Logic Puzzles: A New Test-Suite for Compositional Semantics and Reasoning

Iddo Lev
Computer Science Department
Stanford University
iddolev@stanford.edu

Abstract

This paper suggests the creation of a new reasoning test collection for natural language understanding systems. The test-suite would consist of logic puzzles and would pose a new challenge in Natural Language and Reasoning. It would also present an incentive for developing domain-independent computational knowledge of structural (compositional) semantics. Such knowledge would be useful in both precise understanding applications and statistical NLP. The unique characteristics and benefits of this test-suite are explained and compared with the FraCaS and RTE test-suites, and interesting phenomena in the puzzles are surveyed.

1 Introduction

Test suites are useful for pushing research in NLP forward as they present an incentive for developing computerized NL-analysis capabilities that can be useful in NLP applications. They also provide evaluation metrics for those capabilities and a way to compare the performance of different systems.

In recent years, a number of test-suites have been developed which target text meaning and understanding, including: MUC, TREC QA, FraCaS, and RTE. However, there is room for creating a new test-suite that would target both a high level of precision in understanding complex semantic constructions and more complex reasoning. This paper proposes such a test-suite based on logic puzzles.

Section 2 briefly reviews existing semantic test-suites. Section 3 discusses the importance of structural semantics for NLP. Section 4 proposes a new test-suite of logic puzzles and explains why it is particularly suitable for developing knowledge of structural semantics. Section 5 surveys some interesting semantic and other phenomena that appear in logic puzzle texts. Section 6 discusses how systems for solving logic puzzles could be evaluated, and section 7 concludes.

My hope is that this paper will inspire many researchers to accept the challenge of constructing a system for solving logic puzzles and to develop the computational knowledge of structural semantics necessary for it. This paper therefore does not sketch potential architectures for a puzzle solving system because the goal here is to just introduce the task and to leave it to researchers to come up with different solutions.

[0]I am grateful to Stanley Peters for many useful discussions that eventually led to this paper. Thanks also to Lauri Karttunen and Bill MacCartney for useful comments on a draft of this paper.

2 Background: Existing Test-Suites

The MUC and TREC QA test-suites[1] focus on information retrieval and extraction and on simple-fact queries whose answers explicitly appear in the texts. These tasks rely on very little if any reasoning.

Two NL understanding test-suites that do rely on reasoning are FraCaS and RTE. The FraCaS test-suite [Fra96, pp.63-120] is a set of simple text-understanding tests. The computer's task is to answer a question based on information given in a text consisting of a small number of sentences, usually one or two. The order of the sentences is often unimportant, although sometimes it is important due to anaphoric relations between them. The questions mostly test the computer's ability to identify logical entailments, equivalences, and contradictions between the meanings of sentences. Here are some examples:

(1) A Swede won a Nobel prize.
Every Swede is a Scandinavian.
Did a Scandinavian win a Nobel prize? [Yes]

(2) A Scandinavian won a Nobel prize.
Every Swede is a Scandinavian.
Did a Swede win a Nobel prize? [Don't know]

(3) John went to Paris by car, and Bill by train.
Did Bill go to Paris by train? [Yes]

(4) The PC-6082 is faster than the ITEL-XZ.
The PC-6082 is slow.
Is the ITEL-XZ fast? [No]

This collection of text-question pairs is designed to target specific phenomena in structural semantics[2] as well as in syntax and discourse, including quantifiers, negation, plurals, anaphora, ellipsis, adjectives, comparatives, and more.

A more recent test-suite was made available in the Pascal Recognizing Textual Entailment (RTE) challenge [DGM05].[3] The task is similar to the FraCaS task in that the computer needs to decide whether the meaning of a hypothesis sentence can be inferred from a given short text. Here are some examples from the first RTE test-suite:

(5) T: Mexico City has a very bad pollution problem because the mountains around the city act as walls and block in dust and smog.
H: Poor air circulation out of the mountain-walled Mexico City aggravates pollution.
[Follows]

[1] See http://www-nlpir.nist.gov/related_projects/muc/ and http://trec.nist.gov/data/qa.html.

[2] The term *compositional semantics* is usually used instead. I prefer the term *structural semantics* as it distinguishes this branch of semantics from lexical semantics, which also addresses issues of composition, e.g. noun-noun compounds or the alteration in meaning that verbs and nouns undergo when they are combined in a metonymy.

[3] See also http://www.pascal-network.org/Challenges/RTE/Introduction/.

(6) T: While civilians ran for cover or fled to the countryside, Russian forces were seen edging their artillery guns closer to Grozny, and Chechen fighters were offering little resistance.
H: Grozny is the capital of Chechnya. [Does not Follow]

(7) T: The Mona Lisa, painted by Leonardo da Vinci from 1503-1506, hangs in Paris' Louvre Museum.
H: The Mona Lisa is in France. [Follows]

There are at least two important differences between these two test-suites. First, the sentences in the FraCaS test-suite were manually constructed and were deliberately designed to test particular NL phenomena. In contrast, the RTE test-suite is based on naturally-occurring texts from news sources (with some modifications).

Second, the notion of inference in the FraCaS test-suite is mostly strict logical or linguistic entailment, and it does not assume any background knowledge (if such knowledge is needed, it is stated explicitly in the text, as in (1)-(2) above). In contrast, the inference targeted by the RTE test-suite is based on speaker meaning and is more fuzzy, plausible, non-monotonic. It also assumes some background knowledge that a common person who reads the newspaper possesses (facts such as "Paris is the capital of France"). However, a hypothesis should not be accepted as inferrable from a text if it is directly inferrable from just the background knowledge – compare (6) and (7) above. It is less clear how to define the notion of inference in RTE compared to FraCaS, though there is a high inter-annotator agreement on the test-suite [DGM05]. See [ZKC05, Man06, CKZ06] for further discussions.

3 Computational Knowledge of Structural Semantics

The main benefit of the test-suite to be proposed below is the incentive it will present for developing computational knowledge of structural semantics in the setting of a reasoning task. Because this kind of knowledge is largely neglected in contemporary mainstream NLP, I spend a fair amount of space discussing its importance.

3.1 What Is Knowledge of Structural Semantics?

Roughly speaking, structural semantics deals with the literal meaning of sentences but abstracted away from the concepts that open-category (non-functional) words are related to. It is similar to the semantics of a formal language, where what the symbols in the vocabulary stand for is irrelevant for defining the semantics of the language abstractly, and the only thing that matters is the syntax of the language and the meaning of the logical connectives. Structural semantics explains how the structural meaning of a sentence is related to the syntactic structure of the sentence and the meaning of functional units (the equivalent of logical connectives in a formal language), which include morphological markers, quantifiers, determiners, logical connectives, modals, auxiliary words, pronouns and relative pronouns, some prepositions, and ellipsis markers. Topics in structural semantics include:

- Analysis of the meaning of these functional units and their contribution to the sentence's meaning.

