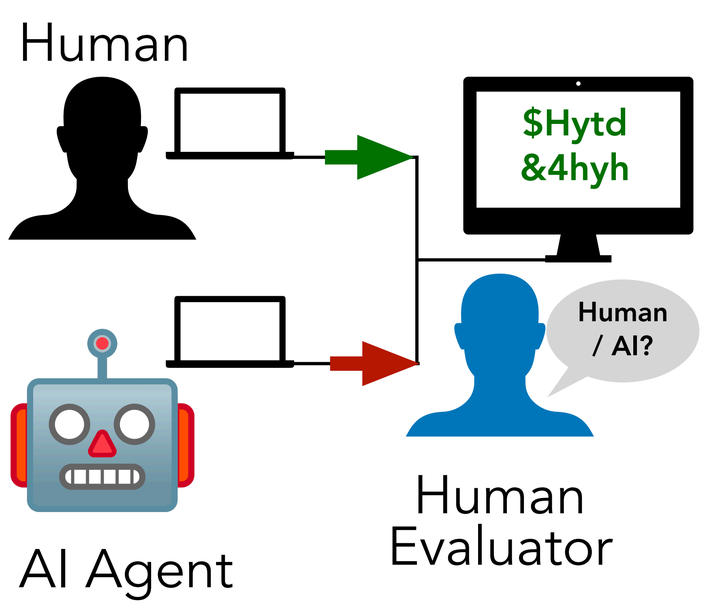
Pseudointelligence: A Unifying Framework for Language Model Evaluation

Shikhar Murty*, Orr Paradise*, Pratyusha Sharma*

Stanford NLP · MIT · Berkeley UNIVERSITY OF CALIFORNIA

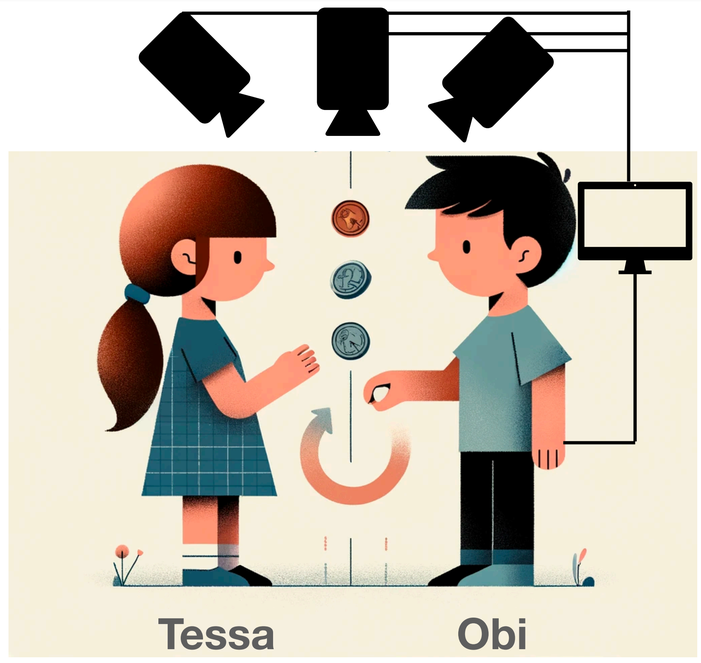
The Turing Test



Case 1: Obi is calling the coin based only on the info available to him from eye sight.

Case 2: Obi has access to sensors that measure the initial state of Tessa's coin, and a computer that performs complex calculations in milliseconds.

Pseudorandomness



Definition [Yao '82, Blum Micali 84]
Distribution $\mathscr{P}$ is ε-pseudorandom against a class of distinguishers D if for every $d \in D$:

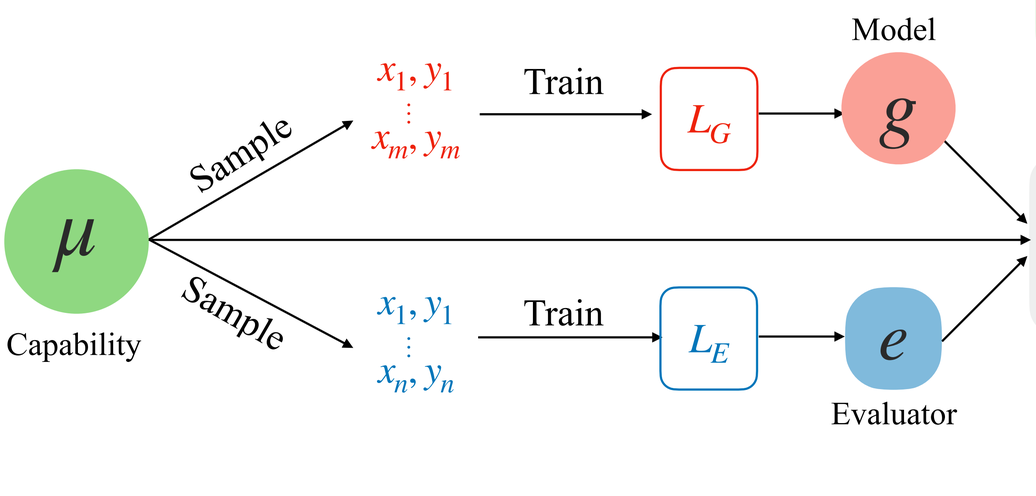
$$\left| \Pr_{x \leftarrow \mathscr{P}} [d(x) \text{ accepts}] - \Pr_{x \leftarrow \mathscr{U}} [d(x) \text{ accepts}] \right| < \epsilon .$$

