Higher-order logical inference with compositional semantics

Koji Mineshima[1,3]
mineshima.koji@ocha.ac.jp

Pascual Martínez-Gómez[1,2,3]
pascual@nii.ac.jp

Yusuke Miyao[2]
yusuke@nii.ac.jp

Daisuke Bekki[1,2,3]
bekki@is.ocha.ac.jp

[1]Ochanomizu University
Tokyo, Japan

[2]National Institute of Informatics
Tokyo, Japan

[3]CREST, JST
Saitama, Japan

Abstract

We present a higher-order inference system based on a formal compositional semantics and the wide-coverage CCG parser. We develop an improved method to bridge between the parser and semantic composition. The system is evaluated on the FraCaS test suite. In contrast to the widely held view that higher-order logic is unsuitable for efficient logical inferences, the results show that a system based on a reasonably-sized semantic lexicon and a manageable number of non-first-order axioms enables efficient logical inferences, including those concerned with generalized quantifiers and intensional operators, and outperforms the state-of-the-art first-order inference system.

1 Introduction

Entailment relations are of central importance in the enterprise of both formal and computational semantics. Traditionally, formal semanticists have concentrated on a relatively small set of linguistic inferences. However, since the emergence of statistical parsers based on sophisticated syntactic theories (Clark and Curran, 2007), an open domain system has been developed that supports certain degree of robust semantic interpretation with wide coverage (Bos et al., 2004). It is then reasonable to expect that a state-of-the-art formal semantics provides an accurate computational basis of natural language inferences.

However, there are still obstacles in the way of achieving this goal. One is that the statistical parsers on which semantic interpretations rely do not necessarily reflect the best syntactic analysis as assumed in the formal semantics literature (Honnibal et al., 2010). Another persistent problem is the gap between the logics employed in the two communities; while it is generally assumed among formal semanticists that adequate semantic representations for natural language demand higher-order logic or type theory (Carpenter, 1997), the dominant view in computational linguistics is that inferences based on higher-order logic are hopelessly inefficient for practical applications (Bos, 2009a). Accordingly, it is claimed that some approximation of higher-order representations in terms of first-order logic (Hobbs, 1985), or a more efficient "natural logic" system based on surface structures is needed. However, it is often not a trivial task to give an approximation of rich higher-order information within a first-order language (Pulman, 2007). Moreover, the coverage of existing natural logic systems is limited to single-premise inferences (MacCartney and Manning, 2008).

In this paper, we first present an improved compositional semantics that fills the gap between the parser syntax and a composition derivation. We then develop an inference system that is capable of higher-order inferences in natural languages. We combine a state-of-the-art higher-order proof system (Coq) with a wide-coverage parser based on a modern syntactic theory (Combinatory Categorial Grammar, CCG). The system is designed to handle multi-premise inferences as well as single-premise ones.

We test our system on the FraCaS test suite (Cooper et al., 1994), which is suitable for evaluating the linguistic coverage of an inference system. The experiments show that our higher-order system outperforms the state-of-the-art first-order system with respect to the speed and accuracy of making logical inferences.

2 CCG and Compositional Semantics

As an initial step of compositional semantics, we use the C&C parser (Clark and Curran, 2007), a statistical CCG parser trained on CCGbank (Hockenmaier and Steedman, 2007). Parser out-

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 2055–2061,
Lisbon, Portugal, 17-21 September 2015. ©2015 Association for Computational Linguistics.

category : $S\backslash NP$
semantics : $\lambda Q.Q(\lambda x.\mathsf{True})(\lambda x.\mathsf{E}(x))$

Figure 1: Schematic lexical entry (semantic template) for intransitive verbs. E is a position in which a particular lexical item appears.

category : NP/N
semantics : $\lambda F \lambda G \lambda H.\forall x(Fx \wedge Gx \rightarrow Hx)$
surf : every

Figure 2: The lexical entry for determiner *every*

puts are mapped onto semantic representations in a standard way (Bos, 2008), using λ-calculus as an interface between syntax and semantics.

The strategy we use to build a semantic lexicon is similar to that of Bos et al. (2004). A lexical entry for each open word class consists of a syntactic category in CCG (possibly with syntactic features) and a semantic representation encoded as a λ-term. Fig. 1 gives an example.[1] For a limited number of closed words such as logical or functional expressions, a λ-term is directly assigned to a surface form (see Fig. 2). The output formula is obtained by combining each λ-term in accordance with meaning composition rules and then by applying β-conversion.