- How the meaning of a phrase is related to the meanings of its parts (the structural syntax-semantics interface). This is non trivial for many constructions, including relative clauses, coordination, comparative constructions, and due to scope ambiguities of quantifiers, modals, negation, etc.

- The influence of functional units and syntax on various topics in semantics which are also affected by lexical semantics and by context, including tense/aspect/event structure, anaphora, the reconstruction of missing semantic material in ellipsis constructions, as well as presuppositions and scalar implicatures.

- Ambiguity management for scope ambiguities, plurality ambiguities, and the context-independent aspects of anaphora and ellipsis ambiguities.

As a simple illustration, the inference in (8), which is based only on structural semantics, does not require knowing the meaning of the words *pencheon* and *sopomous*. It only requires knowing that they are a noun and an adjective, respectively. To utilize such tests, one also needs to have an accurate knowledge of morphology and syntax as prerequisites for the structural semantics stage.

(8) Every pencheon is sopomous.
John is a pencheon.
⇒ John is sopomous.

In contrast, the inference in (9)a is not *merely* a structural semantic inference because it depends on the meaning of the words *man* and *human* and on world knowledge about the connection between those meanings (i.e. between the concepts that those words stand for). However, if this knowledge is given to us in explicit form, as in (9)b, the pattern in (9)a can still provide a test for structural semantic knowledge.[4]

(9) a. Every human likes watermelons.
⇒ Every man likes watermelons.

b. $\forall x.[man(x) \rightarrow human(x)]$

The inference in (10)a and the difference between (10)a and (10)b depend not only on the meaning of the words but also on linguistic knowledge about argument realization – how thematic roles are connected to syntactic roles. This knowledge is intimately connected to issues of knowledge representation and the ontology, it is complex and very wide in scope, and is not considered part of structural semantics.

(10) a. John loaded the wagon with hay.
⇒ The wagon became full [at that time].

b. John loaded hay on the wagon.
$\not\Rightarrow$ The wagon became full [at that time].

[4] An NL sentence expressing background knowledge may itself introduce ambiguities and require more world knowledge. Since the goal here is to test only the understanding of (9)a, the background knowledge in (9)b is formalized rather than given as an NL sentence.

3.2 Importance of Structural Semantic Knowledge

Knowledge of structural semantics, by itself, is insufficient for most NL inferences in practice. Most inferences that are needed in applications depend heavily on lexical semantics and on domain and general world knowledge. However, structural semantics is a necessary component in any system that does interesting inference based on the *meaning* of NL input and the *information* conveyed by it. Without this knowledge, one is limited to just the meaning of words and only very rudimentary combinations of them.

Furthermore, this knowledge is largely domain-independent, and so it possesses a level of generality higher than other kinds of knowledge. Thus, once it is developed, it would have a large impact on many NLP applications, and it could be used for different purposes with little customization (the main customization might be adding frequencies of various phenomena, which may differ across domains). The next section surveys some of these applications.

Finally, of all the kinds of semantic knowledge needed in a sophisticated NL understanding system (including lexical knowledge and world-knowledge in ontology and facts), the body of structural semantic knowledge has the smallest size, and so we have a good chance to capture all or almost all of it through a concentrated collaborative effort in a reasonable amount of time.

3.3 Applications

3.3.1 Precise Understanding

Applications that aim at a high-quality precise understanding of NL in combination with conceptual knowledge and inference will benefit from a more complete and precise structural semantic knowledge, as it will extend the range of phenomena that can be handled and raise the level of understanding.

One example is applications that accept controlled language as input, i.e. a restricted subset of NL that is so precisely delineated that in effect it becomes a formal language that just looks (almost) like English.[5] Such languages have been used in manuals and other technical texts in various area of industry (such as the aero-space industry[6]) and in other specification tasks (see e.g. [FKS06][7]). While these applications differ from other NLP applications in that they stipulate away any NL ambiguity, they do share the need for high-quality, broad knowledge of structural semantics.

Another area that could directly benefit from high-precision knowledge of structural semantics is natural language interfaces to databases (e.g. [And02]). Questions such as (11) require understanding comparative and superlative constructions, as well as being aware of scope ambiguities of semantic operators (is the $70,000 salary in each year or total over the two years?).

(11) a. Which department has the largest number of employees?
b. How many employees were paid a salary higher than $70,000 over the last two years?

[5] See e.g. http://www.ics.mq.edu.au/∼rolfs/controlled-natural-languages/.
[6] See http://www.boeing.com/phantom/sechecker/se.html.
[7] See also http://www.ifi.unizh.ch/attempto/.

A real-world application that has some similarity to solving logic puzzles is understanding regulation texts, such as a website that describes which collection of courses a college student must take in order to fulfil the requirements of a study program, as in (12) below.[8] As with logic puzzles, regulation texts describe general rules and conditions that must be met. The computer should be able to answer questions about these rules as well as questions based on an additional description of a particular real or hypothetical situation (e.g. check whether the set of courses a student has taken conforms to the regulations). Also like logic puzzles, answers to such questions rarely if ever appear explicitly in the text, and must be inferred from it. In order to build a system that understands such NL regulation texts, translates them to representations that the computer could reason with, and answers NL and non-NL queries about these regulations, a detailed knowledge of structural semantics is needed. For example, understanding (12) must rely on knowledge of quantifiers and numeric expressions in their various manifestations (including e.g. "X or higher"), generic as opposed to existential quantification, modals ("must", "may"), and more.

(12) A candidate is required to complete a program of 45 units. At least 36 of these must be graded units, passed with an average 3.0 (B) grade point average (GPA) or higher. The 45 units may include no more than 21 units of courses from those listed below in Requirements 1 and 2.[9]

There has also been research interest in solving exams that were originally designed for humans, including analogical reasoning [TL05], general reading comprehension [HLBB99], and advanced placement tests in particular areas such as chemistry. Recently, a knowledge-based system developed in project HALO[10] was able to get a reasonably high score on a chemistry AP test after the text was encoded in a knowledge representation language. A natural next step to work on is to try giving the computer the ability to read and understand the NL text directly and translate it to the knowledge representations. In order to create accurate enough representations, the computer needs, among other things, a very high quality knowledge of structural semantics.

3.3.2 Statistical NLP

The need for structural semantics is by no means restricted to applications of precise understanding. For example, the matching between queries and texts that existing question-answering systems (such as [PH01]) calculate is based mainly on word alterations and matching syntactic structures. The quality of this matching would be improved if it also relied on knowledge of structural semantics. This knowledge would be used to help capture and represent more precisely the meaning and information that are actually conveyed by the texts and to perform higher-level reasoning. While arguably such knowledge may not be needed for answering simple who-did-what-to-whom factoid questions whose answers appear explicitly in the text (shallower levels of analysis may suffice for that), it is necessary for being able to answer questions that rely on *combining* the information that is conveyed by several sentences. For example:

[8] For a general motivation for the field of computational law, see http://complaw.stanford.edu.

