Foundation Models
(Large Pre-trained Models)

Sarah Pratt
Language Models

Encoders vs Decoders vs Encoder-Decoder Models
Prompting (zero-shot, in-context, chain-of-thought)

Vision + Language Models

CLIP training + inference
Results + Robustness
My prior work
Language Models
Language Models

It’s cold today! Don’t forget to wear a ______.
The ____________ is a popular tourist attraction in Seattle.
I missed ___ bus.
I had 3 pencils and lost one so now I have ______ pencils.
Language Models

It’s cold today! Don’t forget to wear a jacket.

The ____________ is a popular tourist attraction in Seattle.

I missed ___ bus.

I had 3 pencils and lost one so now I have ______ pencils.
Language Models

It’s cold today! Don’t forget to wear a jacket.

The **Space Needle** is a popular tourist attraction in Seattle.

I missed ___ bus.

I had 3 pencils and lost one so now I have ______ pencils.
Language Models

It’s cold today! Don’t forget to wear a jacket.

The Space Needle is a popular tourist attraction in Seattle.

I missed the bus.

I had 3 pencils and lost one so now I have ______ pencils.
Language Models

It’s cold today! Don’t forget to wear a jacket.

The Space Needle is a popular tourist attraction in Seattle.

I missed the bus.

I had 3 pencils and lost one so now I have two pencils.
Encoder Only: Capture the meaning of an entire sequence

I love cake
**Encoder Only:** Capture the meaning of an entire sequence

I love cake

**Decoder Only:** Generate text based on previously generated text

I love
**Encoder Only:** Capture the meaning of an entire sequence

I love cake

**Decoder Only:** Generate text based on previously generated text

I love ___

**Encoder-Decoder:** Generate text based on previously generated text and the meaning of a separate sequence

I love cake me gusta
**Encoder Only:** Capture the meaning of an entire sequence

I love cake

**Decoder Only:** Generate text based on previously generated text

I love

**Encoder-Decoder:** Generate text based on previously generated text and the meaning of a separate sequence

I love cake me gusta
Encoder Only: Capture the meaning of an entire sequence
**Encoder Only:** Capture the meaning of an entire sequence

**Example Model:** BERT
**Encoder Only:** Capture the meaning of an entire sequence

**Example Model:** BERT
Encoder Only: Capture the meaning of an entire sequence

Example Model: BERT

Input: Text sequence
Encoder Only: Capture the meaning of an entire sequence

Example Model: BERT

Input: Text sequence

Input = text token embeddings (and positional embedding)
Encoder Only: Capture the meaning of an entire sequence

Example Model: BERT

Input: Text sequence

Output: Feature Vector

Outputs: context vectors: \( y \) (shape: \( D_v \))
**Encoder Only:** Capture the meaning of an entire sequence

**Example Model:** BERT

**Input:** Text sequence

**Output:** Feature Vector

What information do the $y$ vectors contain?

Outputs: context vectors: $y$ (shape: $D_v$)
**Encoder Only:** Capture the meaning of an entire sequence

**Example Model:** BERT

**Input:** Text sequence

**Output:** Feature Vector

What information do the y vectors contain?

Nothing, yet!
**Encoder Only:** Capture the meaning of an entire sequence

**Example Model:** BERT

**Input:** Text sequence

**Output:** Feature Vector
**Encoder Only:** Capture the meaning of an entire sequence

**Example Model:** BERT

**Input:** Text sequence

**Output:** Feature Vector

What information do the y vectors contain?
**Encoder Only:** Capture the meaning of an entire sequence

**Example Model:** BERT

**Input:** Text sequence

**Output:** Feature Vector

What information do the y vectors contain?

Just copying input
**Encoder Only:** Capture the meaning of an entire sequence

**Example Model:** BERT

**Input:** Text sequence

**Output:** Feature Vector

How to we force this model to learn semantic/factual/grammatical/logical information?
Language Models

It’s cold today! Don’t forget to wear a jacket.

