Advanced behavioral evaluation of NLU models

Christopher Potts

Stanford Linguistics

CS224u: Natural language understanding

Overview

Varieties of evaluation

Behavioral

- Standard (“IID”; Independent and Identically Distributed)
- Exploratory
- Hypothesis-driven
- Challenge
- Adversarial
- Security-oriented

Structural

- Probing
- Feature attribution
- Interventions

Standard evaluations

1. Create a dataset from a single process.
2. Divide the dataset into disjoint train and test sets, and set the test set aside.
3. Develop a system on the train set.
4. Only after all development is complete, evaluate the system based on accuracy on the test set.
5. Report the results as providing an estimate of the system's capacity to generalize.

Adversarial evaluations

1. Create a dataset by whatever means you like.
2. Develop and assess the system using that dataset, according to whatever protocols you choose.
3. Develop a new test dataset of examples that you suspect or know will be challenging given your system and the original dataset.
4. Only after all system development is complete, evaluate the system based on accuracy on the new test dataset.
5. Report the results as providing an estimate of the system's capacity to generalize.

A bit of history

VOL. LIX. No. 236.] [October, 1950

MIND

A QUARTERLY REVIEW

OF

PSYCHOLOGY AND PHILOSOPHY

I.—COMPUTING MACHINERY AND INTELLIGENCE

BY A. M. TURING

A bit of history

VOL. LIX. No. 236.] [October, 1950

MIND

A QUARTERLY REVIEW

OF

PSYCH

I.—C

COGNITIVE PSYCHOLOGY 3, 1–191 (1972)

Understanding Natural Language

TERRY WINOGRAD[1]

Massachusetts Institute of Technology
Cambridge, Massachusetts

A bit of history

VOL. LIX. No. 236.] [October, 1950

MIND

A QUARTERLY REVIEW

OF

PSYCH

I.—C

COGNITIVE PSYCHOLOGY 3, 1–191 (1972)

Understanding Natural Language

TERRY

Massachusetts I

Cambridg

On our best behaviour

Hector J. Levesque

Dept. of Computer Science

University of Toronto

Toronto, Ontario

Canada M5S 3A6

hector@cs.toronto.edu

Winograd sentences

1. The trophy doesn't fit into the brown suitcase because it's too **small**. What is too small?
   The suitcase / The trophy
2. The trophy doesn't fit into the brown suitcase because it's too **large**. What is too large?
   The suitcase / **The trophy**
3. The council refused the demonstrators a permit because they **feared** violence. Who **feared** violence?
   The council / The demonstrators
4. The council refused the demonstrators a permit because they **advocated** violence. Who **advocated** violence?
   The council / **The demonstrators**

Winograd 1972; Levesque 2013

Levesque’s (2013) adversarial framing

Could a crocodile run a steeplechase?

“The intent here is clear. The question can be answered by thinking it through: a crocodile has short legs; the hedges in a steeplechase would be too tall for the crocodile to jump over; so no, a crocodile cannot run a steeplechase.”

Foiling cheap tricks

“Can we find questions where cheap tricks like this will not be sufficient to produce the desired behaviour? This unfortunately has no easy answer. The best we can do, perhaps, is to come up with a suite of multiple-choice questions carefully and then study the sorts of computer programs that might be able to answer them.”

Analytical considerations

Key questions

What can behavioral testing tell us?

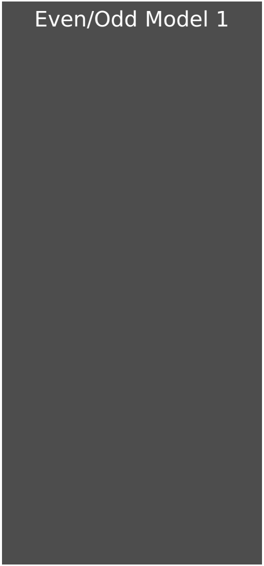
(And what can't it tell us?)

No need to be adversarial

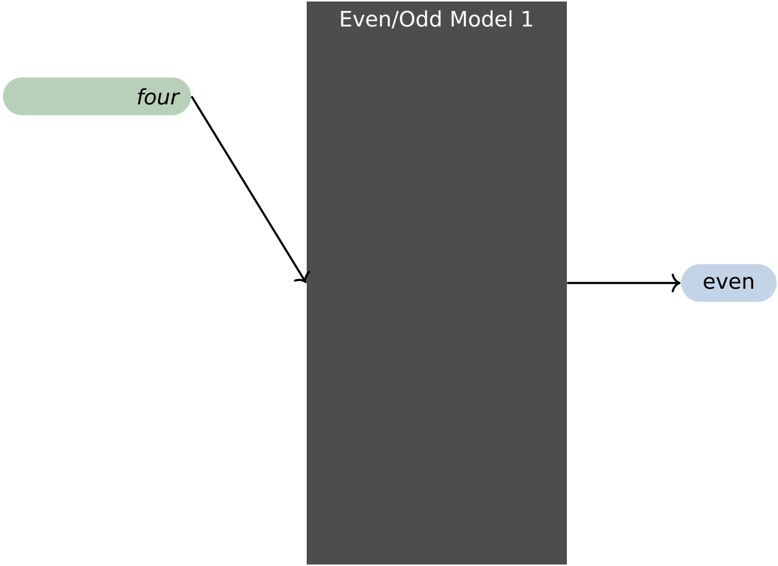
Here are some questions that start off exploratory and end up being adversarial:

1. Has my system learned anything about numerical terms?
2. Does my system understand how negation works?
3. Does my system work with a new style or genre?
4. This system is supposed to know about numerical terms, but here are some test cases that are outside of its training experiences for such terms. . .
5. When applied to invented genres, does my system produce socially problematic (e.g., stereotyped) outputs?
6. Are their patterns of random inputs that lead my system to produce problematic outputs?