[9] From: http://cs.stanford.edu/Degrees/mscs/degree.php.

[10] http://www.projecthalo.com/

(13) T: [Some NL texts talking about U.S. presidents, e.g.:]
George Walker Bush (born July 6, 1946) is the 43rd President of the United States, inaugurated on January 20, 2001. He was re-elected in 2004 and is currently serving his second term.[11]
Q: Was the tenth president of the United States to be re-elected a Democrat or a Republican?

It is unlikely that the answer to this question appears explicitly in some text as "The tenth president of the United States who got re-elected was a Democrat" or some variation on it. Rather, the computer must understand what *tenth* means and how it combines with the meaning of the VP complement to constrain the choice of president, as well as understand the meaning of the connective *or*. It also has to calculate the answer based on various pieces of information conveyed throughout the given texts.

Knowledge of structural semantics would be useful for many RTE-like questions. In the following examples, the text is taken without modification from the first RTE test-suite, while the question is new. In example (14), the computer needs to know what *more than* means in order to answer correctly (the answer would be *Does not follow* if *80 kilometers* was replaced with *120 kilometers*). In (16), the computer needs to know how to instantiate a statement quantified by *each* as well as to know about implications between numeric quantifiers. The last two examples require understanding of conditionals.

(14) T: In any case, the fact that this week Michael Melvill, a 63-year-old civilian pilot, guided a tiny rocket-ship more than 100 kilometres above the Earth and then glided it safely back to Earth, is a cause for celebration.
H: A tiny rocket-ship was guided more than 80 kilometers above the Earth. [Follows]

(15) T: Hyperhidrosis is more common than you think. This condition affects 1 out of 25 people.
H: Hyperhidrosis affects 5% of the people. [Does not follow]

(16) T: Each year, 26,000 people are killed or mutilated by landmines of which 8,000 are children.
H: In 2005, at least two thousand children were injured by landmines. [Follows]

(17) T: Things would be different if Microsoft was located in Georgia.
H: Microsoft's corporate headquarters are not located in Georgia. [Follows]

(18) T1: According to Viroj Laohaphan, director general of Thailand's Customs Department, if this amount of heroin had left Thailand, it would have killed an uncountable number of people.
T2: Thailand's Customs Department reported today that they had failed to stop a very large shipment of heroin from leaving Thailand.
H: An uncountable number of people are going to be killed. [Follows]

NLP systems that go beyond syntactic and lexical knowledge to actually knowing what functional words mean and how they tie together different pieces of information would enjoy a competitive edge over similar systems that do not use such knowledge.

[11] From http://en.wikipedia.org/wiki/George_W._Bush.

3.3.3 Knowledge Acquisition from NL Texts

The more structural language knowledge a computer has, semantics included, the better able it is to automatically acquire factual knowledge correctly from crawling texts on the internet. Possessing only knowledge of syntax allows only rudimentary acquisition of simple patterns of facts that appear explicitly in the text; but having more sophisticated semantic representations and inference can allow the computer to combine separate pieces of information that appear throughout the texts. This is precisely the utility of semantic representations, that they capture the *content*, i.e. meaning, of a text, and make it possible to relate the pieces of information to each other, merge them, compute entailments between them, etc. These are not possible if one relies only on the *form*, i.e. syntax, of the texts.

3.4 Annotation and Pre-Annotation Research

Given that knowledge of structural semantics is important for NLP, one could suggest creating a Semantic Treebank, i.e. a corpus similar to the Penn Treebank except sentences would be annotated not only with syntactic structures but also with semantic representations of their meanings. This is a very important goal indeed, which would allow progress in NLP of the kind that the Penn Treebank allowed in syntax. Let us look at what is needed for such an effort.

Around 1990, researchers decided to start a huge effort of annotation in the form of the Penn Treebank, to be used for statistically-based parsing. To achieve that, they first spent time writing a 300-page manual that essentially codified a syntactic grammar and provided guidance to the annotators. Once the annotation effort that was based on this document ended, the resulting corpus was available for researchers to develop probabilistic machine learning algorithms that could learn from these representations.

In order to develop probabilistic models for structural semantic knowledge, an effort similar to the Penn Treebank is required. However, around 1990 there was a more-or-less broad consensus on what syntactic representations should be used, at least for the core part of English, thanks to several decades of research on syntax in linguistics and on rule-based parsers in NLP that preceded the treebank effort. In contrast, the levels of consensus, understanding, and coverage in structural semantics in linguistics and computational linguistics today are less than they were for syntax in the 1990s. A good attempt to reach some consensus was the FraCaS project (the entire project, not just the test-suite it produced), but many issues still remain unresolved, and more progress on them is needed before good annotation schemes can be developed. In particular, there is still no *unified* semantic representation that is worked out in enough detail and that covers the main phenomena of structural semantics.

It is very important to invest the time and resources to do the necessary preliminary linguistic and computational research in structural semantics and then the research for developing a good annotation scheme, parallelling the research that went into the 300-page Penn Treebank annotation document as well as the linguistic research that preceded it. Otherwise, the very expensive cost (time and money) of annotation will be a wasted effort since the quality of the results it could produce would be low and would not justify the cost.

In fact, even in syntax, there is still much room for improvement in the Penn Treebank itself. The analysis of some syntactic phenomena there is an approximation that could be improved (e.g. the internal structure of noun phrases), or is completely missing (e.g. correct analysis of ellipsis). It is especially important to get high-quality parse trees since they are the basis for doing structural semantics. Taking structural semantics and the structural syntax-semantics interface into account will help point out gaps in the current annotation scheme that would not be felt as important when only the syntax level is considered. For that, we need to pay closer attention to linguistic theories of syntax, structural semantics, and the syntax-semantics interface, and recent developments in those areas.

3.5 Existing Work

There are precise computational grammars today that do incorporate some structural semantics. For example, [Bos04], as well as broad-scale grammars: the English Resource Grammar[12] uses Minimal Recursion Semantics [CFPS05], and the XLE[13] uses precise knowledge representations [BCC$^+$05] (as well as Glue Semantics [Dal01] in some versions). However, the semantics side is not nearly well-developed as the morphological and syntactic sides of the grammars, and the resulting semantic representations have not been sufficiently tested as far as their ability to support correct inference. Much work remains to be done, but the research mentioned above, together with the extensive research in structural semantics in linguistics during the last three decades, brought us to a good place from which this line of work can be continued.