The Space Needle is a popular tourist attraction in Seattle.

I missed the bus.

I had 3 pencils and lost one so now I have two pencils.
Language Models

It’s cold today! Don’t forget to wear a jacket.   **Semantic**
The **Space Needle** is a popular tourist attraction in Seattle.   **Factual**
I missed **the** bus.   **Grammatical**
I had 3 pencils and lost one so now I have **two** pencils.   **Logical**
**Encoder Only:** Capture the meaning of an entire sequence

**Example Model:** BERT

**Input:** Text sequence

**Output:** Feature Vector
**Encoder Only:** Capture the meaning of an entire sequence

**Example Model:** BERT

**Input:** Text sequence

**Output:** Feature Vector

Randomly select 15% of tokens.

- 80% - [MASK]
- 10% - random token
- 10% - keep same
**Encoder Only:** Capture the meaning of an entire sequence

I love cake

**Decoder Only:** Generate text based on previously generated text

I love

**Encoder-Decoder:** Generate text based on previously generated text and the meaning of a separate sequence

I love cake me gusta
Decoder Only: Generate text based on previously generated text
**Decoder Only:** Generate text based on previously generated text
**Decoder Only:** Generate text based on previously generated text

**Input:** Text sequence
Decoder Only: Generate text based on previously generated text

**Input:** Text sequence

**Output:** Completed text sequence
**Decoder Only:** Generate text based on previously generated text

**Input:** Text sequence

**Output:** Completed text sequence

Cons: Need to process entire sentence in order to get loss from one word - not very much signal for the amount of processing
Decoder Only: Generate text based on previously generated text

Input: Text sequence
Output: Completed text sequence

Cons: Need to process entire sentence in order to get loss from one word - not very much signal for the amount of processing

Solution: predict each word given previous words so far
**Decoder Only:** Generate text based on previously generated text

**Input:** Text sequence

**Output:** Completed text sequence

Cons: Need to process entire sentence in order to get loss from one word - not very much signal for the amount of processing

Solution: predict each word given previous words so far
Decoder Only: Generate text based on previously generated text

**Input:** Text sequence

**Output:** Completed text sequence
**Decoder Only:** Generate text based on previously generated text

**Input:** Text sequence  
**Output:** Completed text sequence

What’s wrong with this?
**Decoder Only:** Generate text based on previously generated text

**Input:** Text sequence

**Output:** Completed text sequence

What’s wrong with this?

It can see the answer!
**Decoder Only:** Generate text based on previously generated text

**Input:** Text sequence

**Output:** Completed text sequence

What’s wrong with this?

It can see the answer!

**Solution:** zero out values from future words
Decoder Only: Generate text based on previously generated text

**Input:** Text sequence

**Output:** Completed text sequence

What’s wrong with this?

It can see the answer!

Solution: zero out values from future words
Decoder Only: Inference
Decoder Only: Inference

I Love

I

L

y_0 y_1

mul(→) + add (↑)

softmax (↑)

I

I

V_0 V_1

V_0

V_1

I

I

k_0 k_1

k_0

k_1

I

I

q_0 q_1

q_0

q_1

[Start]
Decoder Only: Inference

\[
\text{mul(→) + add (↑)}
\]

\[
y_0, y_1
\]

\[
\text{softmax (↑)}
\]

\[
y_0, y_1
\]

\[
k_0, k_1, k_2
\]

\[
q_0, q_1, q_2
\]

\[
\text{I Love Cake}
\]

\[
\text{Attention}
\]

\[
\text{Alignment}
\]
**Encoder Only:** Capture the meaning of an entire sequence

```
I  love  cake
```

**Decoder Only:** Generate text based on previously generated text

```
I  love  
```

**Encoder-Decoder:** Generate text based on previously generated text and the meaning of a separate sequence

```
I  love  cake  me  gusta
```
**Encoder-Decoder**: Generate text based on previously generated text and the meaning of a separate sequence.
**Encoder-Decoder**: Generate text based on previously generated text and the meaning of a separate sequence.
**Encoder-Decoder**: Generate text based on previously generated text and the meaning of a separate sequence
**Encoder-Decoder:** Generate text based on previously generated text and the meaning of a separate sequence.
Which of the three options is GPT?
Which of the three options is GPT?  