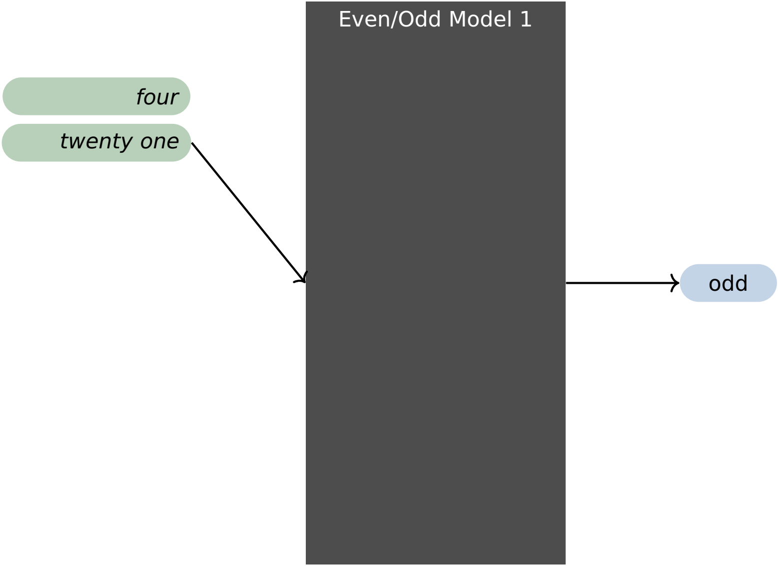
Limits of behavioral testing



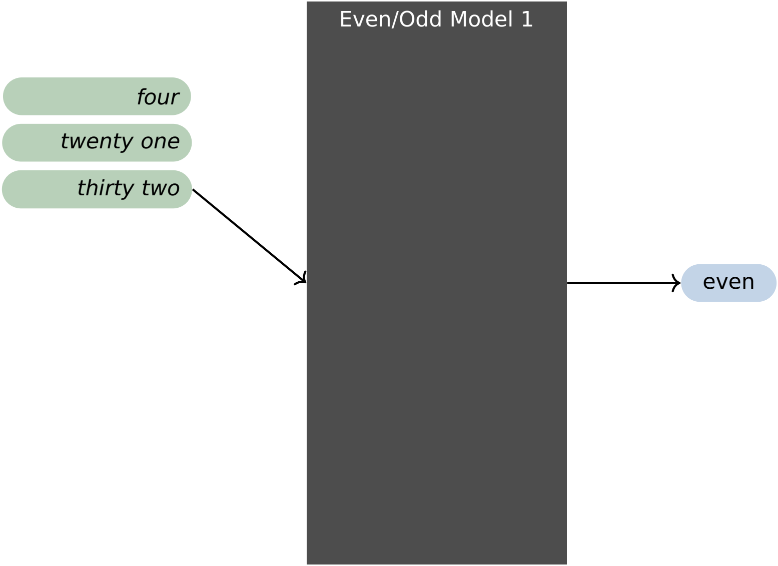
Limits of behavioral testing



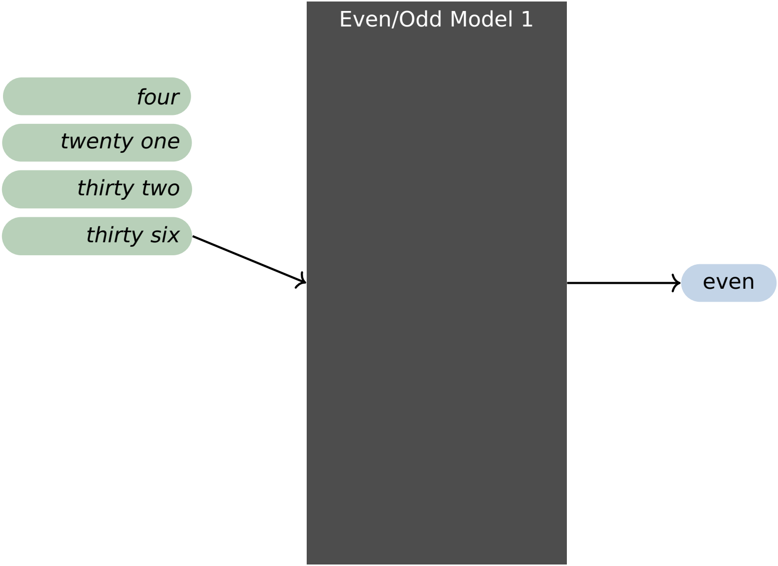
Limits of behavioral testing



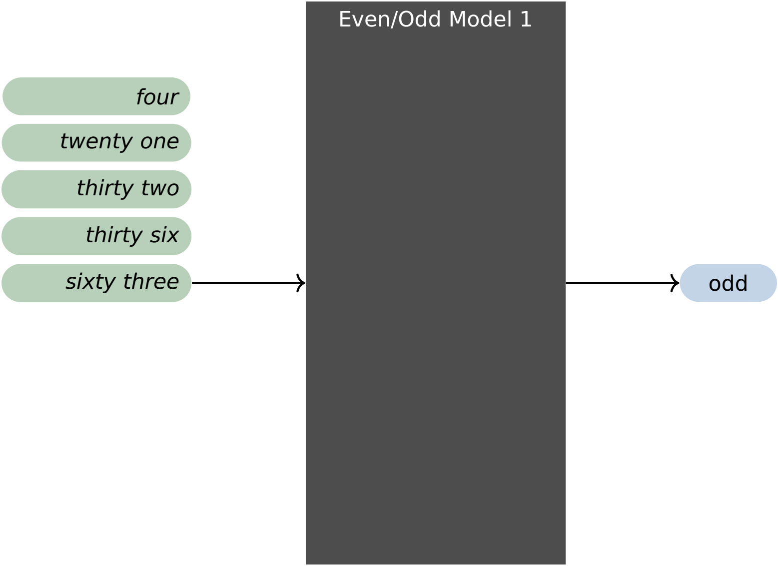
Limits of behavioral testing



Limits of behavioral testing



Limits of behavioral testing



Limits of behavioral testing

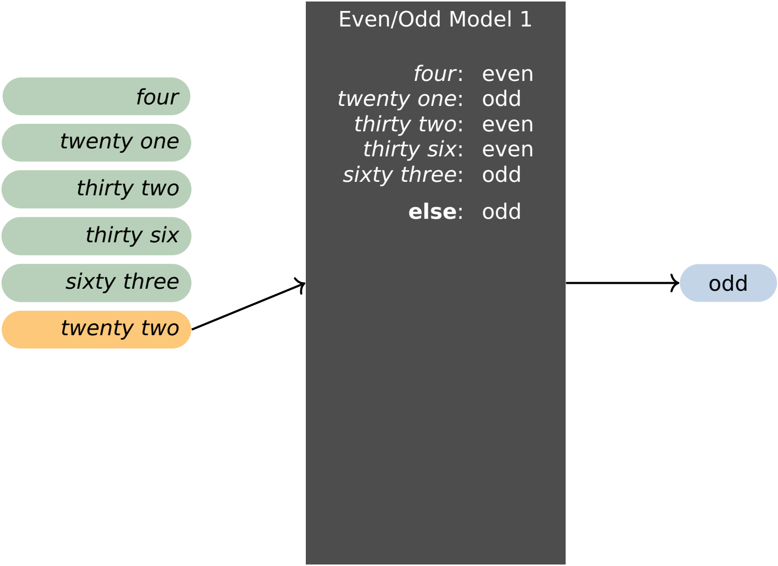
four
twenty one
thirty two
thirty six
sixty three →

Even/Odd Model 1

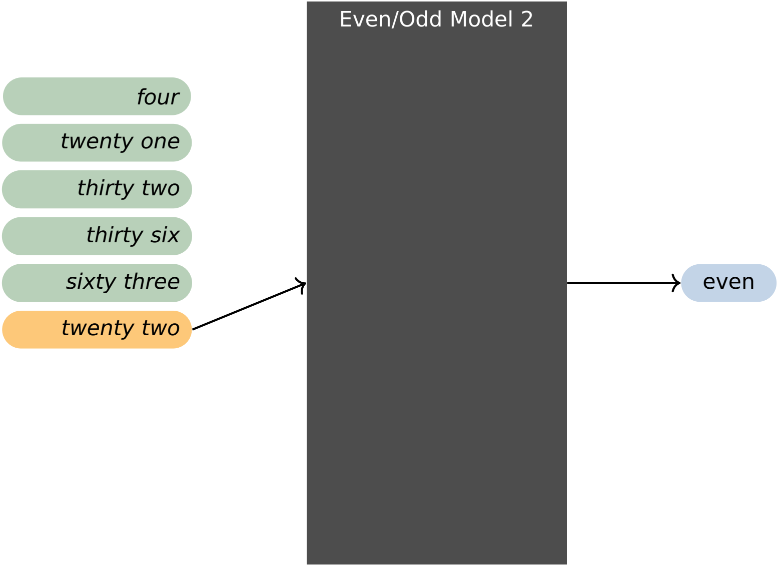
four:	even
twenty one:	odd
thirty two:	even
thirty six:	even
sixty three:	odd
else:	odd

→ odd

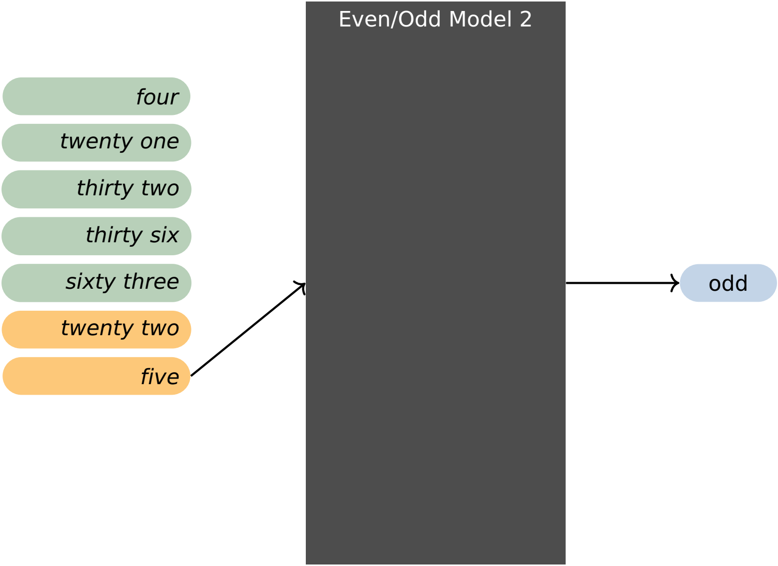
Limits of behavioral testing



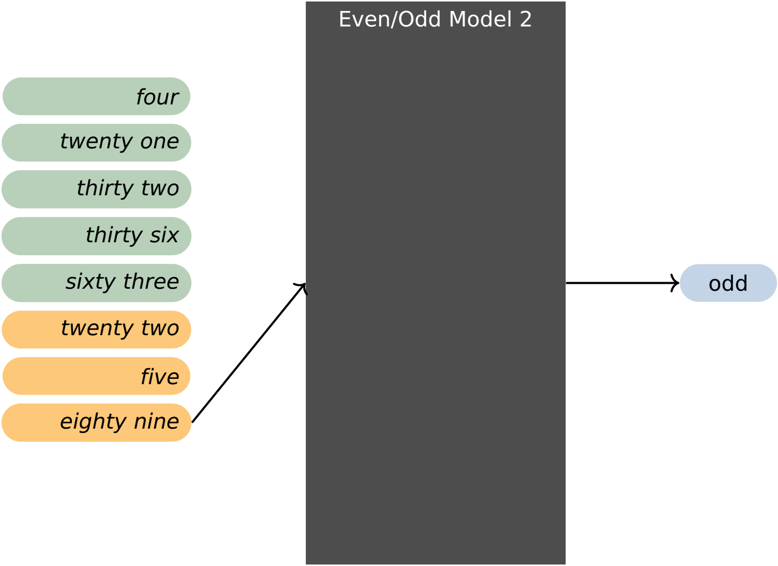
Limits of behavioral testing



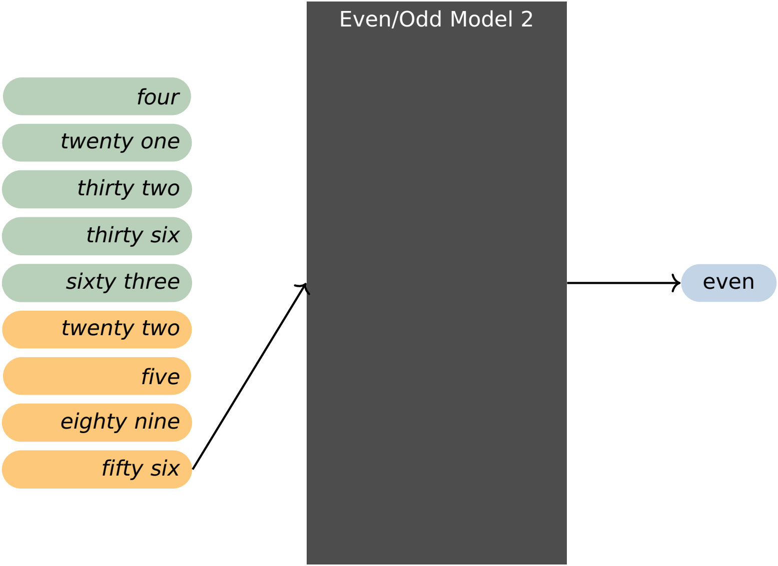
Limits of behavioral testing



Limits of behavioral testing



Limits of behavioral testing



Limits of behavioral testing

four
twenty one
thirty two
thirty six
sixty three
twenty two
five
eighty nine
fifty six

Even/Odd Model 2

$$d = \begin{bmatrix} \textit{one}: & \text{odd} \\ \textit{two}: & \text{even} \\ \textit{three}: & \text{odd} \\ \textit{four}: & \text{even} \\ \textit{five}: & \text{odd} \\ \textit{six}: & \text{even} \\ \textit{seven}: & \text{odd} \\ \textit{eight}: & \text{even} \\ \textit{nine}: & \text{odd} \\ \textbf{else}: & \text{odd} \end{bmatrix}$$

return
d[input final token]

even

Limits of behavioral testing

four
twenty one
thirty two
thirty six
sixty three
twenty two
five
eighty nine
fifty six
sixteen