4 A New Test-Suite

The reasoning required for both the FraCaS and RTE tasks is quite local, involving only a small number of steps. In this paper, a new test-suite for semantics is proposed, which requires more complex reasoning. This test-suite would consist of logic puzzles of the kind appearing in LSAT (and GRE in the past). An example is shown in Figure 1.[14]

What is the rationale for this test suite? Is it a useful research strategy to spend time working on this task? What could be gained?

No one yet has built a system that can take an unseen logic puzzle text from the GRE/LSAT genre and understand and solve it correctly (although there are some initial attempts [LMML04][15]). The test-suite would pose a new challenge in Natural Language and Reasoning which would be interesting and fun to work on. But the main benefit of the test-suite is that it would present an incentive for developing domain-independent computational knowledge of structural (compositional) semantics. As we have seen in section 3, such knowledge would be useful in both precise understanding applications and statistical NLP. Hence, while solving logic puzzles is not itself a real-world application,

[12] http://lingo.stanford.edu/erg.html

[13] http://www2.parc.com/istl/groups/nltt/xle/

[14] Chris Manning was the one who originally came up with the idea of trying to create a system that can solve logic puzzles from their textual descriptions.

[15] See also the PULC project: http://www.stanford.edu/~iddolev/pulc.

Preamble: Six sculptures – C, D, E, F, G, and H – are to be exhibited in rooms 1, 2, and 3 of an art gallery. The exhibition must conform to the following conditions:

(1) Sculptures C and E may not be exhibited in the same room.
(2) Sculptures D and G must be exhibited in the same room.
(3) If sculptures E and F are exhibited in the same room, no other sculpture may be exhibited in that room.
(4) At least one sculpture must be exhibited in each room, and no more than three sculptures may be exhibited in any room.

Question 1: If sculpture D is exhibited in room 3 and sculptures E and F are exhibited in room 1, which of the following may be true?

(A) Sculpture C is exhibited in room 1.
(B) No more than 2 sculptures are exhibited in room 3.
(C) Sculptures F and H are exhibited in the same room.
(D) Three sculptures are exhibited in room 2.
(E) Sculpture G is exhibited in room 2.

Question 2: If sculpture G is exhibited in room 1, which of the following may NOT be a complete list of the sculpture(s) exhibited in room 2?

(A) Sculpture C (B) Sculptures E and H (C)...

Adapted from [Web99].

Figure 1: Example of a logic puzzle text

it does have a direct bearing on related real-world tasks as well as on other scientific endeavors and longer term contributions to NLP.

There are several reasons why a logic puzzles test-suite is particularly suited as an incentive for developing structural semantic knowledge, and more generally, as a target domain for research in computational semantics and reasoning:

Naturalness: There are no a-priori restrictions on allowable syntactic and semantic constructions in logic puzzle texts, and the situations described are diverse. (The frequency of various NL constructions in the logic-puzzle genre may be different from other genres, but that is true for many other data sets). Furthermore, unlike the FraCaS test-suite, the logic puzzles are not hand-crafted for particular linguistic phenomena but are naturally occurring. They are an instance of "*found* test material" in the sense of [HLBB99]: puzzle texts were developed with a goal independent of the evaluation of NLP systems and so they provide a more realistic evaluation framework than specially-designed tests such as TREC QA, FraCaS, and RTE.

AI Task and Background Knowledge: Whereas for humans the language understanding part of logic puzzles is trivial but the reasoning is difficult, for computers it is clearly the reverse. It is straightforward for a computer to solve a puzzle once it is formalized, so the research effort can concentrate almost exclusively on the NL-understanding parts without needing to worry about a difficult back-end AI problem. Moreover, the inference required for solving logic puzzles relies to a very large extent on the structure of the language and on the semantics of functional words, and relies very little on an open-ended repository of lexical semantics or general world knowledge. Only a small core of world knowledge (prominently, temporal and spatial entailments)

is typically needed to solve a puzzle, and this knowledge is not tied to the particular domain of logic puzzles (this will be discussed in section 5.2.3). Thus there is great promise that the goal of solving logic puzzles could be achieved with very high precision much before sophisticated understanding of more general texts could be hoped for. (In contrast, the RTE task relies on a very large and unbounded body of world knowledge).

Precise Semantic Understanding: Answers to puzzle questions never explicitly appear in the text and must be *logically inferred* from it, so there is very little opportunity to use existing superficial analysis methods of information-extraction and question-answering which rely on the assumption that the answer to a question exists more-or-less explicitly in the text. Furthermore, because of the nature of logic puzzles, which target very exact answers, there is virtually no substitute for deep understanding of the text's meaning with very high precision, as almost any form of guessing would likely lead to a wrong answer. As section 5 will show, successfully solving a puzzle requires *precise* understanding of the meaning of semantic phenomena such as quantifiers and comparatives, in contrast with much current NLP work that just ignores such items. The computer needs to understand the meaning of sentences precisely also because the information conveyed in several sentences and questions must be *combined* correctly. In light of these stringent requirements, if the computer does manage to solve correctly a fair number of puzzles, it must mean that it has a very good knowledge of syntax and structural semantics.

Clear Context for Meanings: There are often debates in the linguistic literature about what all the possible meanings of a sentence are, and this question is often hard to resolve because the sentences are plucked out of context. This problem also arises sometimes in the FraCaS and RTE test-suites. But the logic puzzles texts provide a very clear context (thanks to the consensus on correct answers), thus helping us to decide on the meaning of individual sentences. In fact, the restricted nature of the task should allow the computer to actually resolve all ambiguities conclusively, based on a combination of reasoning, limited domain knowledge, and the fact that puzzle texts are designed to yield exactly one correct answer to each multiple-choice question. Thus, the computer can rule out an interpretation of the text if it leads to zero or two or more correct choices for some question.

Reasoning: Semantic representations are worth only as much as they are able to correctly support inference based on the information they are supposed to encode. Therefore, the logic puzzles task has an advantage over a corpus consisting of texts annotated with semantic representations. In the latter case, human designers might create representations that seem good to them but that are insufficient for supporting computational inference, whereas in the logic puzzles task, the representations' inference merit is tested.

Puzzle texts consist of more than just the one or two sentences that FraCaS and RTE texts have, and so the reasoning required to solve a puzzle requires a more challenging integration of the information from the entire text as well as several steps of inference. The inference is mostly logical, but it is more interesting than that: It uses some core lexical and world knowledge, and it also rests on further assumptions such as the unique names assumption and domain closure assumption.

1. Ginger, over the course of an average work-week, wanted to see how much she spent on lunch daily. On Monday and Thursday, she spent $5.43 total. On Tuesday and Wednesday, she spent $3.54 on each day. On Friday, she spent $7.89 on lunch. What was her average daily cost?

(A) $3.19 (B) $3.75 ...