$\mathscr{U}$: Unif. distr. over a finite set

- Decades of extensive research
- At the foundation of modern crypto

Pseudointelligence: Meta-evaluation meets pseudorandomness

Sampling, training and distinguishing



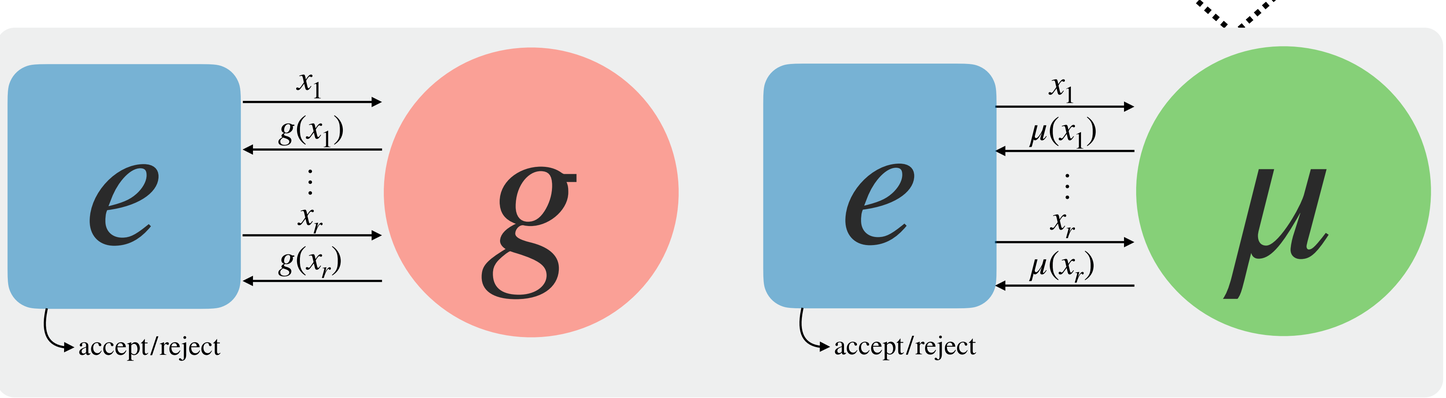
μ Capability
$\mathscr{M}$ Capability class

g Model
L_G Model learner
$m = m(\varepsilon, \delta)$ Model sample complexity

e Evaluator
L_E Evaluator learner
$n = n(\varepsilon, \delta)$ Evaluator sample complexity
$r = r(\varepsilon, \delta)$ Evaluation round complexity

$$\text{dist}_e(g, \mu) < \varepsilon$$

The **dist**inguishing experiment



Definition

L_G is *pseudointellgent* w.r.t L_E and $\mathscr{M}$ if

$$\forall \mu \in \mathscr{M},\ \forall \varepsilon, \delta \in (0,1) \quad \Pr\left[\text{dist}_e(g, \mu) \leq \varepsilon\right] \geq 1 - \delta .$$

Over samples, learned model, learned evaluator

$\left|\Pr[e \text{ accepts } g] - \Pr[e \text{ accepts } \mu]\right|$

Over queries and responses

Key resources modeled:

- Sample complexity
- Learning comput. power
- Learner expressivity
- Forward-pass complexity

Are we evaluating Language Models Correctly?

Casting current LM evaluation into pseudo-intelligence

Dynamic / Adversarial Evaluation:

- L_E uses auxiliary model $\hat{g}$ to search for challenge examples in set seed S.
- Based on the quality of $\hat{g}$, we can get increasingly harder datasets.
- Central resources: size of seed set, complexity of $\hat{g}$.

Model-based evaluation

- LMs are used to generate evaluation sets based on templates.
- Optionally, model generated test sets can be filtered out by human raters.
- Central resources: size of LM, number of queries to human raters.

Self-evaluation:

- Here the model is pitted against itself, serving as both the evaluator and the generator.
- Our framework makes self-evaluation invalid since L_E and L_G must receive i.i.d training samples, so self-evaluation cannot be used as claim of model capability.

FAQs

Differences from the Turing Test?
Pseudointelligence is a complexity-theoretic analogue of the Turing Test, though evaluators need not be human.

Differences from PAC Learning?
PAC learning only has a learner. Pseudo-intelligence is defined with respect to a learner, **and** a (learned) evaluator that operates over multiple non i.i.d rounds.

What's the optimal evaluator?
There is no One True Evaluator.

…then what is missing in LM evaluation?
Evaluators whose resources are tied-to (and scale up with) the resources of the LM. Proven or empirically-verified scaling laws.