

# SETI: Systematicity Evaluation of Textual Inference

**Xiyan Fu**

Dept. of Computational Linguistics  
Heidelberg University  
fu@cl.uni-heidelberg.de

**Anette Frank**

Dept. of Computational Linguistics  
Heidelberg University  
frank@cl.uni-heidelberg.de

## Abstract

We propose SETI (Systematicity Evaluation of Textual Inference), a novel and comprehensive benchmark designed for evaluating pre-trained language models (PLMs) for their systematicity capabilities in the domain of textual inference. Specifically, SETI offers three different NLI tasks and corresponding datasets to evaluate various types of systematicity in reasoning processes. In order to solve these tasks, models are required to perform compositional inference based on known primitive constituents. We conduct experiments of SETI on six widely used PLMs. Results show that various PLMs are able to solve *unseen compositional inferences* when having encountered the knowledge of how to combine primitives, with good performance. However, they are considerably limited when this knowledge is unknown to the model (40-100 % points decrease). Furthermore, we find that PLMs can improve drastically once exposed to crucial compositional knowledge in minimalistic shots. These findings position SETI as the first benchmark for measuring the future progress of PLMs in achieving systematicity generalization in the textual inference.

## 1 Introduction

Natural Language Inference (NLI) determines whether a *hypothesis* follows from a *premise* (Dagan et al., 2013; Bowman et al., 2015; Williams et al., 2018) and has been explored for decades. Existing large pre-trained language models (PLMs) have shown remarkable performance on this task (Devlin et al., 2019; Raffel et al., 2019; Lan et al., 2020). To better assess the true capabilities of models to perform NLI, various associated tasks and benchmarks have been proposed. These works concentrate on exploring how models make predictions, e.g. by establishing ‘hard’ NLI datasets (Koreeda and Manning, 2021) or asking models to ‘explain’ their predictions through highlighting (Camburu et al., 2018), or by generating plausible















Task	Train		Test
	Primitive Concepts	Compositional Concepts	Compositional Concepts
Task1	 	 	 
Task2		 	 
Task3			

Table 1: Illustrating three visual tasks realizing different forms of *systematicity* in compositional generalization.

explanations (Bhagavatula et al., 2020). But little is known about how well such models are able to address *compositional generalization*.

Compositional generalization focuses on how to combine primitive units to predict larger compounds (Hupkes et al., 2020). A key property underlying compositional generalization is *systematicity* (Fodor and Pylyshyn, 1988), a hallmark of human cognition. Systematicity concerns the ability of (re)combining known constituents and composing rules. For example, humans who understand ‘red apple’ and ‘green train’ are able to conceptualize ‘red train’ by recombining ‘red’ and ‘train’ into a new concept. Similar effects of systematicity (generalization) can be studied in Natural Language Understanding (NLU) (Lake and Baroni, 2018; Kim and Linzen, 2020). Since PLMs have achieved results on par with human performance by fitting NLI training data (Wang et al., 2019), we aim to evaluate to what extent these models can master different types of *systematicity* in textual inference.

We propose a novel benchmark SETI (*Systematicity Evaluation of Textual Inference*), which extensively explores systematicity in NLI. SETI contains three interrelated yet independent tasks covering various types of systematicity: 1) **Task1: primitives** → **compositions** aims to evaluate if models can perform compositional inference if primitive constituents of the given inference task have been learned independently. 2) **Task2: compositions** → **compositions** aims to evaluate if models can perform novel compositional inferences if

their constituents have been learned in other compositions. 3) **Task3: primitives and compositions** → **compositions** aims to evaluate if models can perform novel compositional inferences if one primitive constituent has been learned independently, while the other has only been encountered in compositions. SETI can be used to explore systematicity in NLI *comprehensively* since it considers all possibilities of how to construct a novel composition from known constituent types, derived from the ‘permutation and combination’<sup>1</sup> theory acting between primitives and compositions. We introduce these tasks in detail in Section §3. To make the instantiations of systematicity covered in SETI easily accessible, we indicate three analogous visual tasks in Table 1. They test if models can understand: i) a novel compositional concept *red apple* – given the primitive concepts <sup>2</sup>*red* and *apple* have been learned independently; ii) a novel compositional concept *red train* – given constituent concepts *red* and *train* have been encountered in compositions *red apple* and *green train*; iii) a novel compositional concept *red train* – given *red* has been learned independently, and *train* in the compositional concept *green train*.

To apply SETI in practice, we define veridical inference (Karttunen and Peters, 1979; Ross and Pavlick, 2019) and natural inference as *primitives*, and their combinations as *compositions*. For each systematicity task setting, we provide two instantiations: *trivial* and *non-trivial*, depending on the variety of instances presented to the model in training. While both settings fulfill the given task requirements, the *non-trivial* setting is more challenging because the compositional inference knowledge of how to combine constituents is not seen in training.

We evaluate six well-known PLMs on all SETI tasks. They show good performance in *trivial* settings, but inferior results in *non-trivial* settings, for all tasks. This indicates that models can generalize well to unseen compositions when constituents and compositional knowledge are known, while they are limited when they lack knowledge about how to compose constituents. Hence, we further explore whether, and to what extent we can enhance the systematicity capabilities. Our experiments indicate

<sup>1</sup>While permutations make sense in a setting that deals with grammaticality, this does not hold for inference, hence we do not consider permutation order for SETI.

<sup>2</sup>In the visual domain, primitive concepts such as colors and object properties seldomly occur independently of objects, instead they occur in composition with objects. Here, they are only used for task clarifications.

that all PLMs benefit greatly from being exposed to minimal doses of relevant compositional instances.

Our main contributions are as follows:

- i) We introduce SETI (Systematicity Evaluation of Textual Inference), which to our knowledge is the first benchmark to comprehensively evaluate the systematicity capabilities of PLMs when performing NLI.
- ii) We provide datasets for three NLI challenge tasks that evaluate systematicity, with controlled splits for seen vs. unseen information.
- iii) We conduct experiments for six widely used PLMs. The results indicate that models generalize well to unseen compositions if they have previously acquired relevant compositional inference knowledge, but are limited when lacking such knowledge.