There is a non-trivial gap between the parser output and the standard CCG-syntax as presented in Steedman (2000). Due to this gap, it is not straightforward to obtain desirable semantic representations for a wide range of constructions. One major difference from the standard CCG-syntax is the treatment of post-NP modifiers; for instance, the relative clause *who works* is assigned not the category $N\backslash N$, but the category $NP\backslash NP$, which applies to the whole NP. To derive correct truth-conditions for quantificational sentences, we assign to determiners a semantic term having an extra predicate variable as shown in Fig. 2, namely, $\lambda F \lambda G \lambda H.\forall x(Fx \wedge Gx \rightarrow Hx)$, in a similar way to the continuation semantics for event predicates (Bos, 2009b; Champollion, 2015). The extra predicate variable G can be filled by the semantically empty predicate $\lambda x.\mathsf{True}$ in a verb phrase (see Fig. 1). Fig. 3 gives an example derivation.

Note that the changes in the lexical entries as illustrated in Fig. 1 and Fig. 2 are made for the correct semantic parsing, namely, the compositional derivation of semantic representations. Usually, inferences are conducted on those output semantic representations in which additional complexities, such as lambda operators and extra predicate variables, disappear. Accordingly, the changes in the lexical entries do not affect the efficiency of inferences.

Examples	**Semantic Types**
most	$(\mathsf{E} \rightarrow \mathsf{Prop}) \rightarrow (\mathsf{E} \rightarrow \mathsf{Prop}) \rightarrow \mathsf{Prop}$
might	$\mathsf{Prop} \rightarrow \mathsf{Prop}$
true	$\mathsf{Prop} \rightarrow \mathsf{Prop}$
manage	$\mathsf{Prop} \rightarrow \mathsf{E} \rightarrow \mathsf{Prop}$
believe	$\mathsf{Prop} \rightarrow \mathsf{E} \rightarrow \mathsf{Prop}$

Table 1: A classification of key linguistic elements having higher-order denotations.

The present analysis of post NP-modifiers can also handle non-restrictive relative clauses such as "*the president, who*...". In this case, the modifier "*who*..." can be taken to apply to the whole NP *the president*, thus its syntactic category can be regarded as $NP\backslash NP$, not as $N\backslash N$. Thus, although the $NP\backslash NP$ analysis of relative clauses is a non-standard one, it has an advantage in that it provides a unified treatment of restrictive and non-restrictive relative clauses.

3 Representation and Inference in HOL

We present a higher-order representation language and describe apparently higher-order phenomena that have received attention in formal semantics.

3.1 Semantic representations in HOL

We use the language of higher-order logic (HOL) with two basic types, E for entities and Prop for propositions. Here we distinguish between propositions and truth-values, as is standard in modern type theory (Ranta, 1994; Luo, 2012). Key higher-order constructs are summarized in Table 1.[2] A first-order language can be taken as a fragment of this language. Thus, adopting a higher-order language does not lead to the loss of the expressive power of the first-order language.

Apart from sub-sentential utterances such as short answers to *wh*-questions (Ginzburg, 2005), there are important constructions that are naturally

[1] Here we use a non-standard semantics for intransitive verbs. We will explain it in the next paragraph.

[2] We write a function from objects of type A to objects of type B as $A \rightarrow B$. Here $\rightarrow$ is right-associative: $A \rightarrow B \rightarrow C$ means $A \rightarrow (B \rightarrow C)$. We use the symbol $\rightarrow$ both for logical implication and function-type constructor, following the so-called Curry-Howard isomorphism (Carpenter, 1997).

