Implementing Natural Language Inference for comparatives

Izumi Haruta[1], *Koji Mineshima*[2], *and Daisuke Bekki*[1]
[1] Ochanomizu University, Japan
[2] Keio University, Japan

ABSTRACT

This paper presents a computational framework for Natural Language Inference (NLI) using logic-based semantic representations and theorem-proving. We focus on logical inferences with comparatives and other related constructions in English, which are known for their structural complexity and difficulty in performing efficient reasoning. Using the so-called A-not-A analysis of comparatives, we implement a fully automated system to map various comparative constructions to semantic representations in typed first-order logic via Combinatory Categorial Grammar parsers and to prove entailment relations via a theorem prover. We evaluate the system on a variety of NLI benchmarks that contain challenging inferences, in comparison with other recent logic-based systems and neural NLI models.

Keywords: comparatives, compositional semantics, theorem proving, Combinatory Categorial Grammar, Natural Language Inference

INTRODUCTION 1

Natural Language Inference (NLI), which is also called Recognizing Textual Entailment, is the task of determining whether a text entails a hypothesis. It is a method widely used for evaluating systems in Natural Language Processing (NLP). In recent years, with the development of large datasets such as Stanford Natural Language Inference (SNLI;

Bowman *et al.* 2015) and Multi-Genre Natural Language Inference (MultiNLI; Williams *et al.* 2018), it has been used as one of the methods for evaluating the performance of deep learning (DL) models.

NLI can be characterized as a *black-box* type evaluation in the sense that it does not matter what the internal structure of the evaluated system is (Bos 2008a). Thus, it does not matter whether the system to be evaluated is based on DL or on parsing and logic. In fact, the FraCaS project (Cooper *et al.* 1996), one of the origins of NLI benchmarks, was developed to evaluate a pipeline of syntax, semantics, and inference systems based on linguistic theories. The goal was to make a meaningful comparison and evaluation of various frameworks of formal syntax and semantics (cf. Morrill and Valentín 2016).

How well can current linguistic and logical theories solve NLI benchmarks including FraCaS and others that contain challenging semantic phenomena? The purpose of this paper is to address this question. The question has important implications both in the context of NLP and theoretical linguistics. In the context of NLP, a logic-based approach to NLI can provide a basis for a more explanatory and interpretable alternative to DL-based approaches. In the context of theoretical linguistics, it has the significance of systematically testing and evaluating linguistic theories using NLI benchmarks well-designed by linguists.

In this paper, we introduce a logic-based framework for NLI, focusing on comparatives and other related constructions in English, including adjectives, adverbs, numerals, and generalized quantifiers. Comparative constructions have been actively studied in formal semantics yet still pose a challenge to computational approaches (Pulman 2007). Our system has a pipeline consisting of syntactic parsing based on Combinatory Categorial Grammar (CCG; Steedman 1996, 2000), compositional mapping of parsed trees to logical forms, and theorem-proving in a First-Order Logic (FOL) setting. In this respect, the system is transparent, allowing us to examine what happens at each step of parsing (syntax), semantic analysis (semantics), and theorem proving (logic).

Each linguistic phenomenon we are concerned with in this paper has been largely tackled by a separate semantic theory, for example, event semantics for verbs, degree semantics for adjectives, and theories of generalized quantifier for noun phrases (see Section 2 for the detail of each theory). What is needed here is to put together these different theories, to formulate the resulting system as a computational model, and to empirically evaluate its prediction. Note also that it is often the case that computational implementation of existing theories is not a trivial task but one that requires additional substantial work, to decide things for which the published papers do not specify the details. In this respect, there is a large gap between formal semantics and its computational implementation. We also emphasize the importance of a fully-automated NLI system for evaluating a linguistic theory: if you throw an inference in natural language to the system, it can immediately compute the logical forms and evaluate the entailment relation, thus facilitating to make a prediction of the theory in an easy and quick way.

Our system is designed to have a reasonable expressive power to represent various comparative constructions without compromising the efficiency of automated theorem proving. The results of the evaluation on various datasets, including FraCaS, show that our system is capable of solving complex logical reasoning with high accuracy. We also compare our system with existing logic-based systems and current state-of-the-art DL models. All code and evaluation results are publicly available.[1]

Our contributions are summarized as follow:

- We propose semantic representations (logical forms) for various comparative constructions and related constructions in English, including generalized quantifiers, numerals, and adverbs, using a uniform representation language in typed FOL that is suitable for automated theorem proving (Section 2).
- We implement a compositional semantics for these constructions in the framework of CCG (Section 3).
- We evaluate our system on various NLI datasets including FraCaS that contain complex logical inferences with comparatives and other linguistic phenomena, in comparison with other logic-based systems and DL-based NLI models (Section 4).

[1] `https://github.com/izumi-h/ccgcomp`

2 SEMANTIC REPRESENTATIONS

In this section, we first introduce our representation language, in comparison with other approaches (Section 2.1). Then we present the semantic representations of various gradable constructions, in particular, adjectives (Section 2.2), comparatives (Section 2.3), adverbs (Section 2.4), and generalized quantifiers (Section 2.5).

2.1 *Representation language: Typed FOL*

As a representation language, we use the Typed First-Order Form (TFF) of the Thousands of Problems for Theorem Provers (TPTP) format (Sutcliffe *et al.* 2012; Sutcliffe 2017). TPTP is a library of problems for automated theorem proving systems. TFF is a formal expression in FOL with equality and arithmetic operations. TFF extends the language of FOL with the notion of types. It has predefined basic types for entity (e) and truth-value (t), and arithmetic types for integers, rational numbers, and real numbers.[2] We use integers as the type of degrees (d), although we can instead use other arithmetic types (rational numbers or real numbers) in the implementation. In addition, we use the type of *events* (v) as a user-defined type. Thus, the semantic type τ of an expression is defined by the following rule:

$$\tau ::= e \mid t \mid v \mid d \mid \tau \rightarrow \tau$$

Here $\tau \rightarrow \tau$ is a function type, where $\rightarrow$ is right-associative. Thus $t \rightarrow t \rightarrow t$ means $t \rightarrow (t \rightarrow t)$.

Note that although we use λ-calculus for semantic composition as will be explained in Section 3, the language of TFF does not allow the use of λ-abstraction. Thus, λ-terms can only appear in the process of a compositional derivation but not in the resulting logical form. Whether this language has a sufficient descriptive capacity is an empirical question, and we will show through evaluation by NLI benchmarks that the language is expressive enough to represent various linguistic phenomena we deal with in this paper.

[2] TFF uses the notations `$i` for individuals and `$o` for truth-values (booleans). We instead use e and t in this paper.

Other representation languages used in the logic-based approaches to NLI include (i) Higher-Order Logic (HOL), (ii) FOL, and (iii) Type Theory. Regarding (i), Mineshima *et al.* (2015) and Abzianidze (2015, 2016) propose an NLI system combining CCG parsers with provers specialized for natural languages using a controlled fragment of HOL. Although HOL is expressive enough to handle complex expressions such as generalized quantifiers, provers based on HOL are less efficient than those based on FOL and tend to rely on hand-coded rules, causing scalability issues.

For (ii), Bos (2008b) and Martínez-Gómez *et al.* (2017) present NLI systems based on standard FOL. While theorem provers based on FOL are more efficient than HOL, the expressive power is limited so that there are linguistic phenomena that resist straightforward treatment in FOL. A notable exception is Hahn and Richter (2016), which introduces a method to encode HOL constructions in natural languages in FOL Henkin Semantics. However it is not extended to complex phenomena such as comparatives covered in FraCaS. Perhaps the approach that is closest to ours is that of Pulman (2018), which presents methods to approximate some higher-order inferences with adjectives in a first-order setting. Compared with these previous works, our system has broader coverage, handling a variety of inferences with adjectives, comparatives, generalized quantifiers, numerals, and adverbs from a unified perspective.

For (iii), Chatzikyriakidis and Luo (2014), Bernardy and Chatzikyriakidis (2017) and Chatzikyriakidis and Bernardy (2019) present a type-theoretic system using Coq as a proof assistant for NLI, tackling problems in FraCaS. However they inherit the disadvantages of HOL in that the theorem proving is not computationally efficient; in fact, the theorem-proving component of these type-theoretic systems is not fully automated, due in part to the fact that there is no decision procedure for HOL. Thus, it cannot be used as part of a system that would be comparable to logic-based NLI systems studied in the context of natural language processing (NLP). By contrast, TFF, which is adopted in our approach, has computational efficiency and expressive power in that it can handle equality and arithmetic operations implemented in automated theorem provers. It is a language that suits the purpose of our study. We emphasize the importance of building a fully automated NLI system, which allows us to build a system usable in NLP applications and to compute the predictions of each formal semantic theory quickly and precisely. This would be an initial step towards establishing a meaningful and systematic way to evaluate each linguistic framework.

2.2 *Adjectives*

We start with the analysis of adjectives in our framework. This serves as a basis for developing computational degree-based semantics for other gradable constructions.

2.2.1 Gradable adjectives

We introduce the phenomenon of GRADABILITY and present an analysis of gradable adjectives in degree-based semantics.[3]

(1) My car is *expensive*. (Gradable)
   a. My car is <u>very</u> *expensive*.
   b. My car is <u>more</u> *expensive* than yours.

(2) My pet is *four-legged*. (Non-gradable)
   a. # My pet is <u>very</u> *four-legged*.
   b. # My pet is <u>more</u> *four-legged* than yours.

Expensive and *tall* are gradable adjectives, and can take degree modifiers such as *very* and have comparative form as in (1a) and (1b). On the other hand, *four-legged* is not a gradable adjective; the sentences (2a) and (2b) are not felicitous.

In degree-based semantics, gradable adjectives can be treated as two-place predicates that take entity and degree (Cresswell 1976). For instance, *John is 5 feet tall*, containing the specific numerical expression *5 feet*, is analyzed as $\mathsf{tall}(\mathsf{john}, \mathsf{5\ feet})$, where $\mathsf{tall}(x, \delta)$ is read as "x is *at least* as tall as degree δ" (Klein 1991).[4] For simplicity, we do not consider the internal structure of a measure phrase such as *5 feet* and write as $\mathsf{tall}(\mathsf{john}, 5)$, where 5 is treated as an integer.

[3] See Lassiter (2015) and Morzycki (2016) for an overview of degree-based semantics.

[4] For an explanation of why $\mathsf{tall}(x, \delta)$ is not treated as "x is *exactly* as tall as δ", see Section 3.2.

Positive form and comparison class 2.2.2

The positive form of a gradable adjective is regarded as involving comparison to some threshold that can be inferred from the context of the utterance. We write $\theta_F(A)$ to denote the contextually specified threshold for a predicate F given a set A, which is called a COMPARISON CLASS (Klein 1980, 1982). When a comparison class is implicit, as in (3a) and (4a), we use the universal set U as a default comparison class.[5] We often abbreviate $\theta_F(\mathsf{U})$ as θ_F. Thus, (3a) is represented as (3b), which means the height of Mary is more than or equal to the threshold θ_{tall}.

(3) a. Mary is tall.
b. $\mathsf{tall}(\mathsf{mary}, \theta_{\mathsf{tall}})$

We semantically distinguish the positive adjective *tall* from its antonym *short*, which we call a negative adjective. The logical form of (4a), where a negative adjective *short* appears, is (4b); we take it that (4b) means that the height of Mary is *less than* the threshold θ_{short}.[6]

(4) a. Mary is short.
b. $\mathsf{short}(\mathsf{mary}, \theta_{\mathsf{short}})$

A threshold can be explicitly constrained by an NP modified by a gradable adjective. Thus, (5a) can be interpreted as (5b) relative to an explicit comparison class, namely, the sets of animals.[7]

(5) a. Mickey is a small animal. (FraCaS-204)
b. $\mathsf{small}(\mathsf{mickey}, \theta_{\mathsf{small}}(\mathsf{animal})) \wedge \mathsf{animal}(\mathsf{mickey})$

For positive gradable adjectives, if $\mathsf{tall}(x, \delta)$ is true, then x satisfies all heights below δ. By contrast, for negative gradable adjectives,

[5] In this study, we do not consider the context-sensitivity of an implicit comparison class. See Narisawa *et al.* (2013) and Pezzelle and Fernández (2019) for work on this topic in computational linguistics.

[6] We do not claim that this analysis can fully address the subtle inferences about antonyms (cf. Lehrer and Lehrer 1982). A more detailed analysis of antonyms is left for future work.

[7] Here and henceforth, when an example appears in FraCaS dataset (Cooper *et al.* 1996), we refer to the ID of the sentence in the dataset.

if $\mathsf{short}(x, \delta)$ is true, then x satisfies all the heights δ or above. To formalize these properties, we postulate the following axioms for each positive adjective P and negative adjective N:

(up) $\forall x \forall \delta_1 (P(x, \delta_1) \rightarrow \forall \delta_2 ((\delta_2 \leq \delta_1) \rightarrow P(x, \delta_2)))$

(down) $\forall x \forall \delta_1 (N(x, \delta_1) \rightarrow \forall \delta_2 ((\delta_1 \leq \delta_2) \rightarrow N(x, \delta_2)))$

2.2.3 Privative adjectives

Apart from gradable and non-gradable adjectives, *former* and *fake* are classified as **privative** adjectives (Kamp 1975). For a privative adjective *Adj* and a noun phrase *N*, the intersection of $[\![Adj\ N]\!]$ and $[\![N]\!]$ is empty. For example, (6) holds for the privative adjective *former* and the noun phrase *student.*[8]

(6) $[\![\text{former student}]\!] \cap [\![\text{student}]\!] = \emptyset$

(6) can be expressed as an axiom in our system using a predicate variable F in the following way:

(7) $\forall x (\mathsf{former}(F(x)) \rightarrow \neg F(x))$

For instance, (8a) is mapped to (8b). By using (7), (8a) contradicts *Peter is a student.*

(8) a. Peter is a former student.
  b. $\mathsf{former}(\mathsf{student}(\mathsf{peter}))$

2.3 *Adjectival comparatives*

Next, we consider adjectival comparatives using the analysis of gradable adjectives described in the previous sections.

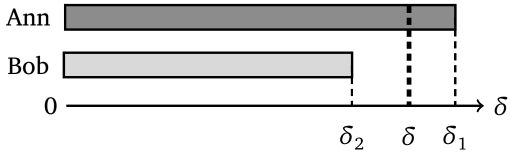
[8] The truth condition of *former* may involve temporal semantics, which we neglect in order to avoid complicating the whole system.