## 2 Related Work

**Textual Inference** Natural Language Inference (NLI) involves reasoning across a *premise* and *hypothesis*, determining the inferential relationship holding between them (Dagan et al., 2013; Bowman et al., 2015; Williams et al., 2018). As one of the major tasks for establishing Natural Language Understanding (NLU), NLI has been widely explored for decades. Recently, large pre-trained language models (Devlin et al., 2019; Raffel et al., 2019; Lan et al., 2020) exhibit remarkable performance on NLI tasks, on par with humans. To better explore the true NLI capabilities of models, various associated tasks and benchmarks have been proposed. Some work has probed NLI models by constructing hypothesis-only baselines (Glockner et al., 2018; Liu et al., 2020), finding that models capture undesired biases. McCoy et al. (2019); Zhou and Bansal (2020); Gubelmann et al. (2022) reveal that models rely on heuristics, e.g., lexical overlap, subsequence heuristics, etc. Nie et al. (2020); Chien and Kalita (2020) evaluate models in adversarial settings and show robustness improvement by training on additional adversarial data. Others focus on explainable NLI, such as highlighting input words that are essential for the label (Camburu et al., 2018), or generating plausible explanations (Bhagavatula et al., 2020). In this work, we focus on exploring the compositional generalization abilities of PLMs when performing textual inference.

**Systematicity** Systematicity is a crucial property of compositionality, which was first introduced in

cognitive science (Fodor and Pylyshyn, 1988) and recently formalized in Hupkes et al. (2020). It is the ability to make use of known concepts to produce novel concept combinations that have not been encountered before. Recently, systematicity has been widely explored in domains such as image caption generation (Nikolaus et al., 2019), visual attribute recognition (Misra et al., 2017; Li et al., 2020), question answering (Keysers et al., 2020; Liu et al., 2022) and semantic parsing (Lake and Baroni, 2018; Finegan-Dollak et al., 2018; Kim and Linzen, 2020; Zheng and Lapata, 2022). In this work, we focus on systematicity in the domain of textual inference.

Existing works that evaluate systematicity in textual inference only focus on one specific type. For example, Yanaka et al. (2021) evaluates systematicity by testing the transitivity of inference relations. Others conduct experiments on novel compositions involving specific linguistic phenomena, such as systematicity of predicate replacements and embedding quantifiers (Yanaka et al., 2020), systematicity when combining lexical entailment and negation (Geiger et al., 2020), and systematicity of quantifiers, negation and concerning the order between premises and hypotheses (Goodwin et al., 2020).

Compared to prior work, we propose a *comprehensive* systematicity evaluation benchmark SETI, which: i) covers the full spectrum of systematicity; ii) evaluates various PLMs; and iii) showcases how PLMs can overcome limitations in systematicity.

### 3 Reasoning Tasks for Systematicity

We now define primitive and compositional inferences and introduce three NLI systematicity tasks.

#### 3.1 Primitive and Compositional Inferences

Among various textual inference types, we select *veridical inference* and *natural inference* as two primitive inference tasks<sup>3</sup>, since they can be flexibly scaled to compositional inferences. Table 2 shows relevant notation and corresponding examples of the two primitive inference types.

**Veridical Inference** Veridical inference is strongly determined by the lexical meaning of sentence embedding verbs. In the context of a *veridical* verb we can infer that the proposition it takes as complement is taken to hold true. By contrast, in the context of a *non-veridical* verb, we *can not*

<sup>3</sup>We adopt binary labels (entailment/non-entailment) by grouping contradiction and neutral as non-entailment.

infer that the proposition it takes as complement is taken to hold true. (Karttunen, 1971; Ross and Pavlick, 2019).  $PI_{ver}$  in Table 2 shows examples of both verb classes. The verb “realize” in the premise “Someone realizes that a man is eating pizza” is *veridical* in relation to the embedded proposition “A man is eating pizza”, since speakers cannot say the premise unless they believe the latter proposition to be true. In contrast, “hope” is *non-veridical*, since the premise “Someone hopes that a man is eating pizza” does not license the equivalent conclusion towards the hypothesis “A man is eating pizza”. In our work, we emphasize veridicality in verb-complement constructions and formulate their inference potential in an NLI setting, as premise-hypothesis pairs, as established by Ross and Pavlick (2019). Specifically, the premises of all veridical inference samples follow the template “Someone  $f_v/f_{nv}$  that  $s$ ”, where  $f_v$  and  $f_{nv}$  represent veridical and non-veridical complement embedding verbs, respectively. We denote samples of entailed vs. non-entailed veridical inferences as  $f_v(s) \rightarrow s$  and  $f_{nv}(s) \not\rightarrow s$ , respectively.

**Natural Inference** A pair of sentences is considered a true entailment if we are able to infer the hypothesis based on the premise.  $PI_{nat}$  in Table 2 shows examples. We categorize natural inference samples into two groups: 1) *lexically-based inferences* typically build on lexical inference knowledge captured in lexical meaning relations, e.g., hypernymy  $boy \rightarrow kid$  in “A boy is jumping into the water”  $\rightarrow$  “A kid is jumping into the water”. 2) *structure-based inferences* involve structural changes, e.g., from active to passive voice and vice versa, as in “The detective follows the man”  $\rightarrow$  “The man is being followed by the detective”. We restrict natural inferences to these two types to facilitate controlled data creation. We denote entailed and non-entailed samples from these two groups as:  $s \xrightarrow{lex} s'$ ,  $s \not\xrightarrow{lex} s'$  and  $s \xrightarrow{stru} s'$ ,  $s \not\xrightarrow{stru} s'$ .

**Composing veridical and natural inference** To evaluate the compositional generalization ability of models, we construct *compositional inferences*  $CI_{ver\_nat}$  (CI) by combining *primitive* veridical inference  $PI_{ver}$  and natural inference  $PI_{nat}$ , following Yanaka et al. (2021) (see Table 3).