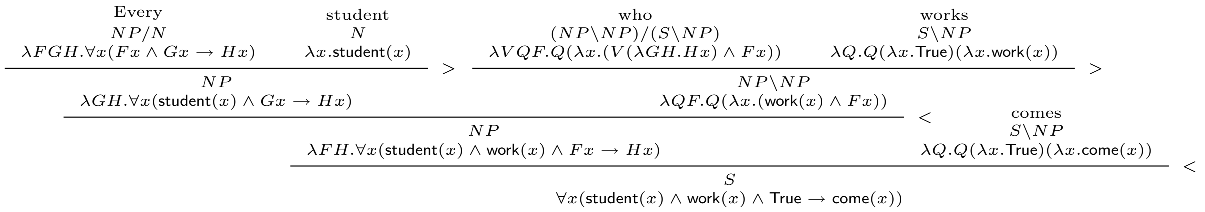
Every
NP/N
$\lambda FGH.\forall x(Fx \wedge Gx \rightarrow Hx)$
student
N
$\lambda x.\mathsf{student}(x)$
$>$
who
$(NP\backslash NP)/(S\backslash NP)$
$\lambda VQF.Q(\lambda x.(V(\lambda GH.Hx) \wedge Fx))$
works
$S\backslash NP$
$\lambda Q.Q(\lambda x.\mathsf{True})(\lambda x.\mathsf{work}(x))$
$>$
NP
$\lambda GH.\forall x(\mathsf{student}(x) \wedge Gx \rightarrow Hx)$
$NP\backslash NP$
$\lambda QF.Q(\lambda x.(\mathsf{work}(x) \wedge Fx))$
$<$
comes
$S\backslash NP$
$\lambda Q.Q(\lambda x.\mathsf{True})(\lambda x.\mathsf{come}(x))$
NP
$\lambda FH.\forall x(\mathsf{student}(x) \wedge \mathsf{work}(x) \wedge Fx \rightarrow Hx)$
$<$
S
$\forall x(\mathsf{student}(x) \wedge \mathsf{work}(x) \wedge \mathsf{True} \rightarrow \mathsf{come}(x))$

Figure 3: A CCG derivation of the semantic representation for the sentence *Every student who works comes*. $\lambda FGH.X$ is an abbreviation for $\lambda F\lambda G\lambda H.X$. "True" denotes the tautology, hence the final formula is equivalent to $\forall x(\mathsf{student}(x) \wedge \mathsf{work}(x) \rightarrow \mathsf{come}(x))$.

represented in higher-order languages.[3]

Generalized quantifiers A classical example of non-first-orderizable expressions is a proportional generalized quantifier like *most* and *half of* (Barwise and Cooper, 1981). Model-theoretically, they denote relations between sets. We represent them as a two-place higher-order predicate taking first-order predicates as arguments. For instance, *Most students work* is represented as follows.

(1) $\mathsf{most}(\lambda x.\mathsf{student}(x), \lambda x.\mathsf{work}(x))$

Here, most is a higher-order predicate in the sense that it takes first-order predicates $\lambda x.\mathsf{student}(x)$ and $\lambda x.\mathsf{work}(x)$ as arguments. We take the entailment patterns governing *most* as axioms, along the same lines of natural logic and monotonicity calculus (Icard and Moss, 2014), where determiners are taken as primitive two-place operators.

Standard quantifiers like *every* and *some* could also be treated as binary operators in the same way as the binary most in (1). But we choose to adopt the first-order decomposition in such cases (see Fig. 2 for the lexical entry of *every*).

Modals Modal auxiliary expressions like *might*, *must* and *can* are represented as unary sentential operators. For instance, the sentence *Some student might come* is represented as:

(2) $\exists x(\mathsf{student}(x) \wedge \mathsf{might}(\mathsf{come}(x)))$.

An important inference role of such a modal operator is to distinguish modal contexts from actual contexts and thus block an inference from one context to another ($\mathsf{might}\ A$ does not entail A).

Alternatives to the higher-order approach include the first-order decomposition of modal operators using world variables (Blackburn et al., 2001) and the first-order modal semantic representations implemented in Boxer (Bos, 2005). We prefer the higher-order approach, because the first-order approaches introduce additional quantifiers and variables at the level of the semantic representations on which one makes inferences.

Veridical and anti-veridical predicates A sentential operator O is *veridical* if $O(A)$ entails A, and *anti-veridical* if $O(A)$ entails $\neg A$. While modal auxiliary verbs like *might* are neither veridical nor anti-veridical, there is a class of expressions licensing these patterns of inference. Typical examples are adjectives taking an embedded proposition, such as *true/correct* and *false/incorrect*. Note that sentences like *Everything/what he said is false* involve a quantification over propositions, which is problematic for the first-order approach.

