Exploring Transitivity in Neural NLI Models through Veridicality

Hitomi Yanaka[1], **Koji Mineshima**[2], **and Kentaro Inui**[3,1]
[1]RIKEN, [2]Keio University, [3]Tohoku University
hitomi.yanaka@riken.jp, minesima@abelard.flet.keio.ac.jp,
inui@ecei.tohoku.ac.jp

Abstract

Despite the recent success of deep neural networks in natural language processing, the extent to which they can demonstrate human-like generalization capacities for natural language understanding remains unclear. We explore this issue in the domain of natural language inference (NLI), focusing on the *transitivity* of inference relations, a fundamental property for systematically drawing inferences. A model capturing transitivity can compose basic inference patterns and draw new inferences. We introduce an analysis method using synthetic and naturalistic NLI datasets involving clause-embedding verbs to evaluate whether models can perform transitivity inferences composed of veridical inferences and arbitrary inference types. We find that current NLI models do not perform consistently well on transitivity inference tasks, suggesting that they lack the generalization capacity for drawing composite inferences from provided training examples. The data and code for our analysis are publicly available at https://github.com/verypluming/transitivity.

1 Introduction

Deep neural networks (DNNs) have shown impressive performance in many natural language processing tasks. In particular, DNN models pre-trained with large-scale data such as BERT (Devlin et al., 2019) have achieved high accuracy in various benchmark tasks (Wang et al., 2019a,b), which suggests that they might possess some generalization capacities that are a hallmark of human cognition. However, recent analyses (Talmor and Berant, 2019; Liu et al., 2019; McCoy et al., 2019) have shown that high accuracy on a test set drawn from the same distribution as the training set does not always indicate that the model has obtained the intended ability, so it remains unclear to what extent DNN models can learn the systematic generalization in natural language from training instances.

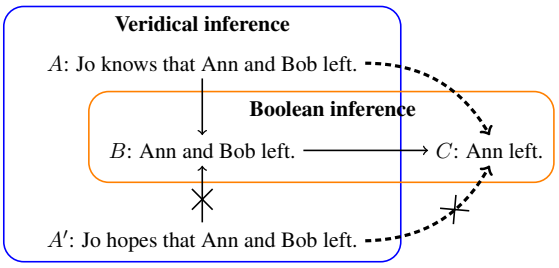

Figure 1: Illustration of transitivity inferences (indicated by ⇢) composed of two basic inferences, veridical and Boolean. Arrows indicate *entailment* and arrows with a cross (×) indicate *non-entailment*.

Central to human-like generalization capacities is the fact that ability to understand a given sentence is related to ability to understand other sentences, called *systematicity* of human cognition in Fodor and Pylyshyn (1988). Thus, if speakers understand the meaning of the sentence *Ann loves Bob*, they must also understand the meaning of structurally related sentences such as *Bob loves Ann*. We explore whether DNN models possess this type of generalization capacity in the domain of natural language inference (NLI), which is the task to judge whether a premise entails a hypothesis (Dagan et al., 2013; Bowman et al., 2015a).

A key property underlying systematicity of drawing inferences is the *transitivity* of inference relations, illustrated in Figure 1. Schematically, if a model learns a basic inference pattern from A to B and one from B to C, it should be able to compose the two patterns to draw a new inference from A to C. If a model lacks this generalization capacity, it must memorize an exponential number of inference combinations independently of basic patterns.

Among the various inference patterns, we focus on transitivity inferences that combine *veridical* in-

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 920–934
April 19 - 23, 2021.

ferences with other types. In veridical inferences, one must distinguish two entailment types. For example, the verb *know* is called **veridical** in that "x *knows that* P" entails that P is true, while the verb *hope* is called **non-veridical** since "x *hopes that* P" does not entail that P is true. Veridical inferences can relatively easily compose transitivity inferences at scale by embedding various inference types into clause-embedding verbs. For instance, as Figure 1 shows, if a model has the ability to perform both Boolean inference and veridical inference, it is desirable to have the ability to combine both types to make a chained inference.

Such transitivity inferences are by no means trivial. For instance, if the premise is changed to *Jo knows that Ann* ***or*** *Bob left*, it does not follow that *Bob left*, even though the veridical verb *know* appears. Models relying on shallow heuristics such as lexical overlap can wrongly predict *entailment* in this case. To correctly handle such composite inferences, models must capture structural relations between veridical inferences and various kinds of embedded inference.

Previous studies on the generalization capacities of NLI models have addressed how models could learn inferences with various challenging linguistic phenomena (Bowman et al., 2015b; Dasgupta et al., 2018; Geiger et al., 2019, 2020; Yanaka et al., 2019a,b; Richardson et al., 2020). However, these studies have focused on the linguistic phenomena in isolation, and thus do not address how a model could learn the *interactions* between them. Our aim is to fill this gap by presenting a method for probing generalization capacity of DNN models performing transitivity inferences.

This study provides three main contributions. First, we create and publicly release two types of NLI datasets for testing model ability to perform transitivity inferences: a fully synthetic dataset that combines veridical inferences and Boolean inferences, and a naturalistic dataset that combines veridical inferences with lexical and structural inferences. Second, we use these datasets to systematically expose models to basic inference patterns and test them on a variety of combinations. This will demonstrate that the models lack the ability to capture transitivity of inference. Third, we investigate whether data augmentation with new combination patterns helps models to learn transitivity. Experiments show that the data augmentation improves model performance on similar combinations, regardless of the existence of basic inference patterns in the training set. These results suggest there is much room for improving the generalization capacities of DNN models for combining basic inferential abilities.

2 Related Work

Transitivity The transitivity of entailment relations, which derives $A \rightarrow C$ from $A \rightarrow B$ and $B \rightarrow C$, is incorporated into logic-based NLI systems using automated theorem proving (Abzianidze, 2015; Mineshima et al., 2015). This is a basic property of formal logic, also known as *syllogism* in traditional logic or the *cut rule* in proof theory (Troelstra and Schwichtenberg, 2000; van Dalen, 2013). Transitivity inference in its various forms has also been widely studied as a fundamental property of human reasoning in cognitive psychology (Johnson-Laird and Byrne, 1991; Khemlani and Johnson-Laird, 2012). In the context of NLP, previous works have proposed a method for training models with transitivity constraints in multi-hop reasoning tasks (Asai and Hajishirzi, 2020) and temporal relation extraction tasks (Ning et al., 2017). Clark et al. (2020) investigated a transformer's ability to perform a chain of reasoning where reasoning rules are explicitly given. In this work, we study model ability to learn transitivity of entailment relations from training examples, rather than explicitly providing rules.

Systematicity There has been extensive discussion of whether neural networks (aka Connectionist models) can exhibit systematicity of cognitive capacities (Fodor and Pylyshyn, 1988; Marcus, 2003). Recent works have explored whether modern neural networks can learn systematicity in semantic parsing tasks (Lake and Baroni, 2017; Baroni, 2020; Kim and Linzen, 2020) and question answering tasks (Sinha et al., 2019), whereas our focus is the systematicity in NLI.

