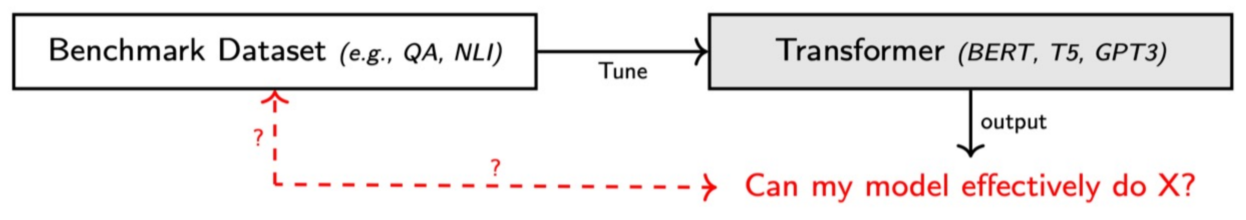
Pushing the Limits of Rule Reasoning in Transformers through Natural Language Satisfiability

Kyle Richardson, Ashish Sabharwal

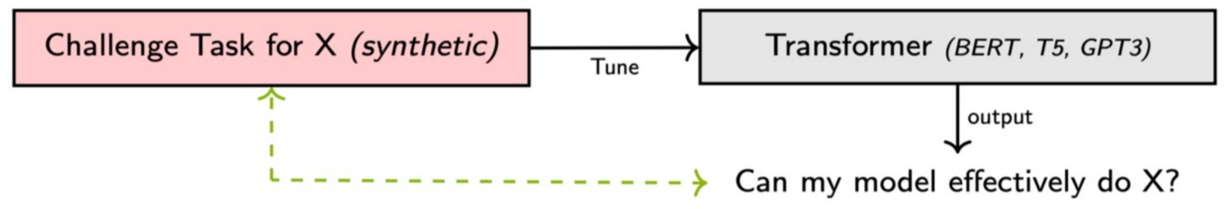
1. Question: What types of algorithmic problems can transformers in NLP solve?

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



Problem: Models and datasets are black boxes, hard to interpret; serious issue for understanding model safety and correctness.

Behavioral Testing *(This work)*



Behavioral Testing: understanding model behavior/competence and limitations through systematically constructed challenge tasks.

- **Rule Reasoning (RuleTaker):** can transformers learn correct deductive reasoning over logical theories expressed in natural language? (Clark et al. 2020)
- **Why Logic?** Fundamental to other forms of reasoning, basic *information-aggregation (IA)* problem, understand limits of IA in transformers.

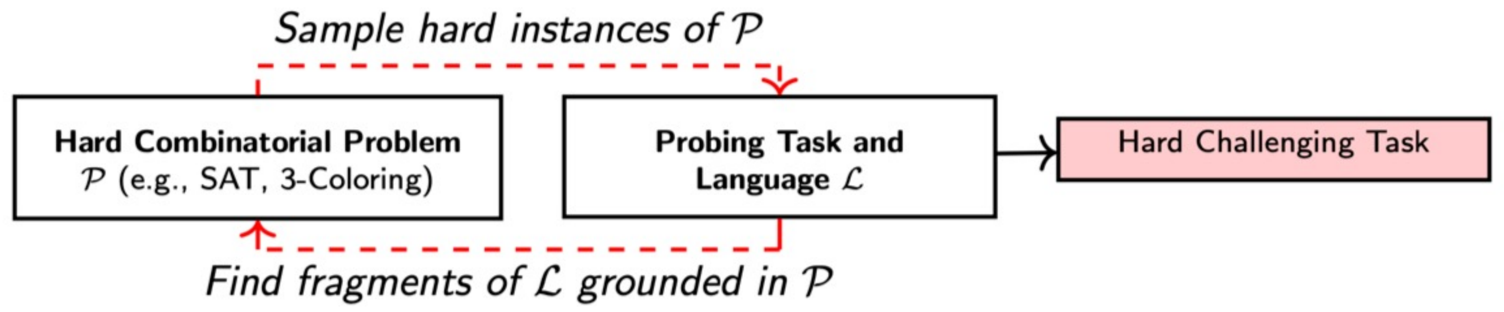
NL Theory (RuleTaker)	$\Gamma_{\mathbf{NL}}$ ={ Bob is round. Alan is blue, rough and young. **If someone is round then they are big.** All rough people are green. **Big people are not green.** }
NL Query	Bob is not green? ✓

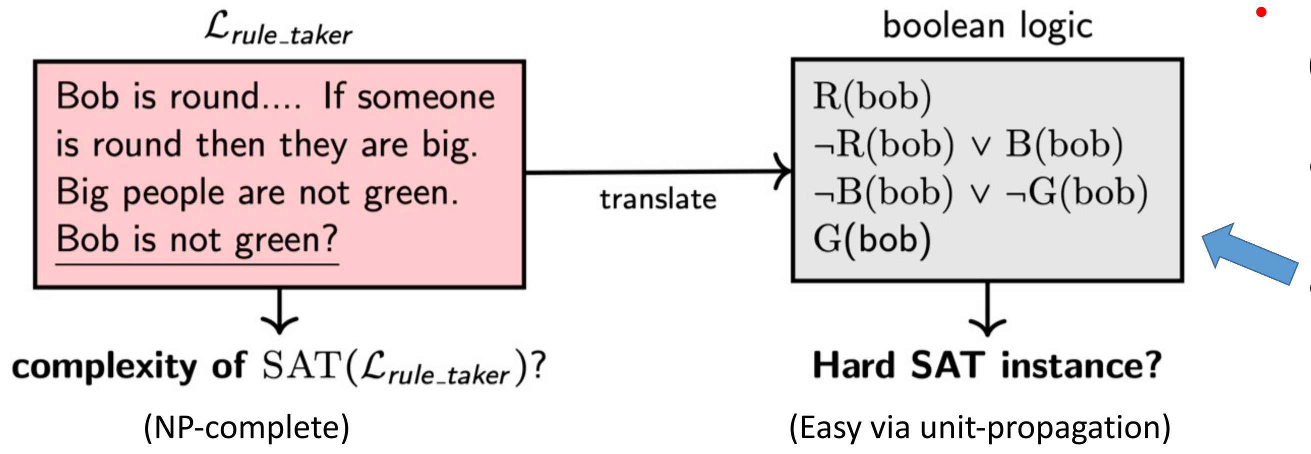
- **Desiderata:** Behavioral tests should faithfully capture the target problem space, include the hardest cases for results to be meaningful.