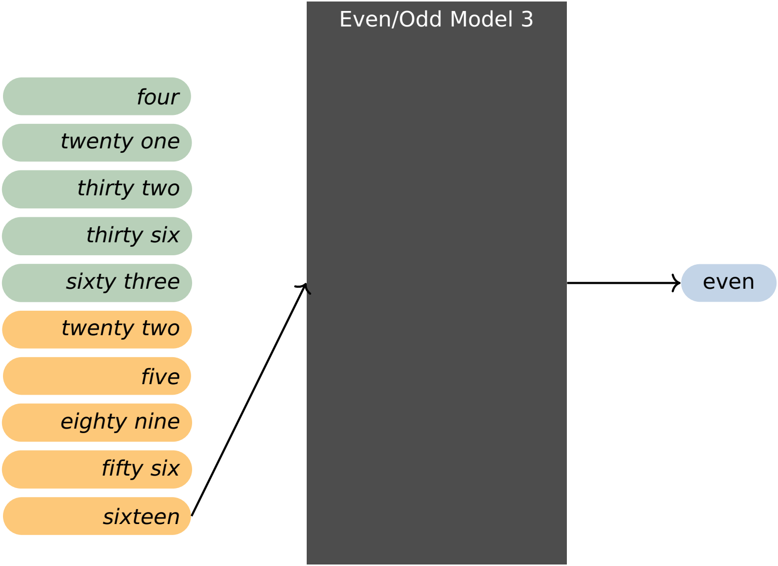
Even/Odd Model 2

$$d = \begin{bmatrix} \textit{one}: & \text{odd} \\ \textit{two}: & \text{even} \\ \textit{three}: & \text{odd} \\ \textit{four}: & \text{even} \\ \textit{five}: & \text{odd} \\ \textit{six}: & \text{even} \\ \textit{seven}: & \text{odd} \\ \textit{eight}: & \text{even} \\ \textit{nine}: & \text{odd} \\ \textbf{else}: & \text{odd} \end{bmatrix}$$

return
$d[\textit{input final token}]$

$\textit{sixteen} \rightarrow$ Model 2 $\rightarrow$ odd

Limits of behavioral testing



Metrics

The limitations of accuracy-based metrics are generally left unaddressed by the methods we will explore here, but these limitations should be brought in!

Model failing or dataset failing?

Liu et al. (2019)

“What should we conclude when a system fails on a challenge dataset? In some cases, a challenge might exploit blind spots in the design of the original dataset (*dataset weakness*). In others, the challenge might expose an inherent inability of a particular model family to handle certain natural language phenomena (*model weakness*). These are, of course, not mutually exclusive.”

Model failing or dataset failing?

Geiger et al. (2019)

However, for any evaluation method, we should ask whether it is fair. Has the model been shown data sufficient to support the kind of generalization we are asking of it? Unless we can say “yes” with complete certainty, we can’t be sure whether a failed evaluation traces to a model limitation or a data limitation that no model could overcome.

Model failing or dataset failing?

3 5 7 . . .

Model failing or dataset failing?

3 5 7 . . .

What number comes next?

Model failing or dataset failing?

p	q	
T	T	T
T	F	
F	T	T
F	F	

Model failing or dataset failing?

p	q	
T	T	T
T	F	
F	T	T
F	F	

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Inoculation by fine-tuning

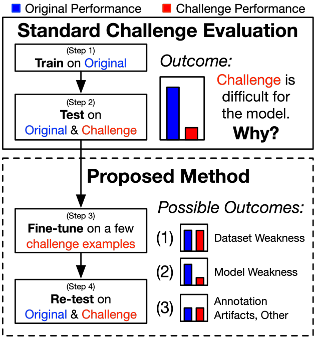
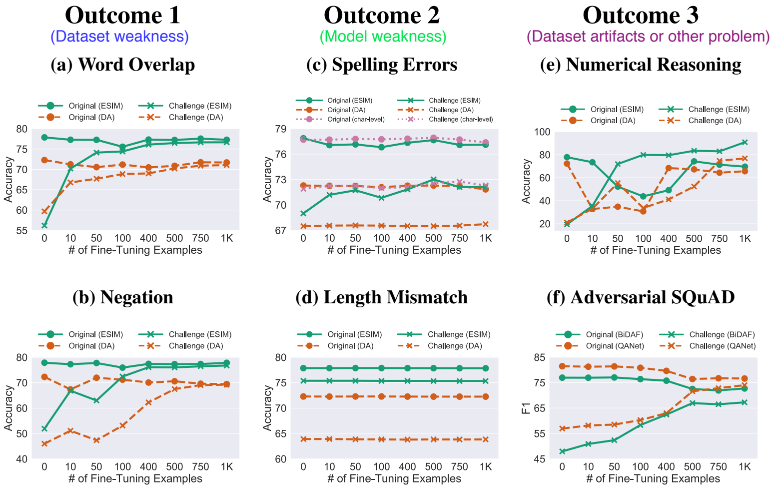


Figure 1: An illustration of the standard challenge evaluation procedure (e.g., Jia and Liang, 2017) and our proposed analysis method. "Original" refers to the a standard dataset (e.g., SQuAD) and "Challenge" refers to the challenge dataset (e.g., Adversarial SQuAD). Outcomes are discussed in Section 2.

Liu et al. 2019

Inoculation by fine-tuning



Liu et al. 2019

Negation as a learning target

Intuitive learning target

If *A entails B* then *not-B entails not-A*

Observation

Top-performing NLI models fail to achieve the learning target (Yanaka et al. 2019, 2020; Hossain et al. 2020; Geiger et al. 2020b).

Tempting conclusion

Top-performing models are incapable of learning negation.

Dataset observation

Negation is severely under-represented in NLI benchmarks.

MoNLI dataset construction

Positive MoNLI (PMoNLI; 1,476 examples)

SNLI hypothesis (A)	Food was served.
WordNet	pizza ⊏ food
New example (B)	Pizza was served.
Positive MoNLI	(A) **neutral** (B)
Positive MoNLI	(B) **entailment** (A)

Negative MoNLI (PMoNLI; 1,202 examples)

SNLI hypothesis (A)	The children are **not** holding plants.
WordNet	flowers ⊏ plants
New example (B)	The children are **not** holding flowers.
Negative MoNLI	(A) **entailment** (B)
Negative MoNLI	(B) **neutral** (A)

A systematic generalization task

NMoNLI Train		NMoNLI Test	
person	198	dog	88
instrument	100	building	64
food	94	ball	28
machine	60	car	12
woman	58	mammal	4
music	52	animal	4
tree	52		
boat	46		
fruit	42		
produce	40		
fish	40		
plant	38		
jewelry	36		
anything	34		
hat	20		
man	20		
horse	16		
gun	12		
adult	10		
shirt	8		
shoe	6		
store	6		
cake	4		
individual	4		
clothe	2		
weapon	2		
creature	2		

Our models know these lexical relations (high Positive MoNLI accuracy) and will be compelled to combine this knowledge with what they learn about negation during Negative MoNLI fine-tuning.

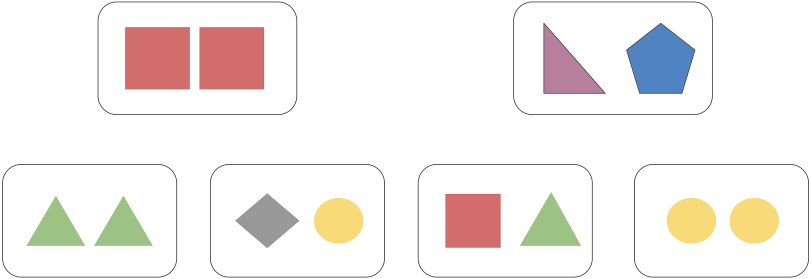
MoNLI as challenge dataset

Model	Input pretrain	NLI train data	No MoNLI fine-tuning SNLI	No MoNLI fine-tuning PMoNLI	No MoNLI fine-tuning NMoNLI	With NMoNLI fine-tuning SNLI	With NMoNLI fine-tuning NMoNLI
BiLSTM	GloVe	SNLI train	81.6	73.2	37.9	74.6	93.5
ESIM	GloVe	SNLI train	87.9	86.6	39.4	56.9	96.2
BERT	BERT	SNLI train	90.8	94.4	2.2	90.5	90.0

Diagnosis: Dataset failing!

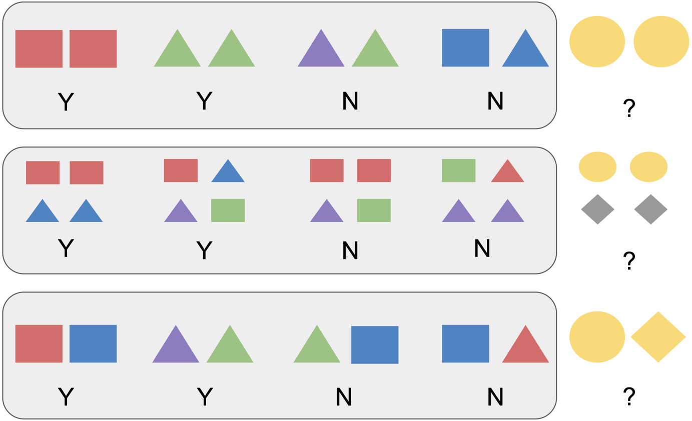
Geiger et al. 2020b

Reminder: Biological creatures are amazing



Premack 1983; Wasserman et al. 2017; Geiger et al. 2020a

Reminder: Biological creatures are amazing



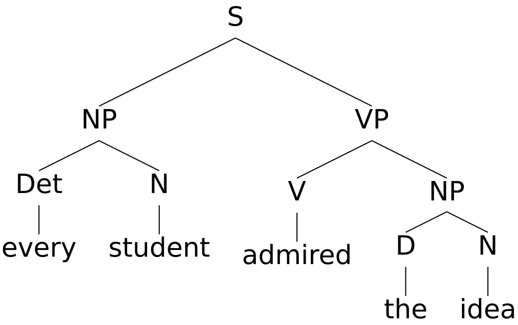
Premack 1983; Wasserman et al. 2017; Geiger et al. 2020a

Compositionality

Informal statement

Compositionality

The meaning of a phrase is a function of the meanings of its immediate syntactic constituents and the way they are combined.



The usual motivation

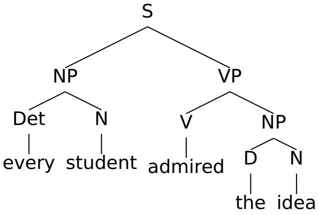
1. Modeling all meaningful units

   $[\![\textit{every}]\!] = \lambda f \; \lambda g \; \forall x \; ((f \; x) \rightarrow (g \; x))$

2. “Infinite” capacity

3. Creativity

4. Systematicity



Compositionality or systematicity?

Fodor and Pylyshyn (1988:37):

“What we mean when we say that linguistic capacities are *systematic* is that the ability to produce/understand some sentences is *intrinsically* connected to the ability to produce/understand certain others.”