For such compositions to be valid, the hypothesis of a veridical inference must match the premise of a natural inference. This matching condition serves as a crucial link to perform transitive inference. Table 3 shows how a compositional inference

	Primitive Inference Types		Examples (premise $\rightarrow$ hypothesis)
Veridical Inference $PI_{ver}$	veridical $f_v(s) \rightarrow s$	non-veridical $f_{nv}(s) \nrightarrow s$	Someone realizes that a man is eating a pizza $\rightarrow$ A man is eating a pizza Someone hopes that a man is eating a pizza $\nrightarrow$ A man is eating a pizza
Natural Inference $PI_{nat}$	lexically-based inference rule $\xrightarrow{lex}$	entailment $s \xrightarrow{lex} s'$ non-entailment $s \not\xrightarrow{lex} s'$	A boy is jumping into the water $\rightarrow$ A kid is jumping into the water A woman is smiling $\nrightarrow$ A man is smiling
	structure-based inference rule $\xrightarrow{stru}$	entailment $s \xrightarrow{stru} s'$ non-entailment $s \not\xrightarrow{stru} s'$	The detective follows a man $\rightarrow$ A man is being followed by the detective A fish is being sliced by a man $\nrightarrow$ A cat is jumping into a box

Table 2: Examples of *primitive* veridical ( $PI_{ver}$ ) and natural ( $PI_{nat}$ ) inferences.  $s, s'$  represent distinct sentences.

$PI_{ver}$	$PI_{nat}$	$CI_{ver,nat}(CI)$	Composed Rules	Examples (premise $\rightarrow$ hypothesis)
$f_v(s) \rightarrow s$	$s \xrightarrow{lex} s'$	$f_v(s) \xrightarrow{lex^+} s'$	① $True \wedge True \rightarrow True$	He realizes a boy is jumping into the water $\rightarrow$ A kid is jumping into the water
$f_v(s) \rightarrow s$	$s \not\xrightarrow{lex} s'$	$f_v(s) \xrightarrow{lex^-} s'$	② $True \wedge False \rightarrow False$	He realizes a woman smiling $\nrightarrow$ A man is smiling
$f_{nv}(s) \nrightarrow s$	$s \xrightarrow{lex} s'$	$f_{nv}(s) \xrightarrow{lex^+} s'$	③ $False \wedge True \rightarrow False$	He hopes a boy is jumping into the water $\nrightarrow$ A kid is jumping into the water
$f_{nv}(s) \nrightarrow s$	$s \not\xrightarrow{lex} s'$	$f_{nv}(s) \xrightarrow{lex^-} s'$	④ $False \wedge False \rightarrow False$	He hopes a woman is smiling $\nrightarrow$ A man is smiling

Table 3: Examples of *compositional* inferences  $CI$  obtained by combining veridical and natural inference (we use  $\xrightarrow{lex}$  as example;  $\xrightarrow{stru}$  works analogously). For  $CI$ , the label ( $\rightarrow/\nrightarrow$ ) is decided by the Boolean ‘Composed Rules’. We use  $lex^+$  and  $lex^-$  to indicate the label of its  $PI_{nat}$  component being *True* or *False*, respectively.

sample ‘*He realizes a boy is jumping into the water*’  $\rightarrow$  ‘*A kid is jumping into the water*’ is composed from  $PI_{ver}$  ‘*He realizes a boy is jumping into the water*’  $\rightarrow$  ‘*A boy is jumping into the water*’ and  $PI_{nat}$  ‘*A boy is jumping into the water*’  $\rightarrow$  ‘*A kid is jumping into the water*’. This reasoning process we denote as:  $f_v(s) \rightarrow s \wedge s \rightarrow s' \Rightarrow f_v(s) \rightarrow s'$ .

In this way, we construct four types of compositional inferences  $CI$  from primitive  $PI_{ver}$  and  $PI_{nat}$  inferences, where Boolean logical rules (Table 3, col. 3) decide the label of  $CI$ , i.e., whether it yields entailment or non-entailment. In case both veridical  $PI_{ver}$  and natural inference  $PI_{nat}$  resolve to *True*,  $CI$  yields entailment, given the Boolean logic rule  $True \wedge True \rightarrow True$  (rule ①). By contrast, if  $PI_{nat}$  yields non-entailment, the compositional veridical inference  $CI$  will fail (rule ②). However, compositional inference with non-veridical verbs invariably yields non-entailment, no matter whether  $PI_{nat}$  resolves to *True* or *False*. This is again due to Boolean logic (rules ③, ④):  $False \wedge (True \vee False) \rightarrow False$ . In conclusion, the first two cases of  $CI$  are more complex, since models need to follow Boolean logic, while a model could exploit shortcuts and invariantly predict non-entailment with non-entailing verbs in  $PI_{ver}$ .

### 3.2 SETI Tasks

Having characterized the two types of primitive inferences we will use in our experiments, along with ways of composing them, we will now spell out i)

how to define increasingly difficult generalization tasks targeting systematicity, with ii) appropriate specifications of train and test settings, to guarantee proper assessment of a model’s generalizing capacities. Table 4 presents examples.

**Task1: primitives  $\rightarrow$  compositions** aims to evaluate whether a model can perform a compositional inference  $CI$  if its (primitive) constituent inferences  $PI_x$  and  $PI_y$ , have been learned independently, while their combination is unseen in training. Hence, *Train and Test sets* ( $D_{train|test}$ ) consist of instances  $e$  and  $\tilde{e}$ :

$$\begin{aligned} D_{train} &= \{e \mid e \in PI_x \vee e \in PI_y\} \\ D_{test} &= \{\tilde{e} \mid \tilde{e} \in CI\} \end{aligned} \quad (1)$$

We select *veridical* inference and *lexically-based* natural inference as primitive inferences, and combinations of these two primitives as compositional inferences, as formally specified below:

$$\begin{aligned} PI_x &= PI_{ver} = \{f_v(s) \rightarrow s, f_{nv}(s) \nrightarrow s\} \\ PI_y &= PI_{lex} = \{s \xrightarrow{lex} s', s \not\xrightarrow{lex} s'\} \\ CI &= \{f_v(s) \xrightarrow{lex^+} s', f_v(s) \xrightarrow{lex^-} s', \\ &\quad f_{nv}(s) \not\xrightarrow{lex^+} s', f_{nv}(s) \not\xrightarrow{lex^-} s'\} \end{aligned} \quad (2)$$

Here, sentences ( $s$  and  $s'$ ) of composed inferences  $CI$  are constrained to match sentences of their primitive constituents  $PI_{ver \vee lex}$ . This is a **trivial** setting since the challenge is restricted to classifying compositional inference from seen primitive inferences.