In works related to the systematicity in NLI, Goodwin et al. (2020), Yanaka et al. (2020), and Geiger et al. (2020) used a manually constructed NLI dataset of monotonicity inferences with and without negation (e.g., *The child is not holding plants* $\rightarrow$ *The child is not holding flowers*) to examine DNN models' generalization capacities. While these approaches concentrate on monotonicity inferences involving quantifiers and negative expressions, our method using veridical inference is general in that it can be applied to any entailment

relation that combines basic inference patterns; we generate composite inferences by embedding various types of sentences into clause-embedding verbs.

Fodor and Pylyshyn (1988) distinguished systematicity (roughly, the ability to understand sentences that are structurally related to each other) from productivity (the ability to understand an infinite set of sentences), claiming that systematicity poses a serious challenge to neural network models. Yanaka et al. (2020) tested both systematicity and productivity of DNN models with a synthetic dataset of monotonicity inferences for upward (e.g., *some, at least three*) and downward (e.g., *few, at most three*) quantifiers, where handling productivity (recursion) makes sentences more involved (e.g., iterated relative clauses and negation). Focusing on systematicity rather than productivity allows testing models with more natural and less complicated data, as compared to sentences appearing in monotonicity inferences.

Veridicality Veridical inferences, including those licensed by factive and implicative verbs, have been intensively studied in the literature of semantics and pragmatics (Karttunen and Peters, 1979; Beaver, 2001). Recent work has revealed graded and context-sensitive aspects of veridicality inferences, creating veridicality judgement datasets (de Marneffe et al., 2012; White and Rawlins, 2018; White et al., 2018). While we use only a subset of veridical predicates discussed in the literature, our method can be extended to more complex inferences, such as factive presupposition.

Ross and Pavlick (2019) presented a naturalistic veridicality dataset and compared the predictions of a BERT-based NLI model and human judgements. These previous studies on veridicality inferences have tended to focus on relations between whole sentences (e.g., *Jo remembered that there was a wild deer jumping a fence*) and its embedded material (e.g., *There was a wild deer jumping a fence*). By contrast, we consider the interactions of veridicality inferences and other inference types (see Section 3.2), including cases where the embedded material is further paraphrased via linguistic phenomena (e.g., *Jo remembered that there was a wild deer jumping a fence* ⇒ *An animal was jumping*). We also collect human judgements on our dataset and compare them with model predictions (see Section 4.4).

Probing NLI models Many studies of probing NLI models have found that current models often fail on linguistically challenging (adversarial) inferences (Rozen et al., 2019; Nie et al., 2019; Yanaka et al., 2019a; Richardson et al., 2020), learning undesired biases (Glockner et al., 2018; Poliak et al., 2018; Tsuchiya, 2018; Liu et al., 2019), and heuristics (McCoy et al., 2019). Our approach also provides adversarial test sets against such heuristics by considering combinations of veridical inferences and diverse (lexical, structural, and logical) types of inferences.

One way to learn challenging inferences is data augmentation, and prior studies (Yanaka et al., 2019b; Richardson et al., 2020; Min et al., 2020) have shown that data augmentation with synthesized datasets improves performance with challenging linguistic phenomena. However, it remains unclear whether data augmentation can help models learn *composite* inferences mixing several inference types from training instances. We address this question in Section 4.3.

3 Dataset

3.1 Overview

To investigate whether models can capture transitivity, we consider two basic inference patterns and their combinations. The first basic pattern, $\mathcal{I}_1$, is veridical inference. We write $f(s_1) \rightarrow s_1$ to denote a schematic veridical inference, where f is a clause-embedding verb and s_1 is the embedded clause. For instance, in the case of the inference pattern $A \rightarrow B$ in Figure 1, "*Jo knows that x*" corresponds to $f(x)$ and "*Ann and Bob left*" to s_1.

The second basic pattern, $\mathcal{I}_2$, provides an inference from the embedded material. We denote a premise-hypothesis pair of this second inference by $s_1 \rightarrow s_2$. Given two inferences $f(s_1) \rightarrow s_1$ in $\mathcal{I}_1$ and $s_1 \rightarrow s_2$ in $\mathcal{I}_2$, we consider a new inference $f(s_1) \rightarrow s_2$, where premise $f(s_1)$ is the same as that of $\mathcal{I}_1$ and hypothesis s_2 is the same as that of $\mathcal{I}_2$. See Table 1 and Table 2 for some examples of inferences $f(s_1) \rightarrow s_1$, $s_1 \rightarrow s_2$, and $f(s_1) \rightarrow s_2$. In this work, we consider binary labels, *entailment* and *non-entailment*, denoted by *yes* and *unk*, respectively. As Table 3 shows, the gold label on the $f(s_1) \rightarrow s_2$ pattern can be determined from those of the basic patterns $f(s_1) \rightarrow s_1$ and $s_1 \rightarrow s_2$, following the transitivity of entailment relations.

We train models with the first and second patterns, $f(s_1) \rightarrow s_1$ and $s_1 \rightarrow s_2$, and then test them

f	$f(s_1) \rightarrow s_1$	$s_1 \rightarrow s_2$	$f(s_1) \rightarrow s_2$	**Example**
V	*yes*	*yes*	*yes*	$f(s_1)$: Someone **noticed** that [Henry and Daniel found Elliot, John and Fred]. s_1: Henry and Daniel found Elliot, John and Fred. s_2: Henry found John.
NV	*unk*	*yes*	*unk*	$f(s_1)$: Someone **expects** that [Tom and Ann admire Greg and Fred]. s_1: Tom and Ann admire Greg and Fred. s_2: Tom admires Greg.
NV	*unk*	*unk*	*unk*	$f(s_1)$: Someone **argued** that [it was not the case that Greg hated John or Elliot]. s_1: It was not the case that Greg hated John or Elliot. s_2: Greg hated John.

Table 1: Examples from our fully synthetic transitivity inference datasets. V and NV indicate types of clause-embedding verbs (veridical/non-veridical); *yes* means *entailment* and *unk* means *non-entailment*.

ID	f	$f(s_1) \rightarrow s_1$	$s_1 \rightarrow s_2$	$f(s_1) \rightarrow s_2$	**Example**
2299	V	*yes*	*yes*	*yes*	$f(s_1)$: Someone **realized** that [a boy was playing a guitar]. s_1: A boy was playing a guitar. s_2: A kid was playing a guitar.
2049	V	*yes*	*unk*	*unk*	$f(s_1)$: Someone **remembered** that [a cat was playing with a device]. s_1: A cat was playing with a device. s_2: The boy was enthusiastically playing in the mud.
5024	NV	*unk*	*yes*	*unk*	$f(s_1)$: Someone **doubts** that [the woman is putting makeup on the man]. s_1: The woman is putting makeup on the man. s_2: A man's face is being painted by a woman.

Table 2: Examples from our naturalistic transitivity inference datasets. V and NV indicate types of clause-embedding verbs (veridical/non-veridical); *yes* means *entailment* and *unk* means *non-entailment*. **ID** indicates the original ID of $s_1 \rightarrow s_2$ in the SICK dataset.

$f(s_1) \rightarrow s_1$	$s_1 \rightarrow s_2$	$f(s_1) \rightarrow s_2$
yes	*yes*	***yes***
yes	*unk*	***unk***
unk	*yes*	***unk***
unk	*unk*	***unk***

Table 3: Rule for determining the $f(s_1) \rightarrow s_2$ label from the basic patterns $f(s_1) \rightarrow s_1$ and $s_1 \rightarrow s_2$.