A-not-A analysis 2.3.1

To begin with, we introduce the so-called A-not-A analysis (Seuren 1973; Klein 1980, 1982, 1991; Schwarzschild 2008) for comparatives in degree-based semantics.[9]

(9) a. Ann is taller than Bob is.

b. $\exists\delta(\mathsf{tall}(\mathsf{ann},\delta) \wedge \neg\mathsf{tall}(\mathsf{bob},\delta))$



According to this analysis, (9a) is analyzed as (9b), where (9a) is interpreted as saying that there exists a degree δ of height that Ann satisfies but Bob does not. As shown in the figure in (9), together with the Consistency Postulate (CP) explained below, this guarantees that Ann's height is greater than Bob's height. More generally, if an adjective F is associated with a degree such as heights and weights, we can say "A is more F than B is" is true if and only if there exists a threshold δ that A satisfies but B does not. A-not-A analysis makes it possible to derive entailment relations between various comparative constructions in a simple way using FOL theorem provers.[10]

We show the logical forms for other basic comparative constructions under A-not-A analysis.

(10) a. Tom is taller than Mary. (Increasing)

b. $\exists\delta(\mathsf{tall}(\mathsf{tom},\delta) \wedge \neg\mathsf{tall}(\mathsf{mary},\delta))$

(11) a. Harry is less tall than Ken. (Decreasing)

b. $\exists\delta(\neg\mathsf{tall}(\mathsf{harry},\delta) \wedge \mathsf{tall}(\mathsf{ken},\delta))$

(12) a. Tom is as tall as Mary. (Equatives)

b. $\forall\delta(\mathsf{tall}(\mathsf{mary},\delta) \rightarrow \mathsf{tall}(\mathsf{tom},\delta))$

The sentence (11a) is a construction representing that the height of Harry is less than that of Ken. The sentence (12a) is interpreted as

[9] A version of this analysis is called *delineation analysis*, which goes back to Lewis (1972).

[10] This possibility is also suggested by Pulman (2007).

"Tom is *at least* as tall as Mary", which means the height of Tom is greater than or equal to that of Mary. This reading is captured by mapping (12a) to (12b). The sentence (12a) can also be interpreted as "Tom is *exactly* as tall as Mary". See Section 3.2 for a discussion on how to derive this strong reading in our setting.

In A-not-A analysis, there is an axiom called Consistency Postulate (CP), which formalizes the relation between the degrees of two entities under A-not-A analysis (Klein 1980, 1982). It asserts that if there is a degree satisfied by x but not by y, then every degree satisfied by y is satisfied by x as well.

(CP) $\forall x \forall y(\exists \delta(A(x,\delta) \wedge \neg A(y,\delta)) \rightarrow \forall \delta(A(y,\delta) \rightarrow A(x,\delta)))$,
where A is an arbitrary gradable adjective.

The axiom (CP) can be deduced as a derivable rule of (up) and (down):

PROPOSITION 1 *(CP) follows from (up) and (down).*

PROOF Consider the case where A is a positive adjective. Suppose there exists δ_0 such that $A(x,\delta_0)$ holds but $A(y,\delta_0)$ does not. Also let δ be arbitrary and suppose $A(y,\delta)$. To show $A(x,\delta)$, let us assume $\delta_0 < \delta$ for the sake of contradiction. By (up) and $A(y,\delta)$, we have $A(y,\delta_0)$, but this is the contradiction. Hence, $\delta \leq \delta_0$ holds, and by (up) we have $A(x,\delta)$. Thus, $A(y,\delta) \rightarrow A(x,\delta)$ holds for any δ. When A is a negative adjective, by using (down) instead of (up) we get the same conclusion. Hence we obtain (CP). □

2.3.2 Measure phrases and differential comparatives

The sentence (13a) contains the measure phrase *2 inches* before the comparative form *taller* of the gradable adjective *tall* and mentions the difference in height between Ken and Harry. Such constructions are known as DIFFERENTIAL COMPARATIVES. (13a) means the height of Ken is *2 inches or greater* than the height of Harry. Thus differential comparatives can be handled by extending the analysis of equatives such as the sentence (12a). (13a) is mapped to the logical form (13b).

(13) a. Ken is 2 inches taller than Harry.
b. $\forall \delta(\mathsf{tall}(\mathsf{harry},\delta) \rightarrow \mathsf{tall}(\mathsf{ken},\delta+2))$

Note that if (13a) is mapped to $\exists\delta(\mathsf{tall}(\mathsf{ken}, \delta + 2) \wedge \neg\mathsf{tall}(\mathsf{harry}, \delta))$, then the meaning that the difference in height between Ken and Harry is exactly 2 inches is missing.

To derive inferences with measure phrases, we define the axioms (sup) and (inf) that formalize supremum and infimum on degree, respectively.

(sup) $\forall x \exists\delta_1(P(x, \delta_1) \wedge \neg\exists\delta_2((\delta_1 < \delta_2) \wedge P(x, \delta_2)))$

(inf) $\forall x \exists\delta_1(N(x, \delta_1) \wedge \neg\exists\delta_2((\delta_2 < \delta_1) \wedge N(x, \delta_2)))$

The import of (sup) is expressed as follows. Assume we are given some assignment of values to variable x and P. Then there is a value δ_1 that makes $P(x, \delta_1)$ true, but there is no value δ_2 that is more than δ_1 and makes $P(x, \delta_2)$ true. Thus, the inference from (13a) to *Ken is taller than Harry* follows from (sup).

PROPOSITION 2 *From* $\forall\delta(\mathsf{tall}(\mathsf{harry}, \delta) \rightarrow \mathsf{tall}(\mathsf{ken}, \delta + 2))$*, it follows that* $\exists\delta(\mathsf{tall}(\mathsf{ken}, \delta) \wedge \neg\mathsf{tall}(\mathsf{harry}, \delta))$.

PROOF By (sup), there exists δ_0 such that $\mathsf{tall}(\mathsf{harry}, \delta_0)$ and there is no δ_1 such that $\delta_0 < \delta_1$ and $\mathsf{tall}(\mathsf{harry}, \delta_1)$. Since $\delta_0 < \delta_0 + 2$, it follows that $\neg\mathsf{tall}(\mathsf{harry}, \delta_0 + 2)$. By the premise, we have $\mathsf{tall}(\mathsf{ken}, \delta_0 + 2)$. Hence, we have $\exists\delta(\mathsf{tall}(\mathsf{ken}, \delta) \wedge \neg\mathsf{tall}(\mathsf{harry}, \delta))$. □

Finally, consider the construction with a measure phrase in a *than*-clause. The sentence (14a) includes the measure phrase *4 feet* in the *than*-clause. It has the same meaning as "Ken is more than 4 feet tall" and is mapped to (14b). Here, instead of comparing the degree of two entities, we compare the height of Ken with the specific value *4 feet*.

(14) a. Ken is taller than 4 feet.

b. $\exists\delta(\mathsf{tall}(\mathsf{ken}, \delta) \wedge (4 < \delta))$

Extensional and intensional comparison classes 2.3.3

Gradable expressions can be divided into extensional and intensional adjectives (Kamp 1975; Partee 2007):

(15) All dogs are animals.

a. $\Rightarrow$ All *fat* dogs are *fat* animals. (Extensional)

b. $\not\Rightarrow$ All *clever* dogs are *clever* animals. (Intensional)

Fat and *tall* are **extensional** adjectives and license the inference in (15a). In contrast, *clever* and *skillful* are **intensional** adjectives, which do not allow the same pattern of inference. Thus, (15b) does not hold.

The difference between extensional and intensional adjectives also arises in reasoning with comparative expressions. Consider the following:

(16) a. John is a fatter politician than Bill.
$\Rightarrow$ John is fatter than Bill. (FraCaS-216)

b. John is a cleverer politician than Bill.
$\nRightarrow$ John is cleverer than Bill. (FraCaS-217)

The sentences in (16a) involve the comparative form *fatter* of the extensional adjective *fat*. The adjective *fat* is classified as an extensional adjective since *fat as a politician* does not make sense.[11] Accordingly, *John is a fatter politician than Bill* can be decomposed into *John is a politician and fatter than Bill*. Thus the inference in (16a) holds. On the other hand, the inference (16b), which contains the comparative form *cleverer* of the intensional adjective *clever*, does not hold. This is because even if John is cleverer than Bill as a politician, we do not know the relation between John and Bill with respect to the cleverness in other domains. For extensional adjectives, the sentence (17a) is mapped to the logical form (17b).

(17) a. John is a fatter politician than Bill.

b. $\mathsf{politician}(\mathsf{john}) \wedge \mathsf{politician}(\mathsf{bill})$
$\wedge \exists\delta(\mathsf{fat}(\mathsf{john},\delta) \wedge \neg\mathsf{fat}(\mathsf{bill},\delta))$

(18) a. John is fatter than Bill.

b. $\exists\delta(\mathsf{fat}(\mathsf{john},\delta) \wedge \neg\mathsf{fat}(\mathsf{bill},\delta))$

For intensional adjectives $\mathsf{clever}(x,\delta)$, we extend its second argument to take an intensional comparison class; in the second argument of the intensional adjectives we use a two-place function for a *noun parameter* $\lambda N\delta.\mathsf{np}(N,\delta)$.[12] The type of $\mathsf{np}(N,\delta)$ is degree. For

[11] Note that it is meaningful to say *fat for a politician*, so the adjective *fat* can take a comparison class and is context-sensitive (cf. Partee 2007).

[12] Throughout the paper, we abbreviate $\lambda X_1 \lambda X_2 \ldots \lambda X_n.M$ as $\lambda X_1 X_2 \ldots X_n.M$.

instance, $\mathsf{clever}(x, \mathsf{np}(\mathsf{politician}, \delta))$ is intended to mean that x is clever as a politician (at least) to degree δ. The sentence (19a) is mapped to the logical form (19b). (19a) means that John is cleverer than Bill as a politician, and thus it does not entail (20a), which means that John is cleverer than Bill for any extension U.

(19) a. John is a cleverer politician than Bill.

b. $\mathsf{politician}(\mathsf{john}) \wedge \mathsf{politician}(\mathsf{bill})$
$\wedge \exists\delta(\mathsf{clever}(\mathsf{john}, \mathsf{np}(\mathsf{politician}, \delta))$
$\wedge \neg\mathsf{clever}(\mathsf{bill}, \mathsf{np}(\mathsf{politician}, \delta)))$

(20) a. John is cleverer than Bill.

b. $\exists\delta(\mathsf{clever}(\mathsf{john}, \mathsf{np}(\mathsf{U}, \delta)) \wedge \neg\mathsf{clever}(\mathsf{bill}, \mathsf{np}(\mathsf{U}, \delta)))$

2.3.4 Degree modifiers

Consider the case where an adjective appears with degree modifiers such as *very* and *much*. The following two sentences (21a) and (22a) are examples:

(21) a. Peter is fat.

b. $\mathsf{fat}(\mathsf{peter}, \theta_{\mathsf{fat}})$

(22) a. Peter is *very* fat.

b. $\exists\delta(\mathsf{fat}(\mathsf{peter}, \delta) \wedge (\theta_{\mathsf{fat}} + \delta' \leq \delta))$

The sentence (21a) is represented as (21b), which means that Peter meets the threshold θ_{fat}. In (22a), the degree modifier *very* appears preceding the adjective, which emphasizes the degree that Peter is fat. In this case, we set the lower bound on Peter's weight as $\theta_{\mathsf{fat}} + \delta'$ for a constant δ' such that $0 < \delta'$ and map (22a) to (22b).

As mentioned in Section 2.2.2, we consider not only positive gradable adjectives such as *fat* but also negative gradable adjectives such as *small*. (23a) is interpreted as (23b), where the size of the room satisfies a value less than the threshold θ_{small}. The sentence (24a) emphasizes the small size of this room. In this case, we interpret the size that the room satisfies as being less than $\theta_{\mathsf{small}} - \delta'$, and express it as (24b).

(23) a. This room is small.

b. $\exists x(\mathsf{room}(x) \wedge \mathsf{small}(x, \theta_{\mathsf{small}}))$

(24) a. This room is *very* small.

b. $\exists x(\mathsf{room}(x) \wedge \exists \delta(\mathsf{small}(x, \delta) \wedge (\delta \leq \theta_{\mathsf{small}} - \delta')))$

A sentence with the degree modifier *much* such as (25a) is interpreted as having a difference of at least a fixed value δ' between the degrees satisfied by the two entities being compared. It is represented as (25b) in a similar way to the analysis of (13).

(25) a. David is *much* taller than Jim.

b. $\forall \delta(\mathsf{tall}(\mathsf{jim}, \delta) \rightarrow \mathsf{tall}(\mathsf{david}, \delta + \delta'))$

2.4 *Adverbial comparatives*

In the previous sections, we analyzed comparative expressions of adjectives using a theory based on degree-based semantics, which was developed for analyzing adjectives and comparatives. In formal semantics, there is another semantic framework, event semantics, used largely to account for the semantics of verb phrases and adverbial modifiers (Davidson 1967; Parsons 1990). To address comparative expressions of adverbs, it is necessary to present a theory that incorporates not only degree semantics but also event semantics. Building on the work in Haruta *et al.* (2020), we combine the two semantic theories and extend the theory of A-not-A analysis with comparative constructions of adverbs.

2.4.1 Adverbs in event semantics

To handle adverbial expressions, we adopt a standard neo-Davidsonian event semantics (Parsons 1990), which analyzes sentences as involving quantification over events. For example, the sentence (26a) is analyzed as (26b), where subj is a function term that associates an event to its subject.

(26) a. John ran.

b. $\exists e(\mathsf{run}(e) \wedge (\mathsf{subj}(e) = \mathsf{john}))$

A sentence containing an adverb like (27a) is analyzed as (27b), where the adverb *slowly* acts as a predicate of an event.

