Lipstick on a Pig:
Using Language Models as Few-Shot Learners

Sameer Singh
sameersingh.org
Natural Language Processing Pipeline
Natural Language Processing Pipeline

Unlabeled Data

→

Pre-Training

→

Finetuning

Task Data

→

Task Model TM
Natural Language Processing Pipeline

Unlabeled Data

Pre-Training

Finetuning

Task Model

Task Data
What’s next? Get rid of finetuning!

Diagram:
- Unlabeled Data → Pre-Training → Prompting → Task Model (TM)
Manual Prompts: Sentiment Analysis

Input: Amazing movie!  Sentiment: [MASK]

Language model (LM)

\[ P(\text{“positive”}) > P(\text{“negative”}) \]
In-Context Learning (Few-Shot Learning!)

Task Model (TM)

Input: Subpar acting. Sentiment: negative
Input: Beautiful film. Sentiment: positive
Input: Amazing movie! Sentiment: [MASK]

Language model LM

$p(\text{"positive"}) > p(\text{"negative"})$
Why is in-context learning interesting?

• Academically interesting
  • What do language models learn? How do we control them?

• Practically relevant (with GPT-3)
  • effective with ~0-16 examples
  • serve one model for many tasks
  • no ML expertise needed

• Related to other ways of adapting language models
  • AutoPrompt*: customized phrases to adapt LMs
  • Prompt/prefix tuning: continuous changes to input/weights
  • Increasingly more accurate and useful

Today’s Talk

How robust are these capabilities to the pretraining data?

How robust are these capabilities to the pretraining data?

What are the biases introduced by this format?

Input: Subpar acting. Sentiment: Negative
Input: Beautiful film. Sentiment: Positive
Input: Amazing. Sentiment: [MASK]

Language model
LM

P(“positive”) > P(“negative”)
How robust are these capabilities to the pretraining data?

Input: Subpar acting. Sentiment: Negative
Input: Beautiful film. Sentiment: Positive
Input: Amazing. Sentiment: [MASK]

Language model LM

\[ P(\text{"positive"}) > P(\text{"negative"}) \]

Components Of The Prompt

Prompt Format

Input: Subpar acting. Sentiment: negative
Input: Beautiful film. Sentiment: positive
Input: Amazing. Sentiment:

Q: What’s the sentiment of “Subpar acting”? A: negative
Q: What’s the sentiment of “Beautiful film”? A: positive
Q: What’s the sentiment of “Amazing”? A:
## Components Of The Prompt

### Training Example Selection

<table>
<thead>
<tr>
<th>Input</th>
<th>Sentiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subpar acting.</td>
<td>negative</td>
</tr>
<tr>
<td>Beautiful film.</td>
<td>positive</td>
</tr>
<tr>
<td>Amazing.</td>
<td></td>
</tr>
<tr>
<td>Good film.</td>
<td>positive</td>
</tr>
<tr>
<td>Don’t watch.</td>
<td>negative</td>
</tr>
<tr>
<td>Amazing.</td>
<td></td>
</tr>
</tbody>
</table>
Components Of The Prompt

Training Example Perturbation

<table>
<thead>
<tr>
<th>Input</th>
<th>Sentiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subpar acting.</td>
<td>negative</td>
</tr>
<tr>
<td>Beautiful film.</td>
<td>positive</td>
</tr>
<tr>
<td>Amazing.</td>
<td></td>
</tr>
<tr>
<td>Beautiful film.</td>
<td>positive</td>
</tr>
<tr>
<td>Subpar acting.</td>
<td>negative</td>
</tr>
<tr>
<td>Amazing.</td>
<td></td>
</tr>
</tbody>
</table>
Accuracy Is Highly Sensitive To Prompt Design

Example Permutation Impacts Accuracy
Accuracy Is Highly Sensitive To Prompt Design

Example Selection Impacts Accuracy
Accuracy Is Highly Sensitive To Prompt Design

Format #1
- Input: Subpar acting.
- Sentiment: negative
- Input: Beautiful film.
- Sentiment: positive
- Input: Amazing.