Type of f	**Verbs**
Veridical	realize, acknowledge, remember, note, find, notice, learn, see, reveal, discover, understand, know, admit, recognize, observe
Non-veridical	feel, claim, doubt, hope, predict, imply, suspect, wish, think, believe, hear, expect, estimate, assume, argue

Table 4: Clause-embedding verbs used for our dataset.

on a set of the composite inferences $f(s_1) \rightarrow s_2$ that combines them. Note that due to how they are constructed, the training and test sets do not overlap. Model capable of applying the transitivity inference from $f(s_1) \rightarrow s_1$ and $s_1 \rightarrow s_2$ to $f(s_1) \rightarrow s_2$ should consistently predict the correct label of $f(s_1) \rightarrow s_2$ for any combination of $f(s_1) \rightarrow s_1$ and $s_1 \rightarrow s_2$.

3.2 Data creation

We generate basic inferences $f(s_1) \rightarrow s_1$ and $s_1 \rightarrow s_2$ and combine them to produce transitivity inferences $f(s_1) \rightarrow s_2$. To test diverse inference patterns, we consider two types of the second basic inference $s_1 \rightarrow s_2$: synthesized Boolean inferences and naturalistic inferences using an existing NLI dataset, SICK (Marelli et al., 2014), which contains lexical inferences (e.g., *boy* $\rightarrow$ *kid* in ID 2299 in Table 2) and structural inferences (e.g., active-passive alternation in ID 5024 in Table 2). Since the ratio of the gold labels (*yes* and *unk*) is set to 1 : 1 in both basic inference sets, the ratio of the gold labels for the transitivity test set is 1 : 3 by the rule in Table 3. We reserve 10% of the basic inference set for the validation set.

Clause-embedding verbs We focus on clause-embedding verbs that take tensed subordinate clauses. Specifically, we collect 67 verbs appearing in both MegaVeridicality2 (White et al., 2018) and the verb veridicality dataset (Ross and Pavlick, 2019). As Table 4 shows, we select a final set of 30 clause-embedding verbs.

Following a previous study (White et al., 2018),

we slot a clause-embedding verb f into a template with the form "Someone f that s_1" and generate premise $f(s_1)$ of veridical inference to avoid confounds introduced by world knowledge and pragmatic inference in the main clause. The clause-embedding verb f is in past or present tense, and we inflect the verb in the complement s_1 to match the tense of f.

When measuring the extent to which models can learn transitivity of entailment relations from training instances, it is desirable to determine the gold labels of composite inferences from those of basic inferences. Thus, we take the labels of veridical inference datasets predicted by the veridical and non-veridical distinction in lexical semantics as the gold standard. In addition, veridical inferences are sensitive to context, influenced by world knowledge and pragmatic factors (de Marneffe et al., 2012). Accordingly, we also present additional experiments to take into account such complexity of veridical inferences in Section 4.2.

Boolean inference To provide a fully synthetic transitivity inference dataset, we generate Boolean inferences with conjunction, disjunction, and negation. The data generation process is similar to the one in Yanaka et al. (2020): sentences are generated using a context-free grammar (CFG) associated with semantic composition rules in lambda-calculus. We first generate a set of premise sentences by the CFG rules and translate each sentence s_1 into a first-order-logic (FOL) formula F_1 in accordance with semantic composition rules specified in the CFG rules. Appendix A provides a set of CFG rules and semantic composition rules. We randomly select one of the atomic subformulas appearing in F_1 and take its positive or negative form, which we denote by F_2. Then we convert F_2 to a sentence s_2 using the same grammar. We set s_2 as a hypothesis.

The gold label for inference pair $s_1 \rightarrow s_2$ is determined by checking whether formula F_1 entails formula F_2 using an FOL theorem prover. The gold labels for $f(s_1) \rightarrow s_1$ and $f(s_1) \rightarrow s_2$ pairs are automatically determined according to the veridicality of a clause-embedding verb and the rule in Table 3, respectively. To restrict the complexity of generated sentences, we set the maximum number of logical connectives appearing in formula F_1 to 6.

Table 1 illustrates examples of fully synthetic transitivity inference datasets. We generate 3,000 Boolean inference examples $s_1 \rightarrow s_2$, 6,000 veridical inference examples $f(s_1) \rightarrow s_1$, and 6,000 composite inference examples $f(s_1) \rightarrow s_2$.

Naturalistic inference To generate a naturalistic transitivity inference dataset, we collect an example $s_1 \rightarrow s_2$ of naturalistic inference from the SICK dataset, which is constructed from existing sentences (image descriptions given by different people) and covers various lexical and structural phenomena. (1) is an example of lexical inference (*brush* $\rightarrow$ *comb*) in SICK, whose label is *yes*.

(1) s_1: A person is brushing a cat.
s_2: A person is combing the fur of a cat.

By selecting a clause-embedding verb f and an embedded sentence s_1, we generate a new sentence $f(s_1)$. As shown in (2), we construct a veridical inference example $f(s_1) \rightarrow s_1$ by setting $f(s_1)$ as a premise and s_1 as a hypothesis.

(2) $f(s_1)$: Someone **sees** that a person is brushing a cat.
s_1: A person is brushing a cat. (*yes*)

Likewise, as in (3), we can obtain a composite inference example $f(s_1) \rightarrow s_2$ whose label is *yes*:

(3) $f(s_1)$: Someone **sees** that a person is brushing a cat.
s_2: A person is combing the fur of a cat.

Table 2 illustrates examples of naturalistic transitivity inference datasets. We sample 1,000 naturalistic inference examples $s_1 \rightarrow s_2$ from the SICK training set and obtain 30,000 veridical inference examples $f(s_1) \rightarrow s_1$ and 30,000 composite inference examples $f(s_1) \rightarrow s_2$.

4 Experiments and Analysis

We analyze whether models trained with the basic inference set can consistently perform composite inferences on the test set. We use two DNN models, BERT and LSTM, which are known to perform well with linguistic phenomena such as subject-verb agreement and hierarchical and structural probing tasks (Linzen et al., 2016; Weiss et al., 2018; Kuncoro et al., 2018).

4.1 Experimental setup

In all experiments, we train each model for 25 epochs or until convergence and select the best-performing model based on its accuracy on the validation set. We perform five runs and report the average and standard deviation of their accuracies.