Decoder Only!
Which of the three options is GPT?  Decoder Only!

**Encoder-Decoder:** Generate text based on previously generated text and the meaning of a separate sequence sequence

**Decoder Only:** Generate text based on previously generated text
GPT-3

1. Translate English to French: task description
2. cheese => prompt

Brown et al “Language Models are Few-Shot Learners”
GPT-3

1. Translate English to French: cheese =>

| Please unscramble the letters into a word, and write that word: |
| taefed = |
| defeat |

Q: What is \((2 \times 4) \times 6\)?
A: 48

Q: ‘Nude Descending A Staircase’ is perhaps the most famous painting by which 20th century artist?
A: MARCEL DUCHAMP

Brown et al “Language Models are Few-Shot Learners”
GPT-3

1. Translate English to French: cheese

2. Please unscramble the letters into a word, and write that word:

   taefed =

   defeat

Q: What is \(2 \times 4 \times 6\)?

A:

48

Q: ‘Nude Descending A Staircase’ is perhaps the most famous painting by which 20th century artist?

A:

MARCEL DUCHAMP

Dataset

- Common Crawl (filtered)
- WebText2
- Books1
- Books2
- Wikipedia

Brown et al “Language Models are Few-Shot Learners”
Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

1. Translate English to French:  \( \rightarrow \) task description
2. cheese =>  \( \rightarrow \) prompt
GPT-3

Zero-shot
The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

1. **Translate English to French:**
2. cheese => ........................................

One-shot
In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

1. **Translate English to French:**
2. sea otter => loutre de mer
3. cheese => ........................................

Few-shot
In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

1. **Translate English to French:**
2. sea otter => loutre de mer
3. peppermint => menthe poivrée
4. plush giraffe => girafe peluche
5. cheese => ........................................
GPT-3

**Zero-shot**

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1. Translate English to French:  
   cheese => ...........................................
```

**One-shot**

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1. Translate English to French:  
2.  sea otter => loutre de mer 
3.  cheese => ...........................................
```

**Few-shot**

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1. Translate English to French:  
2.  sea otter => loutre de mer 
3.  peppermint => menthe poivrée 
4.  plush giraffe => girafe peluche 
5.  cheese => ...........................................
```

“Context”

Brown et al “Language Models are Few-Shot Learners”
In-Context Learning

Zero-shot
The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

One-shot
In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1 Translate English to French:
2 sea otter => loutre de mer
3 cheese =>
```