(27) a. John ran *slowly*.

b. $\exists e(\mathsf{run}(e) \wedge (\mathsf{subj}(e) = \mathsf{john}) \wedge \mathsf{slowly}(e))$

This allows us to derive an inference from (27a) to (26a), i.e., an inference to drop adverbial phrases.[13]

Combining event semantics and degree semantics 2.4.2

To correctly derive entailment relations between sentences with gradable adverbials and comparative expressions of adverbs, we apply the same analysis to gradable adverbials such as *slowly* and *fast* as to gradable adjectives. The following examples show logical forms of basic constructions, where adverbs like *loudly* are treated as binary predicates of an event and a degree:

(28) a. John shouted *loudly*. (Positive)

b. $\exists e(\mathsf{shout}(e) \wedge (\mathsf{subj}(e) = \mathsf{john}) \wedge \mathsf{loud}(e, \theta_{\mathsf{loud}}))$

(29) a. Jim sang *better than* Mary. (Comparative)

b. $\exists e_1 \exists e_2(\mathsf{sing}(e_1) \wedge (\mathsf{subj}(e_1) = \mathsf{jim}) \wedge \mathsf{sing}(e_2)$
$\wedge (\mathsf{subj}(e_2) = \mathsf{mary}) \wedge \exists \delta(\mathsf{good}(e_1, \delta) \wedge \neg \mathsf{good}(e_2, \delta)))$

(30) a. Bob drove *as carefully* as John. (Equative)

b. $\exists e_1 \exists e_2(\mathsf{drive}(e_1) \wedge (\mathsf{subj}(e_1) = \mathsf{bob}) \wedge \mathsf{drive}(e_2)$
$\wedge (\mathsf{subj}(e_2) = \mathsf{john}) \wedge \forall \delta(\mathsf{careful}(e_2, \delta) \to \mathsf{careful}(e_1, \delta)))$

The sentence (28a) contains the adverbial phrase *loudly*, which is analyzed as $\mathsf{loud}(e, \theta_{\mathsf{loud}})$ as in (28b). This means that John's shouting is at least as loud as a certain threshold θ_{loud}, which we take to be the same logical form as the positive form of gradable adjectives. To treat predicates for adverbs in the same way as those for adjectives, we convert a gradable adverb (e.g., *loudly*) to its adjectival form (e.g., *loud*) in the logical form. The sentence (29a) is the adverbial comparative construction with the comparative form *better*. The logical form (29b) means there exists a degree of "goodness" δ such that event e_1 satisfies, but e_2 does not. Similarly, we can assign an appropriate logical form to the sentence (30a) by extending the analyses for adjectival comparatives as described in Section 2.3.

[13] In this study, we do not introduce event variables to adjectives and adverbs themselves. For instance, *Tim is tall* is analyzed as $\mathsf{tall}(\mathsf{tim}, \theta_{\mathsf{tall}})$ not as $\exists e(\mathsf{tall}(e, \theta_{\mathsf{tall}}) \wedge (\mathsf{subj}(e) = \mathsf{tim}))$, where e quantifiers over underlying *states* denoted by *tall*. We do not pursue this alternative analysis here; see Parsons (1990, Chap.10) for some discussion.

2.5 *Generalized quantifiers*

We extend the analysis of comparatives by the degree semantics described above to generalized quantifiers. In the traditional analysis (Barwise and Cooper 1981), generalized quantifiers such as *many, few, more than,* and *most* are analyzed as denoting a relation between sets. Alternatively, these quantifiers can be analyzed as adjectives in degree semantics (Partee 1988; Rett 2018) and the proportional quantifier *most* as the superlative form of *many* (Hackl 2000; Szabolcsi 2010). We implement this alternative analysis in our computational framework.

2.5.1 Numerical adjectives

We represent a numerical adjective such as *ten* in *ten orders* by the predicate $\mathsf{many}(x, n)$, which means that the cardinality of x is at least n, where x ranges over pluralities and n is a positive integer (Hackl 2000). The following shows the logical forms of some typical sentences involving numerical adjectives.

(31) a. Ann won ten orders.

b. $\exists x(\mathsf{order}(x) \wedge \mathsf{many}(x, 10) \wedge \exists e(\mathsf{win}(e) \wedge (\mathsf{subj}(e) = \mathsf{ann}) \wedge (\mathsf{obj}(e) = x)))$

(32) a. Ann won many orders.

b. $\exists \delta \exists x(\mathsf{order}(x) \wedge \mathsf{many}(x, \delta) \wedge (\theta_{\mathsf{many}}(\mathsf{order}) < \delta) \wedge \exists e(\mathsf{win}(e) \wedge (\mathsf{subj}(e) = \mathsf{ann}) \wedge (\mathsf{obj}(e) = x)))$

(33) a. Ann won more orders than Harry.

b. $\exists \delta(\exists x(\mathsf{order}(x) \wedge \mathsf{many}(x, \delta) \wedge \exists e(\mathsf{win}(e) \wedge (\mathsf{subj}(e) = \mathsf{ann}) \wedge (\mathsf{obj}(e) = x))) \wedge \neg \exists y(\mathsf{order}(y) \wedge \mathsf{many}(y, \delta) \wedge \exists e(\mathsf{win}(e) \wedge (\mathsf{subj}(e) = \mathsf{harry}) \wedge (\mathsf{obj}(e) = y))))$

As mentioned in the previous section, a sentence like *John is 5 feet tall* is mapped to the logical form $\mathsf{tall}(\mathsf{john}, 5)$ using the binary predicate of the adjective *tall*. In a similar vein, the sentence (31a) is mapped to the logical form (31b), taking the adjective *many* to be hidden between *ten* and *orders* (see Section 3.2 for a compositional derivation). In the case of (32a), we take *many* as the positive form of the adjective and introduce the threshold $\theta_{\mathsf{many}}(\mathsf{order})$ in the logical form (32b). In the

Table 1: Logical forms of some constructions with numerical adjectives

Sentence	Logical form
Mary won at least eleven orders.	$\exists x(\mathsf{order}(x) \wedge \mathsf{many}(x, 11)$ $\wedge \exists e(\mathsf{win}(e) \wedge (\mathsf{subj}(e) = \mathsf{mary}) \wedge (\mathsf{obj}(e) = x)))$
Mary sold 20 more books than John.	$\forall \delta(\exists x(\mathsf{book}(x) \wedge \mathsf{many}(x, \delta)$ $\wedge \exists e(\mathsf{sell}(e) \wedge (\mathsf{subj}(e) = \mathsf{john}) \wedge (\mathsf{obj}(e) = x)))$ $\rightarrow \exists x(\mathsf{book}(x) \wedge \mathsf{many}(x, \delta + 20)$ $\wedge \exists e(\mathsf{sell}(e) \wedge (\mathsf{subj}(e) = \mathsf{mary}) \wedge (\mathsf{obj}(e) = x))))$
John won twice as many orders than Ann.	$\forall \delta(\exists x(\mathsf{order}(x) \wedge \mathsf{many}(x, \delta)$ $\wedge \exists e(\mathsf{win}(e) \wedge (\mathsf{subj}(e) = \mathsf{john}) \wedge (\mathsf{obj}(e) = x)))$ $\rightarrow \exists x(\mathsf{order}(x) \wedge \mathsf{many}(x, \delta \times 2)$ $\wedge \exists e(\mathsf{win}(e) \wedge (\mathsf{subj}(e) = \mathsf{ann}) \wedge (\mathsf{obj}(e) = x))))$
Bob won more orders than Luis lost.	$\exists \delta(\exists x(\mathsf{order}(x) \wedge \mathsf{many}(x, \delta)$ $\wedge \exists e(\mathsf{win}(e) \wedge (\mathsf{subj}(e) = \mathsf{bob}) \wedge (\mathsf{obj}(e) = x)))$ $\wedge \neg \exists x(\mathsf{order}(x) \wedge \mathsf{many}(x, \delta)$ $\wedge \exists e(\mathsf{lost}(e) \wedge (\mathsf{subj}(e) = \mathsf{luis}) \wedge (\mathsf{obj}(e) = x))))$
More than five campers caught a cold.	$\exists x \exists \delta(\mathsf{camper}(x) \wedge \mathsf{many}(x, \delta) \wedge (\delta > 5)$ $\wedge \exists y(\mathsf{cold}(y) \wedge \exists e(\mathsf{catch}(e) \wedge (\mathsf{subj}(e) = x)$ $\wedge (\mathsf{obj}(e) = y))))$

case of (33a), *more* is analyzed as the comparative form of *many*; the logical form (33b) says that there exists a positive integer δ such that Ann won (at least) δ-many orders but Harry did not. Table 1 shows some more examples of logical forms of constructions with numerical adjectives.

Comparative quantificational determiners 2.5.2

We also use the predicate $\mathsf{many}(x, n)$ to analyze proportional quantifiers such as *most* and *at least half of*. For example, the sentence (34a) is analyzed as meaning "More than half of *A* is *B*", following the standard truth-condition (Barwise and Cooper 1981), and can be represented as (34b). The logical form in (34b) implies that there are more red apples than non-red apples. The sentence (35a) with *at most half of* is analyzed as meaning "Less than or equal to half of A is B", and is mapped to the logical form with the negation in (35b).[14]

(34) a. *Most* apples are red.

b. $\exists\delta(\exists x(\mathsf{apple}(x) \land \mathsf{red}(x) \land \mathsf{many}(x,\delta))$
$\land \neg\exists x(\mathsf{apple}(x) \land \neg\mathsf{red}(x) \land \mathsf{many}(x,\delta)))$

(35) a. *At most half of* apples are red.

b. $\neg\exists\delta(\exists x(\mathsf{apple}(x) \land \mathsf{red}(x) \land \mathsf{many}(x,\delta))$
$\land \neg\exists x(\mathsf{apple}(x) \land \neg\mathsf{red}(x) \land \mathsf{many}(x,\delta)))$

This analysis correctly captures the monotonicity property of *most*, according to which *most* is right-upward monotone;[15] thus (34a) entails *Most apples are red or green*. Likewise, *at most half of* in (35a) is right-downward monotone, which is captured in the logical form (35b). Similarly, the sentence (36a) can be analyzed as meaning "More than or equal to half of A is B" and is represented as (36b). The sentence (37a) with *less than half of* is mapped to (37b). Since *less than half of* is also a downward quantifier, we give it the logical form with negation.

(36) a. *At least half of* apples are red.

b. $\forall\delta(\exists x(\mathsf{apple}(x) \land \neg\mathsf{red}(x) \land \mathsf{many}(x,\delta))$
$\rightarrow \exists x(\mathsf{apple}(x) \land \mathsf{red}(x) \land \mathsf{many}(x,\delta)))$

(37) a. *Less than half of* apples are red.

b. $\neg\forall\delta(\exists x(\mathsf{apple}(x) \land \neg\mathsf{red}(x) \land \mathsf{many}(x,\delta))$
$\rightarrow \exists x(\mathsf{apple}(x) \land \mathsf{red}(x) \land \mathsf{many}(x,\delta)))$

---

[14] Since we assume each variable can stand for pluralities, $\mathsf{red}(x)$ should be interpreted as distributive, meaning that each atomic part of x satisfies the predicate *red* (Link 1983). Similarly, $\neg\mathsf{red}(x)$ should be interpreted as meaning that each atomic part of x does not satisfy *red*, where the negation is treated as a predicate modifier. However, it is beyond the scope of this paper to implement the distinction between collective and distributive predication, so we leave a full treatment of the semantics of pluralities to future work.

[15] Let Q be a quantifier and A and B be its restrictor and nuclear scope, respectively. The quantifier Q is *right-upward monotone* if $Q(A,B)$ and $B \subseteq C$ entail $Q(A,C)$; Q is *right-downward monotone* if $Q(A,B)$ and $C \subseteq B$ entail $Q(A,C)$. For the classification of generalized quantifiers and monotonicity properties, see e.g., Barwise and Cooper (1981) and Westerstaåhl (2007).

ID	Premises and hypothesis	Gold label
253	*P*: At most half of the students take the class. *H*: Less than half of the students take the class.	Unknown
254	*P*: Most students take the class. *H*: None of the students take the class.	No
255	*P*: Less than half of the students take the class. *H*: Most students take the class.	No
256	*P*: More than half of the students take the class. *H*: Most students take the class.	Yes
257	*P*: Most students take the class. *H*: At least half of the students take the class.	Yes

Table 2: Examples of entailment problems for generalized quantifiers from CAD

The above analysis shows that monotonicity inferences with proportional quantifiers can be handled in typed FOL with arithmetic by assigning logical forms based on A-not-A analysis. Table 2 shows some examples of entailment relations with sentences containing the expressions described above. These are extracted from CAD dataset we will use for evaluation (see Section 4.2).

2.5.3 Comparatives and quantifiers

When determiners such as *all* or *some* appear in *than*-clauses, we need to consider the scope of the corresponding quantifiers (Larson 1988). As examples, (38a) and (39a) are assigned the logical forms in (38b) and (39b), respectively.

(38) a. Mary is taller than every student.

b. $\forall y(\mathsf{student}(y) \rightarrow \exists \delta(\mathsf{tall}(\mathsf{mary}, \delta) \wedge \neg \mathsf{tall}(y, \delta)))$

(39) a. Mary is taller than some student.

b. $\exists y(\mathsf{student}(y) \wedge \exists \delta(\mathsf{tall}(\mathsf{mary}, \delta) \wedge \neg \mathsf{tall}(y, \delta)))$

Conjunction (*and*) and disjunction (*or*) appearing in a *than*-clause show different behaviors in scope taking, as pointed out in Larson (1988). For instance, in (40a), the conjunction *and* takes wide scope over the main clause, whereas in (41a), the disjunction *or* can take narrow scope. Thus, we can infer *Mary is taller than Harry* from both (40a) and (41a). These readings are represented as in (40b) and (41b), respectively.

(40) a. Mary is taller than Harry and Bob.

b. $\exists\delta(\mathsf{tall}(\mathsf{mary},\delta) \wedge \neg\mathsf{tall}(\mathsf{harry},\delta))$
$\wedge \exists\delta(\mathsf{tall}(\mathsf{mary},\delta) \wedge \neg\mathsf{tall}(\mathsf{bob},\delta))$

(41) a. Mary is taller than Harry or Bob.

b. $\exists\delta(\mathsf{tall}(\mathsf{mary},\delta) \wedge \neg(\mathsf{tall}(\mathsf{harry},\delta) \vee \mathsf{tall}(\mathsf{bob},\delta)))$

The quantifiers in the *than*-clause as in the sentences (38a), (39a), and (40a) need to take wide scope, while that in (41a) needs to take narrow scope. To derive this kind of scope ambiguity is not the focus of the current study and remains unsolved in our implementation. We use a fixed scope relation for quantifiers in *than*-clauses and take the wide scope reading as in (38a), (39a), and (40a) as a default reading.

3 COMPOSITIONAL SEMANTICS

In this section, we present an overview of compositional semantics that maps various comparative constructions in English to logical forms. We use CCG as a syntactic framework, a lexicalized grammar formalism that provides a transparent syntax-semantics interface (Steedman 1996, 2000). To implement a fully automated system, we use off-the-shelf CCG parsers (Clark and Curran 2007; Lewis and Steedman 2014; Yoshikawa *et al.* 2017), which are based on English CCGBank (Hockenmaier and Steedman 2007). Though it has been pointed out that there is room to improve English CCGBank with respect to the analysis of comparative constructions (Honnibal *et al.* 2010), it provides a reasonably fine-grained and rich syntactic structure that derives the type of logical forms suitable for our purposes, as we will show below. A point of using existing resources such as CCGBank is to make explicit what can be done in currently available treebanks and parsers. This would make clear the potentials and limitations of the current English CCGBank, thereby contributing to the acceleration of the study of computational semantics based on treebanks.