Tasks	Examples (premise $\rightarrow$ hypothesis)
Task1	$D_{train}$ $PI_x$ : Someone realizes a boy is jumping into the water $\rightarrow$ A boy is jumping into the water $PI_y$ : A boy is jumping into the water $\rightarrow$ A kid is jumping into the water
	$D_{test}$ $CI$ : Someone realizes a boy is jumping into the water $\rightarrow$ A kid is jumping into the water
Task2	$D_{train}$ $CI_x$ : Someone realizes a boy is jumping into the water $\rightarrow$ A kid is jumping into the water $CI_y$ : Someone hopes a woman is eating a pizza $\rightarrow$ A man is eating a pizza
	$D_{test}$ $CI$ : Someone hopes a boy is jumping into the water $\rightarrow$ A kid is jumping into the water
Task3	$D_{train}$ $PI$ : A man is driving a car $\rightarrow$ A car is being driven by a man $CI$ : Someone realizes a boy is jumping into the water $\rightarrow$ A kid is jumping into the water
	$D_{test}$ $CI$ : Someone realizes a man is driving a car $\rightarrow$ A car is being driven by a man

Table 4: Examples of three systematicity tasks from SETI. For each task, we select one sample from the *trivial* setting for representation.

However, overlaps of words between  $PI_{nat}$  and  $CI$  bear a risk of shortcuts (Sanchez et al., 2018). Hence, we also evaluate compositional inferences in a **non-trivial** setting, where sentences used in compositional inferences in  $D_{test}$  are constrained to differ from sentences used in primitive constituents in  $D_{train}$ . This is doable if we guarantee that instances from  $PI_{nat}$  and  $CI$  share the same inference rules  $lex_x$ . For example, we provide ‘A boy is jumping into the water  $\rightarrow$  A kid is jumping into the water’ in  $PI_{nat}$ ; and ‘Someone  $f_v/f_{nv}$  a boy is playing in the mud  $\rightarrow$  A kid is playing in the mud’ in  $CI$ . In this way, models can retain the knowledge of  $PI_{lex}$  by using the same inference rules, e.g., rule  $x : boy \rightarrow kid$ , while we inhibit shortcuts by using different contexts in the test set.

**Task2: compositions  $\rightarrow$  compositions** aims to evaluate if a model is able to predict unseen compositional inferences  $CI_{test}$  whose constituting primitives have been encountered in other compositional inferences  $CI_{train}$  in training. *Train and Test sets* ( $D_{train|test}$ ) consist of instances  $e$  and  $\tilde{e}$ :

$$\begin{aligned} D_{train} &= \{e \mid e \in CI_{train}\} \\ D_{test} &= \{\tilde{e} \mid \tilde{e} \in CI_{test}\} \end{aligned} \quad (3)$$

We construct specific types of compositional training instances by combining *veridical* inference with *lexical* natural inference, and *non-veridical* with *structural* natural inference, see (4). To evaluate if models can generalize to novel compositions, we switch the constituents (primitive inference types) seen in training to unseen compositional inferences in testing. I.e., we evaluate *veridical* inference with *structural* natural inference, and *non-veridical* inference with *lexical* natural infer-

ence.  $CI_{train}$  and  $CI_{test}$  are specified as:

$$\begin{aligned} CI_{train} &= \{f_v(s) \xrightarrow{lex^+} s', f_v(s) \xrightarrow{lex^-} s', \\ &\quad f_{nv}(s) \xrightarrow{stru^+} s', f_{nv}(s) \xrightarrow{stru^-} s'\} \\ CI_{test} &= \{f_v(s) \xrightarrow{stru^+} s', f_v(s) \xrightarrow{stru^-} s', \\ &\quad f_{nv}(s) \xrightarrow{lex^+} s', f_{nv}(s) \xrightarrow{lex^-} s'\} \end{aligned} \quad (4)$$

This is a **trivial** setting, given that four composition rules (①②③④ in Table 3) have been instantiated in the training samples. The challenge is restricted to correctly classifying novel compositions from known primitives.

To further explore if models can generalize to novel compositions based on unseen composition rules we propose a **non-trivial** setting. Here, a model must combine *entailed veridical* inference with *entailed natural* inference, and *non-veridical* inference with *non-entailed natural* inference. With this, only rules ① and ④ are instantiated by the training samples. In testing we confront the model with composition instances unseen in training, by switching constituents, so that we test for the unseen rules ② and ③: we compose *entailed veridical* with *non-entailed natural* inference, and *non-veridical* with *entailed natural* inference.  $CI_{train}$  and  $CI_{test}$  are defined as:

$$\begin{aligned} CI_{train} &= \{f_v(s) \xrightarrow{nat^+} s', f_{nv}(s) \xrightarrow{nat^-} s'\} \\ CI_{test} &= \{f_v(s) \xrightarrow{nat^-} s', f_{nv}(s) \xrightarrow{nat^+} s'\} \end{aligned} \quad (5)$$

We expected this to be an intractable challenge, since models are now required to classify novel compositions, where identical primitives have been encountered in training compositions, but the required composition rules of tested compositions are not instantiated in the training data.