1. Sandy loves the puppy.
2. The puppy loves Sandy.
3. the turtle $\sim$ the puppy
4. The turtle loves the puppy.
5. The puppy loves the turtle.
6. The turtle loves Sandy.
7. . . .

A worrisome lack of systematicity

Example		Gold	Prediction
The bakery	sells a mean apple pie.	pos	pos
They	sell a mean apple pie.	pos	pos
She	sells a mean apple pie.	pos	neg
He	sells a mean apple pie.	pos	neg

Compositionality by design

SHRDLU

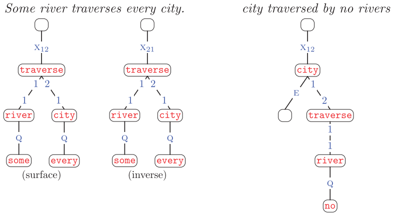
```
(THNOT
    (THPROG  (X2)     (THGOAL(#IS  $?X2  #PYRAMID))
                      (THGOAL(#SUPPORT  $?X1  $?X2)))))
                      "which supports no pyramids"
(THNOT
    (THPROG  (X2)     (THGOAL(#IS  $?X2  #PYRAMID))
                      (THNOT
                          (THGOAL(#SUPPORT  $?X1  $?X2)))))
                    "which supports every pyramid"
```

FIG. 52—Quantifiers.

Chat-80

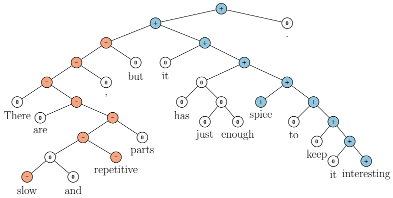
```
/* Sentences */
sentence(S) --> declarative(S), terminator(.) .
sentence(S) --> wh_question(S), terminator(?) .
sentence(S) --> yn_question(S), terminator(?) .
sentence(S) --> imperative(S), terminator(!) .

/* Noun Phrase */
np(np(Agmt,Pronoun,[]),Agmt,NPCase,def,_,Set,Nil) -->
  {is_pp(Set)},
  pers_pron(Pronoun,Agmt,Case),
  {empty(Nil), role(Case,decl,NPCase)}.

/* Prepositional Phrase */
pp(pp(Prep,Arg),Case,Set,Mask) -->
  prep(Prep),
  {prep_case(NPCase)},
  np(Arg,_,NPCase,_,Case,Set,Mask).
```

DCS

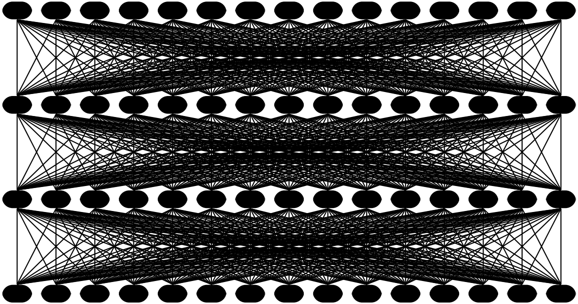


(c) Quantifier scope ambiguity (Q, Q) (d) Quantification (Q, E)

SST



No compositionality/systematicity guarantees!



Can we pose behavioral tests that will assess whether models like this have found systematicity solutions?

COGS and ReCOGS

COGS: A Compositional Generalization Challenge Based on Semantic Interpretation

Najoung Kim
Johns Hopkins University
n.kim@jhu.edu

Tal Linzen
New York University
linzen@nyu.edu

ReCOGS: How Incidental Details of a Logical Form Overshadow an Evaluation of Semantic Interpretation

Zhengxuan Wu **Christopher D. Manning** **Christopher Potts**
Stanford University
{wuzhengx, manning, cgpotts}@stanford.edu

Task

COGS

1. - Input: A rose was helped by a dog .
   - Output: `rose ( x _ 1 ) AND help . theme ( x _ 3 , x _ 1 ) AND help . agent ( x _ 3 , x _ 6 ) AND dog ( x _ 6 )`
2. - Input: The sailor dusted a boy .
   - Output: `* sailor ( x _ 1 ) ; dust . agent ( x _ 2 , x _ 1 ) AND dust . theme ( x _ 2 , x _ 4 ) AND boy ( x _ 4 )`

ReCOGS

1. - Input: A rose was helped by a dog .
   - Output: `rose ( 53 ) ; dog ( 38 ) ; help ( 7 ) AND theme ( 7 , 53 ) AND agent ( 7 , 38 )`
2. - Input: The sailor dusted a boy .
   - Output: `* sailor ( 48 ) ; boy ( 53 ) ; dust ( 10 ) AND agent ( 10 , 48 ) AND theme ( 10 , 53 )`

Motivations

1. Humans easily interpret novel combinations of familiar elements in ways that are systematic.
2. Compositionality is an explanation for this capability.
3. Can our best models generalize this way?
4. Have they too found compositional solutions?

The COGS and ReCOGS tasks are behavioral tests that seek to resolve 3, and the hope is that this can inform 4.

Understanding COGS logical forms

1. Verbs specify primitive events that have their own core conceptual structure and can involve one more more obligatory or optional roles.
   a. Emma broke a vase:
      `vase ( x _ 3 ) ; break . agent ( x _ 2 , Emma ) AND break . theme ( x _ 2 , x _ 3 )`
   b. The vase broke:
      `vase ( x _ 3 ) ; break . theme ( x _ 2 , x _ 1 )`

2. Variable numbering is determined by linear position in the input sentence.

3. All variables are bound; free variables are existentially bound with widest scope:
   a. `dog ( x _ 1 ) AND run . agent ( x _ 2 , x _ 1 )`
   b. ∃x _ 1 ∃x _ 2 `dog ( x _ 1 ) AND run . agent ( x _ 2 , x _ 1 )`

4. Definite descriptions are marked with *:
   a. The sailor ran.
   b. `* sailor ( x _ 1 ) ; run . agent ( x _ 2 , x _ 1 )`

COGS splits

1. Train: 24,000 examples plus 155 primitives
2. Dev: 10,000 examples
3. Test: 10,000 examples
4. Gen: 21,000 examples

Kim and Linzen 2020

Generalization categories

Case	Training	Generalization
S.3.1. Novel Combination of Familiar Primitives and Grammatical Roles		
Subject → Object (common noun)	A **hedgehog** ate the cake.	The baby liked the **hedgehog**.
Subject → Object (proper noun)	**Lina** gave the cake to Olivia.	A hero shortened **Lina**.
Object → Subject (common noun)	Henry liked a **cockroach**.	The **cockroach** ate the bat.
Object → Subject (proper noun)	The creature grew **Charlie**.	**Charlie** worshipped the cake.
Primitive noun → Subject (common noun)	**shark**	A **shark** examined the child.
Primitive noun → Subject (proper noun)	**Paula**	**Paula** sketched William.
Primitive noun → Object (common noun)	**shark**	A chief heard the **shark**.
Primitive noun → Object (proper noun)	**Paula**	The child helped **Paula**.
Primitive verb → Infinitival argument	**crawl**	A baby planned to **crawl**.
S.3.2. Novel Combination Modified Phrases and Grammatical Roles		
Object modification → Subject modification	Noah ate **the cake on the plate.**	**The cake on the table** burned.
S.3.3. Deeper Recursion		
Depth generalization: Sentential complements	Emma said **that** Noah knew **that** the cat danced.	Emma said **that** Noah knew **that** Lucas saw **that** the cat danced.
Depth generalization: PP modifiers	Ava saw the ball **in the bottle on the table**.	Ava saw the ball **in the bottle on the table on the floor**.
S.3.4. Verb Argument Structure Alternation		
Active → Passive	The crocodile **blessed** William.	A muffin **was blessed**.
Passive → Active	The book **was squeezed**.	The girl **squeezed** the strawberry.
Object-omitted transitive → Transitive	Emily **baked**.	The giraffe **baked a cake**.
Unaccusative → Transitive	The glass **shattered**.	Liam **shatterd** the jigsaw.
Double object dative → PP dative	The girl **teleported** Liam the cookie.	Benjamin **teleported** the cake **to** Isabella.
PP dative → Double Object Dative	Jane shipped the cake to John.	Jane shipped John the cake.
S.3.5. Verb Class		
Agent NP → Unaccusative subject	The **cobra** helped a dog.	The cobra **froze**.
Theme NP → Object-omitted transitive subject	The hippo **decomposed**.	The hippo **painted**.
Theme NP → Unergative subject	The hippo **decomposed**.	The hippo **giggled**.

Kim and Linzen 2020

Synthetic leaderboard

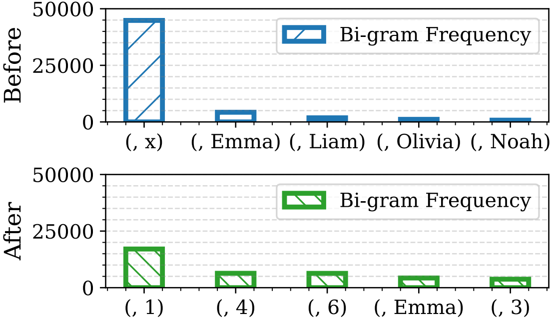
Model		STRUCT		LEX	Overall
	Obj PP → Subj PP	CP Recursion	PP Recursion		%
BART (Lewis et al. 2019)	0	0	12	91	79†
BART+syn (Lewis et al. 2019)	0	5	8	80	80†
T5 (Raffel et al. 2019)	0	0	9	97	83†
Kim and Linzen 2020	0	0	0	73	63
Ontanon et al. 2022	0	0	0	53	48
Akyurek and Andreas 2021	0	0	1	96	82
Conklin et al. 2021	0	0	0	88	75
Csordás et al. 2021	0	0	0	95	81
Zheng and Lapata 2022	0	25	35	99	88‡

†Results are copied from Yao and Koller (2022). ‡Model uses pretrained weights and is hyperparameter tuned using data sampled from the generalization splits.

Wu et al. 2023

Why removing redundant tokens matters

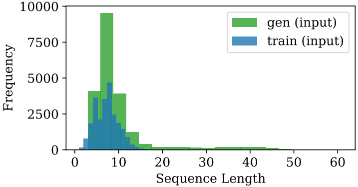
COGS: `kitten ( x _ 1 )` COGS: `kitten ( 1 )`



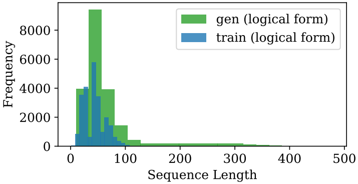
Wu et al. 2023

What is behind the 0s for CP/PP recursion?