Data			Model					
$f(s_1) \to s_1$	$s_1 \to s_2$	$f(s_1) \to s_2$	**LSTM-M**	**LSTM-B**	**LSTM-M&B**	**BERT-M**	**BERT-B**	**BERT-M&B**
yes	*yes*	***yes***	74.2 ± 2.0	89.0 ± 9.1	87.9 ± 3.7	66.3 ± 3.4	100.0 ± 0.0	100.0 ± 0.0
yes	*unk*	***unk***	16.0 ± 4.3	6.3 ± 12.8	60.0 ± 10.2	4.9 ± 1.5	0.4 ± 0.7	60.5 ± 0.6
unk	*yes*	***unk***	14.7 ± 3.8	93.4 ± 8.3	89.0 ± 9.5	12.6 ± 4.8	99.4 ± 9.0	92.9 ± 3.6
unk	*unk*	***unk***	17.8 ± 5.5	92.1 ± 7.2	99.7 ± 0.5	13.2 ± 3.4	99.5 ± 0.5	99.9 ± 0.0
Test Overall			30.9 ± 3.2	70.2 ± 3.4	84.2 ± 1.2	24.4 ± 1.6	75.7 ± 0.4	88.3 ± 0.9
Validation ($f(s_1) \to s_1$)			50.5 ± 1.7	93.3 ± 11.1	91.4 ± 5.7	68.1 ± 1.3	99.2 ± 0.2	98.3 ± 0.3
Validation ($s_1 \to s_2$)			41.5 ± 3.4	89.2 ± 3.4	85.2 ± 1.2	54.4 ± 2.3	100.0 ± 0.0	99.4 ± 0.5

Table 5: Accuracies for the fully synthetic transitivity test set and the validation set. **-B** indicates a model trained with the basic inference set, **-M** indicates a model trained with MNLI, and **-M&B** indicates a model trained with MNLI mixed with the basic inference set. The label *yes* means *entailment*, and *unk* means *non-entailment*.

Data			Model					
$f(s_1) \to s_1$	$s_1 \to s_2$	$f(s_1) \to s_2$	**LSTM-M**	**LSTM-B**	**LSTM-M&B**	**BERT-M**	**BERT-B**	**BERT-M&B**
yes	*yes*	***yes***	64.6 ± 12.1	97.1 ± 2.7	100.0 ± 0.1	85.9 ± 1.1	100.0 ± 0.0	100.0 ± 0.0
yes	*unk*	***unk***	45.6 ± 10.5	0.0 ± 0.0	3.6 ± 1.4	28.4 ± 0.9	8.9 ± 7.8	22.3 ± 13.6
unk	*yes*	***unk***	24.4 ± 12.1	97.1 ± 2.7	99.7 ± 0.5	13.3 ± 1.7	100.0 ± 0.0	100.0 ± 0.0
unk	*unk*	***unk***	45.4 ± 11.2	97.3 ± 2.6	99.9 ± 0.1	31.1 ± 0.9	100.0 ± 0.0	100.0 ± 0.0
Test Overall			45.0 ± 5.5	72.9 ± 2.0	75.8 ± 0.5	39.7 ± 0.2	77.2 ± 2.0	80.6 ± 3.4
Validation ($f(s_1) \to s_1$)			46.2 ± 1.2	82.1 ± 3.3	89.8 ± 6.5	68.7 ± 1.6	99.2 ± 0.0	97.1 ± 0.3
Validation ($s_1 \to s_2$)			58.0 ± 1.0	81.9 ± 3.0	82.1 ± 1.4	62.0 ± 1.0	89.1 ± 2.0	91.0 ± 0.0

Table 6: Accuracies for the naturalistic transitivity test set and the validation set.

LSTM We use an LSTM (Hochreiter and Schmidhuber, 1997) model, where each premise and hypothesis is processed as a sequence of words using RNN with LSTM cells, and the final hidden state of each serves as its representation. The model concatenates the premise and hypothesis representations and passes the result to three hidden layers followed by a two-way softmax classifier. The model is initialized with 300-dimensional GloVe vectors (Pennington et al., 2014) and optimized using Adam (Kingma and Ba, 2015). We search dropout probabilities of $[0, 0.1, 0.2]$ on the output.

BERT We use the base-uncased pretrained BERT (Devlin et al., 2019) model[1], fine-tuned for the NLI classification task on training data in the standard way. When fine-tuning BERT, we search dropout probabilities of $[0, 0.1, 0.2]$ on the output, and hyperparameters are the same as those commonly used for MultiNLI.

4.2 Testing transitivity

We first evaluate whether the models trained with basic inferences $f(s_1) \to s_1$ and $s_1 \to s_2$ can consistently make judgements on the composite inferences $f(s_1) \to s_2$. As a previous work (Ross and Pavlick, 2019) reported that a BERT model trained with the benchmark NLI dataset MultiNLI (MNLI; Williams et al., 2018) is sensitive to verb veridicality, we regard the accuracy of models trained with MNLI as a baseline. We also analyze models trained with MNLI mixed with the basic inference set.

Table 5 shows accuracies for the fully synthetic transitivity test set that combines veridical and Boolean inferences. Models trained with the basic inference set achieved over 80% accuracy on the test cases, except for cases where $f(s_1) \to s_1$ is *yes* and $s_1 \to s_2$ is *unk*. Table 6 shows accuracies for the naturalistic transitivity test set. Again, models trained with the basic inference set performed substantially below chance for the cases $f(s_1) \to s_2$, where $f(s_1) \to s_1$ is *yes* and $s_1 \to s_2$ is *unk*. This suggests that while the models achieve over 80% accuracy on both $f(s_1) \to s_1$ and $s_1 \to s_2$ validation sets, they do not apply transitivity inference from the inferences $f(s_1) \to s_1$ and $s_1 \to s_2$, but rather predict the label for the composite inference $f(s_1) \to s_2$ by judging whether it is similar to the veridical inference $f(s_1) \to s_1$ in the training set.

Accuracy of models trained with MNLI was low because they predicted *yes* for many examples where correct labels were *unk*, as in (4).

(4) $f(s_1)$: Someone wished that **John saw Tom** or Greg.
s_2: John saw Tom. (*unk*)

[1]We use the Pytorch implementation of BERT released at https://github.com/huggingface/transformers.

Type	Templates
Pronoun	At that moment, we f that s
Pronoun	Then he f that s
Specific group	The customers f that s
Specific group	Some economists f that s
Proper noun	Hanson f that s

Table 7: Examples of additional templates used for generating veridical inference datasets. Here f is a place for a veridical verb and s for an embedded sentence.

The models predicted *yes* for over 80% of the fully synthetic transitivity test set and more than 60% of the naturalistic transitivity test set. These results are consistent with the findings in McCoy et al. (2019), namely, that models trained with MNLI tend to predict entailment relations when the hypothesis is a subsequence of the premise, as in (4).

When models are trained with MNLI mixed with the basic inference set, they seem to improve performance on the fully synthetic transitivity test set. One reason for this result is that the models might use heuristics to make predictions for some *unk* examples in the fully synthetic inference set. Error analysis shows that the models tend to predict *unk* when either a premise or a hypothesis contains a negation like (5).

(5) $f(s_1)$: Someone knew that Fred praised Henry or Ann.
s_2: Fred did **not** praise Ann. (*unk*)

These heuristics might be related to the annotation artifact (Gururangan et al., 2018) in MNLI, because an inference example involving negation words tends to be a *contradiction*[2]. Moreover, models can memorize the basic inference set regardless of the existence of MNLI in the training set, so performance seems to be better.

Note that models trained with MNLI mixed with the basic inference set still failed on the naturalistic transitivity inference $f(s_1) \rightarrow s_2$ where $f(s_1) \rightarrow s_1$ is *yes* and $s_1 \rightarrow s_2$ is *unk*. Since the naturalistic basic inference examples $s_1 \rightarrow s_2$ contain various linguistic phenomena, models cannot rely on the heuristics for such examples.

Is poor performance of transitivity inference due to overfitting on verbs? To determine whether models do not overfit on clause-embedding verbs, we analyze the models under

[2]We use binary labels (*entailment*/*non-entailment*) and take *contradiction* as *non-entailment*.