Few-shot
In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1 Translate English to French:
2 sea otter => loutre de mer
3 peppermint => menthe poivrée
4 plush giraffe => girafe peluche
5 cheese =>
```

Brown et al. "Language Models are Few-Shot Learners"
Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?
Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ✗
Standard Prompting

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

A: The answer is 27.  

Chain-of-Thought Prompting

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.
**Standard Prompting**

**Model Input**

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

**Model Output**

A: The answer is 27. ✗

---

**Chain-of-Thought Prompting**

**Model Input**

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

**Model Output**

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✓
Language Models

Encoders vs Decoders vs Encoder-Decoder Models
Prompting (zero-shot, in-context, chain-of-thought)

Vision + Language Models

CLIP training + inference
Results + Robustness
My prior work
Vision + Language
Vision + Language

**Text pretraining task:** Given previous words, pick the next word
Text pretraining task: Given previous words, pick the next word

Text + Vision: Where can we get large amounts of image and text data?
Vision + Language

**Text pretraining task:** Given previous words, pick the next word

**Text + Vision:** Where can we get large amounts of image and text data?

The western slope of Mount Rainier in 2005

https://en.wikipedia.org/wiki/Mount_Rainier
SimCLR
SimCLR

list of positive pairs
SimCLR

Each 2k and 2k + 1 element is a positive pair
CLIP Training
The western slope of Mount Rainier in 2005

A Pallas's cat at Rotterdam Zoo

A very unusual example of a diagonal window set in a brick wall
The western slope of Mount Rainier in 2005

A Pallas's cat at Rotterdam Zoo

A very unusual example of a diagonal window set in a brick wall

CLIP Training
The western slope of Mount Rainier in 2005

A Pallas's cat at Rotterdam Zoo

A very unusual example of a diagonal window set in a brick wall
The western slope of Mount Rainier in 2005

A Pallas's cat at Rotterdam Zoo

A very unusual example of a diagonal window set in a brick wall

A Pallas's cat at Rotterdam Zoo

The western slope of Mount Rainier in 2005

A very unusual example of a diagonal window set in a brick wall

CLIP Training
The western slope of Mount Rainier in 2005

A Pallas's cat at Rotterdam Zoo

A very unusual example of a diagonal window set in a brick wall
CLIP Training

- The western slope of Mount Rainier in 2005
- A Pallas’s cat at Rotterdam Zoo
- A very unusual example of a diagonal window set in a brick wall

Mathematical equation:
\[
\sum_{i=1}^{n} - \log \left( \frac{e^{u_i v_j}}{\sum_{j=1}^{n} e^{u_i v_j}} \right)
\]
The western slope of Mount Rainier in 2005

A Pallas’s cat at Rotterdam Zoo

A very unusual example of a diagonal window set in a brick wall

CLIP Training

\[ \sum_{i=1}^{n} - \log \left( \frac{e^{\langle u_i, v_i \rangle}}{\sum_{j=1}^{n} e^{\langle u_i, v_j \rangle}} \right) \]
The western slope of Mount Rainier in 2005

A very unusual example of a diagonal window set in a brick wall

A Pallas's cat at Rotterdam Zoo

CLIP Training
CLIP Training (from the CLIP paper)

1. Contrastive pre-training

Radford et al “Learning Transferable Visual Models From Natural Language Supervision”
CLIP Inference (from the CLIP paper)
**CLIP Inference (from the CLIP paper)**

2. Create dataset classifier from label text

```
plane
car
dog

...,

bird

---

a photo of a (object).