Format #2
- Subpar acting. I hated the movie.
- Beautiful film. I liked the movie.
- Amazing.

Format #10
- Review: Subpar acting. Stars: 0
- Review: Beautiful film. Stars: 5
- Review: Amazing.

Example Format Impacts Accuracy
In-Context Learning

Input: Meh movie. Sentiment: Negative
Input: Subpar acting. Sentiment: Negative
Input: Beautiful film. Sentiment: Positive
Input: Amazing. Sentiment: [MASK]

Language model
LM

$P(\text{"good"]) > P(\text{"bad"})$

Format matters
Selection matters
Order matters
Word choice matters
Majority Label Bias

Frequency of Positive Test Predictions

- 100 4/4 Positive
-  56 3/4 Positive
-  37 2/4 Positive
-   20 1/4 Positive
-    0 0/4 Positive

Frequent training answers dominate predictions
Recency Bias

Frequency of Positive Predictions

- NPPP: 90
- PNPP: 62
- PPNP: 60
- PPPN: 12

Examples near end of prompt dominate predictions
Common Token Bias

What topic is the following text about? The Model T was released by Ford in 1908. Answer:

```
Common n-grams dominate predictions
```

<table>
<thead>
<tr>
<th>Token</th>
<th>Web (%)</th>
<th>Label (%)</th>
<th>Prediction (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>book</td>
<td>0.026</td>
<td>9</td>
<td>29</td>
</tr>
<tr>
<td>transportation</td>
<td>0.0000006</td>
<td>9</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Token</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>book</td>
<td>0.35</td>
</tr>
<tr>
<td>transportation</td>
<td>0.23</td>
</tr>
<tr>
<td>school</td>
<td>0.11</td>
</tr>
<tr>
<td>village</td>
<td>0.03</td>
</tr>
<tr>
<td>company</td>
<td>0.02</td>
</tr>
</tbody>
</table>
Contextual Calibration of Language Models

Input: Subpar acting. Sentiment: Negative
Input: Beautiful film. Sentiment: Positive
Input: Amazing. Sentiment: _______

Input: Subpar acting. Sentiment: Negative
Input: Beautiful film. Sentiment: Positive
Input: N/A. Sentiment: _______

“meaningless” input, but full context
More Accurate and Stable!

11 different datasets, 0-16 shots, GPT-2 and GPT-3 models

Different Training Examples

Improved mean and worst accuracy
Reduced variance for selection and ordering

Different Prompt Formats

Reduced variance for formats
Contextual Calibration for In-context Learning

+ *extremely* simple fix
+ boosts accuracy, reduces variance

- Calibration doesn’t completely solve brittleness
- Independent of the pretraining corpus
Today’s Talk

How robust are these capabilities to the pretraining data?

What are the biases introduced by this format?

Input: Subpar acting. Sentiment: Negative
Input: Beautiful film. Sentiment: Positive
Input: Amazing. Sentiment: [MASK]

Unlabeled Data
Pre-Training
Language model LM

$P(“positive”) > P(“negative”)
Y. Razeghi, R. Logan, M. Gardner, S. Singh.
Impact of Pretraining Term Frequencies on Few-Shot Reasoning
ArXiV. 2022

How robust are these capabilities to the pretraining data?

What are the biases introduced by this format?

Input: Subpar acting. Sentiment: Negative
Input: Beautiful film. Sentiment: Positive
Input: Amazing. Sentiment: [MASK]

Unlabeled Data

Pre-Training

Language model LM

P(“positive”) > P(“negative”)
Reasoning and In-context Learning

• Instead of downstream classification, let’s focus on Reasoning
  • Difficult to define precisely, but it’s about inference
  • Go beyond regurgitation of what it has already seen
  • Feels different from memorization of facts

• Language Models need to perform reasoning
  - Went for a long lunch today, it lasted ______.
  - Alex loves chewing bones, which is not a surprise, given that he’s a ______.
  - I wanted it in 10 days, but it took 2 weeks, which made me ______.