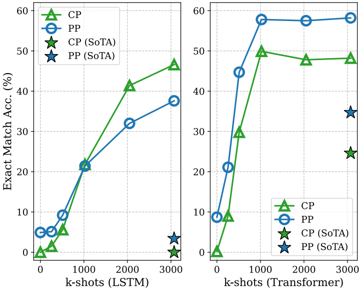
Input sentences



Output LFs



To decouple length from depth, we concatenate existing examples and reindex the variable names to cover the variable names seen at test time.



Wu et al. 2023

What is behind the 0s for PP modifiers?

Hypothesis

The train data *teach* the model that PPs occur only with a specific set of variables and positions. When models learn this lesson, they struggle with examples that contradict it.

Variant	**Sentence**	**Logical Form**
Preposing + Interjection	**The box in the tent** Emma was **um um** lended .	`* box ( x _ 1 ) ; * tent ( x _ 4 ) ; box . nmod . in ( x _ 1 , x _ 4 ) AND lend . theme ( x _ 7 , x _ 1 ) AND lend . recipient ( x _ 7 , Emma )`
Participial VP (*Subj*)	A leaf **painting the spaceship** froze .	`* spaceship ( x _ 4 ) ; leaf ( x _ 1 ) AND leaf . acl . paint ( x _ 1 , x _ 4 ) AND freeze . theme ( x _ 5 , x _ 1 )`

Result

Large performance increases for LSTMs and Transformers.

Wu et al. 2023

Modifications for ReCOGS

Wu et al. 2023

Modifications for ReCOGS

Input Sentence: Mia ate a cake .

Wu et al. 2023

Modifications for ReCOGS

Input Sentence: Mia ate a cake .

COGS LF: `eat . agent ( x _ 1 , Mia ) AND eat . theme ( x _ 1, x _ 3 ) AND cake ( x _ 3 )`

Wu et al. 2023

Modifications for ReCOGS

Input Sentence: Mia ate a cake .

COGS LF: `eat . agent ( x _ 1 , Mia ) AND eat . theme ( x _ 1, x _ 3 ) AND cake ( x _ 3 )`

↓

Redundant Token Removal

Wu et al. 2023

Modifications for ReCOGS

Input Sentence: Mia ate a cake .

COGS LF: `eat . agent ( x _ 1 , Mia ) AND eat . theme ( x _ 1, x _ 3 ) AND cake ( x _ 3 )`

↓

Redundant Token Removal

↓

Meaning-Preserving Data Augmentation

Wu et al. 2023

Modifications for ReCOGS

Input Sentence: Mia ate a cake .

COGS LF: `eat . agent ( x _ 1 , Mia ) AND eat . theme ( x _ 1, x _ 3 ) AND cake ( x _ 3 )`

↓

Redundant Token Removal

↓

Meaning-Preserving Data Augmentation

↓

Arbitrary Variable Renaming

Wu et al. 2023

Modifications for ReCOGS

Input Sentence: Mia ate a cake .

COGS LF: `eat . agent ( x _ 1 , Mia ) AND eat . theme ( x _ 1, x _ 3 ) AND cake ( x _ 3 )`

↓

Redundant Token Removal

↓

Meaning-Preserving Data Augmentation

↓

Arbitrary Variable Renaming

ReCOGS LF: `Mia ( 3 ) ; cake ( 21 ) ; eat ( 6 ) AND agent ( 6 , 3 ) AND theme ( 6 , 21 )`

Wu et al. 2023

Modifications for ReCOGS

Input Sentence: Mia ate a cake .

COGS LF: eat . agent (x _ 1 , Mia) AND eat . theme (x _ 1, x _ 3) AND cake (x _ 3)

↓

Redundant Token Removal

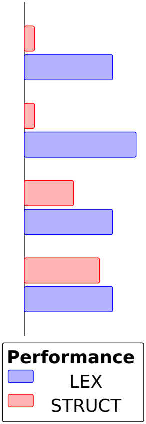
↓

Meaning-Preserving Data Augmentation

↓

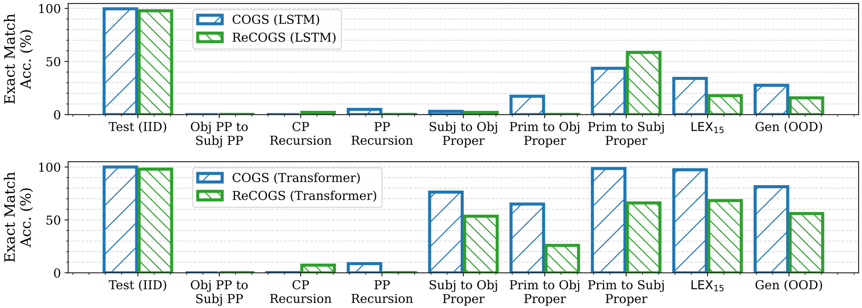
Arbitrary Variable Renaming

ReCOGS LF: Mia (3) ; cake (21) ; eat (6) AND agent (6 , 3) AND theme (6 , 21)



Wu et al. 2023

ReCOGS results



Wu et al. 2023

Conceptual questions

1. How can we test for *meaning* if we are predicting *logical forms*?
2. What is a *fair* generalization test in the current context?
   a. Models are shown a world that manifests specific restrictions.
   b. In some cases we want them not to learn those restrictions.
   c. In other cases we do want them to learn those restrictions.
3. What are the limits of compositionality *for humans* and how should that inform our generalization tests?
4. If we have goals that are not supported by our datasets but that seem like good goals for models to reach, how should we express that in our tasks and our models?

Adversarial testing

SQUaD leaderboards

Leaderboard

SQuAD2.0 tests the ability of a system to not only answer reading comprehension questions, but also abstain when presented with a question that cannot be answered based on the provided paragraph.

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Jun 04, 2021	IE-Net (ensemble) *RICOH_SRCB_DML*	**90.939**	**93.214**
2 Feb 21, 2021	FPNet (ensemble) *Ant Service Intelligence Team*	90.871	93.183
3 May 16, 2021	IE-NetV2 (ensemble) *RICOH_SRCB_DML*	90.860	93.100
4 Apr 06, 2020	SA-Net on Albert (ensemble) *QIANXIN*	90.724	93.011
5 May 05, 2020	SA-Net-V2 (ensemble) *QIANXIN*	90.679	92.948
5 Apr 05, 2020	Retro-Reader (ensemble) *Shanghai Jiao Tong University* http://arxiv.org/abs/2001.09694	90.578	92.978
	⋮		
31 Nov 12, 2019	RoBERTa+Verify (single model) *CW*	86.448	89.586
31 Mar 15, 2019	BERT + ConvLSTM + MTL + Verifier (ensemble) *Layer 6 AI*	86.730	89.286

Rajpurkar et al. 2016

SQUaD adversarial testing

Passage

Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager.

Question

What is the name of the quarterback who was 38 in Super Bowl XXXIII?

Jia and Liang 2017

SQUaD adversarial testing

Passage

Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager.

Question

What is the name of the quarterback who was 38 in Super Bowl XXXIII?

Answer

John Elway

Jia and Liang 2017

SQUaD adversarial testing

Passage

Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. Quarterback Leland Stanford had jersey number 37 in Champ Bowl XXXIV.

Question

What is the name of the quarterback who was 38 in Super Bowl XXXIII?

Answer

John Elway

Jia and Liang 2017

SQUaD adversarial testing

Passage

Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. Quarterback Leland Stanford had jersey number 37 in Champ Bowl XXXIV.

Question

What is the name of the quarterback who was 38 in Super Bowl XXXIII?

Answer

John Elway Model: Leland Stanford

Jia and Liang 2017

SQUaD adversarial testing

Passage

Quarterback Leland Stanford had jersey number 37 in Champ Bowl XXXIV. Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager.

Question

What is the name of the quarterback who was 38 in Super Bowl XXXIII?

Answer

John Elway

Jia and Liang 2017

SQUaD adversarial testing

Passage

Quarterback Leland Stanford had jersey number 37 in Champ Bowl XXXIV. Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager.

Question

What is the name of the quarterback who was 38 in Super Bowl XXXIII?

Answer

John Elway   Model: Leland Stanford

Jia and Liang 2017

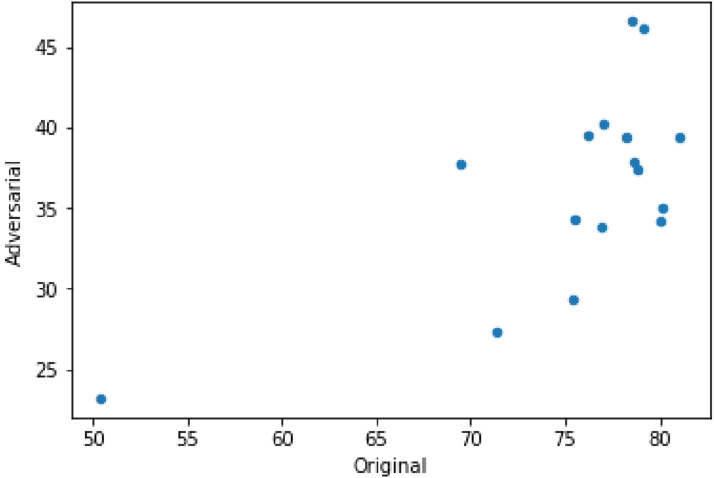
SQUaD adversarial testing

System	Original	Adversarial
ReasoNet-E	81.1	39.4
SEDT-E	80.1	35.0
BiDAF-E	80.0	34.2
Mnemonic-E	79.1	46.2
Ruminating	78.8	37.4
jNet	78.6	37.9
Mnemonic-S	78.5	46.6
ReasoNet-S	78.2	39.4
MPCM-S	77.0	40.3
SEDT-S	76.9	33.9
RaSOR	76.2	39.5
BiDAF-S	75.5	34.3
Match-E	75.4	29.4
Match-S	71.4	27.3
DCR	69.4	37.8
Logistic	50.4	23.2