2. During a 5-day festival, the number of visitors tripled each day. If the festival opened on a Thursday with 345 visitors, what was the number of visitors on that Sunday?

(A) 345 (B) 1,035 ...

Adapted from http://www.testprepreview.com.

Figure 2: Examples of simple math question texts

Clear Metric: The answers to some questions in the FraCaS test-suite depend on which reading of the text is selected, and there is no complete consensus about the correct answer to some pairs in the RTE test-suite, partly because of ambiguities and partly because of differences in assumed background knowledge. In contrast, there is full consensus on the correct answer to logic puzzles – they are deliberately and carefully designed that way. They therefore provide an extremely clear metric of evaluation for the level of success of a puzzle-solving system.

Another potential test-suite similar to logic puzzles could consist of math questions such as those on the SAT. Examples are shown in Figure 2. Although solving such questions requires some domain knowledge of math, this knowledge is simple enough that it does not pose a large AI problem.[16]

5 Phenomena in Logic Puzzles

All examples given below are taken directly or adapted slightly from logic puzzles that appeared in real GRE and LSAT exams and preparation material. This is by no means an exhaustive survey. It is just intended to point out a few interesting cases, and also to demonstrate that precise knowledge of structural semantics is essential for solving logic puzzles and so the puzzles pose an appropriate incentive for developing that knowledge.

As this survey shows, there are a lot of issues to deal with. To be sure, they may require an extensive research over an extended period of time. However, one need not feel that the issues are daunting. The fact that this is a long-term research effort and that solutions will likely require an understanding of formal semantic theory in which not all issues have been worked out yet need not discourage researchers in Computational Linguists and NLP from pursuing this line of work. Knowledge of structural semantics is not going to become practical by itself, someone needs to work on it or it will always remain theoretical and incomplete. But it is a very worthwhile research effort, and there is plenty of existing work in formal and computational semantics that can be used. The following issue are, in my opinion, interesting and exciting to work on.

[16]Solving math puzzles from their English descriptions was one of the tasks that was suggested in discussions of the next DARPA Grand Challenge. But I think that logic puzzles have more interesting NL phenomena than math questions (and their texts are usually longer).

5.1 Linguistic Phenomena

5.1.1 Quantifiers

Quantified expressions play a central role in logic puzzles. They come in many varieties, including simple type $\langle 1,1\rangle$ quantifiers (19)a, boolean combinations of quantifiers (19)b, quantifiers that take more than one restriction set (20), as well as polyadic and resumptive quantifiers which take relations (not just sets) as arguments (21) (see [KW97, PW06]).

(19) a. *Each* table has *at least two* sponsors seated at it, and *each* sponsor is seated at *exactly one* table.
b. *At least two but no more than four* representatives must be on each committee.

(20) *Exactly two more* green candies *than* red candies must be included in a package.
$\Box\forall x.package(x) \rightarrow |\{y.green(y) \wedge candy(y) \wedge included\text{-}in(y,x)\}| = 2 + |\{y.red(y) \wedge candy(y) \wedge included\text{-}in(y,x)\}|$

(21) a. If bill 2 and bill 6 are paid on different days from *each other*, which one of the following must be true?
(*each other* takes as arguments the set $\{bill2, bill6\}$ and the relation $\lambda x\lambda y.[x$ is paid on a different day than $y]$)
b. A documentary film must *always* be displayed before a comedy.
(*always* takes as arguments the set $documentary \times comedy$ and the relation $\lambda x\lambda y.[x$ is displayed before $y]$)

There are also generic statements that involve the indefinite article "a" (22), or a bare plural (23), and they have a universal quantificational force.

(22) a. *A candidate* is required to complete *a program* of 45 units.
b. If *a contestant*'s appetizer recipe does not include paprika, then the contestant's recipe must include which of the following?

(23) *Blue balls* must only be placed in round baskets, but *red balls* may be placed in either round or square baskets.

Quantifiers participate in scope ambiguities, as in (24). The first sentence comes from a puzzle where each room must exhibit at least two sculptures (*at least two* takes narrow scope) while the second comes from a puzzle where there must be at least two representatives that belong to every committee (*at least two* takes wide scope).

(24) a. At least two sculptures must be exhibited in each room.
b. At least two representatives must belong to every committee.

The computer must be aware of this kind of ambiguity and select the correct reading. If it wrongly selects the stronger reading (the reading that entails the other), then it is likely to find the text contradictory, while if it wrongly selects the weaker reading, it is likely to be unable to conclude any of the possible choices in one of the questions.

There have been attempts to resolve scope ambiguities based on general statistics using features such as the quantifier name, its position in the sentence, active/passive voice, etc. [HS03]. This can provide a useful heuristic for ranking possible scopings. In general, however, the ultimate arbitrator on this matter is the meaning of the text, which determines which of the possible scopings is consistent with the rest of the information and yields correct answers. The meaning of the text must be precisely calculated in order to determine this.

Understanding the exact structure and meaning of quantifier expressions is absolutely essential to being able to solve the puzzles correctly. Even a slight confusion between *at least two* and *exactly two*, or between a specific rather than a generic reading of an indefinite article, would produce wrong conditions and would lead to completely incorrect inferences and answers to the puzzles, not just a slight degradation in results.

5.1.2 Collectivity

Most puzzle texts start with a claim that some groups of elements exist and are related to each other. This brings into play a potential ambiguity between distributive, collective, cumulative, and other readings of predicates that have one or more plural arguments (see [Løn97] for a survey). For example, what does a sentence such as the first sentence of Figure 1 actually say in the context of the puzzle? It certainly does not claim that each sculpture is to be exhibited in each room. Perhaps it says that each sculpture is to be exhibited in one of the rooms? It seems doubtful because very similar constructions are used in puzzles where a valid solution may use some but not all of the objects mentioned. It is more likely that the sentence's truth conditions are weaker, and when they are combined with the reader's world knowledge about sculptures (a physical object may not be in more than one location at the same time), a stronger condition is obtained.

There are several additional interesting issues about plurality in the puzzles, including appositions that name the members of the group or their types (25)a,b, *respectively* constructions (25)c, and adjectives that denote collective properties (25)d,e.

(25) a. Six sculptures – C, D, E, F, G, and H – are to be exhibited in rooms 1, 2, and 3 of an art gallery.
b. At a small press, six textbooks, three introductory – F, G, and H – and three advanced – X, Y, and Z – will each be evaluated ...
c. Flights 103 and 104 are scheduled to depart at 11:00 a.m. and 12 noon, respectively.
d. Books G and X may not be evaluated during any two *consecutive* weeks.
e. Sculptures T, S, and P must be on stands that are *immediately adjacent* to one another.