Data			Model	
$f(s_1) \rightarrow s_1$	$s_1 \rightarrow s_2$	$f(s_1) \rightarrow s_2$	**LSTM-B** ($\triangle$)	**BERT-B** ($\triangle$)
yes	*yes*	***yes***	97.9 (+0.8)	100.0 (0.0)
yes	*unk*	***unk***	0.0 (0.0)	2.3 (−6.6)
unk	*yes*	***unk***	99.0 (+1.9)	100.0 (0.0)
unk	*unk*	***unk***	99.2 (+1.9)	100.0 (0.0)

Table 8: Accuracies of models in the setting (I). ($\triangle$) is the difference from the accuracy in Table 6.

Data			Model	
$f(s_1) \rightarrow s_1$	$s_1 \rightarrow s_2$	$f(s_1) \rightarrow s_2$	**LSTM-B** ($\triangle$)	**BERT-B** ($\triangle$)
yes	*yes*	***yes***	90.0 (−7.1)	93.6 (−6.4)
yes	*unk*	***unk***	2.2 (+2.2)	17.9 (+9.0)
unk	*yes*	***unk***	89.9 (−7.2)	94.0 (−6.0)
unk	*unk*	***unk***	98.3 (+1.0)	95.6 (+1.8)

Table 9: Accuracies of models in the setting (II). ($\triangle$) is the difference from the accuracy in Table 6.

two additional settings using naturalistic transitivity datasets: (I) we use various templates other than "Someone f that s_1" to generate the main clause in $f(s_1)$, and (II) we flip the gold labels of 10% veridical inference $f(s_1) \rightarrow s_1$ instances, randomly sampled, instead of using gold labels uniquely fixed from verb types. These two complex settings expose models to more natural evaluation settings that consider the context-sensitive property of veridicality.

For evaluation setting (I) using various templates involving clause-embedding verbs, we manually select forty main clauses of the verb veridicality dataset (Ross and Pavlick, 2019) and provide additional templates. Table 7 shows examples of additional templates involving clause-embedding verbs used for generating veridical inference datasets.

Table 8 and Table 9 show the results for (I) and (II), respectively. These results show the same trends as those in Table 6, indicating that even when we consider the complexity of veridical inference in our analysis, the models fail to consistently perform composite inferences.

4.3 Analysis with data augmentation

We further hypothesize that even if the current models fail to consistently perform composite inferences, data augmentation with a small number of composite inference examples might allow models to learn transitivity inference. Thus, we evaluate models trained with basic inferences $f(s_1) \rightarrow s_1$ and $s_1 \rightarrow s_2$ and with a subset of the composite inferences $f(s_1) \rightarrow s_2$ on a naturalistic inference test set. Considering that models fail on composite inference $f(s_1) \rightarrow s_2$ where f

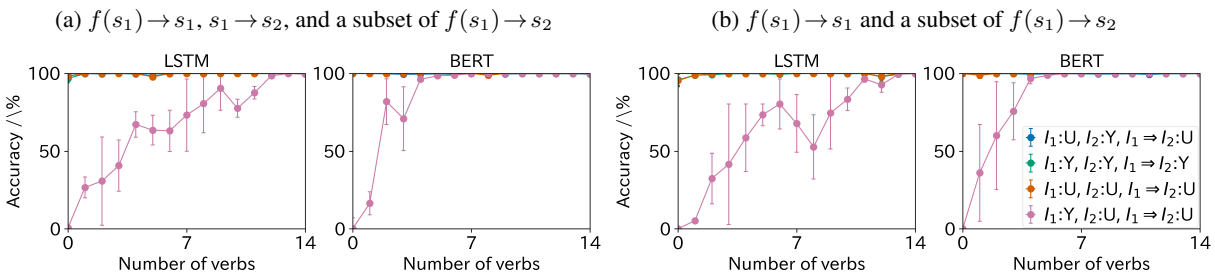

Figure 2: Accuracies of models trained with (a) and (b). I_1 indicates the first basic pattern $f(s_1) \rightarrow s_1$ and I_2 indicates the second basic pattern $s_1 \rightarrow s_2$. Y means *entailment* and U means *non-entailment*. The horizontal axis shows the number of veridical verbs for the additional training set.

Data			Model		Human
$f(s_1) \rightarrow s_1$	$s_1 \rightarrow s_2$	$f(s_1) \rightarrow s_2$	**LSTM-B**	**BERT-B**	
yes	*yes*	***yes***	100.0 ± 2.7	100.0 ± 0.0	98.8
yes	*unk*	***unk***	0.0 ± 0.0	8.9 ± 7.8	98.8
unk	*yes*	***unk***	97.1 ± 2.7	100.0 ± 0.0	44.9
unk	*unk*	***unk***	97.3 ± 2.6	100.0 ± 0.0	99.6
Test Overall			72.9 ± 2.0	77.2 ± 2.0	85.5

Table 10: Comparison between accuracies of humans and the models trained with the basic inference set.

is veridical and $s_1 \rightarrow s_2$ is *unk*, we gradually add veridical verbs (e.g., *know*) one-by-one to generate an additional training set of composite inference $f(s_1) \rightarrow s_2$ and analyze performance on a test set. Figure 2(a) shows that this data augmentation improved performance on test examples $f(s_1) \rightarrow s_2$ where f is veridical and $s_1 \rightarrow s_2$ is *unk*, while maintaining accuracy on the remaining examples in the test set. BERT achieved 100% accuracy over the entire test set by adding composite inferences generated from four veridical verbs, whereas in the case of LSTM twelve veridical verbs were needed to achieve the same accuracy.

To determine whether models augmented with composite inference examples learn the ability to combine basic inferences to perform transitivity inference, we analyze the performance of models where basic inference examples are not included in the training set. Figure 2(b) shows that models trained with only the basic inference set $f(s_1) \rightarrow s_1$ and a subset of the composite inference set $f(s_1) \rightarrow s_2$ also had improved accuracy. This result supports findings that models do not combine the basic inference $f(s_1) \rightarrow s_1$ and $s_1 \rightarrow s_2$, but rather predict the label for a composite inference $f(s_1) \rightarrow s_2$ by judging whether it is similar to inference patterns found in the training set.

4.4 Comparison with humans

To investigate how humans perform on transitivity inference tasks, we collect human judgements for a subset of our naturalistic inference dataset. We asked crowdsourced workers to label 960 transitivity inference examples involving all the clause-embedding verbs in Table 4. Following prior works involving crowdsourced NLI datasets (Zhang et al., 2017; Ross and Pavlick, 2019), we instructed raters to label each premise-hypothesis pair with the degree of entailment on a 5-point Likert scale, with 1 meaning a hypothesis is definitely not true given the premise, and 5 meaning a hypothesis is definitely true. We collected three annotations per pair on Amazon Mechanical Turk (see Appendix D for details), and the inter-rater agreement (the Pearson correlation among raters, averaged across both examples and raters) was 0.76. As model predictions are discrete (*yes* or *unk*), we discretized human scores into evenly sized bins, setting *yes* if the score was 4 or higher and set *unk* if the score was 3 or lower. We assumed the majority of three discretized labels as the final human judgement.