Table 3: Lexical entries for basic categories

Category	Logical form	Example
N	ann	*Ann*
N	$\lambda x.\mathsf{boy}(x)$	*boy*
NP/N	$\lambda FG.\exists x.(F(x) \wedge G(x))$	*a*
NP/N	$\lambda FG.\forall x.(F(x) \rightarrow G(x))$	*every*
$S\backslash NP$	$\lambda Q.Q(\lambda x.\exists e.(\mathsf{run}(e) = x))$	*run*
$S\backslash NP/NP$	$\lambda Q_1 Q_2.Q_1(\lambda y.Q_2(\lambda x.\exists e.\mathsf{love}(e) \wedge (\mathsf{subj}(e) = x) \wedge (\mathsf{obj}(e) = y)))$	*love*

CCG-style Compositional semantics for comparatives 3.1

In CCG-style compositional semantics, the mapping from syntax to semantics is defined by assigning a syntactic category to each word. The logical form of a sentence is then compositionally derived using the standard λ-calculus. In CCGBank, major basic (ground) syntactic categories consist of N (noun), NP (noun phrase), and S (sentence). Functional categories are of the form $X\backslash Y$ and X/Y, which derives an expression of category X when combined with an expression of category Y to its left and right, respectively. Thus, category $S\backslash NP$ expects an expression of category NP to its left and produces an expression of category S, which plays the role of intransitive verbs. Similarly, $S\backslash NP/NP$ is a category for a transitive verb.[16]

There is a correspondence between syntactic categories and semantic types: if E_1 and E_2 are expressions assigned the same category, then the semantic types of E_1 and E_2 necessarily become the same. Table 3 shows a list of major lexical entries with semantic representations.[17]

To see how to derive a logical form from a CCG parsing tree based on English CCGBank, let us start with a simple example:

(42) Ann saw a boy.

[16] \ and / are left-associative; $S\backslash NP/NP$ means $(S\backslash NP)/NP$.

[17] In CCGBank, a proper noun such as *Ann* is assigned the category N and shifted to NP by the unary rule *lex*, to which we assign the semantics $N : \mathsf{ann} \Rightarrow NP : \lambda F.F(\mathsf{ann})$.

$$
\dfrac{
\dfrac{\dfrac{\text{Ann}}{\begin{array}{c}N\\ \mathsf{ann}\end{array}}}{\begin{array}{c}NP\\ \lambda F.F(\mathsf{ann})\end{array}}\,\text{lex}
\qquad
\dfrac{
\dfrac{\text{saw}}{\begin{array}{c}(S\backslash NP)/NP\\ \lambda Q_1Q_2.Q_2(\lambda y.Q_1(\lambda x.\exists e.(\mathsf{see}(e)\\ \wedge(\mathsf{subj}(e)=y)\wedge(\mathsf{obj}(e)=x))))\end{array}}
\quad
\dfrac{
\dfrac{\text{a}}{\begin{array}{c}NP/N\\ \lambda F_1F_2.\exists x.(F_1(x)\\ \wedge F_2(x))\end{array}}
\quad
\dfrac{\text{boy}}{\begin{array}{c}N\\ \lambda x.\mathsf{boy}(x)\end{array}}
}{\begin{array}{c}NP\\ \lambda F_2.\exists x.(\mathsf{boy}(x)\wedge F_2(x))\end{array}}>
}{\begin{array}{c}S\backslash NP\\ \lambda Q_2.Q_2(\lambda y.\exists x.(\mathsf{boy}(x)\wedge \exists e.(\mathsf{see}(e)\wedge(\mathsf{subj}(e)=y)\wedge(\mathsf{obj}(e)=x))))\end{array}}>
}{\begin{array}{c}S\\ \exists x.(\mathsf{boy}(x)\wedge\exists e.(\mathsf{see}(e)\wedge(\mathsf{subj}(e)=\mathsf{ann})\wedge(\mathsf{obj}(e)=x)))\end{array}}<
$$

Figure 1: Parsing tree of *Ann saw a boy*

The parsing tree with logical forms looks as in Figure 1.[18] Here to accommodate our compositional semantics to English CCGBank, it is convenient to use Argument Raising (Hendriks 1993), which assigns a λ-term of the quantifier type $(e \to t) \to t$ to an expression of category *NP*. Thus a transitive verb is assigned a lambda term of type $((e \to t) \to t) \to ((e \to t) \to t) \to t$.

Given this background, let us see how to derive a suitable logical form to adjectival and comparative constructions. Here are three basic constructions with their logical form under our A-not-A analysis.

(43) a. Ann is tall. $\mathsf{tall}(\mathsf{ann}, \theta_{\mathsf{tall}})$

b. Ann is taller than Bob. $\exists\delta(\mathsf{tall}(\mathsf{ann}, \delta) \wedge \neg\mathsf{tall}(\mathsf{bob}, \delta))$

c. Ann is as tall as Bob. $\forall\delta(\mathsf{tall}(\mathsf{bob}, \delta) \to \mathsf{tall}(\mathsf{ann}, \delta))$

To derive these logical forms compositionally, there are two main questions to be addressed: (i) which constituent introduces a degree variable and (ii) how to “saturate” the degree variables in terms of a threshold value as in (43a), existential closure as in (43b), or universal quantification as in (43c). For (i), we take it that adjectives themselves

[18] The variable convention for major semantic types we adopt throughout the paper is as follows. Each variable can be attached subscripts like x_1, x_2.

Variable	Type	Description
x, y, z	e	entities
δ	d	degrees
F, G	$e \to t$	predicates
Q	$(e \to t) \to t$	quantifiers

introduce degree variable.[19] Thus, under the argument raising analysis we adopt, the basic semantic representation for the adjective *tall* is $\lambda Q\delta.Q(\lambda x.\mathsf{tall}(x,\delta))$, though a more complicated form will be needed as explained below. For (ii), we introduce an empty category into the adjunct position (i.e., a position where a measure phrase appears as in *4 feet tall*), to control the compositional derivations of the three types of logical forms.[20] Since English CCGBank does not support this type of empty categories, we insert them in the post-processing process of syntactic parsing. That is, we rewrite each tree in the following way.

- Empty category *pos* for positive form

$$\dfrac{\dfrac{\text{is}}{(S\backslash NP)/(S_{adj}\backslash NP)} \quad \dfrac{\text{tall}}{S_{adj}\backslash NP}}{S\backslash NP}> \quad \dashrightarrow \quad \dfrac{\dfrac{\text{is}}{(S\backslash NP)/(S_{adj}\backslash NP)} \quad \dfrac{\dfrac{\text{pos}}{(S_{adj}\backslash NP)/(S_{adj}\backslash NP)} \quad \dfrac{\text{tall}}{S_{adj}\backslash NP}}{S_{adj}\backslash NP}>}{S\backslash NP}>$$

- Empty category *dgr* for comparative form

$$\dfrac{\dfrac{\text{taller}}{S_{adj}\backslash NP} \quad \dfrac{\text{than Bob}}{(S_{adj}\backslash NP)\backslash(S_{adj}\backslash NP)}}{S_{adj}\backslash NP}< \quad \dashrightarrow \quad \dfrac{\dfrac{\dfrac{\text{dgr}}{(S_{adj}\backslash NP)/(S_{adj}\backslash NP)} \quad \dfrac{\text{taller}}{S_{adj}\backslash NP}}{S_{adj}\backslash NP}> \quad \dfrac{\text{than Bob}}{(S_{adj}\backslash NP)\backslash(S_{adj}\backslash NP)}}{S_{adj}\backslash NP}<$$

- Empty category *dgr2* for equative

$$\dfrac{\dfrac{\text{as tall}}{S_{adj}\backslash NP} \quad \dfrac{\text{as Bob}}{(S_{adj}\backslash NP)\backslash(S_{adj}\backslash NP)}}{S_{adj}\backslash NP}< \quad \dashrightarrow \quad \dfrac{\dfrac{\dfrac{\text{dgr2}}{(S_{adj}\backslash NP)/(S_{adj}\backslash NP)} \quad \dfrac{\text{as tall}}{S_{adj}\backslash NP}}{S_{adj}\backslash NP}> \quad \dfrac{\text{as Bob}}{(S_{adj}\backslash NP)\backslash(S_{adj}\backslash NP)}}{S_{adj}\backslash NP}<$$

---

[19] See Klein (1991), among others. See also Klein (1980, 1982) for views against this type of analysis.

[20] Instead we could introduce type-shifting rules that correspond to the empty categories.

The parsing tree for each sentence in (43) is shown in Figures 2, 3, and 4, respectively.[21] We assign a uniform semantic representation to each adjective, following the strategy of *generalizing to the worst case* (Montague 1970). An adjective (e.g., *tall*) and its comparative form (e.g., *taller*) of category $S_{adj}\backslash NP$ are uniformly assigned the following term:

(44) $\lambda Q\delta HI.Q(I(\lambda x.\mathsf{tall}(x,\delta), H(\mathsf{tall},\delta)))$

This term is combined with the other terms including empty elements to form the relevant logical form as illustrated in Figures 2, 3, and 4. For comparison, Figure 5 shows the parsing tree for the case where the explicit degree modifier *4 feet* appears in the adjunct position.

We introduce two variables H and I in the semantic representation in (44). H can be filled in different ways to control the meaning of a *than*-clause, as illustrated in Figure 2 where there is no *than*-clause or Figure 3 where there is a noun phrase in the *than*-clause. I is used to determine whether the entire logical form is of existential type as in (43b) or of universal type as in (43c). We ascribe the negation in A-not-A analysis to *than*, following the analysis of *than*-clauses as introducing negative contexts as presented in the categorial grammar literature (Hendriks 1995).

$$
\dfrac{
\dfrac{\dfrac{\text{Ann}}{N} \atop \mathsf{ann}}{\begin{matrix} NP \\ \lambda F.F(\mathsf{ann}) \end{matrix}}\,\text{lex}
\qquad
\dfrac{
\dfrac{\text{is}}{\begin{matrix}(S\backslash NP)/(S_{adj}\backslash NP) \\ \lambda AQ.Q(\lambda x.A(\lambda F_1.F_1(x),\delta',H',I'))\end{matrix}}
\quad
\dfrac{
\dfrac{\text{pos}}{\begin{matrix}(S_{adj}\backslash NP)/(S_{adj}\backslash NP) \\ \lambda AQ\delta HI.A(Q,\theta_{\mathsf{tall}},\lambda H_1\delta.\top, \\ \lambda F_1F_2x.F_1(x))\end{matrix}}
\quad
\dfrac{\text{tall}}{\begin{matrix}S_{adj}\backslash NP \\ \lambda Q\delta HI.Q(I(\lambda x.\mathsf{tall}(x,\delta), \\ H(\mathsf{tall},\delta)))\end{matrix}}
}{\begin{matrix}S_{adj}\backslash NP \\ \lambda Q\delta HI.Q(\lambda x.\mathsf{tall}(x,\theta_{\mathsf{tall}}))\end{matrix}}>
}{\begin{matrix}S\backslash NP \\ \lambda Q.Q(\lambda x.\mathsf{tall}(x,\theta_{\mathsf{tall}}))\end{matrix}}>
}{\begin{matrix}S \\ \mathsf{tall}(\mathsf{ann},\theta_{\mathsf{tall}})\end{matrix}}<
$$

Figure 2: Parsing tree of *Ann is tall*

[21] In these semantic representations, δ', H', and I' are constants to be applied to the vacuous λ-abstraction appearing in the term of category $S_{adj}\backslash NP$.

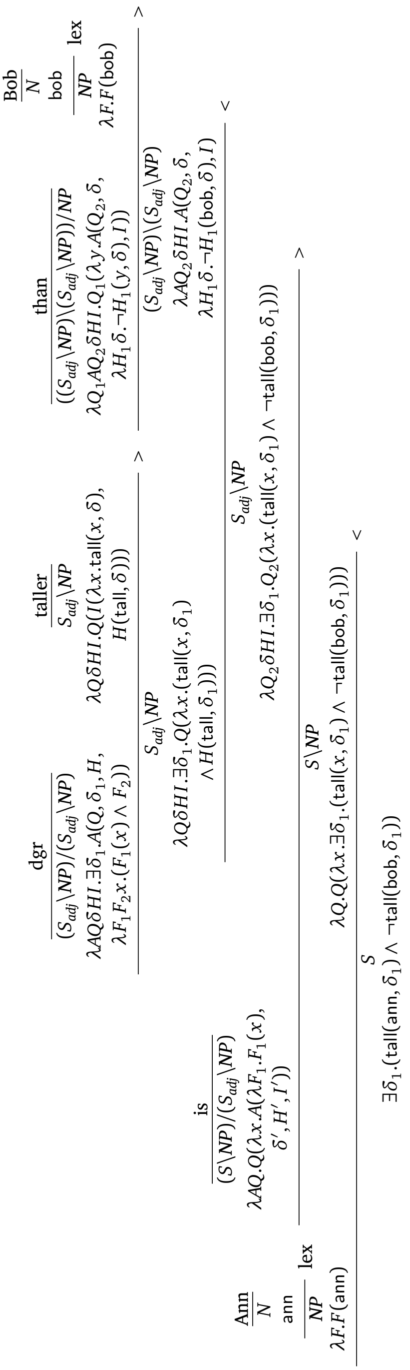


Figure 3: Parsing tree of *Ann is taller than Bob*

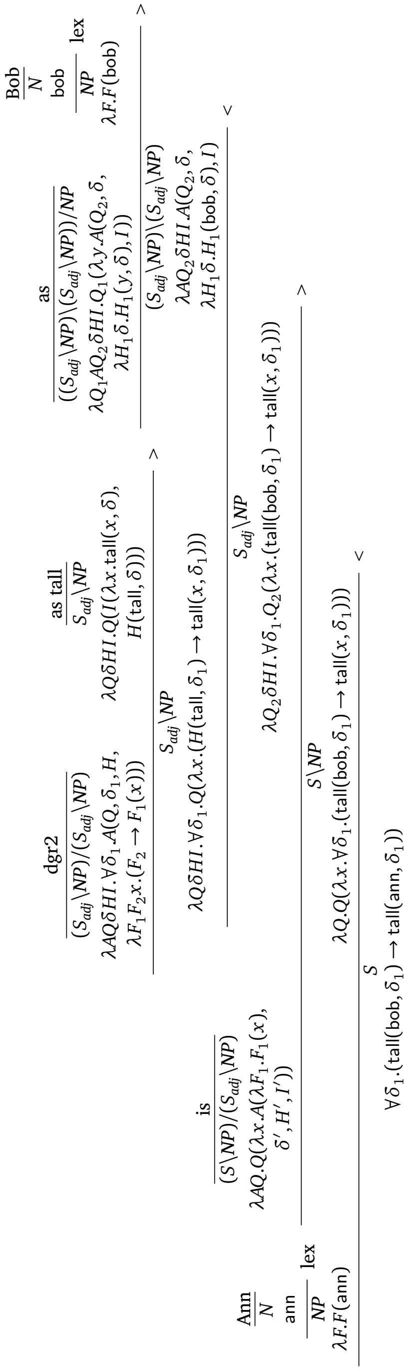


Figure 4: Parsing tree of *Ann is as tall as Bob*

$$\dfrac{\text{4 feet}}{(S_{adj}\backslash NP)/(S_{adj}\backslash NP)} \quad \lambda AQ\delta HI.\exists\delta_1.A(Q,\delta_1,\lambda H_1\delta.\top, \lambda F_1F_2x.(F_1(x) \wedge (\delta_1 = 4)))$$