**Task3: Primitives and Compositions  $\rightarrow$  Compositions** aims to evaluate whether a model is

able to predict an unseen compositional inference  $CI_{test}$  whose one primitive inference  $PI$  has been learned independently, while the other has only been encountered in a compositional inference  $CI_{train}$  in training. Hence, *Train and Test sets* ( $D_{train|test}$ ) consist of instances  $e$  and  $\tilde{e}$ :

$$\begin{aligned} D_{train} &= \{e \mid e \in PI \vee e \in CI_{train}\} \\ D_{test} &= \{\tilde{e} \mid \tilde{e} \in CI_{test}\} \end{aligned} \quad (6)$$

We could choose either veridical or natural inference as a primitive inference  $PI$ . Here, we select *natural inference* as the  $PI$  (veridical inference works analogously). Specifically, we construct  $CI_{train}$  by combining *entailed veridical* inference with *lexically-based* natural inference, and define *structure-based* natural inference as  $PI$ . To evaluate if models can generalize to novel compositional inference, we substitute the lexically-based natural inference component in  $CI_{train}$  with structure-based natural inference to form  $CI_{test}$  instances, as stated below:

$$\begin{aligned} PI &= PI_{stru} = \{s \xrightarrow{stru} s', s \xrightarrow{\cancel{stru}} s'\} \\ CI_{train} &= \{f_v(s) \xrightarrow{lex^+} s', f_v(s) \xrightarrow{\cancel{lex^-}} s'\} \\ CI_{test} &= \{f_v(s) \xrightarrow{stru^+} s', f_v(s) \xrightarrow{\cancel{stru^-}} s'\} \end{aligned} \quad (7)$$

This is again a **trivial** setting, given that the composition rules (①②) required in testing have been exemplified by training samples. That is, the challenge is restricted to correctly classifying novel compositions, where their primitives and the composition rules are known.

Analogous to Task2, we introduce a further **non-trivial** setting to evaluate if models can generalize to novel compositions that test for unseen composition rules. The primitive inference  $PI$  could be either veridical or natural inference.

Hence in one variant, we choose i) *veridical* inference ( $-R_{ver}$ ) as the  $PI$ , and construct the training compositions by combining *entailed veridical* with *entailed lexical* natural inference, while the primitive inference is *non-veridical* inference. For testing, we replace the veridical inference in training compositions with independent non-veridical inferences. This setting is defined below:

$$\begin{aligned} PI &= \{f_{nv}(s) \dashrightarrow s'\} \\ CI_{train} &= \{f_v(s) \xrightarrow{lex^+} s'\} \\ CI_{test} &= \{f_{nv}(s) \xrightarrow{\cancel{lex^+}} s'\} \end{aligned} \quad (8)$$

This setting should be challenging, since models are required to evaluate novel compositions that

Tasks	Composition Generalization			In-distribution				
	type	Train	Dev	Test	Train	Dev	Test	
Task1	trivial	$PI_{ver}$	1680	420	63240	3366	842	63240
		$PI_{nat}$	1686	422				
$\neg$ trivial		$PI_{ver}$	600	150	22620	1203	301	22620
		$PI_{nat}$	603	151				
Task2	trivial	$CI$	25296	6324	31620	25296	6324	31620
	$\neg$ trivial	$CI$	25296	6324	31620	25296	6324	31620
	trivial	$PI_{nat}$	480	120	9000	960	240	9000
		$CI$	480	120				
Task3	$\neg$ trivial	$PI_{ver}$	9048	2262	11310	18096	4524	11310
	$-R_{ver}$	$CI$	9048	2262				
	$\neg$ trivial	$PI_{nat}$	600	150	11310	1203	301	11310
	$-R_{nat}$	$CI$	603	151				

Table 5: Statistics of *compositional generalization controlled* data.  $PI$  and  $CI$  indicate primitive and compositional inferences, respectively. “type” marks the inference types used in train and dev sets.

correspond to the compositional rule ③, while they have only encountered rule ① in training.

As alternative variant ii), we choose *natural inference* ( $-R_{nat}$ ) as the primitive inference  $PI$ . We construct training data for compositions by combining *entailed veridical* with *entailed lexical* natural inference, and define *non-entailed lexical* natural inference as the primitive inference. For testing, we replace the entailed lexical inference in training compositions with independent non-entailed lexical inferences. This is defined below:

$$\begin{aligned} PI &= \{s \xrightarrow{lex} s'\} \\ CI_{train} &= \{f_v(s) \xrightarrow{lex^+} s'\} \\ CI_{test} &= \{f_v(s) \xrightarrow{\cancel{lex^-}} s'\} \end{aligned} \quad (9)$$

This setting is challenging, since models are required to evaluate novel compositions according to rule ②, while having only seen rule ① in training.

## 4 Experimental Setup

### 4.1 Dataset

To evaluate the systematicity capabilities of PLMs on the series of SETI tasks we established above, we construct controlled datasets with instances chosen from established NLI datasets. For primitive inference: 1) *veridical inference*, we select 30 verbs (15 veridical, 15 non-veridical) that appear in both the MegaVeridicality2 (White et al., 2018) and the verb veridicality dataset of Ross and Pavlick (2019), as Yanaka et al. (2021) do (cf. Appendix.A for details). 2) *natural inference*, we extract instances from the SICK dataset (Marelli et al., 2014) that use lexical inferences  $s \xrightarrow{lex} s'$  where sentence pairs

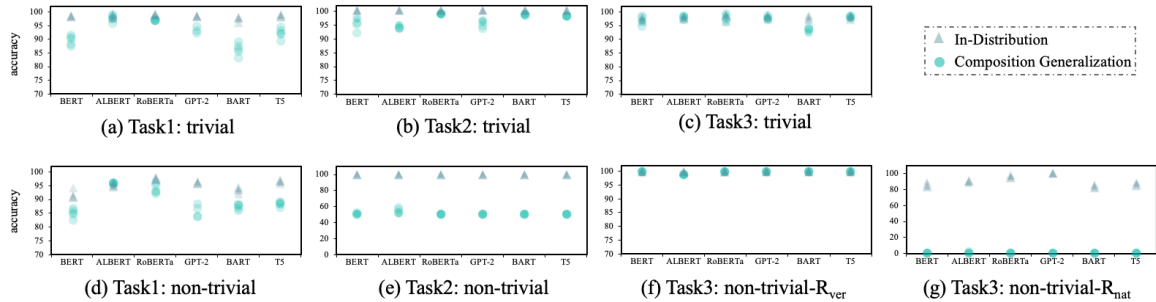


Figure 1: Performance of six PLMs on the SETI benchmark in two configurations: “Compositional Generalization” and “In-Distribution”. For each task setting and PLM, we perform five runs and represent each result by a symbol.

are formed from lexical relations, e.g., synonymy and hyponymy. In addition, we select structural inferences  $s \xrightarrow{stru} s'$  where sentence pairs are derived from each other using the active-passive diathesis. Examples are shown in Table 2. For compositional inferences, we construct instances following §2.1. We combine premises  $f_{v/nv}(s)$  from veridical inferences with hypotheses  $s'$  from natural inference. Boolean logic rules are used to assign labels for these compositional inference instances.