SQUaD adversarial testing

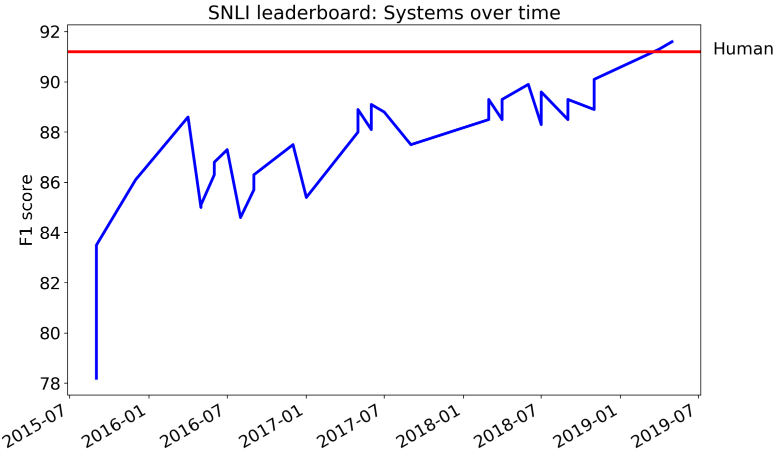
System	Original Rank	Adversarial Rank
ReasoNet-E	1	5
SEDT-E	2	10
BiDAF-E	3	12
Mnemonic-E	4	2
Ruminating	5	9
jNet	6	7
Mnemonic-S	7	1
ReasoNet-S	8	5
MPCM-S	9	3
SEDT-S	10	13
RaSOR	11	4
BiDAF-S	12	11
Match-E	13	14
Match-S	14	15
DCR	15	8
Logistic	16	16

Comparison with regular testing



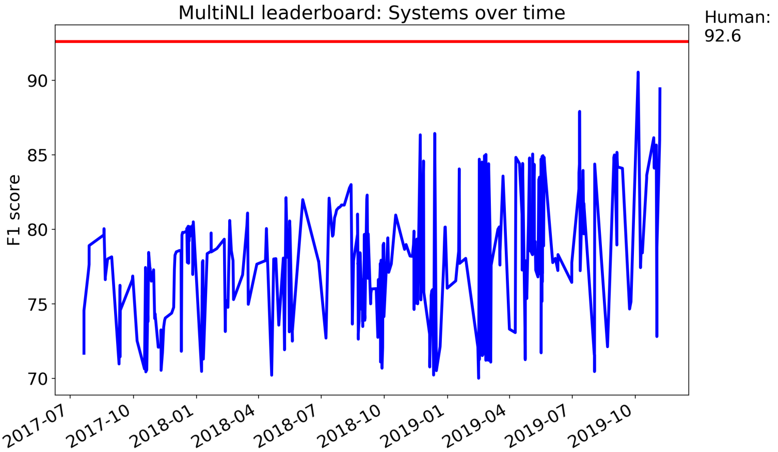
Plot of Original vs. Adversarial scores for SQUaD

Example: NLI



Bowman et al. 2015

Example: NLI



Bowman et al. 2015

An SNLI adversarial evaluation

	Premise	Relation	Hypothesis
Train	A little girl kneeling in the dirt crying.	entails	A little girl is very sad.
Adversarial		entails	A little girl is very unhappy.
Train	An elderly couple are sitting outside a restaurant, enjoying wine.	entails	A couple drinking wine.
Adversarial		neutral	A couple drinking champagne.

Glockner et al. 2018

An SNLI adversarial evaluation

Model	Train set	SNLI test set	New test set	Δ
Decomposable Attention (Parikh et al., 2016)	SNLI	84.7%	51.9%	-32.8
	MultiNLI + SNLI	84.9%	65.8%	-19.1
	SciTail + SNLI	85.0%	49.0%	-36.0
ESIM (Chen et al., 2017)	SNLI	87.9%	65.6%	-22.3
	MultiNLI + SNLI	86.3%	74.9%	-11.4
	SciTail + SNLI	88.3%	67.7%	-20.6
Residual-Stacked-Encoder (Nie and Bansal, 2017)	SNLI	86.0%	62.2%	-23.8
	MultiNLI + SNLI	84.6%	68.2%	-16.8
	SciTail + SNLI	85.0%	60.1%	-24.9
WordNet Baseline	-	-	85.8%	-
KIM (Chen et al., 2018)	SNLI	88.6%	83.5%	-5.1

Models that have access to the resources used to create the adversarial examples

Table 3: Accuracy of various models trained on SNLI or a union of SNLI with another dataset (MultiNLI, SciTail), and tested on the original SNLI test set and the new test set.

An SNLI adversarial evaluation

RoBERTA-MNLI, off-the-shelf

```
[1]: import nli, os, torch
     from sklearn.metrics import classification_report

[2]: # Available from https://github.com/BIU-NLP/Breaking_NLI:
     breaking_nli_src_filename = os.path.join("../new-data/data/dataset.jsonl")
     reader = nli.NLIReader(breaking_nli_src_filename)

[3]: exs = [((ex.sentence1, ex.sentence2), ex.gold_label) for ex in reader.read()]

[4]: X_test_str, y_test = zip(*exs)

[5]: model = torch.hub.load('pytorch/fairseq', 'roberta.large.mnli')
     _ = model.eval()

     Using cache found in /Users/cgpotts/.cache/torch/hub/pytorch_fairseq_master

[6]: X_test = [model.encode(*ex) for ex in X_test_str]

[7]: pred_indices = [model.predict('mnli', ex).argmax() for ex in X_test]

[8]: to_str = {0: 'contradiction', 1: 'neutral', 2: 'entailment'}

[9]: preds = [to_str[c.item()] for c in pred_indices]
```

An SNLI adversarial evaluation

RoBERTA-MNLI, off-the-shelf

```
[10]: print(classification_report(y_test, preds))

                   precision    recall  f1-score   support

    contradiction       0.99      0.97      0.98      7164
       entailment       0.86      1.00      0.92       982
          neutral       0.15      0.15      0.15        47

         accuracy                           0.97      8193
        macro avg       0.67      0.71      0.68      8193
     weighted avg       0.97      0.97      0.97      8193
```

A MultiNLI adversarial evaluation

Category	Premise	Relation	Hypothesis
Antonyms	I love the Cinderella story.	contradicts	I hate the Cinderella story.
Numerical	Tim has 350 pounds of cement in 100, 50, and 25 pound bags.	contradicts	Tim has less than 750 pounds of cement in 100, 50, and 25 pound bags.
Word overlap	Possibly no other country has had such a turbulent history.	entails	The country's history has been turbulent and true is true
Negation	Possibly no other country has had such a turbulent history.	entails	The country's history has been turbulent and false is not true

Also 'Length mismatch' and 'Spelling errors'; Naik et al. 2018

A MultiNLI adversarial evaluation

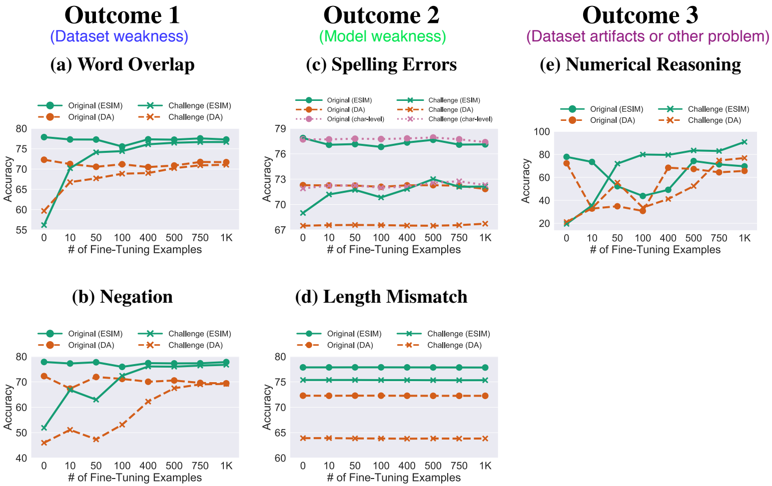
Category	Examples
Antonym	1,561
Length Mismatch	9815
Negation	9,815
Numerical Reasoning	7,596
Spelling Error	35,421
Word Overlap	9,815

Naik et al. 2018

A MultiNLI adversarial evaluation

System	Original MultiNLI Dev		Competence Test			Distraction Test						Noise Test	
			Antonymy		Numerical Reasoning	Word Overlap		Negation		Length Mismatch		Spelling Error	
	Mat	Mis	Mat	Mis		Mat	Mis	Mat	Mis	Mat	Mis	Mat	Mis
NB	74.2	74.8	15.1	19.3	21.2	47.2	47.1	39.5	40.0	48.2	47.3	51.1	49.8
CH	73.7	72.8	11.6	9.3	30.3	58.3	58.4	52.4	52.2	63.7	65.0	68.3	69.1
RC	71.3	71.6	36.4	32.8	30.2	53.7	54.4	49.5	50.4	48.6	49.6	66.6	67.0
IS	70.3	70.6	14.4	10.2	28.8	50.0	50.2	46.8	46.6	58.7	59.4	58.3	59.4
BiLSTM	70.2	70.8	13.2	9.8	31.3	57.0	58.5	51.4	51.9	49.7	51.2	65.0	65.1
CBOW	63.5	64.2	6.3	3.6	30.3	53.6	55.6	43.7	44.2	48.0	49.3	60.3	60.6

A MultiNLI adversarial evaluation



Liu et al. 2019;
Antonym not tested because its label is always ‘contradiction’

Adversarial NLI

Adversarial NLI

Adversarial NLI: A New Benchmark for Natural Language Understanding

Yixin Nie*, Adina Williams†, Emily Dinan†, Mohit Bansal*, Jason Weston†, Douwe Kiela†
*UNC Chapel Hill
†Facebook AI Research

Adversarial NLI: Dataset creation

A direct response to adversarial test failings *NLI datasets:

1. The annotator is presented with a premise sentence and a condition (entailment, contradiction, neutral).
2. The annotator writes a hypothesis.
3. A state-of-the-art model makes a prediction about the premise–hypothesis pair.
4. If the model's prediction matches the condition, the annotator returns to step 2 to try again.
5. If the model was fooled, the premise–hypothesis pair is independently validated by other annotators.

