger, and disgust can be seen as various responses to a threat, given the properties of the threat, where a threat is something that will cause one's goals to be defeated.

I'll close with an obvious question. How long will it be before we are able to automatically analyze texts in the manner I have described? When I wrote my technical report analyzing one paragraph of *Newsweek* in 1976, I thought the answer was that the goal was ten years away. When we began to implement a system based on weighted abduction, I thought the goal was ten years away. So now I will show myself to be consistently optimistic. I think a concerted effort along these lines would yield some measure of success in about ten years.

References

Agar, Michael, and Jerry R. Hobbs. 1982. Interpreting discourse: Coherence and the analysis of ethnographic interviews. *Discourse Processes*, 5(1):1–32.

Alexandre, Francois R. J., Ram Nevatia, Jerry R. Hobbs, and Robert C. Bolles. 2005. VERL: An ontology framework for representing and annotating video events. *IEEE Multimedia*, 12:76–86.

Bos, Johan. 2008. Wide-coverage semantic analysis with Boxer. In J. Bos and R. Delmonte, editors, *Semantics in Text Processing*. Proceedings of STEP 2008 Conference, Research in Computational Semantics, College Publications, Venice, Italy, pages 277–286.

Bruce, Bertram C., and Dennis Newman. 1978. Interacting plans. *Cognitive Science*, 2:195–233.

Charniak, Eugene, and Robert Goldman. 1988. A logic for semantic interpretation. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, NY, pages 87–94.

Cohen, Philip, and C. Raymond Perrault. 1979. Elements of a plan-based theory of speech acts. *Cognitive Science*, 3(3):177–212.

Cole, Peter, editor. 1981. *Radical Pragmatics*. Academic Press, New York.

Cox, P. T., and T. Pietrzykowski. 1986. Causes for events: Their computation and applications. In *Proceedings of Eighth International Conference on Automated Deduction (CADE-8)*, pages 608–621, Oxford.

Davidson, Donald. 1967. The logical form of action sentences. In N. Rescher, editor, *The Logic of Decision and Action*, University of Pittsburgh Press, Pittsburgh, PA, pages 81–95.

Gordon, Andrew, and Jerry R. Hobbs. 2004. Formalizations of commonsense psychology. *AI Magazine*, 25:49–62.

Grice, H. P. 1975. Logic and conversation. In P. Cole and J. Morgan, editors, *Syntax and Semantics*, Academic Press, New York, pages 41–58.

Grimes, Joseph. 1975. *The Thread of Discourse*. Mouton and Company, The Hague, Netherlands.

Hirst, Graeme, 1987. *Semantic Interpretation and the Resolution of Ambiguity*. Cambridge University Press, Cambridge, UK.

Hobbs, Jerry R. 1976. A computational approach to discourse analysis. Research Report 76-2, Department of Computer Sciences, City College, City University of New York, New York, pages 28–38.

Hobbs, Jerry R. 2000. Half orders of magnitude. In *Proceedings of KR-2000 Workshop on Semantic Approximation, Granularity, and Vagueness*, Breckenridge, CO.

Hobbs, Jerry R. 2005. Toward a useful concept of causality for lexical semantics. *Journal of Semantics*, 22:181–209.

Hobbs, Jerry R. 2008. Deep lexical semantics. *Proceedings of 9th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2008)*, Haifa, Israel, pages 183–193.

Hobbs, Jerry R., Douglas E. Appelt, John Bear, David Israel, Megumi Kameyama, Mark Stickel, and Mabry Tyson. 1997. FASTUS: A cascaded finite-state transducer for extracting information from natural-language text. In E. Roche and Y. Schabes, editors, *Finite State Devices for Natural Language Processing*. MIT Press, Cambridge, MA, pages 383–406.

Hobbs, Jerry R., and David Andreoff Evans. 1980. Conversation as planned behavior. *Cognitive Science*, 4(4):349–377.

Hobbs, Jerry R., and Feng Pan. 2004. An ontology of time for the Semantic Web. *ACM Transactions on Asian Language Information Processing*, 3(1):66–85.

Hobbs, Jerry R., Mark Stickel, Douglas Appelt, and Paul Martin. 1993. Interpretation as abduction. *Artificial Intelligence*, 63(1-2):69–142.

Inoue, Naoya, and Kentaro Inui. 2011. ILP-based reasoning for weighted abduction. In *Proceedings of AAAI Workshop on Plan, Activity and Intent Recognition*, San Francisco, CA, pages 25–32.