• And in-context few-shot reasoning is fairly accurate!
  • But how much of this performance is robust reasoning?
Numerical Reasoning

- One of the fundamental reasoning tasks
  - Version of common-sense reasoning
- Piece of the Neural vs Symbolic debate
  - Can LMs learn to multiply numbers?
- Good few-shot performance by big LMs
  - LMs are not explicitly trained for them

Example from GPT-J blog:

Prompt -------
What is 75×10?
--------------
Output:
750
What is -0.002 take away 72.75?
-72.752
Calculate -0.5 - 1039.
-1039.5
What is the difference between -1360 and 2?
1362
What is -27.95 less than -20?
7.95
Calculate -0.3 + -169.
-169.3
What is 0.7 minus 0.05?
0.65
Calculate -2 + 0.0899.
-1.9101
Motivating Example: Multiplication

• Good performance but not always correct

Q: What is 24 times 18? A: 432 ✓ \[ \Omega(24) \approx 10^7 \]
Q: What is 23 times 18? A: 462 ✗ \[ \Omega(23) \approx 10^6 \]

Why does the model perform differently on different instances?

Hypothesis: maybe it depends on unigram statistics in pretraining?
Motivating Example: Multiplication

- First operand: numbers between 0-99
- Accuracy averaged over:
  - 5 choices of training instances
  - second operand: numbers in 1-50

Q: What is 24 times \([x]\)? A: __
Q: What is 23 times \([x]\)? A: __
Motivating Example: Multiplication

- First operand: numbers between 0-99
- Accuracy averaged over:
  - 5 choices of training instances
  - second operands as numbers in 1-50

Q: What is 24 times [x]? A: __
Q: What is 23 times [x]? A: __
Pipeline for Evaluating this Effect

Pretraining Corpus

Count Occurrences → Term Counts

Reasoning Queries:
- $24 \times 18 = ?$ (432)
- $23 \times 18 = ?$ (414)
- 60 hours → mins?
  - (3600)

Prompt Templates:

Q: What is $[x_1]$ times $[x_2]$? A: $[y]$

Q: What is 24 times 18? A:
Analysis of Language Models

Q: What is 24 times 18?
A: 432

Q: What is 23 times 18?
A: 462

Q: What is 24 times 18?
A: 432

Q: What is 23 times 18?
A: 462

Term Counts

<table>
<thead>
<tr>
<th>Term</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>(24)</td>
<td>$10^7$</td>
</tr>
<tr>
<td>(23)</td>
<td>$10^6$</td>
</tr>
<tr>
<td>(60, hour)</td>
<td>$10^6$</td>
</tr>
</tbody>
</table>
Metric: Performance Gap

- Difference in average accuracy of the instances in the top and bottom quantiles of the distribution over term frequencies.

\[ \Delta(\Omega) = \text{Acc}(\Omega_{>90\%}) - \text{Acc}(\Omega_{<10\%}) \]
Experiment Setup

• EleutherAI GPT-models
  • GPT-J-6B
  • GPT-Neo-2.7B
  • GPT-Neo-1.3B

Pretrained on Pile Dataset

• 800GB pretraining corpus
• Publicly available!

Training examples in the prompt:

• Randomly choose $k$ examples
• 5 choice of random seeds
Arithmetic Reasoning

Q: What is 24 plus [x]?
A: __

Q: What is 24 times [x]?
A: __

<table>
<thead>
<tr>
<th>$k$</th>
<th>Multiplication Acc.</th>
<th>$\Delta_1$</th>
<th>Addition Acc.</th>
<th>$\Delta_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5.4</td>
<td>18.0</td>
<td>1.6</td>
<td>8.4</td>
</tr>
<tr>
<td>2</td>
<td>35.9</td>
<td>77.6</td>
<td>88.2</td>
<td>16.8</td>
</tr>
<tr>
<td>4</td>
<td>39.2</td>
<td>70.8</td>
<td>91.4</td>
<td>15.0</td>
</tr>
<tr>
<td>8</td>
<td>42.9</td>
<td>74.6</td>
<td>89.6</td>
<td>16.3</td>
</tr>
<tr>
<td>16</td>
<td>40.9</td>
<td>73.3</td>
<td>88.6</td>
<td>16.4</td>
</tr>
</tbody>
</table>
Q: What is $24 \# [x]$?  A: ___