Table 10 shows that humans generally follow the distinction between veridical and non-veridical verbs traditionally assumed in the lexical semantics, as well as the transitivity of entailment relation. In particular, while as we saw in Section 4.2 the DNN models performed substantially below chance for transitivity inferences where $f(s_1) \rightarrow s_1$ is *yes* and $s_1 \rightarrow s_2$ is *unk*, human performance is near perfect for such inferences.

Interestingly, however, humans tend to predict

incorrect labels for transitivity inferences where the verb f is non-veridical (so $f(s_1) \rightarrow s_1$ is *unk*) and the embedded inference $s_1 \rightarrow s_2$ is *yes*. This might be because a natural complement as in (6) induces veridicality bias (Ross and Pavlick, 2019), that is, no matter whether a complement verb f is veridical or non-veridical, humans tend to decide the truth value of $f(s_1)$ by judging whether its complement s_1 is true. Thus, judgement for $f(s_1) \rightarrow s_2$ coincides with that of $s_1 \rightarrow s_2$ in this case.

(6) $f(s_1)$: Someone **believed** that a man is jumping off a low wall.

s_1: A man is jumping off a low wall.

s_2: A man is jumping a wall.

5 Conclusion

We introduced an analysis method using transitivity inferences for evaluating systematic generalization capacities of NLI models. We found that current NLI models do not perform consistently well on transitivity inference tasks. Furthermore, data augmentation analysis suggested that models can memorize composite inference examples, but do not perform the intended transitivity inferences combining basic inference examples.

Overall, our results indicated that despite the impressive performance of DNN models on standard NLI datasets, there remains much room for improving their systematic generalization capacities with respect to combining basic inferential abilities on various linguistic phenomena. Regarding what is necessary for improving the systematic generalization capacity, one interesting possibility is explicitly feeding some form of logic-guided transitivity rules to models, which is left for future work. Our analysis method using transitivity can be an effective tool for further progress in the study of compositional NLI.

Acknowledgement

We thank the three anonymous reviewers for their helpful comments and suggestions. We are also grateful to Masashi Yoshikawa for helpful discussions. This work was partially supported by the RIKEN-AIST Joint Research Fund (feasibility study) and JSPS KAKENHI Grant Number JP20K19868.

References

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

Akari Asai and Hannaneh Hajishirzi. 2020. Logic-guided data augmentation and regularization for consistent question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5642–5650.

Marco Baroni. 2020. Linguistic generalization and compositionality in modern artificial neural networks. *Philosophical Transactions of the Royal Society B*, 375(1791):20190307.

David Beaver. 2001. *Presupposition and Assertion in Dynamic Semantics*. CSLI Publications.

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

Samuel R. Bowman, Christopher Potts, and Christopher D. Manning. 2015b. Recursive neural networks can learn logical semantics. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 12–21.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence and the 17th Pacific Rim International Conference on Artificial Intelligence (IJCAI-PRICAI)*.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. *Recognizing Textual Entailment: Models and Applications*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Dirk van Dalen. 2013. *Logic and Structure*, 5 edition. Springer.

Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J. Gershman, and Noah D. Goodman. 2018. Evaluating compositionality in sentence embeddings. In *Proceedings of the 40th Annual Conference of the Cognitive Science Society*, pages 1596–1601.

Jacob Devlin, Chang Ming-Wei, Lee Kenton, and Toutanova Kristina. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186.

Jerry A. Fodor and Zenon W. Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.

Atticus Geiger, Ignacio Cases, Lauri Karttunen, and Christopher Potts. 2019. Posing fair generalization tasks for natural language inference. 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)*, pages 4484–4494.

Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 163–173.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 650–655.

Emily Goodwin, Koustuv Sinha, and Timothy J. O'Donnell. 2020. Probing linguistic systematicity. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1958–1969.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 107–112.

Irene Heim and Angelika Kratzer. 1998. *Semantics in Generative Grammar*. Blackwell.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Philip N. Johnson-Laird and Ruth M.J. Byrne. 1991. *Deduction*. Erlbaum.

Lauri Karttunen and Stanley Peters. 1979. Conventional implicatures. In Choon Kyu Oh and David A. Dineen, editors, *Syntax and Semantics 11: Presupposition*, pages 1–56. Academic Press.

Sangeet Khemlani and Philip N Johnson-Laird. 2012. Theories of the syllogism: A meta-analysis. *Psychological bulletin*, 138(3):427–457.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. 2018. LSTMs can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1426–1436.

Brenden M. Lake and Marco Baroni. 2017. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics (TACL)*, 4:521–535.

Nelson F. Liu, Roy Schwartz, and Noah A. Smith. 2019. Inoculation by fine-tuning: A method for analyzing challenge datasets. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2171–2179.

Gary Marcus. 2003. *The Algebraic Mind: Integrating Connectionism and Cognitive Science*. MIT Press.

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)*, pages 216–223.

Marie-Catherine de Marneffe, Christopher D. Manning, and Christopher Potts. 2012. Did it happen? the pragmatic complexity of veridicality assessment. *Computational Linguistics*, 38(2):301–333.

R. Thomas 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)*, pages 3428–3448.

Junghyun Min, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2339–2352.

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), pages 2055–2061.

Richard Montague. 1973. The proper treatment of quantification in ordinary English. In Jaakko Hintikka, Julius M. E. Moravcsik, and Patrick Suppes, editors, *Approaches to Natural Language*, pages 189–224. Reidel, Dordrecht. Reprinted in Richmond H. Thomason (ed.), *Formal Philosophy: Selected Papers of Richard Montague*, 247–270, 1974, New Haven: Yale University Press.

Yixin Nie, Yicheng Wang, and Mohit Bansal. 2019. Analyzing compositionality-sensitivity of NLI models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6867–6874.

Qiang Ning, Zhili Feng, and Dan Roth. 2017. A structured learning approach to temporal relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1027–1037.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 180–191.

Kyle Richardson, Hai Hu, Lawrence S. Moss, and Ashish Sabharwal. 2020. Probing natural language inference models through semantic fragments. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Alexis Ross and Ellie Pavlick. 2019. How well do NLI models capture verb veridicality? 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)*, pages 2230–2240.

Ohad Rozen, Vered Shwartz, Roee Aharoni, and Ido Dagan. 2019. Diversify your datasets: Analyzing generalization via controlled variance in adversarial datasets. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 196–205.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. CLUTRR: A diagnostic benchmark for inductive reasoning from text. 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)*, pages 4506–4515.

Alon Talmor and Jonathan Berant. 2019. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4911–4921.

Anne S. Troelstra and Helmut Schwichtenberg. 2000. *Basic Proof Theory*, 2 edition. Cambridge University Press.

Masatoshi Tsuchiya. 2018. Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC)*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019a. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *Proceedings of Advances in Neural Information Processing Systems 32 (NIPS)*, pages 3266–3280.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2018. On the practical computational power of finite precision RNNs for language recognition. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 740–745.

Aaron Steven White and Kyle Rawlins. 2018. The role of veridicality and factivity in clause selection. In *Proceedings of the 48th Annual Meeting of the North East Linguistic Society, Amherst, MA, USA. GLSA Publications*.

Aaron Steven White, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2018. Lexicosyntactic inference in neural models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4717–4724.

