Pushing the Limits of Rule Reasoning in Transformers through Natural Language Satisfiability

Kyle Richardson and Ashish Sabharwal

Allen Institute for AI

AAAI 2022

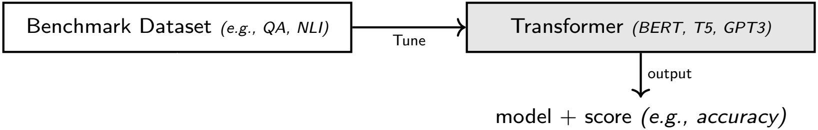
What can models do?

What kind of general (algorithmic) problems can current deep learning models in NLP solve?

What can models do?

What kind of general (algorithmic) problems can current deep learning models in NLP solve?

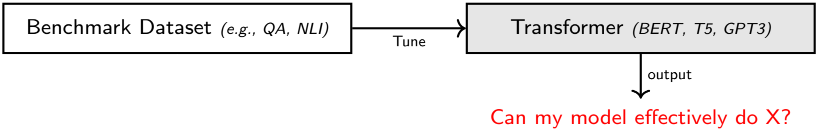
Standard Modeling Pipeline à la Devlin et al. (2018)



What can models do?

What kind of general (algorithmic) problems can current deep learning models in NLP solve?

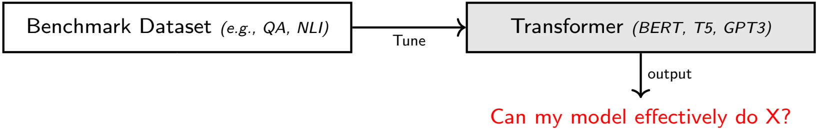
Standard Modeling Pipeline à la Devlin et al. (2018)



What can models do?

What kind of general (algorithmic) problems can current deep learning models in NLP solve?

Standard Modeling Pipeline à la Devlin et al. (2018)

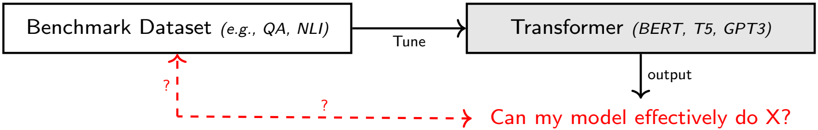


Problem: models and (often) datasets are black boxes, cannot look inside; *A real problem for ensuring model correctness and safety*.

What can models do?

What kind of general (algorithmic) problems can current deep learning models in NLP solve?

Standard Modeling Pipeline à la Devlin et al. (2018)

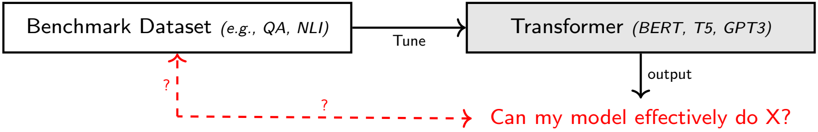


Problem: models and (often) datasets are black boxes, cannot look inside; *A real problem for ensuring model correctness and safety*.

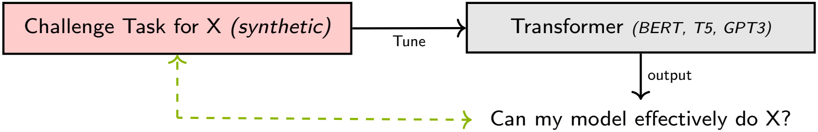
What can models do?

What kind of general (algorithmic) problems can current deep learning models in NLP solve?

Standard Modeling Pipeline à la Devlin et al. (2018)



Behavioral Testing *(This work)*

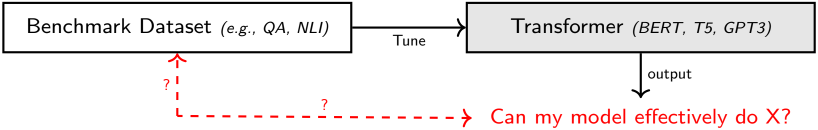


E.g., Can models learn (empirically) to solve hard reasoning puzzles?

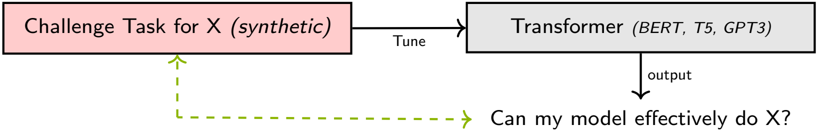
What can models do?

What kind of general (algorithmic) problems can current deep learning models in NLP solve?

Standard Modeling Pipeline à la Devlin et al. (2018)



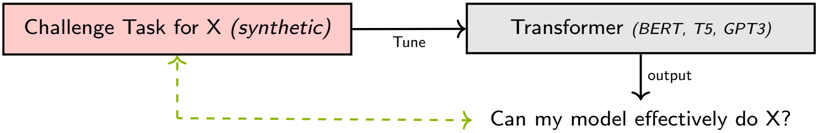
Behavioral Testing *(This work)*



What can models do?

What kind of general (algorithmic) problems can current deep learning models in NLP solve?

Behavioral Testing *(This work)*

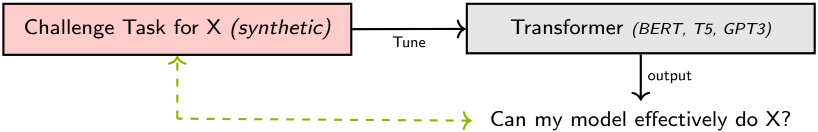


Big Idea: Cannot demonstrate correctness directly, demonstrate *indirectly* using (synthetic) data that is *correct by construction*.

What can models do?

What kind of general (algorithmic) problems can current deep learning models in NLP solve?