<table>
<thead>
<tr>
<th>$k$</th>
<th>Multiplication (#)</th>
<th>Addition (#)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc.</td>
<td>$\Delta_1$</td>
</tr>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>3.1</td>
<td>14.1</td>
</tr>
<tr>
<td>4</td>
<td>5.7</td>
<td>20.9</td>
</tr>
<tr>
<td>8</td>
<td>9.4</td>
<td>31.3</td>
</tr>
<tr>
<td>16</td>
<td>11.0</td>
<td>39.6</td>
</tr>
</tbody>
</table>
Time Unit Conversion

- Minute to Seconds
  Q: What is 24 minutes in seconds? A: __
- Hour to Minutes
  Q: What is 24 hours in minutes? A: __
- Day to Hour
  Q: What is 24 days in hours? A: __
- Week to Day
  Q: What is 24 weeks in days? A: __
- Month to Week
  Q: What is 24 months in weeks? A: __
- Year to Month
  Q: What is 24 years in months? A: __
- Decade to Year
  Q: What is 24 decades in years? A: __
Time Unit Conversion

Year to Month

Decade to Year
## Time Unit Conversion

<table>
<thead>
<tr>
<th>$k$</th>
<th>Min→Sec</th>
<th>Hour→Min</th>
<th>Day→Hour</th>
<th>Week→Day</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc.</td>
<td>$\Delta_{1,2}$</td>
<td>Acc.</td>
<td>$\Delta_{1,2}$</td>
</tr>
<tr>
<td>0</td>
<td>1.3</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>25.5</td>
<td>62.5</td>
<td>19.4</td>
<td>58.0</td>
</tr>
<tr>
<td>4</td>
<td>35.5</td>
<td>60.0</td>
<td>29.1</td>
<td>76.4</td>
</tr>
<tr>
<td>8</td>
<td>49.9</td>
<td>72.1</td>
<td>36.3</td>
<td>74.6</td>
</tr>
<tr>
<td>16</td>
<td>58.4</td>
<td>82.7</td>
<td>42.8</td>
<td>80.1</td>
</tr>
</tbody>
</table>
Effect of Model Size

- As we increase size of model
  - Models get more accurate
  - But, more impacted by pretraining
- Number of shots is inconsistent
  - more training doesn’t lead to robust reasoning by itself
- Difficult to detangle accuracy
  - By scale itself is not a solution
Effect of Pretraining on Reasoning
+ high impact on reasoning performance
+ raises questions about how to design, and evaluate, LMs

- we are not making a causal statement about reasoning
- only evaluated on numerical reasoning
Today’s Talk

How robust are these capabilities to the pretraining data?

What are the biases introduced by this format?

Input: Subpar acting. Sentiment: Negative
Input: Beautiful film. Sentiment: Positive
Input: Amazing. Sentiment: [MASK]

Language model LM

$P(\text{“positive”}) > P(\text{“negative”})$
What Can We Do?

• More **diverse** data is better!
  • Will suffer from Zipf's Law
  • Future is **more unique** than the past

• **Augmentation** during pretraining?
  • Add data to address **specific** reasoning
  • Good for fixing the issues we have **observed**
  • Doesn’t feel like the **end goal**

What Can We Do?

• Maybe scaling further will help?
  • Ultimately, they will just generalize perfectly?

• Neuro-symbolic language modeling?
  • Give LMS access to KGs, calculators, etc.
  • Barack’s Wife Hillary … [ACL 2019] *

• Other losses for pretraining?
  • Should words really compete with each other?

* https://arxiv.org/abs/1906.07241
Thank you!

@sameer_
sameer@uci.edu
sameersingh.org