Notice that all that (25)e says is: there are some stands that are immediately adjacent to one another, and sculptures T, S, and P must be on those stands. In the real world, there might be just two stands that support the three sculptures, or more than three stands, some of which need not even touch any sculpture. It is only the addition of a constraint that there is a one-to-one mapping between sculptures and stands that tells us there are exactly three stands.

The following discussion demonstrates the kind of investigation that needs to be carried out by the designer of a logic puzzles system and shows that knowledge of the linguistic semantic literature is relevant for designing such a system. The issue is what

meaning results from applying a type $\langle 1,1\rangle$ quantifier Q on a noun denoting a set A and on a collective predicate P, i.e. what "Q A P" means. As noted in [vdD93, DKK$^+$98] and [PW06, section 10.4.4], if Q is monotonically increasing in its right argument then the meaning of the combination is (26)a whereas if Q is monotonically decreasing, the meaning is (26)b.

(26) a. $\mathrm{C}^{\uparrow}(Q,A,P) \;:=\; \exists X \subseteq A.[P(X) \wedge Q(A,X)]$
b. $\mathrm{C}^{\downarrow}(Q,A,P) \;:=\; \forall X \subseteq A.[P(X) \rightarrow Q(A,X)] \;\equiv\; \neg\mathrm{C}^{\uparrow}(\neg Q,A,P)$

For example, (27)a means there is a set of students who collaborated and the size of this set is at least 4, while (27)b means that any set of students who collaborated has at most four members.

(27) a. At least four students collaborated.
$\exists X \subseteq \mathit{student}.\ [\mathit{collab}(X) \wedge \mathit{at\text{-}least}[4](\mathit{student},X)]$
b. No more than four students (ever) collaborated.
$\forall X \subseteq \mathit{student}.\ [\mathit{collab}(X) \rightarrow \mathit{at\text{-}most}[4](\mathit{student},X)]$

Now consider the sentence from Figure 1:

(28) No more than three sculptures may be exhibited in any room.

This means: it is not allowed that there would be some room in which more than three sculptures are exhibited. The truth conditions can be written as:

(29) $\neg\Diamond\exists x.\mathit{room}(x) \wedge \mathit{more\text{-}than}[3](\mathit{sculpture}, \lambda y.\mathit{exhibited\text{-}in}(y,x))$

(Here $\Diamond$ encodes the modal "may", and event structure is simplified). While this semantic representation captures the sentence's truth conditions correctly, it cannot be obtained compositionally from the sentence. This is because the negation is separated here from *more-than*[3] whereas "no more than three" should be treated as one unit (it could just as well be replaced by "at most three"). In other words, the following elements are contributed by the basic parts of the sentence:

(30) $\lambda X\lambda P.\mathit{no\text{-}more\text{-}than}[3](X,P)$ $\quad$ $\lambda P.\Diamond P$ $\quad$ $\lambda P.\exists x.\mathit{room}(x) \wedge P(x)$
$\lambda x.\mathit{sculpture}(x)$ $\quad$ $\lambda x\lambda y.\mathit{exhibited\text{-}in}(x,y)$

One can verify that there is no way to combine these five elements to obtain a formula which is logically equivalent to (29). In particular, the obvious combinations won't work:

(31) a. $\mathit{no\text{-}more\text{-}than}[3](\mathit{sculpture}, \lambda y.\Diamond\exists x.\mathit{room}(x) \wedge \mathit{exhibited\text{-}in}(y,x))$
b. $\Diamond\exists x.\mathit{room}(x) \wedge \mathit{no\text{-}more\text{-}than}[3](\mathit{sculpture}, \lambda y.\mathit{exhibited\text{-}in}(y,x))$

The first option says that the total number of sculptures that may be exhibited somewhere does not exceed three. But this is a stronger condition than (29), which allows overall more than three sculptures to be exhibited, as long as no more than three of them are exhibited together. Indeed, in the puzzle in Figure 1, each of the six sculptures must be exhibited somewhere in any valid solution. (31)b is too weak. It says that it is allowed that some room does not exhibit more than three sculptures. But (29) requires all rooms to have that property. All other permutations of (30) fail as well.

How are we to account for this? Upon further reflection, one can realize that the predicate "may be exhibited in any room" can be taken to be a (non-atomic) collective predicate that is true of a group (of sculptures) if it is possible that all members of that group are exhibited in some room at the same time. That collective predicate is combined with "sculptures" and "no more than three" using $\mathrm{C}^{\downarrow}$ since the quantifier is downward monotone:

(32) $\mathrm{C}^{\downarrow}(\textit{no-more-than}[3], \textit{sculpture}, \lambda X.\Diamond\exists r.[\textit{room}(r) \wedge \textit{exhibited-in}(X, r)]) \equiv$
$\forall X \subseteq \textit{sculpture}.[[\Diamond\exists r.\textit{room}(r) \wedge \textit{exhibited-in}(X, r)] \rightarrow \textit{no-more-than}[3](\textit{sculpture}, X)]$

This says that if a group of sculptures may be exhibited in the same room then no more than three sculptures are in that group. We also need to specify that *exhibited-in* is distributive on its first argument, i.e. $\textit{exhibited-in}(X, r) \rightarrow \forall y \in X.\textit{exhibited-in}(y, r)$. One can verify that this analysis gives correct truth conditions (the $\Diamond$ operator can be percolated to a $\Box$ ("must") in front of the formula, and this is good because all conditions in a puzzle are preceded by an implicit $\Box$).

5.1.3 Anaphora

Most (though not all) anaphoric expressions in puzzle texts are intra- rather than inter-sentential. Constructions that are often called "donkey sentences" in the structural semantics literature (e.g. [Chi95]) are common in puzzles, e.g.:

(33) If [sculptures E and F]$_i$ are exhibited in a room$_j$, no other$_i$ sculpture may be exhibited in that$_j$ room.

Sometimes, an expression is anaphoric to an entity that is not overt in the text. In a puzzle that starts with (34)a, the constraint (34)b has an expression *that year* which is anaphoric to the implicit year of the corn planting (as if the antecedent of the conditional were "If the farmer plants corn on some year"), and both need to be quantified. Although no specific group of years is mentioned anywhere in the puzzle, the question (34)c refers to the first year and the third year. In (34)d, "the vegetables" is not anaphoric to a specific group of vegetables but is subordinate to a quantified element (the year) and needs to be accommodated (in the technical sense of presupposition theory, e.g. [vdS92]).

(34) a. A farmer plants only five different kinds of vegetables – beans, corn, kale, peas, and squash. Every year the farmer plants exactly three kinds of vegetables according to the following restrictions:
b. If the farmer plants corn, he also plants beans *that year*.
$\textit{the}(x, farmer(x),\ \forall y.\textit{year}(y) \rightarrow [\textit{plants}(x, \textit{corn}, y) \rightarrow \textit{plants}(x, \textit{beans}, y)])$
c. If the farmer plants beans, corn, and kale in *the first year*, which of the following combinations must be planted in *the third year*?
d. In any year, the farmer plants no more than one of *the vegetables* the farmer planted in the previous year.