Behavioral Testing *(This work)*

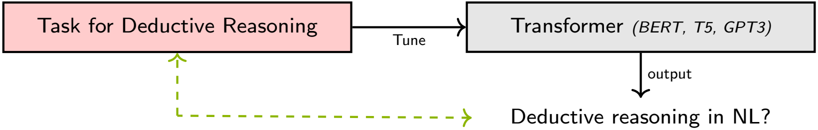


Big Idea: Cannot demonstrate correctness directly, demonstrate *indirectly* using (synthetic) data that is *correct by construction*.

Results only have meaning if data faithfully captures target problem space, **Can we ensure that data is reliable?**

This work: probing deductive reasoning in transformers

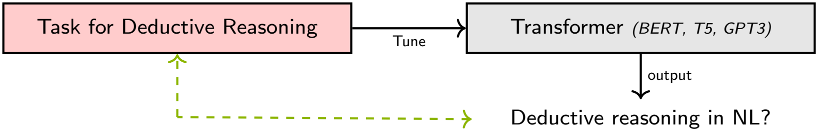
Behavioral Testing of Deductive Reasoning



NL Theory (RuleTaker)	$\Gamma_{\mathbf{NL}}$ ={ Bob is round. Alan is blue, rough and young. **If someone is round then they are big.** All rough people are green. **Big people are not green.** }
NL Query	Bob is not green?

Rule Reasoning: Can models learn to do correct deductive reasoning over **NL Theories** (rules and facts)? Clark et al. (2020)

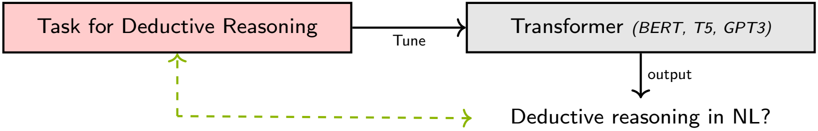
Behavioral Testing of Deductive Reasoning



NL Theory (RuleTaker)	$\Gamma_{\mathbf{NL}}$ ={ Bob is round. Alan is blue, rough and young. **If someone is round then they are big.** All rough people are green. **Big people are not green.** }
NL Query	Bob is not green? ✓

Rule Reasoning: Can models learn to do correct deductive reasoning over **NL Theories** (rules and facts)? Clark et al. (2020)

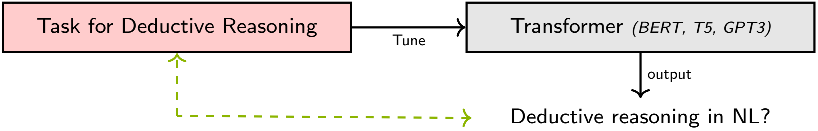
Behavioral Testing of Deductive Reasoning



NL Theory (RuleTaker)	$\Gamma_{\mathbf{NL}}$ ={ Bob is round. Alan is blue, rough and young. **If someone is round then they are big.** All rough people are green. **Big people are not green.** }
NL Query	Bob is not green? ✓

Rule Reasoning: Can models learn to do correct deductive reasoning over **NL Theories** (rules and facts)? Clark et al. (2020)

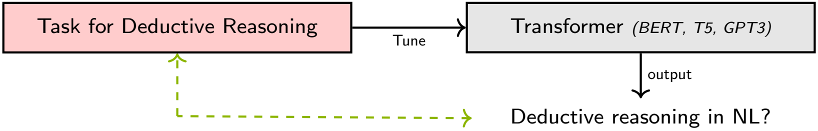
Behavioral Testing of Deductive Reasoning



NL Theory (RuleTaker)	$\Gamma_{\mathbf{NL}}$ ={ Bob is round. Alan is blue, rough and young. **If someone is round then they are big.** All rough people are green. **Big people are not green.** }
NL Query	Bob is not green? ✓

Rule Reasoning: Can models learn to do correct deductive reasoning over **NL Theories** (rules and facts)? Clark et al. (2020)

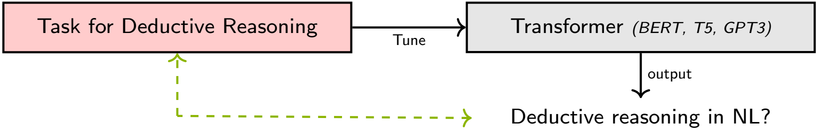
Behavioral Testing of Deductive Reasoning



NL Theory (RuleTaker)	$\Gamma_{\textbf{NL}}$ ={ Bob is round. Alan is blue, rough and young. **If someone is round then they are big.** All rough people are green. **Big people are not green.** }
NL Query	Bob is not green? ✓

Rule Reasoning: Can models learn to do correct deductive reasoning over **NL Theories** (rules and facts)? Clark et al. (2020)

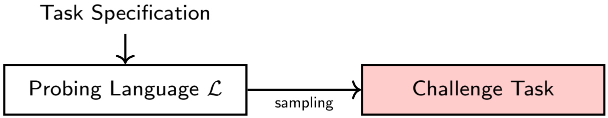
Behavioral Testing of Deductive Reasoning



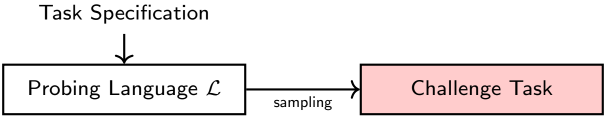
NL Theory (RuleTaker)	$\Gamma_{\mathbf{NL}}$ ={ Bob is round. Alan is blue, rough and young. **If someone is round then they are big.** All rough people are green. **Big people are not green.** }
NL Query	Bob is not green? ✓

Why Logic? fundamental to other forms of reasoning, well-understood, gives insight into the general *information aggregation* capacity of models.

Behavioral Testing of Deductive Reasoning: How?



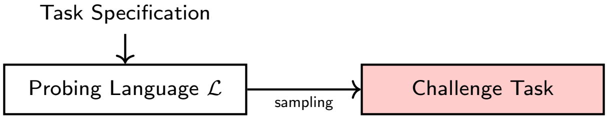
Behavioral Testing of Deductive Reasoning: How?



Want to see if models can solve hard computational problems, test limits.

