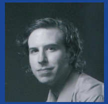
Provable Limitations of Acquiring Meaning from Ungrounded Form: What Will Future Language Models Understand?

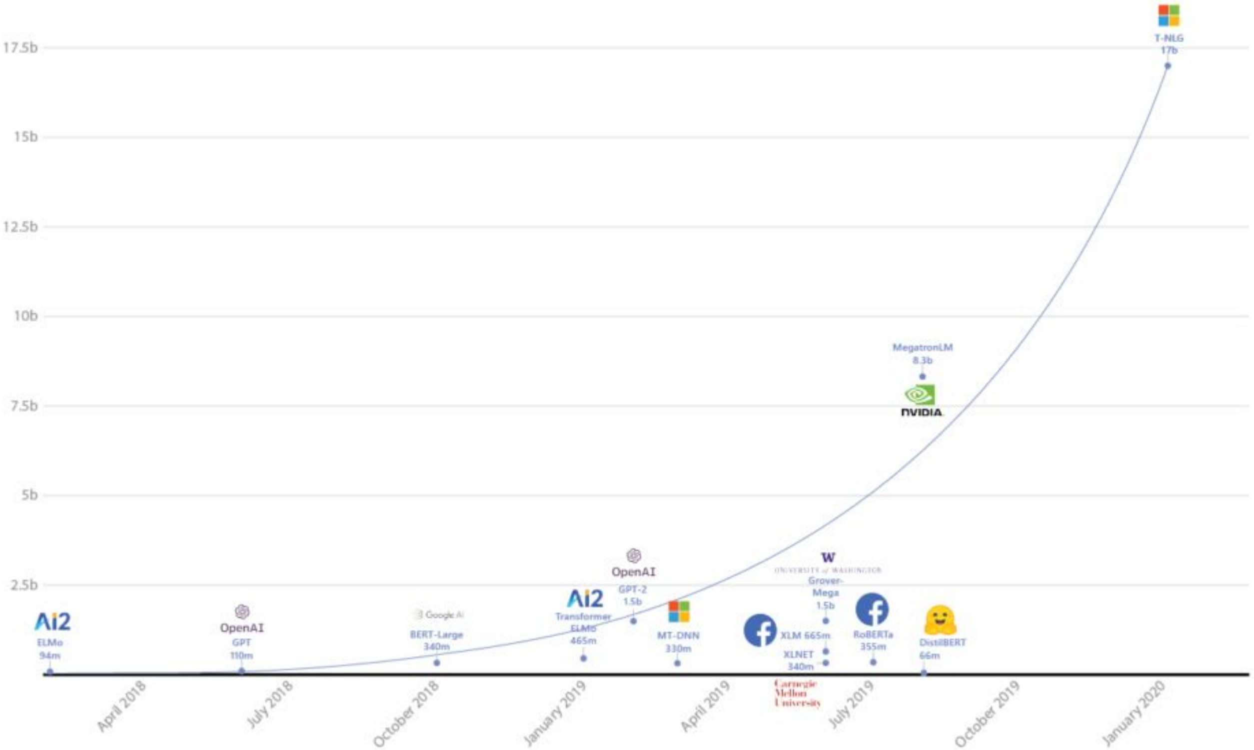

William Merrill


Yoav Goldberg


Roy Schwartz


Noah A. Smith

Pretrained LMs are a successful paradigm for NLP



But is pretraining limited?

"A more fundamental limitation of…scaling up any LM-like model, whether autoregressive or bidirectional – is that it may eventually run into (or could already be running into) the limits of the pretraining objective."

Language Models are Few-Shot Learners

Tom B. Brown* Benjamin Mann* Nick Ryder* Melanie Subbiah*

Jared Kaplan† Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry

Amanda Askell Sandhini Agarwal Ariel Herbert-Voss Gretchen Krueger Tom Henighan

Rewon Child Aditya Ramesh Daniel M. Ziegler Jeffrey Wu Clemens Winter

Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray

Benjamin Chess Jack Clark Christopher Berner

Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

The form/meaning debate

Climbing towards NLU:
On Meaning, Form, and Understanding in the Age of Data

Emily M. Bender
University of Washington
Department of Linguistics
ebender@uw.edu

Alexander Koller
Saarland University
Dept. of Language Science and Technology
koller@coli.uni-saarland.de

- Language models are trained on “form”
- But understanding is about “meaning”

Contributions

We formalize and answer a key thought experiment raised by Bender and Koller:

Q: **In principle, can a LM trained on code learn to execute code?**

A: Under some conditions, yes, but in general, no.

Limitations: results depend on our definitions and assumptions

How could language models learn to execute code?

Which is more likely in Python?

```python
assert 1 + 1 == 4
```

```python
assert 2 + 2 == 4
```

Assertion argument (Michael, 2020; Potts, 2020)

1. **Assumption:** programmers intend to write true assertions

2. Therefore, assertions are more likely to appear in training data if they are true

3. ∴ Language model can learn semantic features of expressions (e.g., which = 4)

```
def context():
    x = 2 + 2
    y = 1 + 1
    assert x == 4
```

But are assertions enough to *execute* code?