Pushing the Limits *(this work) How difficult can we make the problems? General framework for ensuring task hardness and obtaining more reliable empirical performance bounds.*

2. Framework: Pushing the limits by working from known combinatorial problems

- Focus on probing tasks and languages that are grounded in known hard combinatorial problems. **advantages:**
  - *Understand the general complexity of target tasks.*
  - *Work from known hard problem distributions to effectively construct and sample hard challenge tasks.*





- **Example:** Focus on deductive reasoning tasks grounded in Boolean satisfiability (SAT) and 3SAT:
  - ***General task hardness:*** *classical NP-complete problem;* **Known hard** distributions, random kSAT (Selman et al. 1996)
  - ***Can study the complexity of existing deductive reasoning tasks*** *via SAT.*

Observation: *Tasks like RuleTaker focus narrowly on easy deductive reasoning problems, ad-hoc sampling yields easy cases, misleading results*

3. New Tasks for rule reasoning and Natural Language Satisfiability

- **Natural Language Satisfiability (NLSat):** deductive reasoning task that involves determining whether a set of rules in natural language has a satisfying assignment (Pratt-Hartmann 2004); mirrors ordinary propositional SAT.

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:

$(\neg A \wedge \neg B) \rightarrow C \equiv A \vee B \vee C$

If not apple and not carrot then pear. If not apple and carrot then pear. If apple and carrot then steak. (apple=T,carrot=F,pear=T,...)

$(\neg A(j) \wedge \neg B(j)) \rightarrow G(j)$

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,..)

Two rule fragments investigated:

Grounded Rule Language (GRL): translation of 3SAT into logically equivalent NL propositional rules, nouns as variables.

Relative Clause Fragment (RCL): 3SAT clauses to relative clause constructions, nouns as variables (Pratt-Hartmann 2004) .

- **Task Construction and Sampling:** Translated from hard 5->12 variable 3SAT instances sampled from critical phase-change region to corresponding NL rule templates, comparable size to RuleTaker, increased empirical complexity.
- **Out-of-domain set**: evaluate scale-invariance, ability of model to generalize to problems of larger scope/# variables.

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
$\mathbf{GRL\text{-}eval}_{20,50}$	17k	yes	22.0/13.0	29.3/13.0

4. Experiments and General Findings

- **Task:** binary prediction task (**sat** vs. **unsat,** Acc %); standard fine-tuning set up from (Clark et al. 2020). **Transformer models:** T5-large (Raffel et al. 2020) and RoBERTa-large (Liu et al. 2019) .
- Models can robustly solve some new problems (i.i.d setting) despite increased difficulty, **important caveats**:
  - *Clear degradation of performance as a function of # variables; models lack training efficiency.*
  - *Still room for improvement, not a solved task.*

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
$\mathbf{T5}_{5,7,8,10,12}$	98.0	95.4	94.3	90.7	88.3	93.4
$\mathbf{RoBERTa}_{5,7,8,10,12}$	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
$\mathbf{T5}_{16,70}$	95.9	95.3	94.7	92.9	94.7
$\mathbf{RoBERTa}_{16,70}$	96.0	95.9	94.9	94.0	95.2

- **Generalization:** exhibit some scale-invariance**,** still lack the kind of generalization skills we would expect for robust deductive reasoning.

Challenge: *how can we train models to be scale-invariant and robust algorithmic learners?*

	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

- Model performance looks very different depending on training and testing problem distribution, sampling/understanding problem distr. is very important.
  - **Discovering hard instances of existing tasks:** Showed how to retrofit random kSAT instances to find hard RuleTaker instances, more effective sampling.

Results on easy (84.9) vs. hard (72.7) 30 variable problems, distribution matters!

Conclusions

- Investigated the ability of transformers to learn deductive rule reasoning in natural language.
  - ***Novel methodology****: ground tasks in known combinatorial problems, ensure hardness, work from hard problem instances.*
  - ***Pushing the limits****: tested on a new suite of textual deductive reasoning tasks grounded in Boolean SAT, sampled using random kSAT.*
- **General Results***:* Models can solve reasoning tasks that exceed complexity of existing benchmarks, **though**:
  - *Lack robustness and scale-invariance; seem far from learning underlying reasoning algorithms.*
  - *Results are only meaningful with an understanding of the target problem distribution; naïve sampling can yield misleading results and harm model robustness.*

Selected References

Clark, P.; Tafjord, O.; Richardson, K. (2020) **Transformers as Soft Reasoners over Language.** *Proceedings of IJCAI*

Selman, B.; Mitchell, DF, Levesque, HJ. (1996) **Generating hard satisfiability problems.** *Artificial Intelligence Journal.*

Pratt-Hartmann, I. (2004) **Fragments of Language.** *Journal of Language, Logic and Information.*

Raffel, C. et al. (2020) **Exploring the Limits of Transfer Learning with Text-to-Text Transformer.** *Journal of Machine Learning Research*