Important questions about target tasks:

Behavioral Testing of Deductive Reasoning: How?

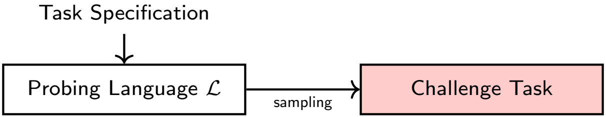


Want to see if models can solve hard computational problems, test limits.

Important questions about target tasks:

1. Is our probing language *able* to express computationally hard problems?

Behavioral Testing of Deductive Reasoning: How?



Want to see if models can solve hard computational problems, test limits.

Important questions about target tasks:

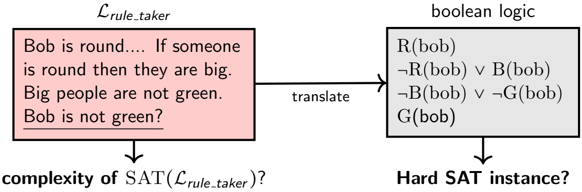
1. Is our probing language *able* to express computationally hard problems?
2. Does the sampling procedure *effectively find* the hard instances?

Are we asking models to solve hard problems?

1. Is our probing language able to express computationally hard problems?
2. Does the sampling procedure effectively find the hard instances?

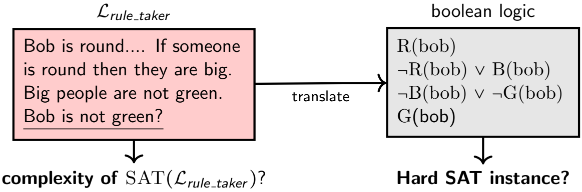
Are we asking models to solve hard problems?

1. Is our probing language able to express computationally hard problems?
2. Does the sampling procedure effectively find the hard instances?



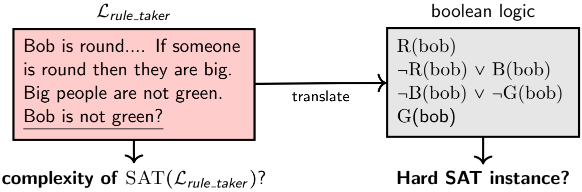
Are we asking models to solve hard problems?

1. Is our probing language able to express computationally hard problems?
2. Does the sampling procedure effectively find the hard instances?



Are we asking models to solve hard problems?

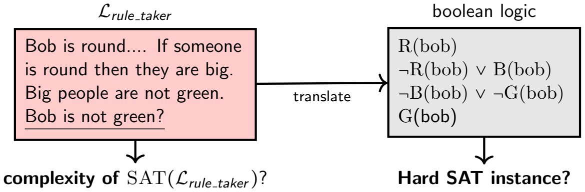
1. Is our probing language able to express computationally hard problems?
2. Does the sampling procedure effectively find the hard instances?



- Seemingly non-trivial problems can be computationally easy; here, easily solvable in linear time (unit-propagation), **common in RuleTaker**.

Are we asking models to solve hard problems?

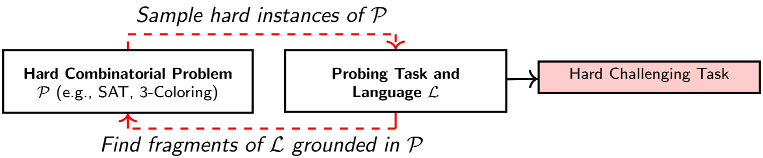
1. Is our probing language able to express computationally hard problems?
2. Does the sampling procedure effectively find the hard instances?



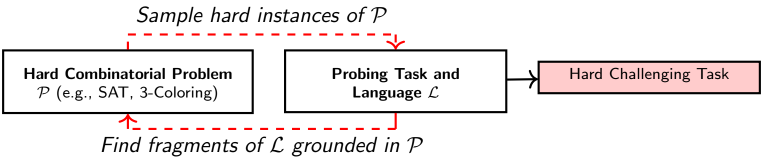
- Seemingly non-trivial problems can be computationally easy; here, easily solvable in linear time (unit-propagation), **common in RuleTaker**.

  Random sampling does not always result in hard instances; yield misleading results / harm model robustness (Shin et al., 2019).

Our Framework

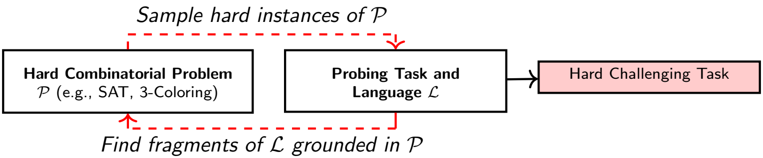


Our Framework



- Ground target probing problems in known *hard* combinatorial problems; *undertstand complexity* and work from *known hard problem distributions*.

Our Framework



- Ground target probing problems in known *hard* combinatorial problems; *undertstand complexity* and work from *known hard problem distributions*.

This work: new hard reasoning tasks for deductive rule reasoning and *natural language satisfiability*, based on boolean SAT and random kSAT.

New Tasks: Natural Language Satisfiability (NLSat)

Natural Language Satisfiability (NLSat)

- Deductive reasoning that involves determining whether a set of rules expressed in natural language has a satisfying assignment.

Natural Language Satisfiability (NLSat)

- Deductive reasoning that involves determining whether a set of rules expressed in natural language has a satisfying assignment.

  Used in linguistics and logic to investigate the complexity of rule fragments of English (Pratt-Hartmann, 2004, 2015).

Natural Language Satisfiability (NLSat)

- Deductive reasoning that involves determining whether a set of rules expressed in natural language has a satisfying assignment.

  Used in linguistics and logic to investigate the complexity of rule fragments of English (Pratt-Hartmann, 2004, 2015).

ordinary boolean SAT:

$$\underbrace{(A \vee B \vee C)}_{\text{clause}} \wedge (A \vee \neg B \vee \underbrace{C}_{+\text{literal}}) \wedge (\neg A \vee \underbrace{\neg B}_{\text{-literal}} \vee D) \quad \textit{(A=T,B=F,C=T,D=T)}$$

Natural Language Satisfiability (NLSat)

- Deductive reasoning that involves determining whether a set of rules expressed in natural language has a satisfying assignment.