The so-called implicative verbs like *manage* and *fail* (Nairn et al., 2006) are also an instance of this class. For example, *Some student manages to come* is formalized as

(3) $\exists x(\mathsf{student}(x) \wedge \mathsf{manage}(x, \mathsf{come}(x)))$

where *manage* is a veridical predicate taking a proposition as the second argument; it licenses an inference to $\exists x(\mathsf{student}(x) \wedge \mathsf{come}(x))$.

Attitude verbs A wide range of propositional attitude verbs such as *believe* and *hope* are similar to modals in that they do not license an inference from attitude contexts to actual contexts. But factives like *know* and *remember* are an exception; they are veridical.[4]

A first-order translation can be given along the lines of Hintikka (1962). (4) is translated as (5).

(4) $\mathsf{know}(\mathsf{john}, \exists x(\mathsf{student}(x) \wedge \mathsf{come}(x)))$

(5) $\forall w_1(R_{\mathsf{john}} w_0\, w_1 \rightarrow$
$\exists x(\mathsf{student}(w_1, x) \wedge \mathsf{come}(w_1, x)))$

[3] See also Blackburn and Bos (2005) for some discussion on inferences that go beyond first-order logic.

[4] Factive predicates show the important inference patterns known as *presupposition projection* (van der Sandt, 1992), which are beyond the scope of this paper.

Inference pattern	**Axiom**
Existential import	$\forall F \forall G\,(\mathsf{most}(F,G)$ $\rightarrow \exists x(Fx \wedge Gx))$
Conservativity	$\forall F \forall G\,(\mathsf{most}(F,G)$ $\rightarrow \mathsf{most}(F, \lambda x.(Fx \wedge Gx)))$
Monotonicity (right-upward)	$\forall F \forall G \forall H\,(\mathsf{most}(F,G)$ $\rightarrow (\forall x(Gx \rightarrow Hx)$ $\rightarrow \mathsf{most}(F,H)))$
Veridicality	$\forall P(\mathsf{true}(P) \rightarrow P)$ $\forall x \forall P(\mathsf{manage}(x,P) \rightarrow P)$ $\forall x \forall P(\mathsf{know}(x,P) \rightarrow P)$
Anti-veridicality	$\forall P(\mathsf{false}(P) \rightarrow \neg P)$ $\forall x \forall P(\mathsf{fail}(x,P) \rightarrow \neg P)$

Table 2: Axioms for non-first-order constructions.

However, one drawback is that the compositional semantics becomes complicated, so we prefer the non-decomposition approach for attitude verbs.

3.2 Inferences in HOL

Following Chatzikyriakidis and Luo (2014), we use a proof-assistant Coq (Castéran and Bertot, 2004) to implement a specialized prover for higher-order features in natural languages, and combine it with efficient first-order inferences. We use Coq's built-in tactics for first-order inferences. Coq also has a language called *Ltac* for user-defined automated tactics (Delahaye, 2000). The additional axioms and tactics specialized for natural language constructions are written in Ltac. We ran Coq fully automated, by feeding to its interactive mode a set of predefined tactics combined with user-defined proof-search tactics.

Table 2 shows the axioms we implemented. Modals and non-veridical predicates (by which we mean predicates that are neither veridical nor anti-veridical) do not have particular axioms, with the consequence that actual and hypothetical contexts are correctly distinguished.

4 Experiments

We evaluated our system on the FraCaS test suite (Cooper et al., 1994), a set of entailment problems that is designed to evaluate theories of formal semantics.[5] We used the version provided by MacCartney and Manning (2007). The whole data set is divided into nine sections, each devoted to linguistically challenging problems. Of these, we used six sections, excluding three sections (nominal anaphora, ellipsis and temporal reference) that involve a task of resolving context-dependency, a task beyond the scope of this paper. Each problem consists of one or more premises, followed by a hypothesis. There are three types of answer: *yes* (the premise set entails the hypothesis), *no* (the premise set entails the negation of the hypothesis), and *unknown* (the premise set entails neither the hypothesis nor its negation). Fig. 4 shows some examples.

Section	#	Ours	Nut	L&S 13	Tian 14
Quantifiers	74	.77	.53	.62	**.80**
Plurals	33	**.67**	.52	–	–
Adjectives	22	**.68**	.32	–	–
Comparatives	31	**.48**	.45	–	–
Verbs	8	.62	.62	–	–
Attitudes	13	**.77**	.46	–	–
Total	181	**.69**	.50	–	–

Table 3: Accuracy on the FraCaS test suite. The first column shows the number of problems. Of the total 188 problems, we excluded seven problems that lack a well-defined answer.