Based on the constructed pool of inference data, we design three **Compositional Generalization** task datasets to evaluate the systematicity of PLMs. Specifically, primitive and compositional inferences data is divided for training  $D_{train}$  and testing  $D_{test}$  in a controlled way, as outlined in Section §3. This ensures that the evaluated models will be exposed to specific types of inference instances in training, while being evaluated on unseen compositional inferences. That is to say, *the testing data is out of distribution from the training data*. In addition, we provide corresponding **In-Distribution** task datasets for comparison. Here, the data is divided into  $D'_{train}$  and  $D'_{test}$  by producing random splits from  $D = D_{train} \cup D_{test}$ . Hence, the evaluated models will, during training, encounter instances of the kind that will be presented in testing. In other words, *the testing data is In-distribution of the training data*. *In-Distribution* data makes it possible to confirm whether the failure of *Compositional Generalization* is due to intractable compositional inference tests or a lack of systematicity. Table 5 shows detailed data statistics for both configurations. For further details see Appendix B.

## 4.2 Evaluated Models

We choose six well-known PLMs for evaluation, of which three are masked language models (*encoder-only*): BERT (Devlin et al., 2019), RoBERTa (Liu

et al., 2019) and ALBERT (Lan et al., 2020); two are denoising autoencoder models (*encoder-decoder*): T5 (Raffel et al., 2020) and BART (Lewis et al., 2020); and one auto-regressive model (*decoder-only*): GPT-2 (Radford et al., 2019). We use standard accuracy as the evaluation metric.

For all PLMs we have chosen *Large* models, with checkpoints from the Hugging Face implementation (Wolf et al., 2020)<sup>4</sup>. We finetuned these models using the Adam Optimizer with batch size of 16. The maximum input token number is limited to 128. For each of the seven task settings, we perform five runs for each PLM, using different seeds. Further details are provided in Appendix C.

## 5 Experiments

### 5.1 Overview Results

Fig.1 illustrates the performance of six well-known PLMs on the SETI benchmark across two data configurations: *Compositional Generalization* and *In-Distribution*. Among seven different task settings, we find the test accuracy of PLMs in *In-Distribution* to be close to 100% in most cases, with a drop to  $\geq 80\%$  in Task1 and Task2, non-trivial, but always stable across five rounds. This indicates that compositional inferences of various types are feasible for the evaluated PLMs if they have seen relevant instances in training. *Compositional Generalization* shows comparable results in trivial settings of Task2 (Fig.1.b) and Task3 (Fig.1.c), but inferior results for most of the remaining settings. This suggests that the evaluated PLMs are lacking systematicity capabilities when encountering unseen compositional inference problems, while achieving remarkable performance in *In-Distribution* by fitting training data.

<sup>4</sup><https://huggingface.co/models>

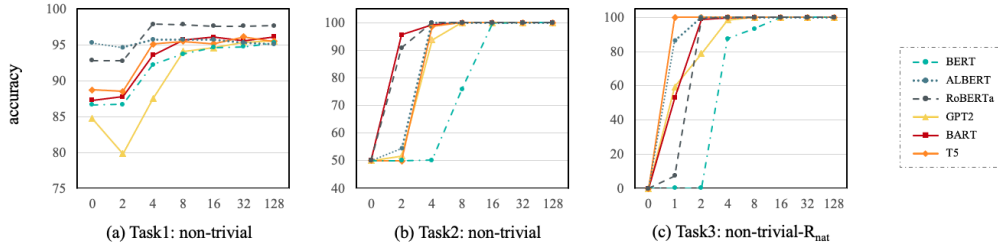


Figure 2: Few-shot performance of six well-known PLMs on three challenging sub-tasks of the SETI benchmark.

Comparing *trivial* and *non-trivial* settings across three tasks, we find: 1) In **Task1**, the test accuracy of *In-Distribution* slightly decreases in the non-trivial setting, which confronts the models with novel contexts for  $PI_{nat}$  inferences within the compositional test cases  $CI$ . This shows that the *non-trivial* setting is more challenging. We also find that the performance of *Compositional Generalization* drops in the non-trivial setting. However, the encoder-only models ALBERT and RoBERTa outperform others substantially, showing strong systematicity generalization ability in both settings. 2) In **Task2**, the test accuracy of all generalization-tested models declines sharply to 50% in the non-trivial setting, no matter how well a model performs in the trivial setting. And this finding holds across different rounds of each PLM, indicating that novel compositional inferences are equally challenging for all evaluated models. 3) In **Task3**, generalization-tested models also show inferior results in the non-trivial setting, while non-trivial- $R_{ver}$  (Fig.1.f) is an exception (100% accuracy). This is expected since in this setting the PLMs can solve unseen compositional problems by exploiting superficial characteristics during training, rather than by generalization, i.e., the capability of systematicity. Specifically, the non-trivial- $R_{ver}$  task evaluates  $f_{nv}(s) \xrightarrow{\text{lex}^+} s'$  given  $f_v(s) \xrightarrow{\text{lex}^+} s'$  and  $f_{nv}(s) \rightarrow s'$ . In this task setting, non-veridical verbs  $f_{nv}$  are only seen in non-veridical inference, which may lure models to predict non-entailment for compositional inferences containing non-veridical verbs, yet without considering the entailment class of the embedded lexical inference.

Across the different tasks, the evaluated PLMs show diverse performance for generalization testing. **Task1** is almost solved by ALBERT and RoBERTa, highlighting that some models are capable of combining different primitive inferences (learned independently) in unseen compositional inferences. However, along with all other PLMs,

none of the two remaining Tasks can be reliably solved in the controlled, non-trivial “Compositional Generalization” setting: i) predicting unseen compositions, the components of which have been learned during training (**Task2**) and ii) determining a novel composition, where one primitive is learned independently, while the other has been encountered in a composition during training (**Task3**).