  Used in linguistics and logic to investigate the complexity of rule fragments of English (Pratt-Hartmann, 2004, 2015).

ordinary boolean SAT:

$$\underbrace{(A \vee B \vee C)}_{\text{clause}} \wedge (A \vee \neg B \vee \underbrace{C}_{\text{+literal}}) \wedge (\neg A \vee \underbrace{\neg B}_{\text{-literal}} \vee D) \quad \textit{(A=T,B=F,C=T,D=T)}$$

Text rendering:

$\overbrace{\text{If not apple and not carrot then pear.}}^{(\neg A \wedge \neg B) \rightarrow C \equiv A \vee B \vee C}$ If not apple and carrot then pear. If apple and carrot then steak. (apple=T,carrot=F,pear=T,...)

Random assignment: variables → nouns (A →**apple**, B →**carrot**, C → **pear**), clauses → rules: *If (not)* $\mathbf{N}_1$ *and (not)* $\mathbf{N}_2$ *then (not)* $\mathbf{N}_3$.

Natural Language Satisfiability (NLSat)

- Deductive reasoning that involves determining whether a set of rules expressed in natural language has a satisfying assignment.

  Used in linguistics and logic to investigate the complexity of rule fragments of English (Pratt-Hartmann, 2004, 2015).

ordinary boolean SAT:

$$\underbrace{(A \vee B \vee C)}_{\text{clause}} \wedge (A \vee \neg B \vee \underbrace{C}_{+\text{literal}}) \wedge (\neg A \vee \underbrace{\neg B}_{\text{-literal}} \vee D) \quad (A{=}T, B{=}F, C{=}T, D{=}T)$$

Text rendering:

$$\overbrace{\text{Everyone who is not a gardener and not a cook is a nurse.}}^{(\neg A(j) \wedge \neg B(j)) \rightarrow G(j)}$$

Everyone who is not a gardener and not a cook is a nurse. Every cook who is not a gardener is a nurse... *John is either a cook or not a nurse or...* (John can be: a gardener, a nurse,..)

Natural Language Satisfiability (NLSat)

- Deductive reasoning that involves determining whether a set of rules expressed in natural language has a satisfying assignment.

  Used in linguistics and logic to investigate the complexity of rule fragments of English (Pratt-Hartmann, 2004, 2015).

ordinary boolean SAT:

$$\underbrace{(A \vee B \vee C)}_{\text{clause}} \wedge (A \vee \neg B \vee \underbrace{C}_{+\text{literal}}) \wedge (\neg A \vee \underbrace{\neg B}_{\text{-literal}} \vee D) \quad \textit{(A=T,B=F,C=T,D=T)}$$

RuleTaker:

If someone is not round and not big then they are green. If something is big and not round round then they are green... $\underbrace{\neg\text{Bob is not green?}}_{\text{negate query: unsat?}}$

Assume finite-domains, *instantiate* quantifiers to translate to propositional logic, in the style of Kautz et al. (1992).

Probing Languages and Rule Fragments

- **Tasks:** Two rule fragments (/ *set of rule templates*) that can be directly translated to/from arbitrary 3SAT, **computationally hard by design**.

Probing Languages and Rule Fragments

- **Tasks:** Two rule fragments (/ *set of rule templates*) that can be directly translated to/from arbitrary 3SAT, **computationally hard by design**.

Grounded Rule Language ($\mathcal{L}_{\text{GRL}}$)

$\underbrace{\text{If carrot and not steak then apples. If apples and grapes then no carrots. If no carrots and no steak then not apples.}}_{\text{grounded prop. rules}}$

Probing Languages and Rule Fragments

- **Tasks:** Two rule fragments (/ *set of rule templates*) that can be directly translated to/from arbitrary 3SAT, **computationally hard by design**.

Grounded Rule Language ($\mathcal{L}_{\mathsf{GRL}}$)

$\underbrace{\text{If carrot and not steak then apples. If apples and grapes then no carrots. If no carrots and no steak then not apples.}}_{\text{grounded prop. rules}}$

Relative clause fragment ($\mathcal{L}_{\mathsf{RCL}}$) (partly from (Pratt-Hartmann, 2004))

$\overbrace{\text{Every doctor who is not a philosopher is a baker.}}^{\text{relative clause construction}}$ No baker who is a gardener is a philosopher... $\underbrace{\text{John is either a doctor or a baker or not a...}}_{\text{disjunctive rules}}$

Sampling: how to find hard problems

- Solving 3SAT is computationally hard under a worst-case analysis; *does not* mean that all problems are hard.

Sampling: how to find hard problems

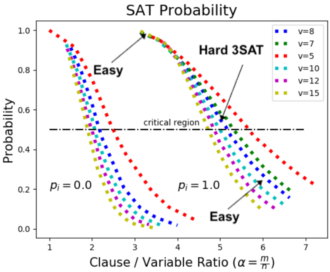
- Solving 3SAT is computationally hard under a worst-case analysis; *does not* mean that all problems are hard.

  Generating hard random kSAT problems is well-studied (Selman et al., 1996), lies at critical thresholds; **can use to sample hard examples**.

Sampling: how to find hard problems

- Solving 3SAT is computationally hard under a worst-case analysis; *does not* mean that all problems are hard.

  Generating hard random *k*SAT problems is well-studied (Selman et al., 1996), lies at critical thresholds; **can use to sample hard examples**.



Hard Sampling: Sampling from *critical regions* of random 3SAT, compare against other sampling strategies (e.g., *easy only*, *random*).

Summary of Datasets and Comparison

Grounded rule **(GRL)** and *Relative clause* **(RCL)** fragments, instances translated from *hard* random 3SAT across different # variables.