Lakoff, George, and Mark Johnson. 1980. *Metaphors We Live By*. University of Chicago Press, Chicago.

Lewis, David. 1979. Scorekeeping in a language game. *Journal of Philosophical Logic*, 6:339–359.

Longacre, Robert. 1976. *An Anatomy of Speech Notions*. The Peter de Ridder Press, Ghent, Belguim.

Mann, William, and Sandra Thompson. 1986. Relational propositions in discourse. *Discourse Processes*, 9(1):57–90.

Moldovan, Dan, Sanda Harabagiu, Roxana Girju, Paul Morarescu, Finley Lacatusu, Adrian Novischi, Adriana Badulescu, Orest Bolohan. 2002. LCC tools for question answering. In *Proceedings of the Eleventh Text Retrieval Conference* (TREC

2002), Gaithersburg, MD. Available at
`http://trec.nist.gov/pubs/trec11/`
`t11_proceedings.html`, 21:1–10.

Montazeri, Niloofar, and Jerry R. Hobbs.
2011. Elaborating a knowledge base for
deep lexical semantics. In *Proceedings
of the Ninth International Conference on
Computational Semantics (IWCS 2011)*,
Oxford, England, pages 195–204.

Mulkar-Mehta, Rutu, Jerry R. Hobbs, and
Eduard Hovy. 2011. Applications and
discovery of granularity structures
in natural language discourse.
In *Proceedings of the 10th Symposium
on Logical Formalizations of
Commonsense Reasoning*. Available at
`http://commonsensereasoning.`
`org/2011/proceedings.html`

Norvig, Peter. 1987. Inference in text
understanding. In *Proceedings,
AAAI-87, Sixth National Conference on
Artificial Intelligence*, pages 561–565,
Seattle, WA.

Ovchinnikova, Ekaterina, Niloofar
Montazeri, Teodor Alexandrov, Jerry R.
Hobbs, Michael C. McCord, and
Rutu Mulkar-Mehta. 2013. Abductive
reasoning with a large knowledge
base for discourse processing.
Computing Meaning, 4:104–124.

Pan, Feng, Rutu Mulkar-Mehta, and Jerry
Hobbs. 2011. Annotating and learning
event durations in text. *Computational
Linguistics*, 37(4):727–752.

Peirce, Charles Sanders. 1955. Abduction
and induction. In Justus Buchler,
editor, *Philosophical Writings of
Peirce*, Dover Books, New York,
pages 150–156.

Pollard, Carl, and Ivan A. Sag. 1994.
Head-Driven Phrase Structure Grammar.
University of Chicago Press and
CSLI Publications.

Pople, Harry E., Jr. 1973, On the
mechanization of abductive logic.

In *Proceedings of Third International Joint
Conference on Artificial Intelligence*,
pages 147–152, Stanford, California.

Richards, I. A. 1936. *The Philosophy of
Rhetoric*, Oxford University Press, Oxford.

Rieger, Charles J., III. 1974. Conceptual
memory: A theory and computer program
for processing the meaning content of
natural language utterances. Memo
AIM-233, Stanford Artificial Intelligence
Laboratory, Stanford University.

Sager, Naomi. 1981. *Natural Language
Information Processing: A Computer
Grammar of English and Its Applications*.
Addison-Wesley, Reading, MA.

Santos, Eugene. 1996. Polynomial solvability
of cost-based abduction. *Artificial
Intelligence*, 86:157–170.

Schubert, Lenhart K. 2002. Can we derive
general world knowledge from texts?
In *Proceedings of the 2nd International
Conference on Human Language Technology
Research (HLT 2002)*, pages 94–97, San
Diego, CA.

Smith, Kate, Ethel B. Magee, and S. S.
Seward, Jr. 1928. *English Grammar: Correct
and Effective Use*. The Athenaeum Press,
Ginn and Co., Boston, MA.

Thagard, Paul R. 1978. The best explanation:
Criteria for theory choice. *The Journal of
Philosophy*, 75(2):76–92.

Thomason, Richmond H. 1985.
Accommodation, conversational planning,
and implicature. In *Proceedings of Workshop
on Theoretical Approaches to Natural
Language Understanding*, Halifax,
Nova Scotia, pages 117–126.

Vico, Giambattista, 1968 [1744]. *The New
Science of Giambattista Vico*, T. Bergin and
M. Frisch, translators. Cornell University
Press, Ithaca, NY.

Wilensky, Robert. 1983. *Planning and
Understanding: A Computational Approach to
Human Reasoning*. Addison-Wesley,
Reading, MA.