$$\dfrac{\text{tall}}{S_{adj}\backslash NP} \quad \lambda Q\delta HI.Q(I(\lambda x.\mathsf{tall}(x,\delta)), H(\mathsf{tall},\delta)))$$

$$\overline{S_{adj}\backslash NP \quad \lambda Q\delta HI.\exists\delta_1.Q(\lambda x.(\mathsf{tall}(x,\delta_1) \wedge (\delta_1 = 4)))} \; >$$

$$\dfrac{\text{is}}{(S\backslash NP)/(S_{adj}\backslash NP)} \quad \lambda AQ.Q(\lambda x.A(\lambda F_1.F_1(x), \delta', H', I'))$$

$$\overline{S\backslash NP \quad \lambda Q.Q(\lambda x.\exists\delta_1.(\mathsf{tall}(x,\delta_1) \wedge (\delta_1 = 4)))} \; >$$

$$\dfrac{\dfrac{\text{Ann}}{N} \quad \mathsf{ann}}{NP} \,\text{lex} \quad \lambda F.F(\mathsf{ann})$$

$$\overline{S \quad \exists\delta_1.(\mathsf{tall}(\mathsf{ann},\delta_1) \wedge (\delta_1 = 4))} \; <$$

Figure 5: Parsing tree for *Ann is 4 feet tall*

3.2 *Generalized quantifiers and numeral adjectives*

Determiners such as *every*, *no*, and *most* are assigned the category NP/N in CCGBank. Table 4 shows some representative examples of lexical entries for determiners. The lexical entry for *most* here derives the desired logical form in (34).

To see how to give a compositional analysis of numeral adjectives in our framework, let us first take a look at modified numerals. Here we need to distinguish three types of NPs according to their *monotonicity* property (Barwise and Cooper 1981), upward monotonic (e.g., *at least two*), downward monotonic (e.g., *at most two*), and non-monotonic (e.g., *exactly two*). Table 5 gives lexical entries for these three types of modifiers. Here we use the category *Num* for numeral expressions such as *two*. For bare numerals like *two* in (45a), we shift the category *Num* to NP/N, which yields the term $\lambda F_1F_2.\exists x(F_1(x) \wedge F_2(x) \wedge \mathsf{many}(x,2))$. This allows us to derive the logical form in (45b):

Table 4: Lexical entries for quantifiers

Expression	Syntactic category	LF
every	NP/N	$\lambda F_1F_2.\forall x(F_1(x) \to F_2(x))$
some	NP/N	$\lambda F_1F_2.\exists x(F_1(x) \wedge F_2(x))$
no	NP/N	$\lambda F_1F_2.\neg\exists x(F_1(x) \wedge F_2(x))$
most	NP/N	$\lambda F_1F_2.\exists\delta(\exists x(F_1(x) \wedge F_2(x) \wedge \mathsf{many}(x,\delta))$ $\wedge\neg\exists y(F_1(y) \wedge \neg F_2(y) \wedge \mathsf{many}(y,\delta)))$

Table 5: Lexical entries for monotonicity

Expression	Syntactic category	Logical form
2	*Num*	2
at least	$(NP/N)/Num$	$\lambda\delta F_1 F_2.\exists x(F_1(x) \wedge F_2(x) \wedge \mathsf{many}(x,\delta))$
at most	$(NP/N)/Num$	$\lambda\delta F_1 F_2.\neg\exists x(F_1(x) \wedge F_2(x) \wedge \mathsf{many}(x,\delta+1))$
exactly	$(NP/N)/Num$	$\lambda\delta F_1 F_2.(\exists x(F_1(x) \wedge F_2(x) \wedge \mathsf{many}(x,\delta))$ $\wedge\forall\delta_1(\exists x(F_1(x) \wedge F_2(x) \wedge \mathsf{many}(x,\delta_1)) \rightarrow (\delta_1 \leq \delta)))$
$\phi_{exactly}$	$(NP/N)/Num$	$\lambda\delta F_1 F_2.(\exists x(F_1(x) \wedge F_2(x) \wedge \mathsf{many}(x,\delta))$ $\wedge\forall\delta_1(\exists x(F_1(x) \wedge F_2(x) \wedge \mathsf{many}(x,\delta_1)) \rightarrow (\delta_1 \leq \delta)))$

(45) a. Mary read two books. (Upward)

b. $\exists x(\mathsf{book}(x) \wedge \mathsf{many}(x,2) \wedge \exists e(\mathsf{read}(e) \wedge (\mathsf{subj}(e) = \mathsf{mary}) \wedge (\mathsf{obj}(e) = x)))$

For numeral modifiers such as *at least*, we give the category $(NP/N)/Num$. Figure 6 shows an example derivation. The following is an example of a sentence involving a downward monotonic modifier *less than*.

(46) a. Mary read less than two books. (Downward)

b. $\neg\exists x(\mathsf{book}(x) \wedge \mathsf{many}(x,2) \wedge \exists e(\mathsf{read}(e) \wedge (\mathsf{subj}(e) = \mathsf{mary}) \wedge (\mathsf{obj}(e) = x)))$

$$\dfrac{\dfrac{\dfrac{\text{at least}}{\begin{array}{c}(NP/N)/Num\\ \lambda\delta F_1 F_2.(\exists x(F_1(x) \wedge F_2(x) \wedge \mathsf{many}(x,\delta))\end{array}} \quad \dfrac{\text{two}}{\begin{array}{c}Num\\ 2\end{array}}}{\begin{array}{c}NP/N\\ \lambda F_1 F_2.(\exists x(F_1(x) \wedge F_2(x) \wedge \mathsf{many}(x,2))\end{array}}> \quad \dfrac{\text{books}}{\begin{array}{c}N\\ \lambda x.\mathsf{book}(x)\end{array}}}{\begin{array}{c}NP\\ \lambda F_2.(\exists x(\mathsf{book}(x) \wedge F_2(x) \wedge \mathsf{many}(x,2))\end{array}}>$$

Figure 6: Parsing tree of *at least two books*

Similarly, we assign syntactic categories like $(NP/N)/Num$ to non-monotonic quantifiers such as *exactly* and *only*. This allows the sentence (47a) to be assigned the complex logical form (47b), which adds the meaning "the number of books Mary read is less than or equal to two" to (45b).

(47) a. Mary read exactly two books. (Non-monotonicity)

b. $\exists x(\mathsf{book}(x) \wedge \mathsf{many}(x, 2) \wedge \exists e(\mathsf{read}(e) \wedge (\mathsf{subj}(e) = \mathsf{mary}) \wedge (\mathsf{obj}(e) = x))) \wedge \forall x \forall \delta(\mathsf{book}(x) \wedge \mathsf{many}(x, \delta) \wedge \exists e(\mathsf{read}(e) \wedge (\mathsf{subj}(e) = \mathsf{mary}) \wedge (\mathsf{obj}(e) = x)) \rightarrow (\delta \leq 2))$

Here (45a) has the *at least* reading glossed as "Mary read *at least* two books". However, it is often natural to interpret (45a) as "Mary read *exactly* three books". This *exactly* reading is usually derived pragmatically as scalar implicature (SI) (Horn 1973; Gazdar 1979; van Rooij and Schulz 2004). To account for this reading, as an initial attempt, we implement the mechanism of scalar implicature in our system. For this purpose, we use empty category $\phi_{exactly}$, which derives the same interpretation as in (47b) for (45a). Thus the system can distinguish two logical forms for a sentence involving a bare numeral, depending on the environment in which it appears.[22]

This type of pragmatic ambiguity is related to the fact that $\mathsf{tall}(x, \delta)$ is not interpreted as "x is *exactly* as tall as δ" but as "x is *at least* as tall as δ", as mentioned in Section 2.3.1. Thus by inserting the $\phi_{exactly}$ operator we can uniformly derive SI readings for sentences with numerical expressions as in (45), equatives as in (48), measure phrases as in (49) and (50).

(48) a. Tom is as tall as Mary.
⇝ Tom is *exactly* as tall as Mary.

b. $\forall \delta(\mathsf{fast}(\mathsf{mary}, \delta) \longleftrightarrow \mathsf{fast}(\mathsf{tom}, \delta))$

(49) a. John is 5 cm shorter than Bob.
⇝ John is *exactly* 5 cm shorter than Bob.

b. $\forall \delta(\mathsf{short}(\mathsf{bob}, \delta) \longleftrightarrow \mathsf{short}(\mathsf{john}, \delta - 5\ \mathsf{cm}))$

(50) a. Bob is 170 cm tall.
⇝ Bob is *exactly* 170 cm tall.

b. $\mathsf{tall}(\mathsf{bob}, 170\ \mathsf{cm}) \wedge \forall \delta(\mathsf{tall}(\mathsf{bob}, \delta) \rightarrow (\delta \leq 170\ \mathsf{cm}))$

On the other hand, negative sentences from (51) to (53) have *at least* reading (see Spector (2013) for an overview). Thus, we do not insert the empty categories in the following constructions.

[22] This strategy is similar to the grammatical encoding of scalar implicature proposed by Chierchia (2004).

(51) a. Peter didn't solve ten problems.

b. $\neg\exists x(\mathsf{problem}(x) \wedge \mathsf{solve}(\mathsf{peter}, x) \wedge \mathsf{many}(x, 10))$

(52) a. Tom is not as tall as Mary.

b. $\neg\forall\delta(\mathsf{tall}(\mathsf{mary}, \delta) \rightarrow \mathsf{tall}(\mathsf{tom}, \delta))$

(51a) can be interpreted to mean that Peter solved no more than nine problems, i.e., the number of problems Peter solved is less than ten. To derive the reading in (51b), we need to assign the *at least* reading to the numeral adjective *ten*. Similarly, the equative construction with the negation in (52a) has the *at least* reading as in (52b).

Such differences in interpretation occur not only in negation but also more generally in downward environments triggered by negative adjectives such as *fewer than five* and *few*, as well as in the antecedent of a conditional and the restrictor of a universal quantifier.[23]

(53) a. Fewer than five children play in the park.

b. Few boys had three cookies.

c. If Andy is 5 feet tall, he is taller than Bob.

d. Every student who solved 10 problems passed.

We apply the same technique to derive two reading of the determiner *any* (Kadmon and Landman 1993), the existential reading as in (54a) and the universal reading as in (54b).

(54) a. Bob did not take any exams. (Existential reading)

b. Any owl hunts mice. (Universal reading)

The existential reading is known to be allowed only if *any* appears within the range of DOWNWARD ENTAILING (DE) operators (DE environments) that reverse the direction of entailment, such as negative expressions (Ladusaw 1979). We assume that there is lexical ambiguity in that *any* as an NPI has an existential meaning (Horn 1973; Ladusaw 1979), while *any* as free choice has a universal meaning (Carlson 1981).

To derive two interpretations, we determine from the CCG parsing trees whether *any* appears in the DE environment. Specifically, when

[23] Note that there is disagreement as to whether hypothetical clauses are truly SI-free; see the discussion in Breheny (2008) and Spector (2013).

any appears in a non-DE environment, we assign a universal meaning ($any_{\forall}$), and when *any* appears in a DE environment, we assign an existential meaning ($any_{\exists}$). This is accomplished in the same way as the process for deriving SIs as described before.

3.3 Compositional event semantics and adverbial comparatives

For the compositional account of adverbs and adverbial comparatives, we basically follow the implementation of compositional event semantics presented in Martínez-Gómez *et al.* (2017), which derives the logical form (55b) from the sentence (55a). The compositional derivation is shown in Figure 7.

(55) a. Tim ran fast.

b. $\exists e(\mathsf{run}(e) \wedge (\mathsf{subj}(e) = \mathsf{tim}) \wedge \mathsf{fast}(e, \theta_{\mathsf{fast}}))$

To derive the logical form in (55b) compositionally, we follow Champollion (2015) to use a continuation variable K which is to be filled in by an adverbial element; If there is no adverbial element as in the root of the parsing tree, it is filled by the constant $\top$ (meaning "true"). We also need to introduce an empty category *pos* that sets the threshold value to θ_{tall}, in a similar way to the treatment of positive adjectives.

4 EXPERIMENTS

We implemented our system and evaluated it on various NLI datasets. All code and data, including visualized CCG parsing trees with logical forms obtained for each dataset, are made publicly available at `https://github.com/izumi-h/ccgcomp`.