## 5.2 Few-shot Evaluation

We conclude from the results shown in Fig 1 that the evaluated PLMs are incapable of performing compositional generalization if they have not encountered crucial compositional inference knowledge during training. Hence, we aim to explore whether, and to what extent we could enhance the systematicity capabilities of the evaluated PLMs, by exposing them to small doses of relevant instances. Specifically, we select three *non-trivial* sub-tasks that expect models to solve compositional inferences without encountering the required inferential knowledge in training. For each such task, we construct a few-shot dataset  $D_{few}$  where each sample (compositional inference, CI) is constructed following §4.1.  $D_{few}$  and  $D_{test}$  contain different data, i.e.,  $D_{test} \cap D_{few} = \emptyset$ . For each task, we evaluate few shot samples from 0 to 128, and each model is fine-tuned for three epochs. By doing so, we expect the models to learn the underlying compositional inference knowledge from the samples given in  $D_{few}$ , so they can finally solve  $D_{test}$ .

Figure 2 shows the few-shot experiment results. Across different tasks, we find that all evaluated PLMs benefit from few-shot samples that teach the model relevant compositional inference knowledge. In **Task1**, most PLMs show a significant performance increase with only four CI samples in  $D_{few}$ . This finding is consistent with the fact that solving  $D_{test}$  requires four different compositional rule types, as shown in definition Table 3. Similarly,  $D_{test}$  from **Task2** and **Task3** require two and one



samples illustrating required, but previously unseen compositional inference knowledge, respectively. We find most evaluated PLMs in Fig 2.b to drastically improve their performance with only two samples, and in Fig.2.c with just a single sample. An exception is BERT, which requires more shots than the number of unseen inference cases.

The above experiment suggests that the evaluated PLMs can greatly benefit from few-shot settings to enhance their systematicity capabilities. It is compelling that the number of samples in  $D_{few}$  needed to reach substantial task performance corresponds to the number of inference knowledge types required to make correct inference predictions, i.e., making it possible to evaluate novel compositions. It will be interesting to study how to identify potentially missing types of compositional inference knowledge for existing PLMs, and how to inject this knowledge in an efficient, data-free method.

## 6 Conclusion

We propose the first *comprehensive* systematicity evaluation benchmark, SETI, applied to Natural Language Inference. Experiments on six widely used PLMs show that they can distinguish novel compositions with known primitives and composing knowledge with high accuracy, but limited when lacking such knowledge. Moreover, we show that models can quickly acquire missing inferential knowledge for systematicity by being presented with *unique* samples representing each missing case of inferential knowledge, in a few-shot setup.

## 7 Limitation

SETI only considers veridical inference and natural inference (including both lexically-based inference and structure-based inference). However, our benchmark SETI can be flexibly extended to more varied reasoning patterns, such as negation, quantifiers, or others. In addition, we evaluate the systematicity capabilities of PLMs on semi-synthetic datasets, which are limited in language variance. Extending our benchmark on manually annotated compositional inference datasets might be a promising future work.

Recently, Hupkes et al. (2020) dissect the notion of compositionality and define five theoretically grounded tests for generalization, in a task-agonistic manner. Our work is limited to evaluating the systematicity of PLMs in textual inference. While the systematicity test is one of the most im-

portant tests, the remaining ones (e.g., *productivity* and *localism*) are still worth to be explored in future works.

## 8 Acknowledgments

We are grateful to three anonymous reviewers for their valuable comments that have helped to improve this paper. This work has been supported through a scholarship provided by the Heidelberg Institute for Theoretical Studies gGmbH.

## References

- Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Wen-tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *International Conference on Learning Representations*.
- 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*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.
- Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31.
- Tiffany Chien and Jugal Kalita. 2020. Adversarial analysis of natural language inference systems. In *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, pages 1–8. IEEE.
- Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies*, 6(4):1–220.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Catherine Finegan-Dollak, Jonathan K. Kummerfeld, Li Zhang, Karthik Ramanathan, Sesh Sadasivam, Rui Zhang, and Dragomir Radev. 2018. [Improving text-to-SQL evaluation methodology](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 351–360, Melbourne, Australia. Association for Computational Linguistics.

- Jerry A Fodor and Zenon W Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.
- 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, Online. Association for Computational Linguistics.
- 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 (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.
- 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*, pages 1958–1969, Online. Association for Computational Linguistics.
- Reto Gubelmann, Christina Niklaus, and Siegfried Handschuh. 2022. [A philosophically-informed contribution to the generalization problem of neural natural language inference: Shallow heuristics, bias, and the varieties of inference](#). In *Proceedings of the 3rd Natural Logic Meets Machine Learning Workshop (NALOMA III)*, pages 38–50, Galway, Ireland. Association for Computational Linguistics.
- Diewu Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795.
- Lauri Karttunen. 1971. Implicative verbs. *Language*, pages 340–358.
- Lauri Karttunen and Stanley Peters. 1979. Conventional implicature. In *Presupposition*, pages 1–56. Brill.
- Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, et al. 2020. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*.
- 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, Online. Association for Computational Linguistics.
- Yuta Koreeda and Christopher Manning. 2021. [ContractNLI: A dataset for document-level natural language inference for contracts](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1907–1919, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pages 2873–2882. PMLR.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Yong-Lu Li, Yue Xu, Xiaohan Mao, and Cewu Lu. 2020. Symmetry and group in attribute-object compositions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11316–11325.
- Linqing Liu, Patrick Lewis, Sebastian Riedel, and Pontus Stenetorp. 2022. [Challenges in generalization in open domain question answering](#). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2014–2029, Seattle, United States. Association for Computational Linguistics.
- Tianyu Liu, Zheng Xin, Baobao Chang, and Zhifang Sui. 2020. [HypoNLI: Exploring the artificial patterns of hypothesis-only bias in natural language inference](#). In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6852–6860, Marseille, France. European Language Resources Association.
- 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)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).
- 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*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