Example (34)d also shows the use of anaphoric adjectives such as *previous*, *next*, *other*, *same*, *different*.

5.1.4 Reduction (Ellipsis and Pro-Form)

Reduction is a phenomenon where part of a sentence is not expressed explicitly because it is very similar to another part of the sentence or discourse. The part which is not expressed is either replaced by a short (one or two words long) *pro-form*, as in (35)a, or altogether omitted or *elided*, as in (35)b.

(35) a. If Zena plays the piano, then Xara *does so*, too.
b. If owls are not in the forest then sparrows are $\triangle$.

One challenge is identifying correctly the sentence's syntactic structure: some of its parts are missing and the remaining sentence is grammatically "defective" in some way. Also, inference from a reduced sentence must proceed as if the missing material was there explicitly, and so it has to be "reconstructed" in some way and become an explicit part of the semantic representation. In puzzle texts, the reduction's antecedent that contains the missing material is usually within the same sentence. Here are some more interesting examples from puzzle texts:

(36) a. In each case, the wife is listed first, the husband $\triangle$ second.
(i.e.: the husband is listed second)

b. Olga's aisle is numbered higher than either $\triangle$ of Kurt's aisles $\triangle$, and $\triangle$ lower than at least one of Larisa's $\triangle$.
= Olga's aisle is numbered higher than either [one] of Kurt's aisles [is numbered], and [Olga's aisle is numbered] lower than at least one of Larisa's [aisles] [is numbered].

c. An animal shelter houses eighteen puppies in three kennels, six puppies per kennel.
= six puppies [are housed] per [each] kennel.

Reduction is particularly common in comparative constructions, which are themselves common in puzzle texts. For example, (37)a is a reduced form of (37)b, and (37)b is of (37)c (see [Lev05b, Lev05a, Lev06] for an analysis).

(37) a. Ella lifted two more boxes than Victor.
b. Ella lifted two more boxes than Victor did.
c. Ella lifted two more boxes than Victor lifted.

5.2 General Knowledge

The crucial knowledge for solving logic puzzles is structural linguistic knowledge, as the puzzles rely on relatively little background knowledge. A puzzle describing a farmer planting various crops (34) would not assume knowledge of which crop is grain and which is a vegetable. If such knowledge is needed for solving the puzzle, it would be stated explicitly. Nonetheless, some more basic knowledge that every human can be assumed to possess may not be stated explicitly, and this issue needs to be addressed.

5.2.1 Lexical Connections

In general NL understanding, the computer needs to know about connections between lexical items, such as synonyms and alternative ways to describe an event. For example:

(38) a. x sold y to z ⇒ z bought y from x
b. x murdered y ⇒ x killed y ⇒ y died

Fortunately, puzzles usually do not rely on such knowledge, and the little that is required is quite simple and can be taken from such resources as WordNet and FrameNet.

5.2.2 Lexical Truth Conditions

Another kind of knowledge that is required more often pertains to the truth conditions imposed by certain words. For example, to solve a question such as (39)a, the computer needs to know what *complete list* means. The adjective does not have a simple intersective semantics, but rather, one needs to spell out its truth conditions: a sculpture is exhibited in room 2 iff it is on the complete list. In this case, a statement such as (39)b can be used with $A = \mathit{sculptue}$ and $P = \lambda x.\mathit{exhibited\text{-}in}(x, \mathit{room}2)$. A similar comment holds for the expression *in common* in (39)c.

(39) a. Which of the following may not be a *complete list* of the sculptures exhibited in room 2?
b. $\mathit{complete}(x) \wedge \mathit{list}(x) \wedge \mathit{of}(x, \mathit{the}(g,\ g \subseteq A \wedge P(g))) \ \Rightarrow\ \forall y.[y \in x \leftrightarrow (y \in A \wedge P(y))]$
c. The two committees must have at least one member *in common*.

The meaning of each such expression is quite specific and non-trivial. It is hard to see how those meanings could be acquired in any other way than carefully analyzing the expressions and formalizing them on a case-by-case basis. Nevertheless, each puzzle requires only a handful of them, and they recur across puzzles, so the effort to formalize these meanings will gradually stabilize after considering a sufficient number of puzzles.

5.2.3 World Knowledge

The puzzle in Figure 1 does not state explicitly that no sculpture may be exhibited in more than one room. Without this piece of knowledge, the explicit conditions in the text are insufficient to yield exactly one answer for each multiple choice question (for question 1, answers B, D and E would all be correct). Human readers know this piece of world knowledge, but it has to be given to a computer in explicit form.

One may use the puzzles test-suite as an incentive for developing such computational world knowledge and for investigating how to integrate such knowledge with NL understanding. But representation and acquisition of commonsense knowledge are very hard AI problems. Fortunately, if one is primarily interested in developing structural semantic knowledge, there is a way to circumvent this problem in some cases.

Puzzle texts usually rely on very few implicit pieces of information. Moreover, implicit world knowledge can often be recast as mathematical properties of the relations mentioned in the puzzle. For instance, the unique location constraint on sculptures is

equivalent to constraining the mapping from sculptures to rooms to be injective (one-to-one); other possible properties are surjective and total. The computer could systematically search for such implicit constraints that, in combination with the explicitly stated constraints, yield exactly one answer per question.

6 Evaluation

How should we evaluate a system for solving logic puzzles? One way is to calculate the percentage of questions it answers correctly. Since each multiple-choice question has five answer choices, the baseline (random guessing) is 20% correct answers. Another simple idea is to count how many puzzles the system solved entirely correctly. This is more stringent and the baseline is much lower (if there were exactly Q questions per puzzle, the baseline would be $(1/5)^Q \cdot 100\%$; but the number of questions per puzzle may vary).

A more illuminating evaluation would investigate the reasons for the system's successes and failures. In particular, it might be that the system correctly translated the text to semantic representations, except for one little fact that was misunderstood and spoiled the entire result, or one novel word or construction that the system does not know about. This kind of failure is likely to happen very often because of the brittle nature of the logic puzzles task, where each constraint must be understood precisely and combined correctly with all the other pieces of information.

One way to carry out such an investigation is to evaluate how a particular class of NL phenomena is understood by the system. This could be done by using simplified logic puzzles that target mainly this class (while relying on other phenomena that are already known to be well-understood by the system). Here is a simple example of a puzzle which tests knowledge of comparatives:

(40) If Mary is taller than Bill and John is taller than Mary then which of the following must be true?
A. John is taller than Bill. B. Mary is taller than John. ...