4.1 System architecture

Figure 8 shows the pipeline of the proposed system. First, the input consists of a set of premises $P_1, \ldots, P_n$ and a hypothesis H, which are mapped to CCG parsing trees. The trees are converted so that they

$$\dfrac{\dfrac{\text{Tim}}{N}}{}\quad \dfrac{\text{tim}}{NP}\ \text{lex} \qquad \lambda F.F(\mathsf{tim})$$

$$\dfrac{\text{ran}}{S\backslash NP} \qquad \lambda QK.Q(\lambda x.\exists e.(\mathsf{run}(e) \wedge (\mathsf{Subj}(e) = x) \wedge K(e)))$$

$$\dfrac{\text{pos}}{((S\backslash NP)\backslash(S\backslash NP))/((S_{adv}\backslash NP)\backslash(S_{adv}\backslash NP))} \qquad \lambda AVQK.Q(\lambda x.A(V, \lambda F.F(x), \theta_{\mathsf{fast}}, \lambda H_1\delta.\top, \lambda A_1 A_2.A_1, K))$$

$$\dfrac{\text{fast}}{(S_{adv}\backslash NP)\backslash(S_{adv}\backslash NP)} \qquad \lambda VQ\delta HIK.V(Q, \lambda e.I((\mathsf{fast}(e,\delta) \wedge K(e)), H(\mathsf{fast},\delta)))$$

$$\dfrac{\text{pos} \quad \text{fast}}{(S\backslash NP)\backslash(S\backslash NP)}\ > \qquad \lambda VQK.Q(\lambda x.V(\lambda F.F(x), \lambda e.(\mathsf{fast}(e,\theta_{\mathsf{fast}}) \wedge K(e))))$$

$$\dfrac{\text{ran} \quad \text{pos fast}}{S\backslash NP}\ < \qquad \lambda QK.Q(\lambda x.\exists e.(\mathsf{run}(e) \wedge (\mathsf{Subj}(e) = x) \wedge \mathsf{fast}(e,\theta_{\mathsf{fast}}) \wedge K(e)))$$

$$\dfrac{\text{Tim} \quad \text{ran pos fast}}{S}\ < \qquad \lambda K.\exists e.(\mathsf{run}(e) \wedge (\mathsf{Subj}(e) = \mathsf{tim}) \wedge \mathsf{fast}(e,\theta_{\mathsf{fast}}) \wedge K(e))$$

Figure 7: Parsing tree for *Tim ran fast*

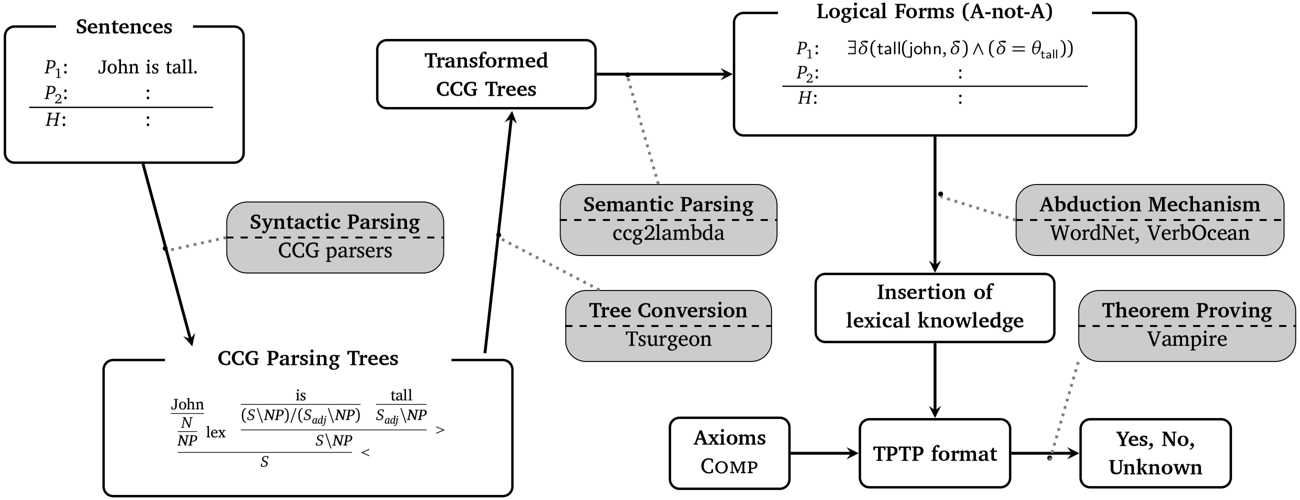


Figure 8: Overview of the proposed system

are suitable for our compositional semantics described in Section 3. The modified trees are mapped to logical forms. Before the process of theorem-proving, the abduction mechanism searches for lexical relations holding on the predicates in the mapped logical forms and introduces them as axioms. Then, a theorem prover checks whether $P_1 \wedge \cdots \wedge P_n \rightarrow H$ holds, potentially with the aid of the axioms. The system outputs *yes* (entailment) if $P_1 \wedge \cdots \wedge P_n \rightarrow H$ can be proved by a theorem prover, and outputs *no* (contradiction) if the negation of the hypothesis (i.e., $P_1 \wedge \cdots \wedge P_n \rightarrow \neg H$) can be proved. If both fail, it tries to construct a counter-model and outputs *unknown* (neutral) if a counter model is found or a timeout occurs.

We build the system on top of off-the-shelf CCG parsers and a theorem prover. To these existing tools, we mainly add three components, (1) rules to transform CCG derivation trees, (2) rules to map CCG derivation trees to logical forms, and (3) axioms for comparatives to derive theorems. We will explain each step in the pipeline in detail.

1. Syntactic parsing To obtain CCG parsing trees we use three CCG parsers to mitigate parsing errors: C&C (Clark and Curran 2007), EasyCCG (Lewis and Steedman 2014), and depccg (Yoshikawa *et al.* 2017). For all parsers, we use the standard model trained on the original CCGBank. We also use POS tagging to supplement the information available from CCG trees. For example, CCG categories do not distinguish positive and comparative forms of adjectives. To remedy this, we use POS tags *JJ* and *JJR* for positive and comparative forms. For POS tagging, we use the C&C POS tagger for C&C and spaCy[24] for depccg.

2. Tree conversion To modify CCG parsing trees, we use Tsurgeon (Levy and Andrew 2006). We use 125 entries (regex rewriting rules) in the Tsurgeon script. In addition to modifying trees, we use the following rules to add information needed to derive logical forms in our compositional semantics. There are five types of rewriting rules.

- Multiword Expression. We add rules to join multiword expressions for determiners; e.g. *a lot of* to *a∼lot∼of* and *a few* to *a∼few*.
- Empty category. We insert empty categories and add syntactic features to CCG categories as described in Section 3.

[24] `https://github.com/explosion/spaCy`

- Adjective type. Based on the analysis presented in Section 2, we classify adjectives into six types: extensional positive (*POS*), extensional negative (*NEG*), intensional positive (*POS-INT*), intensional negative (*NEG-INT*), non-gradable (*PRE*), or non-subsective (*N-SUB*). To classify positive and negative adjectives, we use SentiWordNet (Baccianella *et al.* 2010). For the other types, we prepare hand-rewritten rules for a set of the adjectives appearing in the FraCaS dataset.
- Negative Polarity *any*. We distinguish $any_\forall$ and $any_\exists$ according to its environment as described in Section 3.2.
- Lemmatization. Comparative forms of adverbs are converted to positive forms (e.g., *faster* to *fast*), and positive forms of adverbs are converted to corresponding adjectives (e.g., *slowly* to *slow*). We use the WordNet (Miller 1995) library in NLTK[25] for this conversion.

3. Semantic parsing To implement compositional semantics, we use the semantic parsing platform ccg2lambda (Martínez-Gómez *et al.* 2016), which uses λ-calculus to obtain logical forms. We extend the schematic lexical entries (called semantic templates) for FOL event semantics proposed in Martínez-Gómez *et al.* (2017) to handle linguistic phenomena based on degree-based semantics. In this system, semantic parsing is performed using two different semantic templates to manipulate the scope of negation in logical forms. If input sentences contain the negation *not* or *n't*, the proof is attempted in two different logical forms with negation taking wide scope or narrow scope. The total number of lexical entries assigned to CCG categories is 551, and the number of entries directly assigned to particular words (e.g., *than* and *as* for comparatives and items for quantifiers) is 151.

4. Abduction mechanism To handle basic lexical inferences, we adapt an abduction mechanism presented in Martínez-Gómez *et al.* (2017) to our framework. Given logical forms for premises, the abduction mechanism searches lexical relations from two lexical knowledge bases: WordNet (Miller 1995) and VerbOcean (Chklovski and Pantel 2004). Following Martínez-Gómez *et al.* (2017), we use seven rela-

[25] `https://www.nltk.org/`

tionships such as antonym and hypernym and add the corresponding axioms. The acquisition of antonym relations of gradable adjectives such as *tall* and *short* is also based on the use of this mechanism.

5. Theorem proving For theorem proving, we use a resolution-based FOL prover Vampire 4.4 (Kovács and Voronkov 2013),[26] which accepts TFF forms with arithmetic operations. The proof runs in the automatic modes `casc` and `casc_sat`, which automatically select a series of strategies that attempt to prove a particular problem. While `casc` is aimed at solving theorems, `casc_sat` is aimed at solving satisfiable or non-theorem problems, that is, those problems where there is a model in which the premises are true but the conclusion is false (i.e., there is a counter-model for the inference). In our system, we first try to prove the problem in `casc` mode and then try to prove it again in `casc_sat` mode for any problems that are labeled *unknown*. We set the timeout at 7 sec in `casc` mode and 1 sec in `casc_sat`. We add the four axiom schemata described in Section 2, which we call the axiomatic system COMP, before starting the process of theorem proving. Each axiom scheme is instantiated by gradable adjectives appearing in the target sentences.

We run a process of theorem proving for each of the three parsers and obtain three outputs. If the three outputs are different, we choose the system answer in the following way: if two answers are *yes* (resp. *no*), then the system answer is *yes* (resp. *no*), no matter what the other answer is; if one answer is *yes* (resp. *no*) and the others are *unknown*, the system answer is *yes* (resp. *no*); if all answers are different, then the system answer is *unknown*.

4.2 *Datasets*

For evaluation, we use five NLI datasets containing linguistically challenging problems with quantifiers, adjectives, adverbs, comparatives, and lexical knowledge. Table 6 shows some examples in each dataset.
FraCaS FraCaS (Cooper *et al.* 1996) is a dataset comprising nine sections, each of which contains semantically challenging inferences related to various linguistic phenomena. In this study, we target four

[26] `https://github.com/vprover/vampire`

Table 6: Examples of entailment problems from the FraCaS, MED, SICK, HANS, and CAD datasets. They are solved by our system but not by the DL models

Dataset	Label	ID	Example (premises and hypothesis)	Gold label
FraCaS	*Adj*	209	P_1: Mickey is a small animal. P_2: Dumbo is a large animal. H: Mickey is larger than Dumbo.	No
	Com	241	P_1: ITEL won more orders than APCOM lost. P_2: APCOM lost ten orders. H: ITEL won at least eleven orders.	Yes
MED	*gq*	485	P: Exactly 12 aliens threw some tennis balls. H: Exactly 12 aliens threw some balls.	Unknown
		1021	P: More than five campers have had a sunburn or caught a cold. H: More than five campers have caught a cold.	Unknown
	gqlex	176	P: Few aliens saw birds. H: Few aliens saw doves.	Yes
SICK	–	1357	P: A puppy is repeatedly rolling from side to side on its back. H: A dog is rolling from side to side.	Yes
		4789	P: There is no woman riding on an elephant. H: A woman is opening a soda and drinking it.	Unknown
HANS	–	16005	P: Happy authors advised the artists. H: Authors advised the artists.	Yes
		23990	P: The student recommended the author, or the presidents believed the managers. H: The student recommended the author.	Unknown
CAD	–	001	P_1: John is 5 cm taller than Bob. P_2: Bob is 170 cm tall. H: John is 175 cm tall.	Yes
		103	P_1: Bob is not tall. P_2: John is not tall. H: John is taller than Bob.	Unknown
		115	P: Exactly seven students smiled. H: At most nine students smiled.	Yes
		157	P_1: Ann runs as fast as Luis does. P_2: Ann runs slowly. H: Luis runs fast.	No

sections: Generalized Quantifiers (*GQ*: 73 problems), Adjectives (*Adj*: 22 problems), Comparatives (*Com*: 31 problems), and Attitudes (*Att*: 13 problems). The Comparative section contains a complex inference that requires arithmetic operation, such as ID-241 in Table 6.

MED MED (Yanaka *et al.* 2019) collects problems with monotonicity inferences with generalized quantifiers and lexical knowledge via crowdsourcing. We use a portion of the dataset tagged with *gqlex* and *gq*, those inferences that require lexical knowledge (*gqlex*: 691 problems) and those that do not (*gq*: 498 problems).

SICK We use the 2014 version of SemEval (Marelli *et al.* 2014) of SICK dataset. The dataset contains 4,927 problems for test set. SICK is designed to evaluate compositional inferences involving lexical knowledge and logical operations such as negation and quantifiers.

HANS HANS (McCoy *et al.* 2019) is a dataset containing problems that DL-based systems tend to erroneously output *yes* for cases in which they rely on simple heuristics, for example, problems where the hypothesis is a constituent or a sub-string of the premise, such as disjunctive sentences (e.g., HANS-23990 in Table 6), and problems related to those concerning adjectives and adverbs (e.g., ID-16005 in Table 6). The entire test set contains 30,000 problems, which are divided into entailment (*yes*) and non-entailment (*unknown*) problems.

CAD The above four datasets do not cover linguistically interesting inferences such as ones concerned with adverb phrases (e.g., dropping adverbial phrases and comparative forms of adverbs). Accordingly, we created a new dataset containing 257 inference problems concerning adjectives, comparatives, adverbs, and quantifiers. The dataset also includes problems related to SI (29 problems), to which both gold labels for semantic interpretation and pragmatics interpretation (i.e., those considering SIs) are annotated. We collected a set of inferences (13 problems) from linguistics papers (Klein 1982; Lasersohn 2006) and created more problems by adding negation and degree modifiers (e.g., *very*), changing numerical expressions, replacing positive and negative adjectives (e.g., *large* to *small*), or swapping the premise and hypothesis of an inference. Of the 257 problems, 137 are single-premise problems, and 120 are multi-premise problems. The distribution of gold answer labels is (*yes/no/unknown*) = (110/70/77). All of the gold labels were checked by an expert in linguistics.

Results and discussion 4.3

Tables 7, 8, 9, 10, and 11 show the results of the evaluation. We will describe the details of each result from Section 4.3.1 to Section 4.3.5 below. Since MED and HANS use binary labels (*yes* and *unknown*), for these two datasets we modify the system so that it outputs *yes* if the hypothesis can be proved from the premise; otherwise, the output is *unknown*. *Majority* is the accuracy of the majority baseline. Before looking at the details of the results, let us explain the setting of an ablation analysis and the systems being compared.

Ablation analysis To gain insights into the impact of each component, we performed an ablation analysis on overall performance.