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)*, pages 1112–1122.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, and Kentaro Inui. 2020. Do neural models learn systematicity of monotonicity inference in natural language? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6105–6117.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019a. Can neural networks understand monotonicity reasoning? In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 31–40.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019b. HELP: A dataset for identifying shortcomings of neural models in monotonicity reasoning. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 250–255.

Sheng Zhang, Rachel Rudinger, Kevin Duh, and Benjamin Van Durme. 2017. Ordinal common-sense inference. *Transactions of the Association for Computational Linguistics (TACL)*, 5:379–395.

A Details about the Boolean logic fragment

Table 11 shows the context-free grammar used to generate sentences for Boolean logic reasoning with conjunction, disjunction, and negation. Each rewriting rule is paired with the corresponding semantic composition rule in standard Montagovian semantics to generate the logical form of a sentence (Montague, 1973; Heim and Kratzer, 1998). We use ten items each for proper names (PN), intransitive verbs (IV), and transitive verbs (TV). Each sentence is generated with a verb in the past tense.

For sentences with multiple NPs, we assume the surface-scope reading where the subject NP takes scope over the object NP. For instance, the sentence *Ann and Bob saw Chris or Daniel*, where the subject NP is conjunctive and the object NP is disjunctive, has the logical form $(\mathbf{see}(\mathbf{ann}, \mathbf{chris}) \vee \mathbf{see}(\mathbf{ann}, \mathbf{daniel})) \wedge (\mathbf{see}(\mathbf{bob}, \mathbf{chris}) \vee \mathbf{see}(\mathbf{bob}, \mathbf{daniel}))$.

There are two types of negation, sentential negation (SNEG) and verbal negation (VNEG), which are distinguished with respect to their scope interpretation. Thus, the sentence *Ann and Bob did not swim* has the logical form $\neg\mathbf{swim}(\mathbf{ann}) \wedge \neg\mathbf{swim}(\mathbf{bob})$, while the sentence *It is not the case that Ann and Bob did not swim* has the logical form $\neg(\mathbf{swim}(\mathbf{ann}) \wedge \mathbf{swim}(\mathbf{bob}))$.

To generate a premise-hypothesis pair (s_1, s_2) using this Boolean logic fragment, we first generate a sentence s_1 and derive its logical form F_1 using the grammar in Table 11. We then randomly select one of the atomic formulas appearing in F_1, say A, and takes its positive (A) or negative ($\neg A$) form, which is in turn converted to the hypothesis sentence s_2 using the same grammar. The gold label (*entailment* or *non-entailment*) for the pair (s_1, s_2) is determined by checking whether F_1 logically entails A or $\neg A$ using a first-order-logic theorem prover[3].

B Training details

In all experiments, we trained models on eight NVIDIA DGX-1 Tesla V100 GPUs. The runtime for training each model was about 1-8 hours, depending on the size of the training set. The order of training instances was shuffled for each model.

[3]https://github.com/vprover/vampire

C Supplementary results on the random train-test split

To confirm that our transitivity inference dataset is not excessively difficult, we conducted additional experiments using the random $9:1$ train:test split of transitivity inference ($f(s_1) \rightarrow s_2$) datasets. We evaluate models under two settings: (i) models trained with the train split of transitivity inference datasets and (ii) models trained with the train split mixed with MNLI. Table 12 shows the results on the random train-test split of our full-synthetic transitivity dataset, and Table 13 shows the results on the random train-test split of our naturalistic transitivity dataset. These results showed that regardless of the existence of MNLI in the training set, models achieved perfect performance on our transitivity inference test set with the standard random train-test split setting.

D Human judgement details

Using Amazon Mechanical Turk, we collected human judgements for 960 naturalistic veridical inference examples and 960 naturalistic transitivity inference examples. We required raters to have completed at least 5,000 approved tasks to maintain a 99% approval rating. Raters could indicate by a checkbox that one or both sentences did not make sense, but no rater clicked the checkbox. We collected three annotations per pair and paid $0.06 per labeled pair.

Since humans predict incorrect labels for some composite inference examples $f(s_1) \rightarrow s_2$ where the verb f is non-veridical, we checked the accuracy of human judgement on a set of premise-hypothesis pairs $f(s_1) \rightarrow s_1$ and $f(s_1) \rightarrow s_2$ involving each non-veridical verb, as shown in Table 14. Annotators tended to incorrectly make judgements for both $f(s_1) \rightarrow s_1$ and $f(s_1) \rightarrow s_2$. Regarding accuracy for each non-veridical verb, annotators correctly drew inferences containing *wish* and *hope*, while they tended to draw inferences containing *claim* and *hear* incorrectly.

In comparison with the previous veridicality dataset MegaVeridicality2 (White et al., 2018), the accuracy tended to be lower than that in MegaVeridicality2[4]. As (7) shows, while a simple complement is used for MegaVeridicality2, a natural complement like (8) might induce veridicality

[4]We calculated the percentage of the majority judgement for each verb for ten different annotations in MegaVeridicality2.

Syntax	Semantics
S $\rightarrow$ NP VP_{past}	$[\![\text{S}]\!] = [\![\text{NP}]\!]([\![\text{VP}_{\text{past}}]\!])$
S $\rightarrow$ SNEG S	$[\![\text{S}]\!] = [\![\text{SNEG}]\!]([\![\text{S}]\!])$
NP $\rightarrow$ PN	$[\![\text{NP}]\!] = [\![\text{PN}]\!]$
NP $\rightarrow$ PN CON PN	$[\![\text{NP}]\!] = \lambda F.[\![\text{CON}]\!]([\![\text{PN}]\!](F), [\![\text{PN}]\!](F))$
NP $\rightarrow$ PN , PN , CON PN	$[\![\text{NP}]\!] = \lambda F.[\![\text{CON}]\!]([\![\text{PN}]\!](F), [\![\text{CON}]\!]([\![\text{PN}]\!](F), [\![\text{PN}]\!](F)))$
$\text{VP}_{tense} \rightarrow \text{IV}_{tense}$	$[\![\text{VP}_{tense}]\!] = [\![\text{IV}_{tense}]\!]$
$\text{VP}_{tense} \rightarrow \text{TV}_{tense}$ NP	$[\![\text{VP}_{tense}]\!] = \lambda x.[\![\text{NP}]\!](\lambda y.[\![\text{TV}_{tense}]\!](x, y))$
$\text{VP}_{\text{past}} \rightarrow$ VNEG VP_{base}	$[\![\text{VP}_{\text{past}}]\!] = \lambda x.[\![\text{VNEG}]\!]([\![\text{VP}_{\text{base}}]\!](x))$
PN $\rightarrow$ `Ann` | `Bob` | `Chris` | $\cdots$	$[\![\text{PN}]\!] = \lambda F.F(\mathbf{sym})$
$\text{IV}_{\text{base}} \rightarrow$ `swim` | `drink` | `smoke` | $\cdots$	$[\![\text{IV}_{\text{base}}]\!] = \lambda x.\mathbf{sym}(x)$
$\text{IV}_{\text{past}} \rightarrow$ `swam` | `drank` | `smoked` | $\cdots$	$[\![\text{IV}_{\text{past}}]\!] = \lambda x.\mathbf{sym}(x)$
$\text{TV}_{\text{base}} \rightarrow$ `see` | `visit` | `touch` | $\cdots$	$[\![\text{TV}_{\text{base}}]\!] = \lambda y \lambda x.\mathbf{sym}(x, y)$
$\text{TV}_{\text{past}} \rightarrow$ `saw` | `visited` | `touched` | $\cdots$	$[\![\text{TV}_{\text{past}}]\!] = \lambda y x.\mathbf{sym}(x, y)$
SNEG $\rightarrow$ `it is not the case that`	$[\![\text{SNEG}]\!] = \lambda P.\neg P$
VNEG $\rightarrow$ `did not`	$[\![\text{VNEG}]\!] = \lambda P.\neg P$
CON $\rightarrow$ `and`	$[\![\text{CON}]\!] = \lambda P \lambda Q.P \wedge Q$
CON $\rightarrow$ `or`	$[\![\text{CON}]\!] = \lambda P \lambda Q.P \vee Q$

Table 11: Grammar for the Boolean logic fragment with semantic composition. Feature *tense* for VP is either "base" or "past." In semantic composition, **sym** is the place where the symbol (lemma) for a lexical item appears.