Summary of Datasets and Comparison

Grounded rule **(GRL)** and *Relative clause* **(RCL)** fragments, instances translated from *hard* random 3SAT across different # variables.

parameters: # variables, for RCL: #vars. = # ground variables (after quantifier instantiation), random *var*. → nouns/selection of templates.

Summary of Datasets and Comparison

Grounded rule **(GRL)** and *Relative clause* **(RCL)** fragments, instances translated from *hard* random 3SAT across different # variables.

parameters: # variables, for RCL: #vars. = # ground variables (after quantifier instantiation), random *var*. → nouns/selection of templates.

Language complexity and SAT metrics

Dataset ($\mathbf{d}_{\#vars}$)	**Size**	**Complexity** (NP-complete?)	**Conflicts** **(avg/med.)**	**Decisions** **(avg/med.)**
RuleTaker	130k	yes	0.0,/0.0	6.6/0.0
$\textbf{GRL}_{5,12}$	187k	yes	3.4/4.0	5.4/4.0
$\textbf{RCL}_{16,70}$	219k	yes	7.6/6.0	29.7/6.0
$\textbf{GRL}\text{-eval}_{20,50}$	17k	yes	22.0/13.0	29.3/13.0

Similar in size to RuleTaker,

Summary of Datasets and Comparison

Grounded rule **(GRL)** and *Relative clause* **(RCL)** fragments, instances translated from *hard* random 3SAT across different # variables.

parameters: # variables, for RCL: #vars. = # ground variables (after quantifier instantiation), random *var*. → nouns/selection of templates.

Language complexity and SAT metrics

Dataset ($\mathbf{d}_{\#vars}$)	Size	Complexity (NP-complete?)	Conflicts (avg/med.)	Decisions (avg/med.)
RuleTaker	130k	yes	0.0,/0.0	6.6/0.0
$\mathbf{GRL}_{5,12}$	187k	yes	3.4/4.0	5.4/4.0
$\mathbf{RCL}_{16,70}$	219k	yes	7.6/6.0	29.7/6.0
GRL-eval$_{20,50}$	17k	yes	22.0/13.0	29.3/13.0

Similar in size to RuleTaker, **out of domain evaluation set**,

Summary of Datasets and Comparison

Grounded rule **(GRL)** and *Relative clause* **(RCL)** fragments, instances translated from *hard* random 3SAT across different # variables.

parameters: # variables, for RCL: #vars. = # ground variables (after quantifier instantiation), random *var*. → nouns/selection of templates.

Language complexity and SAT metrics

Dataset ($\mathbf{d}_{\#vars}$)	Size	Complexity (NP-complete?)	Conflicts (avg/med.)	Decisions (avg/med.)
RuleTaker	130k	yes	0.0,/0.0	6.6/0.0
$\mathbf{GRL}_{5,12}$	187k	yes	3.4/4.0	5.4/4.0
$\mathbf{RCL}_{16,70}$	219k	yes	7.6/6.0	29.7/6.0
GRL-eval$_{20,50}$	17k	yes	22.0/13.0	29.3/13.0

Similar in size to RuleTaker, **out of domain evaluation set**, **higher empirical complexity**

RuleTaker: Hard language, empirically involves simplest forms of reasoning, solvable through pre-processing.

Summary of Datasets and Comparison

Grounded rule **(GRL)** and *Relative clause* **(RCL)** fragments, instances translated from *hard* random 3SAT across different # variables.

parameters: # variables, for RCL: #vars. = # ground variables (after quantifier instantiation), random *var*. → nouns/selection of templates.

Language complexity and SAT metrics

Dataset ($\mathbf{d}_{\#vars}$)	Size	Complexity (NP-complete?)	Conflicts (avg/med.)	Decisions (avg/med.)
RuleTaker	130k	yes	0.0,/0.0	6.6/0.0
$\mathbf{GRL}_{5,12}$	187k	yes	3.4/4.0	5.4/4.0
$\mathbf{RCL}_{16,70}$	219k	yes	7.6/6.0	29.7/6.0
GRL-eval$_{20,50}$	17k	yes	22.0/13.0	29.3/13.0

Similar in size to RuleTaker, **out of domain evaluation set**, **higher empirical complexity**

Our datasets: Test a wider-range of reasoning types and difficulty; (*caveat*) still relatively easy SAT problems, highly verbose.

Experimental Setup

- Binary decision task (accuracy % **sat** vs. **unsat**); Two models: **T5-large** (Raffel et al., 2020), **RoBERTa-large** (Liu et al., 2019).

Experimental Setup

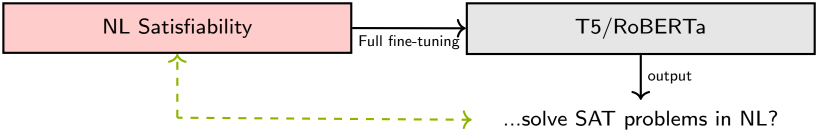
- Binary decision task (accuracy % **sat** vs. **unsat**); Two models: **T5-large** (Raffel et al., 2020), **RoBERTa-large** (Liu et al., 2019).

  Standard fine-tuning set up following Clark et al. (2020), tuned to maximize dev. accuracy.

Experimental Setup

- Binary decision task (accuracy % **sat** vs. **unsat**); Two models: **T5-large** (Raffel et al., 2020), **RoBERTa-large** (Liu et al., 2019).

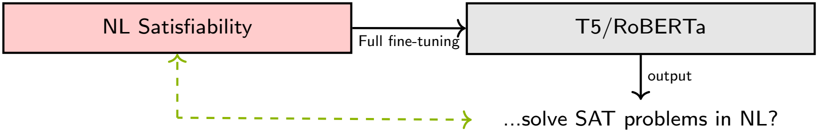
  Standard fine-tuning set up following Clark et al. (2020), tuned to maximize dev. accuracy.



Experimental Setup

- Binary decision task (accuracy % **sat** vs. **unsat**); Two models: **T5-large** (Raffel et al., 2020), **RoBERTa-large** (Liu et al., 2019).