- *Plain* is the accuracy of the system with the transformation of CCG parsing trees only.
- +*abduction* is the accuracy achieved by the insertion of lexical knowledge through the implementation of the abduction mechanism, as described in Section 4.1.
- +*rule* is the accuracy achieved by the addition of hand-coded rules. Some errors were caused by failing to assign correct POS tags and lemmas to comparatives. For example, *cleverer* is wrongly assigned *NN* rather than *JJR* (FraCaS-217). To estimate the upper

FraCaS					
Section		*GQ*	*Adj*	*Com*	*Att*
#All		73	22	31	13
Majority		.49	.41	.61	.62
DL	`RB`	.73	.45	.52	.69
Logic	`MN`	.77	.68	.48	.77
	`LP`	.93	.73	–	**.92**
Ours	plain	.96	.82	**.90**	**.92**
	+abduction	.97	.82	**.90**	**.92**
	+abduction +rule	**.99**	**.95**	**.90**	**.92**

Table 7:
Accuracy on FraCaS dataset

Table 8:
Accuracy on MED dataset

MED			
Label		*gq*	*gqlex*
#All		498	691
Majority		.58	.63
DL	BERT	.56	.58
	BERT+	.54	.68
	RB	.57	.55
Ours	plain	**.97**	.67
	+abduction	**.97**	.91
	+abduction +rule	**.97**	**.92**

Table 9:
Accuracy on SICK dataset

SICK		
#All		4,927
Majority		.57
DL	RB	.56
Logic	LP	.81
	MG	**.83**
Ours	plain	.76
	+abduction	.82
	+abduction +rule	.82

bound on the accuracy of our system by reducing error propagation, we added hand-coded rules to assign correct POS tags and lemmas (23 words). We also added two rules to join multiword expressions to derive correct logical forms (*law lecturer* and *legal authority* in FraCaS-214, 215).

- For CAD, we also experimented with an implementation for SI, as described in Section 3.2. We use 23 rules in Tsurgeon scripts. The accuracy is shown in +*implicature*.

Comparison of existing NLI systems We compare our system with other logic-based systems and recent DL-based systems. For logic-based systems, we mainly compare three systems based on CCG parsers and theorem proving:

- MN (Mineshima *et al.* 2015) uses a CCG parser (C&C; Clark and Curran 2007) and implements a theorem prover for NLI based on HOL. This system uses Coq (Castéran and Bertot 2004), an

HANS			
Gold		*yes*	*unknown*
#All		15,000	15,000
Majority		.50	.50
DL	BF	.87	.61
	RB	**1.0**	.56
Symbolic	GKR4	.84	.59
DL & Symbolic	HNB	.84	.54
	HNX	.83	.25
Ours	plain	.98	**.83**
	+abduction	.98	**.83**
	+abduction +rule	.98	**.83**

Table 10:
Accuracy on HANS dataset

CAD		
#All		257
Majority		.43
DL	RB	.58
Ours	plain	81
	+abduction	.81
	+abduction +rule	.82
	+abduction +rule +implicature	**.92**

Table 11:
Accuracy on CAD dataset

interactive natural deduction theorem prover in a fully automated way.

- LP (Abzianidze 2015, 2016) is a system that uses two CCG parsers (C&C and EasyCCG) and implements a natural logic inference system based on semantic tableau. The system uses the theorem prover for HOL (Abzianidze 2015) based on *natural logic* (Lakoff 1970; van Benthem 1986).
- MG (Martínez-Gómez *et al.* 2017) is a system based on two CCG parsings (C&C and EasyCCG) with compositional event semantics and theorem proving, an updated version of MN.

Table 12 summarizes the characteristics of the logic-based systems, including ours.

For DL-based systems, we compare our system with the following.

Table 12: Existing logic-based NLI systems

System	Proof strategy	Logic	Prover	Abduction	Arithmetic
MN	natural deduction	HOL	Coq		
LP	tableau	Natural Logic/HOL	NLogPro	✓	
MG	natural deduction	FOL	Coq	✓	
Ours	resolution	Typed FOL	Vampire	✓	✓

- BERT shows the performance of a BERT model fine-tuned with MultiNLI, and BERT+ shows that of a BERT model with data augmentation for approximately 36,000 monotonicity inferences in addition to the MultiNLI training set. Both models were tested and reported in Yanaka *et al.* (2019).
- BF is a BiLSTM model trained on MultiNLI, which is a state-of-the-art model on HANS. The model was tested and reported in Yaghoobzadeh *et al.* (2019).
- RB shows that we use a state-of-the-art model RoBERTa (Liu *et al.* 2019) trained on MultiNLI (Williams *et al.* 2018) using the implementation provided in AllenNLP.[27] The accuracies in the table represent those we tested.

In addition, for HANS dataset (see Table 10) we refer to the accuracy of a hybrid system with a symbolic component and a DL component reported in Kalouli *et al.* (2020), where three systems, HNB, HNX, and GKR4 are distinguished.

- HNB uses the Graphical Knowledge Representation (GKR) context graphs (Kalouli and Crouch 2018) to determine whether a given inference is semantically complex or not; for a complex problem, it uses a symbolic component that makes use of multiple graphs to represent sentence information, while for a simple problem, it uses a BERT model for determining the entailment label.
- HNX is a system that uses an XLNet model as the DL-model.
- GKR4 is a system that only uses the symbolic component.

[27] https://github.com/allenai/allennlp

FraCaS 4.3.1

Table 7 shows the results on FraCaS. For comparison, we use the two logic-based systems (`MN` and `LP`) and the DL-based system (`RB`). Our system achieved very high accuracy and outperformed the DL-system by a large margin. Table 6 shows examples that were solved by our system but not by the DL-system. Our system successfully solved inferences such as FraCaS-209 that involve antonyms, which the DL-system found particularly difficult to solve. FraCaS-241 is a complex inference with numerical expressions and clausal comparatives. This problem is solved by our system but by neither of the other logic-based systems, nor by the DL-system.

One problem that our system was not yet able to solve is concerned with comparative ellipsis. The sentence *APCOM has a more important customer than ITEL* (FraCaS-244, 245) can have two interpretations (56*H*) or (57*H*).

(56) *P*: APCOM has a more important customer than ITEL.
H: APCOM has a more important customer than ITEL is. (FraCaS-244, gold label: *yes*)

(57) *P*: APCOM has a more important customer than ITEL.
H: APCOM has a more important customer than ITEL has. (FraCaS-245, gold label: *yes*)

Our system does not have a component to handle this type of comparative ellipsis and can only derive the interpretation in (56*H*), thus failing to provide the correct judgement for FraCaS-245.

MED 4.3.2

Table 8 shows the results on MED. Our system outperformed the DL-based systems. MED-176 and MED-485 in Table 6, which involve a downward quantifier (*few*) and a non-monotonic quantifier (*exactly 12*), respectively, are examples that our system correctly solved but the DL-models did not. For the problems containing lexical inferences in *gqlex*, our system achieved a high improvement in accuracy (67% to 91%) by implementing the abduction mechanism, showing that our system is compatible with lexical knowledge.

4.3.3 SICK

Table 9 shows the results on SICK. Our system outperformed the DL-based system (RB) and achieved comparable results with the logic-based systems (LP and MG). SICK-1357 in Table 6 is an example involving the lexical inference from *puppy* to *dog*. Our system correctly predicted the *yes* label for this problem, while the DL-based system (RB) predicted the *no* label. SICK-4789 in Table 6 contains negation *no*; our system can represent what information is negated by the scope of the negation in the logical form, but DL-based systems tend to answer *no* to such inferences.

One problem that was solved by MG but not by our system is the following.

(58) P: Someone is on a black and white motorcycle and is standing on the seat.
H: A motorcycle rider is standing up on the seat of a white motorcycle. (SICK-199, gold label: *unknown*)

In the case of MG, which implements *on-demand* abduction (an axiom is added during the process of constructing a natural deduction proof), the premise sentence does not generate any axioms, while in our system, the axiom $\forall x(\mathsf{black}(x) \rightarrow \neg\mathsf{white}(x))$ based on the antonym is added before the proof process, making the premise inconsistent with the same entity being white and not white at the same time. Thus, our system incorrectly predicts *yes* by the principle of explosion (i.e., any proposition can be derived from the contradiction).

Another type of error is found in the following problem.

(59) P: A man is holding a small animal in one hand.
H: A man is holding an animal, which is small, in one hand. (SICK-4690, gold label: *yes*)

The gradable adjective *small* in P is a nominal adjective, generating the threshold $\theta_{\mathsf{small}}(\mathsf{animal})$, while that in H is a predicate adjective, generating the threshold $\theta_{\mathsf{small}}(\mathsf{U})$ with the universal set U. Due to this mismatch in the comparison class, the system failed the proof.

Overall, our system achieved performance comparable to that of MG based on event semantics, thus showing the compatibility of event semantics and degree semantics.

HANS 4.3.4

Table 10 shows the results on HANS. We compared our system with the following systems: `BF`, `RB`, `GKR4`, `HNB`, and `HNX`.

McCoy *et al.* (2019) reported that DL-based systems tend to erroneously output *yes* for cases in which the hypothesis was a constituent or a substring of the premise, such as disjunctive sentences (e.g., HANS-23990 in Table 6). To see how a system performs in these cases, we present the accuracy for each gold answer label (*yes* and *unknown*). While accuracy whose gold label is *yes* was close to 100% in both our system and the DL-based system (`RB`), the accuracy of our system was higher than that of `RB` when the label is *unknown* (83% vs. 56%).

One reason for the relatively low accuracy (83%) of our system in comparison with its performance on the other datasets is parse error. HANS contains syntactically complex sentences such as *The author who advised the lawyer supported the athlete* (HANS-12182, *subsequence*), for which the CCG parsers output incorrect parses. For example, in the case of C&C parser, the substring of the sentence, *The author who advised*, is parsed as *NP*, separated from the object noun phrase *the lawyer*. The rest of the sentence, *the lawyer supported the athlete*, is parsed as *S* and shifted to $NP\backslash NP$. For depccg, the sentence *The athletes presented in the library* (HANS-13002) is parsed as *NP* instead of S.

Another type of error is concerned with an inference involving a modal adverb, e.g., the inference from *Probably the secretary admired the athlete* to *The secretary admired the athlete* (HANS-24034). The gold label is *unknown*, but our system predicts *yes* since any adverb can be dropped in the current implementation. A more fine-grained classification of adverbs will be needed to handle this type of inference.

CAD 4.3.5

Table 11 shows the results on CAD. Our system outperformed the DL-based system (`RB`). Our system was able to solve inference involving numerical computations (CAD-001,115) and antonym conversion for adverbs (CAD-157) shown in Table 6, while `RB` incorrectly predicted *unknown* for CAD-001, *no* for CAD-115, and *yes* for CAD-157.

Table 13 shows some example problems from CAD where the gold label changes between semantics and pragmatics. In the setting shown

Table 13: Examples of entailment problems for SI of gradable expressions from CAD

ID	Premises and hypothesis	Gold label	
		Semantics	Pragmatics
002	P_1: John is 5 cm shorter than Bob. P_2: Bob is 170 cm tall. H: John is 165 cm tall.	Unknown	Yes
052	P_1: Bob is much taller than John. P_2: Bob is a 5 feet tall boy. H: John is shorter than 5 feet.	Unknown	Yes
112	P: Bob saw four students. H: Bob saw three students.	Yes	No
145	P: Ann runs as fast as Luis. H: Ann runs faster than Luis.	Unknown	No
245	P: There are a few books. H: There are many books.	Unknown	No

in +*implicature*, our system was able to solve problems involving SIs, which led to the improvement in accuracy. Our system also solved complex inferences (CAD-002,052) that involve antonyms and numerical expressions.

There are still problems that need to be addressed. For example, the sentence *Jones drives more carefully today than yesterday* (CAD-183) conjoins two adverbs *today* and *yesterday* by *than*. The current system does not derive the correct logical form for this type of complex coordinate structure formed by *than*-clauses. Also, in the case of the sentence *Chris is more happy than Alex is sad* (CAD-013), which is an instance of COMPARATIVE SUBDELETION (Bresnan 1975), the clause *Alex is sad* is simply parsed as S and mapped to $\mathsf{sad}(\mathsf{alex}, \theta_{\mathsf{sad}})$, making it impossible to compare it the degrees introduced by the main clause. Further improvement to CCG parsing is needed to handle complex coordinate constructions and comparative subdeletion.

4.3.6 Comparison of CCG parsers

For a comprehensive comparison, Table 14 shows accuracies for each CCG parser at its best performances in our system. It shows that our system achieved the best accuracy with depccg in most datasets. One of the reasons for this is that the tree conversion is designed based on the outputs of depccg. It is also noted that as described in Section 4.1,

Table 14: Accuracy for each CCG parser at the best performances

Parser	FraCaS				MED		SICK	HANS		CAD
	GQ	*Adj*	*Com*	*Att*	*gq*	*gqlex*		*yes*	*unknown*	
Multi	.99	.95	.90	.92	.97	.92	.82	.98	.83	.92
C&C	.82	.86	.61	.69	.93	.88	.76	.80	.85	.52
EasyCCG	**.97**	.86	.55	**.92**	**.97**	.89	**.77**	.93	**.98**	.53
depccg	.96	**.95**	**.90**	**.92**	.96	**.91**	**.77**	**.97**	.95	**.92**

our system prioritizes *yes* (or *no*) rather than *unknown* among the answers given by the three parsers. For this reason, parse errors caused by C&C led to a decrease in overall accuracy in the case of *unknown* problems, as shown in Table 14. It would be necessary to refine the system's answer selection mechanism when multiple parsers are used.

General discussion 4.3.7

FraCaS and CAD are datasets manually constructed by experts; their size is small (FraCaS: 139 , CAD: 257) but contains linguistically challenging inferences. The evaluation of FraCaS and CAD shows that the proposed system can handle the various types of complex inferences discussed in formal semantics, including adjectives, comparatives, and generalized quantifiers.

MED, SICK, and HANS are crowdsourced or automatically generated datasets that are larger in size than FraCaS and CAD (MED: 1,189, SICK: 4,297, HANS: 30,000). The inferences in MED, SICK, and HANS are single-premise inferences, simpler than FraCaS and CAD but containing lexical inferences (MED, SICK) and logical phenomena such as quantification, disjunction, and negation (MED, SICK, HANS). The experimental results for MED, SICK, and HANS indicate that our system can successfully handle these types of inferences.

The ablation study aimed to estimate the effects of three additional mechanisms: (1) abduction (lexical inference) mechanism, (2) hand-written rules for error correction, and (3) mechanisms for handling implicature. The results of the ablation study for each dataset show that the system improved accuracy for the datasets that include lexical inference (indicated by +abduction in MED and SICK) and for the dataset containing implicature (indicated by +implicature in

CAD). These results were more or less expected, but still seem to be meaningful enough to show the effectiveness of the additional components.