Our main results:

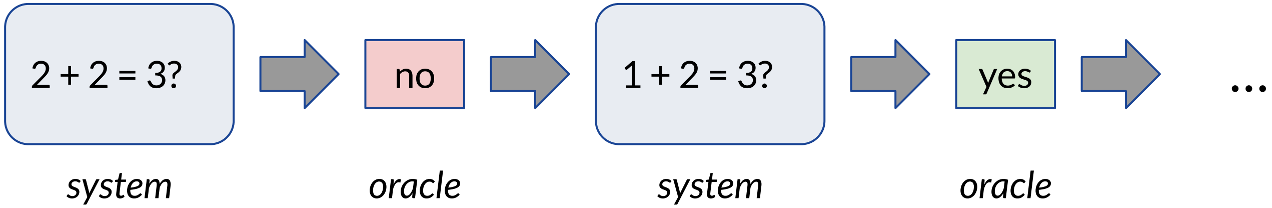
1. Can learn to execute all **transparent languages** using assertions

2. **In general**, cannot learn to execute all languages using assertions

Learning by assertions

Imagine language model learns by *querying* an *assert oracle* function

$$\aleph_L(e, e' \mid \kappa) = \begin{cases} 1 & \text{if } [\![e \mid \kappa]\!]_L = [\![e' \mid \kappa]\!]_L \\ 0 & \text{otherwise.} \end{cases}$$

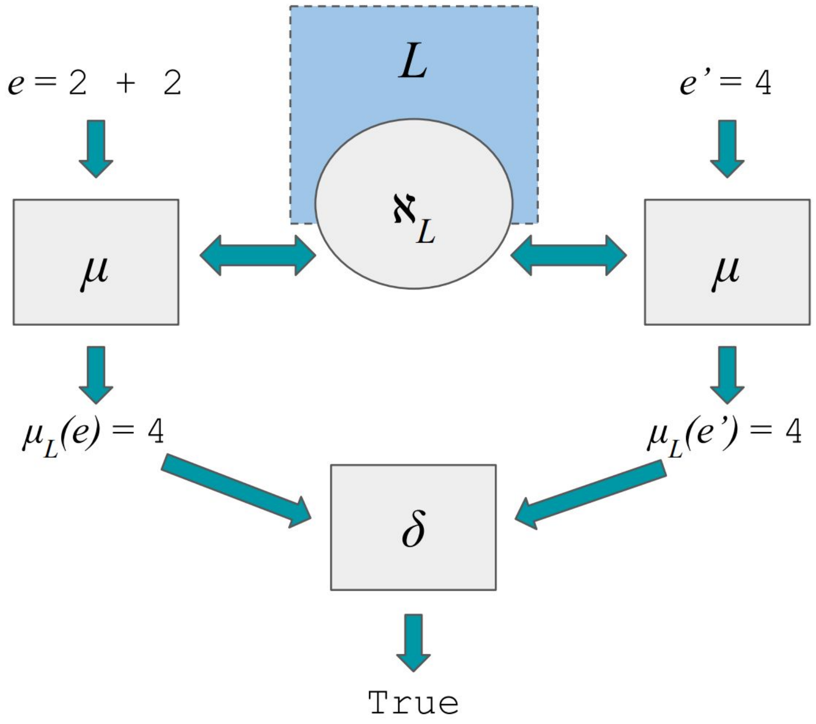


Cf. query learning setup of Angluin (1987)

Defining success: emulation

- A system *emulates* a language if there exists a probe that says whether two expressions are equal given the representations the system produces

- We will analyze what types of languages are *emulatable*



Case 1: transparent languages

A language is transparent if every expression has a canonical semantic value across all contexts

Transparent: $2 + 2$ means 4

Non-transparent: $2 + x$ means ?

#1: transparent languages are emulatable

Theorem 1 (Informal). There exists an algorithm for μ that will emulate any transparent language using assertion queries.

```
from typing import Callable

AssertType = Callable[[str, str, str, str], bool]

def emulate(expr: str, asserteq: AssertType) -> int:
    for idx, cand in enumerate(all_strings()):
        if asserteq(expr, cand, "", ""):
            return idx
```

#2: (some) non-transparent languages are not emulatable

Theorem 2 (Informal). There exists a class of non-transparent languages that no computable function μ can emulate.

Summary of contributions

- First provable “limits of the pretraining objective”
- Formal resolution to the Bender and Koller code thought experiment
    - Transparent yes
    - Non-transparent no

Towards natural language

- Assertion-like contexts exist for natural language also

$$p(\text{New York is urban}) > p(\text{New York is rural})$$

- General takeaway: capabilities of LMs depends on how people create training data (their *intents*, specifically)

(Section 6 of paper: “Towards Natural Language”)

Ongoing research

- Can current LMs learn by assertions?

Zhaofeng Wu, PYI on AllenNLP

- Can an LM trained on natural language implicitly solve NLI?
  - Answer TBD; got some promising theorems
  - Rational Speech Acts theory (Goodman et al., 2016)

Thank you!

Acknowledgments:
Mark-Jan Nederhof, Dana Angluin, Matt Gardner, Eran Yahav, Zachary Tatlock, Kyle Richardson, Ruiqi Zhong, Samuel Bowman, Christopher Potts, Thomas Icard, and Zhaofeng Wu