  Standard fine-tuning set up following Clark et al. (2020), tuned to maximize dev. accuracy.

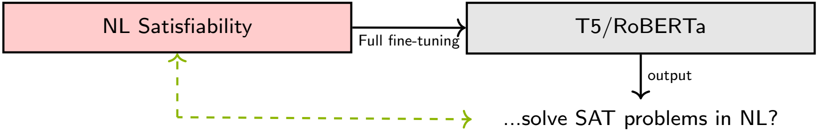


- **I.I.D train/test:** Solving natural language satisfiability problems involving problems with a **fixed number of variables**.

Experimental Setup

- Binary decision task (accuracy % **sat** vs. **unsat**); Two models: **T5-large** (Raffel et al., 2020), **RoBERTa-large** (Liu et al., 2019).

  Standard fine-tuning set up following Clark et al. (2020), tuned to maximize dev. accuracy.



- **I.I.D train/test:** Solving natural language satisfiability problems involving problems with a **fixed number of variables**.
- **Scale-invariance:** testing models on problems of **larger scope** and **differing number of variables**.

Can models solve natural language satisfiability problems?

Can models solve natural language satisfiability problems?

It's complicated

Result 1: Models can solve (some) hard tasks

- Models are capable of solving new problems (i.i.d setting), with degradation of performance as a function of # variables.

Result 1: Models can solve (some) hard tasks

- Models are capable of solving new problems (i.i.d setting), with degradation of performance as a function of # variables.

	Grounded Rule Language **GRL**, Accuracy%					
Model (# variables.)	**5var**	**7var**	**8var**	**10var**	**12var**	**Avg.**
Random	50.0	50.0	50.0	50.0	50.0	50.0
$\textbf{T5}_{5,7,8,10,12}$	98.0	95.4	94.3	90.7	88.3	93.4
$\textbf{RoBERTa}_{5,7,8,10,12}$	96.4	92.0	90.2	85.4	83.4	89.5

Result 1: Models can solve (some) hard tasks

- Models are capable of solving new problems (i.i.d setting), with degradation of performance as a function of # variables.

Grounded Rule Language **GRL**, Accuracy%

Model (# variables.)	**5var**	**7var**	**8var**	**10var**	**12var**	**Avg.**
Random	50.0	50.0	50.0	50.0	50.0	50.0
T5 $_{5,7,8,10,12}$	98.0	95.4	94.3	90.7	88.3	93.4
RoBERTa $_{5,7,8,10,12}$	96.4	92.0	90.2	85.4	83.4	89.5

Result 1: Models can solve (some) hard tasks

- Models are capable of solving new problems (i.i.d setting), with degradation of performance as a function of # variables.

Grounded Rule Language **GRL**, Accuracy%

Model (# variables.)	**5var**	**7var**	**8var**	**10var**	**12var**	**Avg.**
Random	50.0	50.0	50.0	50.0	50.0	50.0
T5$_{5,7,8,10,12}$	98.0	95.4	94.3	90.7	88.3	93.4
RoBERTa$_{5,7,8,10,12}$	96.4	92.0	90.2	85.4	83.4	89.5

Result 1: Models can solve (some) hard tasks

- Models are capable of solving new problems (i.i.d setting), with degradation of performance as a function of # variables.

Grounded Rule Language **GRL**, Accuracy%

Model (# variables.)	**5var**	**7var**	**8var**	**10var**	**12var**	**Avg.**
Random	50.0	50.0	50.0	50.0	50.0	50.0
T5$_{5,7,8,10,12}$	98.0	95.4	94.3	90.7	88.3	93.4
RoBERTa$_{5,7,8,10,12}$	96.4	92.0	90.2	85.4	83.4	89.5

Grounded Relative Clause Language **RCL**, Accuracy%

Model (# ground var.)	**16,21v**	**25,32v**	**35,48v**	**60,70v**	**Avg.**
Random	50.0	50.0	50.0	51.2	50.3
T5$_{16,70}$	95.9	95.3	94.7	92.9	94.7
RoBERTa$_{16,70}$	96.0	95.9	94.9	94.0	95.2

Result 1: Models can solve (some) hard tasks

- Models are capable of solving new problems (i.i.d setting), with degradation of performance as a function of # variables.

Grounded Rule Language **GRL**, Accuracy%

Model (# variables.)	**5var**	**7var**	**8var**	**10var**	**12var**	**Avg.**
Random	50.0	50.0	50.0	50.0	50.0	50.0
T5$_{5,7,8,10,12}$	98.0	95.4	94.3	90.7	88.3	93.4
RoBERTa$_{5,7,8,10,12}$	96.4	92.0	90.2	85.4	83.4	89.5

Grounded Relative Clause Language **RCL**, Accuracy%

Model (# ground var.)	**16,21v**	**25,32v**	**35,48v**	**60,70v**	**Avg.**
Random	50.0	50.0	50.0	51.2	50.3
T5$_{16,70}$	95.9	95.3	94.7	92.9	94.7
RoBERTa$_{16,70}$	96.0	95.9	94.9	94.0	95.2

Can models solve NL satisfiability problem? It depends on the number of variables, still not an entirely solved task. (GRL)

Result 2: Models have limited scale invariance

- Models exhibit some degree of scale invariance, though lack generalization ability expected for robust deductive reasoning.