5 CONCLUSION

We presented a CCG-based compositional semantics and inference system for comparatives and other related constructions. The logical forms used are based on A-not-A analysis in formal semantics and the inference system is combined with the axioms of COMP based on TFF forms acceptable in efficient FOL provers. The entire system is transparently composed of multiple modules and can solve complex inferences in an explanatory manner. The system can handle gradable expressions such as comparatives and adjectives, which are a weakness of conventional logic-based systems. The system can also be extended to handle generalized quantifiers, adverbs, and numerals while maintaining the advantages of the original system for adjectival comparatives. For adverbs in particular, by combining two semantic theories, degree semantics and event semantics, we were able to assign appropriate logical forms to solve complex inferences.

For evaluation, we used various NLI datasets containing linguistically challenging problems. The results showed that our system works well on complex logical inferences for which standard DL-based systems show poor performance. In addition, our system has the advantage that it does not require large amounts of training data, such as SNLI or MultiNLI, as opposed to DL-based systems.

It might be objected that the results on the DL models in Section 4.3 were not surprising, because these models were trained on SNLI and MultiNLI that do not target the logical and numerical inferences we are concerned with in this study. However, it is fair to say that it is challenging to generate effective training data for handling various complex inferences with comparatives, numerals, and generalized quantifiers. This study can also contribute to the study of computational modeling and to the evaluation of formal semantic theories, as well as to the creation of challenging NLI problems that DL-based models need to address.

In addition to the problems we have already mentioned, there are still some unresolved issues in this study. For example, we need to extend our analysis to cover more challenging comparative constructions such as GAPPING (Ross 1970; Hendriks 1995). It would also be interesting to modify CCGbank, which is the training data for CCG parsers, based on the proposed transformation of parsing trees. These are left for future work.

ACKNOWLEDGEMENT

We thank the three reviewers for their valuable comments. This work was supported by JSPS KAKENHI Grant Number JP21K00016, JSPS KAKENHI Grant Number JP18H03284, JST CREST Grant Number JP-MJCR2114, and JST CREST Grant Number JPMJCR20D2, Japan.

REFERENCES

Lasha ABZIANIDZE (2015), A tableau prover for natural logic and lganguage, in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2492–2502.

Lasha ABZIANIDZE (2016), Natural solution to FraCaS entailment problems, in *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics (*SEM)*, pp. 64–74.

Stefano BACCIANELLA, Andrea ESULI, and Fabrizio SEBASTIANI (2010), SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining, in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pp. 2000–2004.

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

Jean-Philippe BERNARDY and Stergios CHATZIKYRIAKIDIS (2017), A type-theoretical system for the FraCaS test suite: Grammatical Framework meets Coq, in *IWCS 2017 – 12th International Conference on Computational Semantics*.

Johan Bos (2008a), Let's not argue about Semantics, in Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, and Daniel Tapias, editors, *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pp. 2835–2840.

Johan Bos (2008b), Wide-coverage semantic analysis with Boxer, in *Proceedings of the 2008 Conference on Semantics in Text Processing (STEP)*, pp. 277–286.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning (2015), A large annotated corpus for learning natural language inference, in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 632–642.

Richard Breheny (2008), A new look at the semantics and pragmatics of numerically quantified noun phrases, *Journal of Semantics*, 25(2):93–139.

Joan W. Bresnan (1975), Comparative deletion and constraints on transformations, *Linguistic Analysis*, 1:25–74.

Greg Carlson (1981), Distribution of free-choice *Any*, in Masek Hendrick and Miller, editors, *Papers from the Seventeenth Regional Meeting of the Chicago Linguistics Society*, 17, pp. 8–23.

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 Jean-Philippe Bernardy (2019), A wide-coverage symbolic natural language inference system, in *Proceedings of the 22nd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pp. 298–303.

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

Gennaro Chierchia (2004), Scalar implicatures, polarity phenomena and the syntax/pragmatics interface, in Adriana Belletti, editor, *Structures and Beyond*, pp. 39–103, Oxford University Press.

Timothy Chklovski and Patrick Pantel (2004), VerbOcean: Mining the web for fine-grained semantic verb relations, in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 33–40.

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 G. Pulman, *et al.* (1996), FraCaS – A framework for computational semantics, *Deliverable D6*.

Max J. CRESSWELL (1976), The semantics of degree, in Barbara PARTEE, editor, *Montague Grammar*, pp. 261–292, Academic Press.

Donald DAVIDSON (1967), The logical form of action sentences, in Nicholas RESCHER, editor, *The Logic of Decision and Action*, pp. 81–95, University of Pittsburgh Press.

Gerald GAZDAR (1979), *Pragmatics: Implicature, Presupposition, and Logical Form*, Academic Press.

Martin HACKL (2000), *Comparative Quantifiers*, Ph.D. thesis, Massachusetts Institute of Technology.

Michael HAHN and Frank RICHTER (2016), Henkin semantics for reasoning with natural language, *Journal of Language Modelling*, 3(2):513–568.

Izumi HARUTA, Koji MINESHIMA, and Daisuke BEKKI (2020), Combining event semantics and degree semantics for natural language inference, in *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, pp. 1758–1764.

Herman HENDRIKS (1993), *Studied Flexibility: Categories and Types in Syntax and Semantics*, Ph.D. thesis, ILLC, University of Amsterdam.

Petra HENDRIKS (1995), *Comparatives and Categorial Grammar*, Ph.D. thesis, University of Groningen.

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)*, pp. 207–215.

Laurence Robert HORN (1973), *On the Semantic Properties of Logical Operators in English*, Ph.D. thesis, University of California.

Nirit KADMON and Fred LANDMAN (1993), Any, *Linguistics and Philosophy*, 16(4):353–422.

Aikaterini-Lida KALOULI and Richard CROUCH (2018), GKR: the graphical knowledge representation for semantic parsing, in *Proceedings of the Workshop on Computational Semantics beyond Events and Roles (SemBEaR)*, pp. 27–37.

Aikaterini-Lida KALOULI, Richard CROUCH, and Valeria DE PAIVA (2020), Hy-NLI: a hybrid system for natural language inference, in *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, pp. 5235–5249.

Hans KAMP (1975), Two theories about adjectives, in Edward L KEENAN, editor, *Formal Semantics of Natural Language*, pp. 123–155, Cambridge University Press.

Ewan KLEIN (1980), A semantics for positive and comparative adjectives, *Linguistics and Philosophy*, 4(1):1–45.

Ewan KLEIN (1982), The interpretation of adjectival comparatives, *Journal of Linguistics*, 18(1):113–136.

Ewan KLEIN (1991), Comparatives, in Arnim VON STECHOW and Dieter WUNDERLICH, editors, *Semantics: An International Handbook of Contemporary Research*, pp. 673–691, de Gruyter.

Laura KOVÁCS and Andrei VORONKOV (2013), First-order theorem proving and Vampire, in *Proceedings of the 25th International Conference on Computer Aided Verification*, volume 8044, pp. 1–35.

William A. LADUSAW (1979), *Polarity Sensitivity as Inherent Scope Relations*, Ph.D. thesis, University of Texas.

George LAKOFF (1970), Linguistics and natural logic, *Synthese*, 22(1-2):151–271.

Richard K. LARSON (1988), Scope and comparatives, *Linguistics and Philosophy*, 11(1):1–26.

Peter N. LASERSOHN (2006), Event-based semantics, in Keith BROWN, editor, *Encyclopedia of Language and Linguistics*, volume 4, pp. 316–320.

Daniel LASSITER (2015), Adjectival modification and gradation, in *The Handbook of Contemporary Semantic Theory*, pp. 141–167, John Wiley & Sons, Ltd.

Adrienne LEHRER and Keith LEHRER (1982), Antonymy, *Linguistics and Philosophy*, 5(4):483–501.

Roger LEVY and Galen ANDREW (2006), Tregex and Tsurgeon: tools for querying and manipulating tree data structures, in *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, pp. 2231–2234.

David LEWIS (1972), General semantics, in Donald DAVIDSON and Gilbert HARMAN, editors, *Semantics of Natural Language*, pp. 169–218, Springer.

Mike LEWIS and Mark STEEDMAN (2014), A* CCG parsing with a supertag-factored model, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 990–1000.

Godehard LINK (1983), The logical analysis of plurals and mass terms: A lattice-theoretic approach, in Paul PORTNER and Barbara H. PARTEE, editors, *Formal Semantics – the Essential Readings*, pp. 127–147, Blackwell.

Yinhan LIU, Myle OTT, Naman GOYAL, Jingfei DU, Mandar JOSHI, Danqi CHEN, Omer LEVY, Mike LEWIS, Luke ZETTLEMOYER, and Veselin STOYANOV (2019), RoBERTa: A robustly optimized BERT pretraining approach, *arXiv preprint arXiv:1907.11692*.

Marco MARELLI, Stefano MENINI, Marco BARONI, Luisa BENTIVOGLI, Raffaella BERNARDI, and Roberto ZAMPARELLI (2014), A SICK cure for the evaluation of compositional distributional semantic models, in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pp. 216–223.

Pascual MARTÍNEZ-GÓMEZ, Koji MINESHIMA, Yusuke MIYAO, and Daisuke BEKKI (2016), ccg2lambda: A compositional semantics system, in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL), System Demonstrations*, pp. 85–90.

Pascual MARTÍNEZ-GÓMEZ, Koji MINESHIMA, Yusuke MIYAO, and Daisuke BEKKI (2017), On-demand injection of lexical knowledge for recognising textual entailment, in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 710–720.

Tom MCCOY, Ellie PAVLICK, and Tal LINZEN (2019), Right for the wrong Reasons: diagnosing syntactic heuristics in natural language inference, in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 3428–3448.

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

Koji MINESHIMA, Pascual MARTÍNEZ-GÓMEZ, Yusuke MIYAO, and Daisuke BEKKI (2015), Higher-order logical inference with compositional semantics, in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2055–2061.

Richard MONTAGUE (1970), Universal grammar, *Theoria*, 36(3):373–398.

Glyn MORRILL and Oriol VALENTÍN (2016), Computational coverage of type logical grammar: The Montague test, *Empirical Issues in Syntax and Semantics*, 11:1–30.

Marcin MORZYCKI (2016), *Modification*, Cambridge University Press.

Katsuma NARISAWA, Yotaro WATANABE, Junta MIZUNO, Naoaki OKAZAKI, and Kentaro INUI (2013), Is a 204 cm man tall or small? Acquisition of numerical common sense from the web, in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 382–391.

Terence PARSONS (1990), *Events in the Semantics of English*, MIT Press.

Barbara H. PARTEE (1988), Many quantifiers, in *Proceedings of the 5th Eastern States Conference on Linguistics (ESCOL)*, pp. 383–402.

Barbara H. PARTEE (2007), Compositionality and coercion in semantics: The dynamics of adjective meaning, in Gerlof BOUMA *et al.*, editors, *Cognitive Foundations of Interpretation*, pp. 145–161.

Sandro PEZZELLE and Raquel FERNÁNDEZ (2019), Is the red square big? MALeViC: Modeling adjectives leveraging visual contexts, in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2865–2876.

Stephen G. PULMAN (2007), Formal and computational semantics: A case study, in Jeroen GEERTZEN, Elias THIJSSE, Harry BUNT, and Amanda SCHIFFRIN, editors, *Proceedings of the Seventh International Workshop on Computational Semantics: IWCS-7, Tilburg, The Netherlands, 2007*, pp. 181–196.

Stephen G. PULMAN (2018), Second order inference in natural language semantics, *Journal of Language Modelling*, 6(1):1–40.

Jessica RETT (2018), The semantics of *many*, *much*, *few*, and *little*, *Language and Linguistics Compass*, 12(1):e12269.

John Robert ROSS (1970), Gapping and the order of constituents, in Manfred BIERWISCH and Karl E. HEIDOLPH, editors, *Progress in Linguistics*, pp. 249–259, De Gruyter Mouton.

Roger SCHWARZSCHILD (2008), The semantics of comparatives and other degree constructions, *Language and Linguistics Compass*, 2(2):308–331.

Pieter A. M. SEUREN (1973), The comparative, in Ferenc KIEFER and Nicolas RUWET, editors, *Generative Grammar in Europe*, pp. 528–564, Riedel.

Benjamin SPECTOR (2013), Bare numerals and scalar implicatures, *Language and Linguistics Compass*, 7(5):273–294.

Mark STEEDMAN (1996), *Surface Structure and Interpretation*, MIT Press.

Mark STEEDMAN (2000), *The Syntactic Process*, MIT Press.

Geoff SUTCLIFFE (2017), The TPTP problem library and associated infrastructure, *Journal of Automated Reasoning*, 59(4):483–502.

Geoff SUTCLIFFE, Stephan SCHULZ, Koen CLAESSEN, and Peter BAUMGARTNER (2012), The TPTP typed first-order form with arithmetic, in Nikolaj BJØRNER and Andrei VORONKOV, editors, *Logic for Programming, Artificial Intelligence, and Reasoning*, pp. 406–419, Springer.

Anna SZABOLCSI (2010), *Quantification*, Cambridge University Press.

Johan VAN BENTHEM (1986), *Essays in Logical Semantics*, Springer.

Robert VAN ROOIJ and Katrin SCHULZ (2004), Exhaustive interpretation of complex sentences, *Journal of Logic, Language and Information*, 13(4):491–519.

Dag WESTERSTAÅHL (2007), Quantifiers in formal and natural languages, in Dov M. GABBAY and Franz GUENTHNER, editors, *Handbook of Philosophical Logic*, volume 14, pp. 223–338, Springer.

Adina WILLIAMS, Nikita NANGIA, and Samuel BOWMAN (2018), A broad-coverage challenge corpus for sentence understanding through inference, in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 1112–1122.

Yadollah YAGHOOBZADEH, Remi TACHET, Timothy J. HAZEN, and Alessandro SORDONI (2019), Robust natural language inference models with example forgetting, *arXiv preprint arXiv:1911.03861*.

Hitomi YANAKA, Koji MINESHIMA, Daisuke BEKKI, Kentaro INUI, Satoshi SEKINE, Lasha ABZIANIDZE, and Johan BOS (2019), Can neural networks understand monotonicity reasoning?, in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 31–40.

Masashi YOSHIKAWA, Hiroshi NOJI, and Yuji MATSUMOTO (2017), A* CCG parsing with a supertag and dependency factored model, in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 277–287.

Izumi Haruta

Ochanomizu University, Japan

Daisuke Bekki

0000-0002-9988-1260

bekki@is.ocha.ac.jp

Ochanomizu University, Japan

Koji Mineshima

0000-0002-2801-9171

minesima@abelard.flet.keio.ac.jp

Keio University, Japan

Izumi Haruta, Koji Mineshima, and Daisuke Bekki (2022), *Implementing Natural Language Inference for comparatives*, Journal of Language Modelling, 10(1):139–191

https://dx.doi.org/10.15398/jlm.v10i1.294