Adversarial NLI: Example

Premise	Hypothesis	Reason	Label	Model
A melee weapon is any weapon used in direct hand-to-hand combat; by contrast with ranged weapons which act at a distance. The term “melee” originates in the 1640s from the French word “mělée”, which refers to hand-to-hand combat, a close quarters battle, a brawl, a confused fight, etc. Melee weapons can be broadly divided into three categories	Melee weapons are good for ranged and hand-to-hand combat.	Melee weapons are good for hand to hand combat, but NOT ranged.	E	N

Adversarial NLI results

Model	Data	A1	A2	A3	ANLI	ANLI-E	SNLI	MNLI-m/-mm
BERT	S,M$^{\star 1}$	00.0	28.9	28.8	19.8	19.9	91.3	86.7 / 86.4
	+A1	44.2	32.6	29.3	35.0	34.2	91.3	86.3 / 86.5
	+A1+A2	57.3	45.2	33.4	44.6	43.2	90.9	86.3 / 86.3
	+A1+A2+A3	57.2	49.0	46.1	50.5	46.3	90.9	85.6 / 85.4
	S,M,F,ANLI	57.4	48.3	43.5	49.3	44.2	90.4	86.0 / 85.8
XLNet	S,M,F,ANLI	67.6	50.7	48.3	55.1	52.0	91.8	89.6 / 89.4
RoBERTa	S,M	47.6	25.4	22.1	31.1	31.4	92.6	90.8 / 90.6
	+F	54.0	24.2	22.4	32.8	33.7	92.7	90.6 / 90.5
	+F+A1$^{\star 2}$	68.7	19.3	22.0	35.8	36.8	92.8	90.9 / 90.7
	+F+A1+A2$^{\star 3}$	71.2	44.3	20.4	43.7	41.4	92.9	91.0 / 90.7
	S,M,F,ANLI	73.8	48.9	44.4	53.7	49.7	92.6	91.0 / 90.6

Table 3: Model Performance. 'Data' refers to training dataset ('S' refers to SNLI, 'M' to MNLI dev (-m=matched, -mm=mismatched), and 'F' to FEVER); 'A1–A3' refer to the rounds respectively. '-E' refers to test set examples written by annotators exclusive to the test set. Datasets marked '$^{\star n}$' were used to train the base model for round n, and their performance on that round is underlined.

A vision for future development

Zellers et al. (2019)

"a path for NLP progress going forward: towards benchmarks that adversarially co-evolve with evolving state-of-the-art models."

Nie et al. (2019)

"This process yields a "moving post" dynamic target for NLU systems, rather than a static benchmark that will eventually saturate."

Dynabench

DynaBench

Rethinking AI Benchmarking

Dynabench is a research platform for dynamic data collection and benchmarking. Static benchmarks have well-known issues: they saturate quickly, are susceptible to overfitting, contain exploitable annotator artifacts and have unclear or imperfect evaluation metrics.

This platform in essence is a scientific experiment: can we make faster progress if we collect data dynamically, with humans and models in the loop, rather than in the old-fashioned static way?

Read more

Dynabench

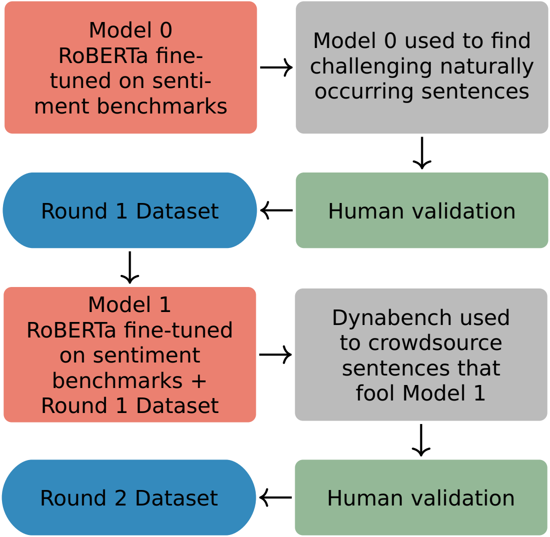
1. NLI (see Nie et al. 2020)
2. QA (see Bartolo et al. 2020)
3. Sentiment (DynaSent; Potts et al. 2021)
4. Hate Speech (Vidgen et al. 2020)

DynaSent

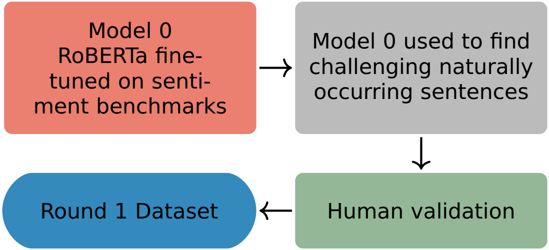
Overview and resources

- Data, code, and models:
  https://github.com/cgpotts/dynasent
- 121,634 sentences, across two rounds, each with 5 gold labels
- Paper: Potts et al. 2021
- Dynabench: https://dynabench.org

DynaSent overview



Round 1


Model 0
RoBERTa fine-tuned on sentiment benchmarks
Model 0 used to find challenging naturally occurring sentences
Human validation
Round 1 Dataset

Model 0: RoBERTa-based classifier

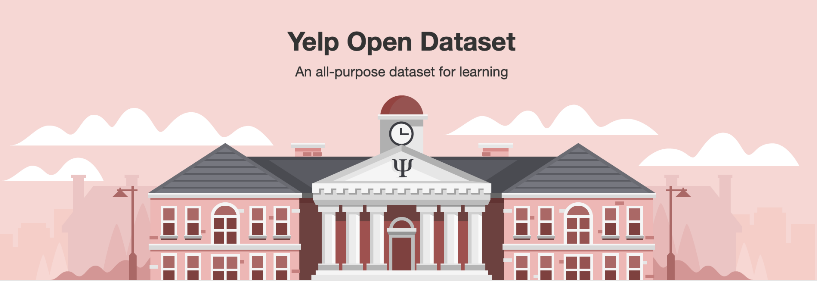
Training data

	CR	IMDB	SST-3	Yelp	Amazon
Positive	2,405	12,500	42,672	260,000	1,200,000
Negative	1,366	12,500	34,944	260,000	1,200,000
Neutral	0	0	81,658	130,000	600,000
Total	3,771	25,000	159,274	650,000	3,000,000

Performance on external assessment datasets

	SST-3		Yelp		Amazon	
	Dev	Test	Dev	Test	Dev	Test
Positive	85.1	89.0	88.3	90.5	89.1	89.4
Negative	84.1	84.1	88.8	89.1	86.6	86.6
Neutral	45.4	43.5	58.2	59.4	53.9	53.7
Macro avg	71.5	72.2	78.4	79.7	76.5	76.6

Harvesting sentences



Favor sentences where the review is 1-star and Model 0 predicts positive, and where the review is 5-star and Model 0 predicts negative.

Validation

Instructions

You will be shown 10 sentences from reviews of products and services. For each, your task is to choose from one of four labels:

- Positive: The sentence conveys information about the author's **positive evaluative sentiment**.
- Negative: The sentence conveys information about the author's **negative evaluative sentiment**.
- No sentiment: The sentence **does not convey anything** about the author's positive or negative sentiment.
- Mixed sentiment: The sentence conveys **a mix of positive and negative sentiment** with **no clear overall sentiment**.

Here are some simple examples of the labels:

- Sentence: This is an under-appreciated little gem of a movie.
  This is Positive because it expresses a positive overall opinion.
- Sentence: I asked for my steak medium-rare, and they delivered this perfectly!
  This is Positive because it puts a positive spin on an aspect of the author's experience.
- Sentence: The screen on this device is a little too bright.
  This is Negative because it negatively evaluates an aspect of the product.
- Sentence: The book is 972 pages long.
  This is No sentiment because it describes a factual matter with no evaluative component.
- Sentence: The waiting room is drab but the examination rooms are cheery enough.
  This is Mixed sentiment because two different sentiment evaluations are balanced against each other.
- Sentence: The entrees are delicious, but the service is so bad that it's not worth going.
  This is Negative because the negative statement outweighs the positive one.

1

Sentence: The host did a great job of making me feel unwanted.

- ○ **Positive: The sentence conveys information about the author's positive evaluative sentiment.**
- ○ **Negative: The sentence conveys information about the author's negative evaluative sentiment.**
- ○ **No sentiment: The sentence does not convey anything about the author's positive or negative sentiment.**
- ○ **Mixed sentiment: The sentence conveys a mix of positive and negative sentiment with no clear overall sentiment.**

Resulting dataset

	Dist	Majority Label		
	Train	Train	Dev	Test
Positive	130,045	21,391	1,200	1,200
Negative	86,486	14,021	1,200	1,200
Neutral	215,935	45,076	1,200	1,200
Mixed	39,829	3,900	0	0
No Majority	–	10,071	0	0
Total	472,295	94,459	3,600	3,600

47% adversarial examples

Model 0 versus the humans

Model 0

	SST-3		Yelp		Amazon		Round 1	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test
Positive	85.1	89.0	88.3	90.5	89.1	89.4	33.3	33.3
Negative	84.1	84.1	88.8	89.1	86.6	86.6	33.3	33.3
Neutral	45.4	43.5	58.2	59.4	53.9	53.7	33.3	33.3
Macro avg	71.5	72.2	78.4	79.7	76.5	76.6	33.3	33.3

Five annotators synthesized from our crowd

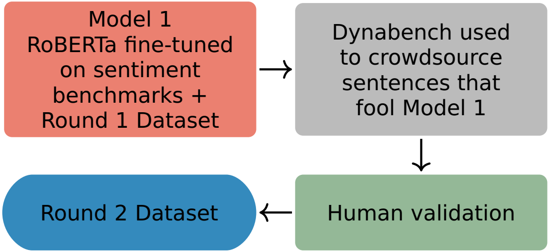
	Dev	Test
Positive	88.1	87.8
Negative	89.2	89.3
Neutral	86.6	86.9
Macro avg	88.0	88.0

Note: 614/1,280 workers *never* disagreed with the majority label.