Data			Model			
$f(s_1) \rightarrow s_1$	$s_1 \rightarrow s_2$	$f(s_1) \rightarrow s_2$	**LSTM-T**	**LSTM-M&T**	**BERT-T**	**BERT-M&T**
yes	*yes*	***yes***	99.8 ± 0.1	100.0 ± 0.1	100.0 ± 0.0	100.0 ± 0.0
yes	*unk*	***unk***	99.3 ± 0.0	99.7 ± 0.1	100.0 ± 0.0	100.0 ± 0.0
unk	*yes*	***unk***	99.4 ± 0.2	99.8 ± 0.1	100.0 ± 0.0	100.0 ± 0.0
unk	*unk*	***unk***	99.6 ± 0.1	100.0 ± 0.1	100.0 ± 0.0	100.0 ± 0.0

Table 12: Accuracies on the random train-test split of our fully synthetic transitivity dataset. **-T** indicates a model trained with the train split of the transitivity inference set, and **-M&T** indicates a model trained with MNLI mixed with the train split.

Data			Model			
$f(s_1) \rightarrow s_1$	$s_1 \rightarrow s_2$	$f(s_1) \rightarrow s_2$	**LSTM-T**	**LSTM-M&T**	**BERT-T**	**BERT-M&T**
yes	*yes*	***yes***	99.2 ± 0.0	100.0 ± 0.1	100.0 ± 0.0	100.0 ± 0.0
yes	*unk*	***unk***	98.4 ± 0.2	99.1 ± 0.1	100.0 ± 0.0	100.0 ± 0.0
unk	*yes*	***unk***	98.3 ± 0.2	99.3 ± 0.1	100.0 ± 0.0	100.0 ± 0.0
unk	*unk*	***unk***	99.6 ± 0.1	100.0 ± 0.1	100.0 ± 0.0	100.0 ± 0.0

Table 13: Accuracies on the random train-test split of our naturalistic transitivity test set.

bias (Ross and Pavlick, 2019), resulting in incorrect judgements on veridical inference. Whether a verb is veridical or non-veridical, humans tend to judge the complement as true.

(7) $f(s_1)$: Someone **believed** that something happened.

s_1: Something happened.

(8) $f(s_1)$: Someone **believed** that a man is jumping off a low wall.

s_1: A man is jumping off a low wall.

s_2: A man is jumping a wall.

E Supplementary results with data augmentation

In Section 4.3, we gradually added a subset of the composite inferences $f(s_1) \rightarrow s_2$ involving a veridical verb (e.g., *know*) to the training set and evaluated the performance of models on a naturalistic inference test set. We also evaluated the performance of models under two conditions: (a) models trained with the basic inference set $s_1 \rightarrow s_2$ and a subset of the composite inference set $f(s_1) \rightarrow s_2$ and (b) models trained with a subset of the composite inference set $f(s_1) \rightarrow s_2$. Figure 3(a) shows that the models significantly improved accuracy on composite inferences except for the test example $f(s_1) \rightarrow s_2$, whose label dif-

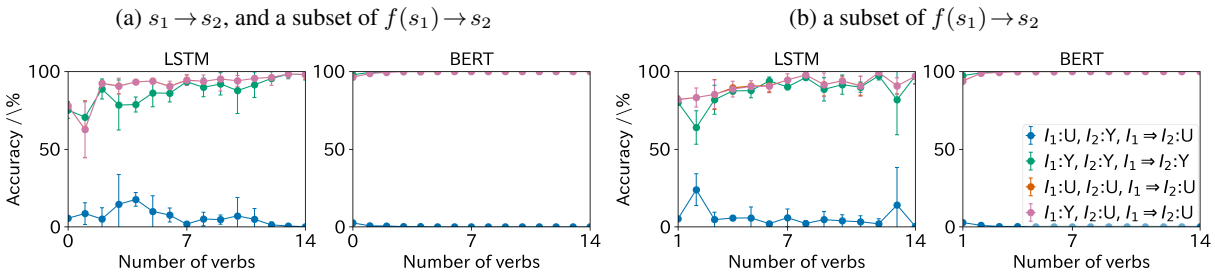

Figure 3: Accuracies of models trained with (a) and (b). I_1 is the first basic pattern $f(s_1) \rightarrow s_1$ and I_2 is the second basic pattern $s_1 \rightarrow s_2$. Y indicates *entailment* and U indicates *non-entailment*. The horizontal axis shows the number of veridical verbs for the additional training set.

Verb	$f(s_1) \rightarrow s_1$	$f(s_1) \rightarrow s_2$	**MegaV2**
argue	34 (-46)	66 (-14)	80
assume	70 (-15)	79 (-6)	85
believe	19 (-71)	59 (-31)	90
claim	15 (-65)	56 (-24)	80
doubt	91 (+9)	96 (+16)	80
estimate	35 (-50)	64 (-21)	85
expect	53 (-27)	70 (-10)	80
feel	42 (-53)	67 (-28)	95
hear	14 (-41)	53 (-2)	55
hope	77 (-8)	92 (+7)	85
imply	18 (-47)	58 (-7)	65
predict	50 (-25)	73 (-2)	75
suspect	48 (-47)	79 (-16)	95
think	18 (-77)	57 (-38)	95
wish	77 (+7)	92 (+22)	70

Table 14: Accuracy (%) of human judgements for each non-veridical verb. MegaV2 indicates the percentage of those annotators who judge each verb to be non-veridical in MegaVeridicality2 (White et al., 2018). A number in parentheses is a difference from the accuracy in MegaVeridicality2.

fered from that of $s_1 \rightarrow s_2$. Moreover, their performance was maintained even without composite inference examples in the training set. This indicates that models predict labels for the composite inference example only by judging whether it is similar to the basic inference example in the training set.

Figure 3(b) shows the result when models are trained only with a subset of the composite inference set $f(s_1) \rightarrow s_2$. As non-veridical verbs are not included in the training set in this setting, the models predict labels for composite inferences involving non-veridical verbs by judging whether they are similar to composite inferences involving veridical verbs in the training set. The models thus fail on composite inference examples $f(s_1) \rightarrow s_2$ where f is non-veridical and $s_1 \rightarrow s_2$ is *yes*. The labels of such non-veridical inference examples are opposite to those of veridical inference examples in the training set.