Currently, our system has 57 templates for general syntactic categories and 80 lexical entries for closed words. In a similar way to Bos et al. (2004), closed words are confined to a limited range of logical and functional expressions such as quantifiers and connectives. These templates and lexical entries are not specific with respect to the FraCaS test suite. We use WordNet (Miller, 1995) as the knowledge base for antonymy; logical axioms relevant to given inferences are extracted from this knowledge base.

We compared our system with the state-of-the-art CCG-based first-order system Boxer (Bos, 2008), which is one of the most well-known logic-based approaches to textual entailment. We used the Nutcracker system based on Boxer that utilizes a first-order prover (*Bliksem*) and a model builder (*Mace*) with the option enabling access to WordNet. We did not use the option enabling modal semantics, since it did not improve the results. All experiments were run on a 4-core@1.8Ghz, 8GB RAM and SSD machine with Ubuntu.

Experimental results are shown in Table 3. Our system improved on Nutcracker. We set a time-out of 30 seconds, after which we output the label "unknown". Nutcracker timed-out in one third of the problems (57 out of 181), whereas there was no time-out in our system.

Table 4 shows parse times and inference times for the FraCaS test suite. The inference speed

[5] Our system will be publicly available at `https://github.com/mynlp/ccg2lambda`.

fracas-067	
Premise 1	All residents of the North American continent can travel freely within Europe.
Premise 2	Every Canadian resident is a resident of the North American continent.
Hypothesis	All Canadian residents can travel freely within Europe.
Answer	Yes
fracas-074	
Premise 1	Most Europeans can travel freely within Europe.
Hypothesis	Most Europeans who are resident outside Europe can travel freely within Europe.
Answer	Unknown

Figure 4: Examples of entailment problems from the FraCaS test suite

Parsing and inference	sec /problem
CCG Parsing (C&C parser)	3.76
Our system with higher-order inference	3.72
Our system with higher-order rules ablated	3.46
Nutcracker with first-order inference (first-order prover + model builder)	11.23

Table 4: Comparison of inference time on the FraCaS test suite. CCG parsing is common to both our system and Nutcracker.

of our system is significantly higher than that of Nutcracker. Our system's total accuracy with higher-order rules is 69%, and drops to 59% when ablating the higher-order rules.

We are aware of two other systems tested on FraCaS that are capable of multiple-premise inferences: the CCG-based first-order system of Lewis and Steedman (2013) and the dependency-based compositional semantics of Tian et al. (2014). These systems were only evaluated on the Quantifier section of FraCaS. As shown in Table 3, our results improve on the former and are comparable with the latter.

Other important studies on FraCaS are those based on natural logic (MacCartney and Manning, 2008; Angeli and Manning, 2014). These systems are designed solely for single-premise inferences and hence are incapable of handling the general case of multiple-premise problems (which cover about 45% of the problems in FraCaS). Our system improves on these natural-logic-based systems by making multiple-premise inferences as well.

Main errors we found are due to various parse errors caused by the CCG parser, including the failure to handle multiwords like *a lot of*. The performance of our system will be further improved with correct syntactic analyses. Our experiments on FraCaS problems do not constitute an evaluation on real texts nor on unseen test data. Note, however, that a benefit of using a linguistically controlled set of entailment problems is that one can check not only whether, but also *how* each semantic phenomenon is handled by the system. In contrast to the widely held view that higher-order logic is less useful in computational linguistics, our results demonstrate the logical capacity of a higher-order inference system integrated with the CCG-based compositional semantics.

5 Conclusion

We have presented a framework for a compositional semantics based on the wide-coverage CCG parser, combined with a higher-order inference system. The experimental results on the FraCaS test suite have shown that a reasonable number of lexical entries and non-first-order axioms enable various logical inferences in an efficient way and outperform the state-of-the-art first-order system. Future work will focus on incorporating a robust model of lexical knowledge (Lewis and Steedman, 2013; Tian et al., 2014) to our framework.

Acknowledgments

We are grateful to Chris Worth, Ribeka Tanaka, and the three anonymous reviewers for their helpful comments and suggestions. This research has been supported by the JST CREST program (Establishment of Knowledge-Intensive Structural Natural Language Processing and Construction of Knowledge Infrastructure).