- Ishan Misra, Abhinav Gupta, and Martial Hebert. 2017. From red wine to red tomato: Composition with context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1792–1801.
- Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. [Adversarial NLI: A new benchmark for natural language understanding](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.
- Mitja Nikolaus, Mostafa Abdou, Matthew Lamm, Rahul Aralikkatte, and Desmond Elliott. 2019. [Compositional generalization in image captioning](#). In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 87–98, Hong Kong, China. Association for Computational Linguistics.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.
- 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, Hong Kong, China. Association for Computational Linguistics.
- Ivan Sanchez, Jeff Mitchell, and Sebastian Riedel. 2018. [Behavior analysis of NLI models: Uncovering the influence of three factors on robustness](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1975–1985, New Orleans, Louisiana. Association for Computational Linguistics.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.
- 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*, pages 4717–4724, Brussels, Belgium. Association for Computational Linguistics.
- 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, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- 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*, pages 6105–6117, Online. Association for Computational Linguistics.
- Hitomi Yanaka, Koji Mineshima, and Kentaro Inui. 2021. [Exploring transitivity in neural NLI models through veridicality](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 920–934, Online. Association for Computational Linguistics.
- Hao Zheng and Mirella Lapata. 2022. [Disentangled sequence to sequence learning for compositional generalization](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4256–4268, Dublin, Ireland. Association for Computational Linguistics.
- Xiang Zhou and Mohit Bansal. 2020. [Towards robustifying NLI models against lexical dataset biases](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8759–8771, Online. Association for Computational Linguistics.

## A Veridical Inference

In order to construct *veridical inference*, we select 30 verbs, including 15 veridical verbs  $f_v$  and 15 non-veridical verbs  $f_{nv}$ . Table 6 show instantiation of selected verbs.

Verb Types	Instantiations
veridical verbs $f_v$	realize, acknowledge, remember, note, find, notice, learn, see, reveal, discover, understand, know, admit, recognize, observe
non-veridical verbs $f_{nv}$	feel, claim, doubt, hope, predict, imply, suspect, wish, think, believe, hear, expect, estimate, assume, argue

Table 6: Instantiation of veridical and non-veridical verbs used for constructing veridical inference.

## B Data Statics

Since we use 30 verbs to construct premises  $f_{v/nv}(s)$  for primitive veridical ( $PI_{ver}$ ) and compositional ( $CI$ ) inferences from the premises ( $s$ ) of natural inferences ( $PI_{nat}$ ), the number of these two inference types is 30 times the amount of  $PI_{nat}$ , respectively. To avoid data biases in composition training, we guarantee the two major types from  $D_{train}$  are balanced by downsampling the extensive inference type. For example, in Task1 trivial setting, we downsample  $PI_{ver}$  to ensure the training data of  $PI_{ver}$  and  $PI_{nat}$  is balanced.

## C Evaluated Pre-train Language Models

We evaluate SETI across six well-known PLMs. Table 7 shows the training objective and parameters of each model. Detailed information and training parameters of each model is:

**BERT** (Devlin et al., 2019) is a bidirectional transformer pre-trained model, trained with masked language modeling and next sentence prediction objectives on a large corpus. We fine-tuned the base-uncased-large version, with the default setting.

**ALBERT** (Lan et al., 2020) build on BERT, and presents two parameter-reduction techniques to lower memory consumption and increase the training speed. We fine-tuned the ALBERT-large version, with the default setting.

**RoBERTa** (Liu et al., 2019) builds on BERT, but is trained without the next-sentence prediction objective and uses much larger data. We fine-tuned a RoBERTa-large version, with the default setting.

**GPT-2** (Radford et al., 2019) is a decoder-only model, pre-trained on a large corpus of English

data in a self-supervised fashion. We fine-tuned the GPT2-large version, with the default setting.

**BART** (Lewis et al., 2020) is an encoder-decoder model. The pretraining task involves randomly shuffling the order of the original sentences and a novel in-filling scheme, where spans of text are replaced with a single mask token. We fine-tuned the BART-large version, with the default setting.

**T5** (Raffel et al., 2020) is an encoder-decoder model which is pre-trained on a multi-task mixture of unsupervised and supervised tasks. Each task is in a text-to-text format. We fine-tuned the T5-large version, with the default setting.

Model	Objective	Parameters	Layers	Type
BERT	MLM+NSP	340M	24	
ALBERT	MLM+SOP	17M	24	Enc
RoBERTa	MLM	355M	24	
GPT-2	LM	774M	36	Dec
BART	DAE	406M	24	
T5	DAE	770M	24	Enc-Dec

Table 7: Overview of PLMs evaluated for systematicity in our work. For training objectives, *MLM* is masked language modeling, *NSP* is next sentence prediction objective, *SOP* is sentence order prediction, *LM* is language modele, and *DAE* is the denoising autoencoder



## ACL 2023 Responsible NLP Checklist

---

### A For every submission:

- A1. Did you describe the limitations of your work?  
*Section 7*
- A2. Did you discuss any potential risks of your work?  
*We do not find any risk in our work so far. We use an open-sourced NLI dataset.*
- A3. Do the abstract and introduction summarize the paper’s main claims?  
*In the abstract and introduction sections.*
- A4. Have you used AI writing assistants when working on this paper?  
*Left blank.*

### B Did you use or create scientific artifacts?

*Section 4.1*

- B1. Did you cite the creators of artifacts you used?  
*Section 4.1*
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?  
*The datasets used in this paper are open-sourced.*
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?  
*Both artifacts are used for research purposes.*
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?  
*The SICK dataset does not contain information about names or uniquely identifies individual people or offensive content.*
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?  
*Section 4.1*
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.  
*Section 4.1*

### C Did you run computational experiments?

*Appendix C*

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?  
*Appendix C*

---

*The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.*

- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?

*Appendix C, Section 4.2*

- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

*Section 4.2, Section 5.1*

- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

*Left blank.*

**D  Did you use human annotators (e.g., crowdworkers) or research with human participants?**

*Left blank.*

- D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

*No response.*

- D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?

*No response.*

- D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

*No response.*

- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

*No response.*

- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

*No response.*