This looks very much like the FraCaS test-suite. Such tests let us know in more detail what the system does and does not know. The semantic understanding of the system must still be tested for its ability to support correct inference in the logic puzzle's task.

Another way would be to add to the definition of the logic puzzles task the requirement that the computer should attempt to detect the limits of its own knowledge and indicate whether it thinks it can definitely answer a question correctly or whether it is missing some piece of linguistic or domain knowledge. For example, the computer could indicate it does not know the meaning of a certain word or how to deal with a certain grammatical construction. Knowing the limits of one's own knowledge and acting accordingly is an important characteristic of intelligent behavior. In some NLP applications, if the computer does not know how to deal with some NL phenomenon, guessing is sometimes helpful and may still give useful results. But because of the extreme precision required in logic puzzles, it seems that guessing and robustness techniques are much less likely to help here compared to other NLP tasks.

An additional kind of measure is how easily system failures can be fixed. Once the basic machinery and a certain critical mass of knowledge have been put into the system,

it should reach a level of competency where failures are more frequently the result of minor lacks of knowledge that are easy to fix. Given that the system failed on a question, the measure would reflect how fast the source of the failure can be detected and how fast the necessary knowledge can be added to fix it (this can be measured by both time and lines of additional code/formulated knowledge). The first question provides a rough measure for the software-engineering quality of the system (how well it is constructed in terms of modularity, intelligibility for human designers, ease of debug and extension, etc.), and the second question provides a rough measure for the system's competency.

7 Conclusion

The time has come to equip our computers with high-quality and broad structural semantic knowledge that goes beyond the levels of morphology and syntax. This will make a lasting contribution to computers' ability to understand and reason with the information conveyed by texts. The test-suite proposed here presents a particularly relevant incentive for that goal, and my hope is that this paper will inspire many researchers to accept the challenge of constructing a system for solving logic puzzles.

References

[And02] Ion Androutsopoulos. *Exploring Time, Tense, and Aspect in Natural Language Database Interfaces*. John Benjamins Publishing Company, 2002.

[BCC+05] Danny Bobrow, Cleo Condoravdi, Richard Crouch, Ronald Kaplan, Lauri Kartunnen, Tracy Holloway King, Valeria de Paiva, and Annie Zaenen. A basic logic for textual inference. In *Proc. of the AAAI Workshop on Inference for Textual Question Answering*, 2005.

[Bos04] Johan Bos. Computational semantics in discourse: Underspecification, resolution, and inference. *Journal of Logic, Language and Information*, 13(2):139–157, 2004.

[CFPS05] Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan Sag. Minimal Recursion Semantics: an introduction. *Research on Language and Computation*, 3:281–332, 2005.

[Chi95] Gennaro Chierchia. *Dynamics of Meaning: Anaphora, Presupposition, and the Theory of Grammar*. University of Chicago Press, 1995.

[CKZ06] Richard Crouch, Lauri Karttunen, and Annie Zaenen. Circumscribing is not excluding: A response to Manning, 2006. Ms., PARC, March, 2006. http://cs224u.stanford.edu/publications/reply-to-manning.pdf.

[Dal01] Mary Dalrymple. *Lexical Functional Grammar*, volume 34 of *Syntax and Semantics Series*. Academic Press, 2001.

[DGM05] Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL recognising textual entailment challenge. In *Proc. of the PASCAL Challenge Workshop on Recognizing Textual Entailment*, 2005.

[DKK+98] Mary Dalrymple, Makoto Kanazawa, Yookyung Kim, Sam Mchombo, and Stanley Peters. Reciprocal expressions and the concept of reciprocity. *Linguistics and Philosophy*, 21(2):159–210, 1998.

[FKS06] Norbert E. Fuchs, Kaarel Kaljurand, and Gerold Schneider. Attempto Controlled English meets the challenges of knowledge representation, reasoning, interoperability and user interfaces. In *Proc. of FLAIRS'2006*, 2006.

[Fra96] FraCaS. Using the framework: Deliverable 16 of the FraCaS project, 1996. http://www.cogsci.ed.ac.uk/∼fracas/.

[HLBB99] Lynette Hirschman, Marc Light, Eric Breck, and John D. Burger. Deep Read: A reading comprehension system. In *Proc. of ACL'99*, pages 325–332, 1999.

[HS03] Derrick Higgins and Jerrold M. Sadock. A machine learning approach to modeling scope preferences. *Computational Linguistics*, 29:73–96, 2003.

[KW97] Edward L. Keenan and Dag Westerståhl. Generalized quantifiers in linguistics and logic. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*, pages 837–893. Elsevier, 1997.

[Lev05a] Iddo Lev. Comparative constructions. Addendum to [Lev05b], 2005.

[Lev05b] Iddo Lev. Gradable comparatives: Syntax and syntax-semantics interface. Paper for Ling221B, Stanford University. http://www.stanford.edu/∼iddolev/, 2005.

[Lev06] Iddo Lev. On the syntax-semantics interface of overt and covert reciprocals. Paper for Ling223B, Stanford University. http://www.stanford.edu/∼iddolev/, 2006.

[LMML04] Iddo Lev, Bill MacCartney, Christopher D. Manning, and Roger Levy. Solving logic puzzles: From robust processing to precise semantics. In *Proc. of the 2nd workshop on text meaning and interpretation, ACL'04*, 2004.

[Løn97] Jan Tore Lønning. Plurals and collectivity. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*, pages 1009–1053. 1997.

[Man06] Christopher D. Manning. Local textual inference: It's hard to circumscribe, but you know it when you see it – and NLP needs it, 2006. Ms., Stanford University, February 25, 2006. http://cs224u.stanford.edu/papers/LocalTextualInference.pdf.

[PH01] Marius Pasca and Sanda M. Harabagiu. High performance question/answering. In *Proc. of SIGIR*, pages 366–374, 2001.

[PW06] Stanley Peters and Dag Westerståhl. *Quantifiers in Language and Logic*. Oxford University Press, 2006.

[TL05] Peter D. Turney and Michael L. Littman. Corpus-based learning of analogies and semantic relations. *Machine Learning*, 60:251–278, 2005.

[vdD93] Jaap van der Does. Sums and quantifiers. *Linguistics and Philosophy*, 16:509–550, 1993.

[vdS92] Rob A. van der Sandt. Presupposition projection as anaphora resolution. *Journal of Semantics*, 9:333–377, 1992.

[Web99] Karl Weber. *The Unofficial Guide to the GRE Test*. ARCO Publishing, 2000 edition, 1999.

[ZKC05] Annie Zaenen, Lauri Karttunen, and Richard Crouch. Local textual inference: Can it be defined or circumscribed? In *Proc. of the ACL workshop on empirical modeling of semantic equivalence and entailment*, pages 31–36, 2005.