References

Gabor Angeli and Christopher D. Manning. 2014. NaturalLI: Natural logic inference for common sense reasoning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2014)*, pages 534–545.

Jon Barwise and Robin Cooper. 1981. Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4(2):159–219.

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

Patrick Blackburn, Maarten de Rijke, and Yde Venema. 2001. *Modal Logic*. Cambridge University Press.

Johan Bos, Stephen Clark, Mark Steedman, James R. Curran, and Julia Hockenmaier. 2004. Wide-coverage semantic representations from a CCG parser. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING-2004)*, pages 104–111.

Johan Bos. 2005. Towards wide-coverage semantic interpretation. In *Proceedings of Sixth International Workshop on Computational Semantics (IWCS-6)*, pages 42–53.

Johan Bos. 2008. Wide-coverage semantic analysis with Boxer. In *Proceedings of the 2008 Conference on Semantics in Text Processing*, pages 277–286.

Johan Bos. 2009a. Applying automated deduction to natural language understanding. *Journal of Applied Logic*, 7(1):100–112.

Johan Bos. 2009b. Towards a large-scale formal semantic lexicon for text processing. In *Proceedings of the Biennal GSCL Conference From Form to Meaning: Processing Texts Automatically*, pages 3–14.

Bob Carpenter. 1997. *Type-Logical Semantics*. MIT Press.

Pierre Castéran and Yves Bertot. 2004. *Interactive Theorem Proving and Program Development. Coq'Art: The Calculus of Inductive Constructions*. Springer.

Lucas Champollion. 2015. The interaction of compositional semantics and event semantics. *Linguistics and Philosophy*, 38(1):31–66.

Stergios Chatzikyriakidis and Zhaohui Luo. 2014. Natural language inference in Coq. *Journal of Logic, Language and Information*, 23(4):441–480.

Stephen Clark and James R Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

Robin Cooper, Richard Crouch, Jan van Eijck, Chris Fox, Josef van Genabith, Jan Jaspers, Hans Kamp, Manfred Pinkal, Massimo Poesio, Stephen Pulman, et al. 1994. FraCaS: A Framework for Computational Semantics. *Deliverable*, D6.

David Delahaye. 2000. A Tactic Language for the System Coq. In *Logic for Programming and Automated Reasoning (LPAR)*, volume 1955 of *Lecture Notes in Computer Science*, pages 85–95. Springer.

Jonathan Ginzburg. 2005. Abstraction and ontology: Questions as propositional abstracts in type theory with records. *Journal of Logic and Computation*, 15(2):113–130.

Jaakko Hintikka. 1962. *Knowledge and Belief*. Cornell University Press.

Jerry R. Hobbs. 1985. Ontological promiscuity. In *Proceedings of the 23rd annual meeting on Association for Computational Linguistics (ACL-1985)*, pages 60–69.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: a corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Matthew Honnibal, James R. Curran, and Johan Bos. 2010. Rebanking CCGbank for improved NP interpretation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-2010)*, pages 207–215.

Thomas Icard and Lawrence Moss. 2014. Recent progress in monotonicity. *Linguistic Issues in Language Technology (LiLT)*, 9.

Mike Lewis and Mark Steedman. 2013. Combining distributional and logical semantics. *Transactions of the Association for Computational Linguistics*, 1:179–192.

Zhaohui Luo. 2012. Formal semantics in modern type theories with coercive subtyping. *Linguistics and Philosophy*, 35(6):491–513.

Bill MacCartney and Christopher D. Manning. 2007. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200.

Bill MacCartney and Christopher D. Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 521–528.

George A Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Rowan Nairn, Cleo Condoravdi, and Lauri Karttunen. 2006. Computing relative polarity for textual inference. *Inference in Computational Semantics (ICoS-5)*, pages 67–76.

Stephen Pulman. 2007. Formal and computational semantics: a case study. In *Proceedings of the Seventh International Workshop on Computational Semantics (IWCS-7)*, pages 181–196.

Aarne Ranta. 1994. *Type-Theoretical Grammar*. Oxford University Press.

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

Ran Tian, Yusuke Miyao, and Takuya Matsuzaki. 2014. Logical inference on dependency-based compositional semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL-2014)*, pages 79–89.

Rob van der Sandt. 1992. Presupposition projection as anaphora resolution. *Journal of Semantics*, 9(4):333–377.