Randomly sampled (short) examples

Sentence	Model 0	Responses
Good food nasty attitude by hostesses .	neg	**mix, mix, mix**, neg, neg
Not much of a cocktail menu that I saw.	neg	**neg, neg, neg, neg, neg**
I scheduled the work for 3 weeks later.	neg	**neu, neu, neu, neu**, pos
I was very mistaken, it was much more!	neg	neg, **pos, pos, pos, pos**
It is a gimmick, but when in Rome, I get it.	neu	**mix, mix, mix**, neu, neu
Probably a little pricey for lunch.	neu	mix, **neg, neg, neg, neg**
But this is strictly just my opinion.	neu	**neu, neu, neu, neu**, pos
The price was okay, not too pricey.	neu	mix, neu, **pos, pos, pos**
The only downside was service was a little slow.	pos	**mix, mix, mix**, neg, neg
However there is a 2 hr seating time limit.	pos	mix, **neg, neg, neg**, neu
With Alex, I never got that feeling.	pos	**neu, neu, neu, neu**, pos
Its ran very well by management.	pos	**pos, pos, pos, pos, pos**

Round 2


Model 1
RoBERTa fine-tuned
on sentiment
benchmarks +
Round 1 Dataset
Dynabench used
to crowdsource
sentences that
fool Model 1
Human validation
Round 2 Dataset

Model 1: RoBERTa-based classifier

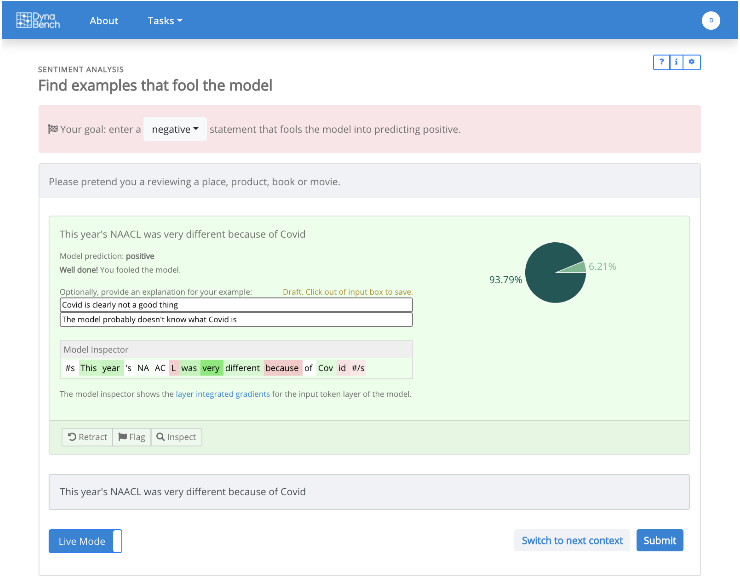
Training data

	CR	IMDB	SST-3	Yelp	Amazon	Round 1
Positive	2,405	12,500	128,016	29,841	133,411	339,748
Negative	1,366	12,500	104,832	30,086	133,267	252,630
Neutral	0	0	244,974	30,073	133,322	431,870
Total	3,771	25,000	477,822	90,000	400,000	1,024,248

Performance on external assessment datasets and Round 1

	SST-3		Yelp		Amazon		Round 1	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test
Positive	84.6	88.6	80.0	83.1	83.3	83.3	81.0	80.4
Negative	82.7	84.4	79.5	79.6	78.7	78.8	80.5	80.2
Neutral	40.0	45.2	56.7	56.6	55.5	55.4	83.1	83.5
Macro avg	69.1	72.7	72.1	73.1	72.5	72.5	81.5	81.4
Model 0	71.5	72.2	78.4	79.7	76.5	76.6	33.3	33.3

Dynabench interface



The prompt condition

guide | info | setting

SENTIMENT ANALYSIS

Find examples that fool the model

Your goal: enter a negative statement that fools the model into predicting positive or neutral.

Inspirational Prompt (you can use this as a starting point but it might not be negative):

The waitress periodically stopped by to say sorry or that it was coming up soon, but we didn't actually get food until almost 7:50.

The waitress periodically stopped by to say sorry in a very nice way, but we didn't actually get food until almost 7:50.

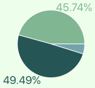
Model prediction: **positive**

You fooled the model! It predicted **positive**, but a person would say this sentence is **negative**.

Thank you! You are **required** to confirm that you judge this sentence to be **negative** before you can submit this HIT!

Yes, I confirm that I judge this sentence to be **negative**.

No, I judge this sentence to be **positive or neutral**.



Inspect

The waitress periodically stopped by to say sorry in a very nice way, but we didn't actually get food until almost 7:50.

Live Mode

Switch to next context | Submit | Tries: 1 / 10

Validation

Same as in Round 1.

Resulting dataset

	Dist	Majority Label		
	Train	Train	Dev	Test
Positive	32,551	6,038	240	240
Negative	24,994	4,579	240	240
Neutral	16,365	2,448	240	240
Mixed	18,765	3,334	0	0
No Majority	–	2,136	0	0
Total	92,675	18,535	720	720

19% adversarial examples

Model 1 versus the humans

Model 1

	SST-3		Yelp		Amazon		Round 1		Round 2	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test
Positive	84.6	88.6	80.0	83.1	83.3	83.3	81.0	80.4	33.3	33.3
Negative	82.7	84.4	79.5	79.6	78.7	78.8	80.5	80.2	33.3	33.3
Neutral	40.0	45.2	56.7	56.6	55.5	55.4	83.1	83.5	33.3	33.3
Macro avg	69.1	72.7	72.1	73.1	72.5	72.5	81.5	81.4	33.3	33.3

Five annotators synthesized from our crowd

	Dev	Test
Positive	91.0	90.9
Negative	91.2	91.0
Neutral	88.9	88.2
Macro avg	90.4	90.0

Note: 116/244 workers *never* disagreed with the majority label.

Randomly sampled (short) examples

Sentence	Model 1	Responses
The place was somewhat good and not well	neg	**mix, mix, mix, mix**, neg
I bought a new car and met with an accident.	neg	**neg, neg, neg, neg, neg**
The retail store is closed for now at least.	neg	**neu, neu, neu, neu, neu**
Prices are basically like garage sale prices.	neg	neg, neu, **pos, pos, pos**
That book was good. I need to get rid of it.	neu	**mix, mix, mix**, neg, pos
I REALLY wanted to like this place	neu	mix, **neg, neg, neg**, pos
I'm going to leave my money for the next vet.	neu	neg, **neu, neu, neu, neu**
once the model made a super decision.	neu	**pos, pos, pos, pos, pos**
I cook my caribbean food and it was okay	pos	**mix, mix, mix**, pos, pos
This concept is really cool in name only.	pos	mix, **neg, neg, neg**, neu
Wow, it'd be super cool if you could join us	pos	**neu, neu, neu, neu**, pos
Knife cut thru it like butter! It was great.	pos	**pos, pos, pos, pos, pos**

Conclusions

Key open questions

1. Can adversarial training improve systems? (See Jia and Liang 2017:§4.6; Alzantot et al. 2018:§4.3; Liu et al. 2019; Iyyer et al. 2018.)
2. What constitutes a *fair* non-IID generalization test?
3. Can hard behavioral testing provide us with the insights we need when it comes to certifying systems as truswothy? If so, which tests? If not, what should be do instead?
4. Are systems finding systematic solutions?
5. Where humans generalize in ways that are unsupported by direct experience, how should AI respond in terms of system design?

References I

Ekin Akyurek and Jacob Andreas. 2021. Lexicon learning for few shot sequence modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4934–4946, Online. Association for Computational Linguistics.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the AI: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678.

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, Stroudsburg, PA. Association for Computational Linguistics.

Henry Conklin, Bailin Wang, Kenny Smith, and Ivan Titov. 2021. Meta-learning to compositionally generalize. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3322–3335, Online. Association for Computational Linguistics.

Róbert Csordás, Kazuki Irie, and Juergen Schmidhuber. 2021. The devil is in the detail: Simple tricks improve systematic generalization of transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 619–634, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Atticus Geiger, Alexandra Carstensen, Michael C. Frank, and Christopher Potts. 2020a. Relational reasoning and generalization using non-symbolic neural networks. Ms., Stanford University.

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 4475–4485, Stroudsburg, PA. Association for Computational Linguistics.

Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020b. 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.

References II

Md Mosharaf Hossain, Venelin Kovatchev, Pranoy Dutta, Tiffany Kao, Elizabeth Wei, and Eduardo Blanco. 2020. An analysis of natural language inference benchmarks through the lens of negation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9106–9118, Online. Association for Computational Linguistics.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. 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 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031. Association for Computational Linguistics.

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.

Hector J. Levesque. 2013. On our best behaviour. In *Proceedings of the Twenty-third International Conference on Artificial Intelligence*, Beijing.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. ArXiv:1910.13461.

Percy Liang, Michael I. Jordan, and Dan Klein. 2013. Learning dependency-based compositional semantics. *Computational Linguistics*, 39(2):389–446.

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, Volume 1 (Long and Short Papers)*, pages 2171–2179, Minneapolis, Minnesota. Association for Computational Linguistics.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2019. Adversarial NLI: A new benchmark for natural language understanding. UNC CHapel Hill and Facebook AI Research.

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.

References III

Santiago Ontanon, Joshua Ainslie, Zachary Fisher, and Vaclav Cvicek. 2022. Making transformers solve compositional tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3591–3607, Dublin, Ireland. Association for Computational Linguistics.

Christopher Potts, Zhengxuan Wu, Atticus Geiger, and Douwe Kiela. 2021. DynaSent: A dynamic benchmark for sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2388–2404, Online. Association for Computational Linguistics.

David Premack. 1983. The codes of man and beasts. *Behavioral and Brain Sciences*, 6(1):125–136.

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*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Stroudsburg, PA. Association for Computational Linguistics.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2020. Learning from the worst: Dynamically generated datasets to improve online hate detection. *arXiv prerint arXiv:2012.15761*.

David H. D. Warren and Fernando C. N. Pereira. 1982. An efficient easily adaptable system for interpreting natural language queries. *American Journal of Computational Linguistics*, 8(3–4):110–122.

Ed Wasserman, Leyre Castro, and Joël Fagot. 2017. Relational thinking in animals and humans: From percepts to concepts. In J. Call, G. M Burghardt, I. M. Pepperberg, C. T. Snowdon, and T. Zentall, editors, *APA Handbook of Comparative Psychology: Perception, Learning, and Cognition*, volume 2. American Psychological Association.

Terry Winograd. 1972. Understanding natural language. *Cognitive Psychology*, 3(1):1–191.

Zhengxuan Wu, Christopher D. Manning, and Christopher Potts. 2023. ReCOGS: How incidental details of a logical form overshadow an evaluation of semantic interpretation. Ms., Stanford University.

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.

References IV

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019. 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 2019)*, pages 250–255, Minneapolis, Minnesota. Association for Computational Linguistics.

Yuekun Yao and Alexander Koller. 2022. Structural generalization is hard for sequence-to-sequence models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5048–5062, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. 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.