---

Text Encoder

---

...,

T_1
T_2
T_3

...,

T_N
```

3. Use for zero-shot prediction

```
---

Image Encoder

---

I_1
I_1 \cdot T_1
I_1 \cdot T_2
I_1 \cdot T_3

...,

I_1 \cdot T_N

---

a photo of a dog.
```

Radford et al “Learning Transferable Visual Models From Natural Language Supervision”
Image Classification on ImageNet
After training on ~1,000,000 labeled ImageNet train images
After training on ~1,000,000 labeled ImageNet train images

After training on 0 labeled ImageNet train images (but many other “unlabeled” images)
<table>
<thead>
<tr>
<th>DATASET</th>
<th>ImageNet</th>
<th>IMAGENET RESNET101</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="ImageNet" /></td>
<td><img src="image2.png" alt="ImageNet" /></td>
<td><img src="image3.png" alt="ImageNet" /></td>
</tr>
<tr>
<td>DATASET</td>
<td>IMAGENET RESNET101</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------</td>
<td></td>
</tr>
<tr>
<td>ImageNet</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ObjectNet</td>
<td>76.2%</td>
<td></td>
</tr>
<tr>
<td>DATASET</td>
<td>RESNET101</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td>ImageNet</td>
<td>76.2%</td>
<td></td>
</tr>
<tr>
<td>ObjectNet</td>
<td>32.6%</td>
<td></td>
</tr>
<tr>
<td>DATASET</td>
<td>IMAGE NET RESNET101</td>
<td>CLIP VIT-L</td>
</tr>
<tr>
<td>----------</td>
<td>---------------------</td>
<td>------------</td>
</tr>
<tr>
<td>ImageNet</td>
<td>76.2%</td>
<td>76.2%</td>
</tr>
<tr>
<td>ObjectNet</td>
<td>32.6%</td>
<td>72.3%</td>
</tr>
<tr>
<td>DATASET</td>
<td>IMAGENET</td>
<td>RESNET101</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------</td>
<td>-----------</td>
</tr>
<tr>
<td>ImageNet</td>
<td>76.2%</td>
<td></td>
</tr>
<tr>
<td>ImageNet V2</td>
<td>64.3%</td>
<td></td>
</tr>
<tr>
<td>ImageNet Rendition</td>
<td>37.7%</td>
<td></td>
</tr>
<tr>
<td>ObjectNet</td>
<td>32.6%</td>
<td></td>
</tr>
<tr>
<td>ImageNet Sketch</td>
<td>25.2%</td>
<td></td>
</tr>
<tr>
<td>ImageNet Adversarial</td>
<td>2.7%</td>
<td></td>
</tr>
</tbody>
</table>
AI-Chaining
(aka a plug for my own past work)
AI-Chaining
(aka a plug for my own past work)
LLM-prompts:

“What does a {lorikeet, marimba, viaduct, papillon} look like?”

GPT-3

Image-prompts:

“A lorikeet is a small to medium-sized parrot with a brightly colored plumage.”

“A marimba is a large wooden percussion instrument that looks like a xylophone.”

“A viaduct is a bridge composed of several spans supported by piers or pillars.”

“A papillon is a small, spaniel-type dog with a long, silky coat and fringed ears.”

Lorikeet  Marimba  Viaduct  Papillon
<table>
<thead>
<tr>
<th></th>
<th>ImageNet</th>
<th>DTD</th>
<th>Stanford Cars</th>
<th>SUN397</th>
<th>Food101</th>
<th>FGVC Aircraft</th>
<th>Oxford Pets</th>
<th>Caltech101</th>
<th>Flowers102</th>
<th>UCF101</th>
<th>Kinetics-700</th>
<th>RESISC45</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
<th>Birdsnap</th>
</tr>
</thead>
<tbody>
<tr>
<td>std</td>
<td>75.54</td>
<td>55.20</td>
<td>77.53</td>
<td>69.31</td>
<td>93.08</td>
<td>32.88</td>
<td>93.33</td>
<td>93.24</td>
<td>78.53</td>
<td>77.45</td>
<td>60.07</td>
<td>71.10</td>
<td>95.59</td>
<td>78.26</td>
<td>50.43</td>
</tr>
<tr>
<td># hw</td>
<td>80</td>
<td>8</td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>34</td>
<td>1</td>
<td>48</td>
<td>28</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>1</td>
</tr>
<tr>
<td>CuPL (base)</td>
<td>76.19</td>
<td>58.90</td>
<td>76.49</td>
<td>72.74</td>
<td>93.33</td>
<td>36.69</td>
<td>93.37</td>
<td>93.45</td>
<td>78.83</td>
<td>77.74</td>
<td>60.24</td>
<td>68.96</td>
<td>95.81</td>
<td>78.47</td>
<td>51.11</td>
</tr>
<tr>
<td>Δ std</td>
<td>+0.65</td>
<td>+3.70</td>
<td>-1.04</td>
<td>+3.43</td>
<td>+0.25</td>
<td>+3.81</td>
<td>+0.04</td>
<td>+0.21</td>
<td>+0.30</td>
<td>+0.29</td>
<td>+0.17</td>
<td>-2.14</td>
<td>+0.22</td>
<td>+0.21</td>
<td>+0.63</td>
</tr>
<tr>
<td># hw</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
Thank you!