Result 2: Models have limited scale invariance

- Models exhibit some degree of scale invariance, though lack generalization ability expected for robust deductive reasoning.

	main **GRL** (i.i.d)				o.o.d eval(20-50 variables)			
Model	**5var**	**8var**	**10var**	**12var**	**20var**	**30var**	**40var**	**50var**
T5	96.2	92.4	87.7	73.6	74.4	67.1	53.5	50.1
(v=8)	94.0	87.9	81.6	74.8	67.5	58.3	51.2	50.0
T5	93.9	92.7	89.7	79.0	78.6	71.2	54.7	50.1
(v=10)	89.7	86.3	82.5	76.7	70.0	60.1	51.4	50.0
T5	94.5	91.5	87.7	77.3	77.8	70.7	53.3	50.0
(v=12)	91.1	84.9	80.7	81.0	70.1	60.3	51.4	50.0
T5	98.6	96.0	92.6	85.0	86.5	84.9	69.8	59.1
(v=5,12)	98.1	93.6	89.6	88.5	80.7	72.7	61.4	51.8

Result 2: Models have limited scale invariance

- Models exhibit some degree of scale invariance, though lack generalization ability expected for robust deductive reasoning.

	main **GRL** (i.i.d)				o.o.d eval(20-50 variables)			
Model	**5var**	**8var**	**10var**	**12var**	**20var**	**30var**	**40var**	**50var**
T5	96.2	92.4	87.7	73.6	74.4	67.1	53.5	50.1
(v=8)	94.0	87.9	81.6	74.8	67.5	58.3	51.2	50.0
T5	93.9	92.7	89.7	79.0	78.6	71.2	54.7	50.1
(v=10)	89.7	86.3	82.5	76.7	70.0	60.1	51.4	50.0
T5	94.5	91.5	87.7	77.3	77.8	70.7	53.3	50.0
(v=12)	91.1	84.9	80.7	81.0	70.1	60.3	51.4	50.0
T5	98.6	96.0	92.6	85.0	86.5	84.9	69.8	59.1
(v=5,12)	98.1	93.6	89.6	88.5	80.7	72.7	61.4	51.8

5-12 variable models can solve 20-30 variable problems far above random chance, large differences between $\frac{easy}{hard}$ distribution.

Result 2: Models have limited scale invariance

- Models exhibit some degree of scale invariance, though lack generalization ability expected for robust deductive reasoning.

	main **GRL** (i.i.d)				o.o.d eval(20-50 variables)			
Model	**5var**	**8var**	**10var**	**12var**	**20var**	**30var**	**40var**	**50var**
T5	96.2	92.4	87.7	73.6	74.4	67.1	53.5	50.1
(v=8)	94.0	87.9	81.6	74.8	67.5	58.3	51.2	50.0
T5	93.9	92.7	89.7	79.0	78.6	71.2	54.7	50.1
(v=10)	89.7	86.3	82.5	76.7	70.0	60.1	51.4	50.0
T5	94.5	91.5	87.7	77.3	77.8	70.7	53.3	50.0
(v=12)	91.1	84.9	80.7	81.0	70.1	60.3	51.4	50.0
T5	98.6	96.0	92.6	85.0	86.5	84.9	69.8	59.1
(v=5,12)	98.1	93.6	89.6	88.5	80.7	72.7	61.4	51.8

5-12 variable models can solve 20-30 variable problems far above random chance, large differences between $\frac{easy}{hard}$ distribution.

Result 2: Models have limited scale invariance

- Models exhibit some degree of scale invariance, though lack generalization ability expected for robust deductive reasoning.

	main **GRL** (i.i.d)				o.o.d eval(20-50 variables)			
Model	**5var**	**8var**	**10var**	**12var**	**20var**	**30var**	**40var**	**50var**
T5	96.2	92.4	87.7	73.6	74.4	67.1	53.5	50.1
(v=8)	94.0	87.9	81.6	74.8	67.5	58.3	51.2	50.0
T5	93.9	92.7	89.7	79.0	78.6	71.2	54.7	50.1
(v=10)	89.7	86.3	82.5	76.7	70.0	60.1	51.4	50.0
T5	94.5	91.5	87.7	77.3	77.8	70.7	53.3	50.0
(v=12)	91.1	84.9	80.7	81.0	70.1	60.3	51.4	50.0
T5	98.6	96.0	92.6	85.0	86.5	84.9	69.8	59.1
(v=5,12)	98.1	93.6	89.6	88.5	80.7	72.7	61.4	51.8

Can models solve NL satisfiability problem? It depends on the distribution of problems being evaluated, results can look very different.

Result 2: Models have limited scale invariance

- Models exhibit some degree of scale invariance, though lack generalization ability expected for robust deductive reasoning.

	main **GRL** (i.i.d)				o.o.d eval(20-50 variables)			
Model	**5var**	**8var**	**10var**	**12var**	**20var**	**30var**	**40var**	**50var**
T5	96.2	92.4	87.7	73.6	74.4	67.1	53.5	50.1
(v=8)	94.0	87.9	81.6	74.8	67.5	58.3	51.2	50.0
T5	93.9	92.7	89.7	79.0	78.6	71.2	54.7	50.1
(v=10)	89.7	86.3	82.5	76.7	70.0	60.1	51.4	50.0
T5	94.5	91.5	87.7	77.3	77.8	70.7	53.3	50.0
(v=12)	91.1	84.9	80.7	81.0	70.1	60.3	51.4	50.0
T5	98.6	96.0	92.6	85.0	86.5	84.9	69.8	59.1
(v=5,12)	98.1	93.6	89.6	88.5	80.7	72.7	61.4	51.8

Result 2: Models have limited scale invariance

- Models exhibit some degree of scale invariance, though lack generalization ability expected for robust deductive reasoning.

	main **GRL** (i.i.d)				o.o.d eval(20-50 variables)			
Model	**5var**	**8var**	**10var**	**12var**	**20var**	**30var**	**40var**	**50var**
T5	96.2	92.4	87.7	73.6	74.4	67.1	53.5	50.1
(v=8)	94.0	87.9	81.6	74.8	67.5	58.3	51.2	50.0
T5	93.9	92.7	89.7	79.0	78.6	71.2	54.7	50.1
(v=10)	89.7	86.3	82.5	76.7	70.0	60.1	51.4	50.0
T5	94.5	91.5	87.7	77.3	77.8	70.7	53.3	50.0
(v=12)	91.1	84.9	80.7	81.0	70.1	60.3	51.4	50.0
T5	98.6	96.0	92.6	85.0	86.5	84.9	69.8	59.1
(v=5,12)	98.1	93.6	89.6	88.5	80.7	72.7	61.4	51.8

Can models solve NL satisfiability problem? Far from learning underlying algorithm, not scale-invariant. **Challenge:** how to improve this?

Effective sampling is important

- Experimented with different sampling strategies: sampling via **hard** vs. **easy** distributions, **naive** sampling (randomly selecting).

	Accuracy%	
Model (*sampling strategy*)	**easy**$_{5,10}$	**hard**$_{5,10}$
T5-GRL v=10 (*biased*)	88.4	77.1
T5-GRL v=10 (*naive*)	89.7	78.7
T5-GRL v=10 (*hard*)	92.4	86.4

Effective sampling is important

- Experimented with different sampling strategies: sampling via **hard** vs. **easy** distributions, **naive** sampling (randomly selecting).

Model (*sampling strategy*)	Accuracy% **easy**$_{5,10}$	**hard**$_{5,10}$
T5-GRL v=10 (*biased*)	88.4	77.1
T5-GRL v=10 (*naive*)	89.7	78.7
T5-GRL v=10 (*hard*)	92.4	86.4

Can models solve NL satisfiability problem? Depends critically on the distribution of SAT problems and on sampling strategy.

Effective sampling is important: RuleTaker

- Random SAT can be *retrofitted* to find *hard* instances in existing tasks such as RuleTaker (Clark et al., 2020) (RT).

		Model accuracy (%)	
evaluation	Majority	**RT**-T5	**RT**-RoBERTa
RuleTaker (RT) *(standard)*	43.0	97.5	98.7
Hard RT (SAT sampling)	50.0	57.7	59.6

Effective sampling is important: RuleTaker

- Random SAT can be *retrofitted* to find *hard* instances in existing tasks such as RuleTaker (Clark et al., 2020) (RT).

	Model accuracy (%)		
evaluation	Majority	**RT**-T5	**RT**-RoBERTa
RuleTaker (RT) *(standard)*	43.0	97.5	98.7
Hard RT (SAT sampling)	50.0	57.7	59.6

Another reminder: Naive sampling can yield misleading results. **Future:** Understanding the exact problem distributions of existing NLP tasks.

Conclusions

- Investigated methodology for probing rule reasoning in pre-trained transformers, map probing tasks to existing combinatorial problems.

Conclusions

- Investigated methodology for probing rule reasoning in pre-trained transformers, map probing tasks to existing combinatorial problems.

  Challenge Task: solving SAT problems in NL, created via hard distributions of random *kSAT*, harder than existing challenges.

Conclusions

- Investigated methodology for probing rule reasoning in pre-trained transformers, map probing tasks to existing combinatorial problems.

  Challenge Task: solving SAT problems in NL, created via hard distributions of random $kSAT$, harder than existing challenges.

- **Findings**: positive results on some sub-sets, though limited scale-invariance and grasp of underlying problem.

Conclusions

- Investigated methodology for probing rule reasoning in pre-trained transformers, map probing tasks to existing combinatorial problems.

  Challenge Task: solving SAT problems in NL, created via hard distributions of random *kSAT*, harder than existing challenges.

- **Findings**: positive results on some sub-sets, though limited scale-invariance and grasp of underlying problem.

  methodological: Results depend critically understanding target problem distribution, effective sampling strategy.

Conclusions

- Investigated methodology for probing rule reasoning in pre-trained transformers, map probing tasks to existing combinatorial problems.

  Challenge Task: solving SAT problems in NL, created via hard distributions of random *kSAT*, harder than existing challenges.

- **Findings**: positive results on some sub-sets, though limited scale-invariance and grasp of underlying problem.

  methodological: Results depend critically understanding target problem distribution, effective sampling strategy.

  open challenge: How to train models to be more robust, scale-invariant for reasoning?

A Final Lesson from empirical SAT

Can SAT solvers (empirically) solve hard SAT problems?

> Random formulas have been used by many researchers to empirically evaluate the performance of SAT testing programs. **The value of such studies depends upon careful selection of of formula distribution**... When using random formulas, an extensive enough study of the distribution's parameter space must be carried out ... if the results are to be meaningful.

Mitchell and Levesque (1996) *Some pitfalls for experiments with random SAT*

A Final Lesson from empirical SAT

Can SAT solvers (empirically) solve hard SAT problems?

> Random formulas have been used by many researchers to empirically evaluate the performance of SAT testing programs. **The value of such studies depends upon careful selection of of formula distribution**... When using random formulas, an extensive enough study of the distribution's parameter space must be carried out ... if the results are to be meaningful.

Mitchell and Levesque (1996) *Some pitfalls for experiments with random SAT*

- **Same for probing**: understanding the target problem distribution and how to sample hard cases is essential for understanding model behavior.

Thank you.

References I

Clark, P., Tafjord, O., and Richardson, K. (2020). Transformers as soft reasoners over language. *Proceedings of IJCAI*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Kautz, H. A., Selman, B., et al. (1992). Planning as satisfiability. In *ECAI*, volume 92, pages 359–363. Citeseer.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Mitchell, D. G. and Levesque, H. J. (1996). Some pitfalls for experimenters with random sat. *Artificial Intelligence*, 81(1-2):111–125.

Pratt-Hartmann, I. (2004). Fragments of language. *Journal of Logic, Language and Information*, 13(2):207–223.

Pratt-Hartmann, I. (2015). Semantic complexity in natural language. *The Handbook of Contemporary Semantic Theory*, page 429.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

References II

Selman, B., Mitchell, D. G., and Levesque, H. J. (1996). Generating hard satisfiability problems. *Artificial intelligence*, 81(1-2):17–29.

Shin, R., Kant, N., Gupta, K., Bender, C., Trabucco, B., Singh, R., and Song, D. (2019). Synthetic datasets for neural program synthesis. *arXiv preprint arXiv:1912.12345*.