Deep Learning

Week [03]: [LLMs]
Logistics

- HW2 is out
  - Due next Thursday

- We released a feedback form on Canvas
  - Due Friday February 23
  - Part of your participation grade!

- We are talking about projects today!
  - HW3 will be a shorter
Project

- Aim: to get hands on experience with implementing modern deep learning methods
- Find a recent deep learning research paper that was published in the last 1-5 years
- Reproduce a specific result from the paper
- To be completed in groups of 2-3
Project Grading

- **Project proposal**
  - We want to make sure the papers you’ve chosen are reasonable
  - Sign up on spreadsheet, we don’t want duplicates!

- **Final presentation**
  - The last 2-3 classes will be devoted to presentations
  - Present the paper you’ve chosen to the class
  - Discuss whether you were able to replicate results from the paper and describe any obstacles

- **Final deliverable**
  - Github repo with re-implementation
  - Readme with method description, instructions to run the code and results
  - More detail will be provided later
A page long project proposal due **March 7**. It should contain the following:

- **Paper selection:**
  - Title, authors, and publication venue of the chosen paper
  - Brief summary of the chosen paper
  - Brief justification of why you choose this paper

- **Result Selection**
  - Tell us which result you want to replicate

- **Re-implementation Plan**
  - Describe architecture, method, and metrics
  - Details about how much compute and time is required to replicate results

**Detailed instruction will be provided on canvas**
Transformer Architecture

Introduced for seq2seq tasks like Machine translation, summarization, question answering, etc.

I like black coffee

Ich mag schwarzen Kaffee

<START> Ich mag schwarzen
Transformer Architecture

Introduced for seq2seq tasks like Machine translation, summarization, question answering, etc.

- Parallel computation!
- Attention within the encoder and decoder

I like black coffee

Ich mag schwarzen Kaffee

<START> Ich mag schwarzen
Transformer Architecture

I like black coffee

Ich mag schwarzen Kaffee
Self-Attention

Queries (Q)  Keys (K)  Values (V)

$W_q$, $W_k$, $W_v$
Self-Attention

$s_1 = \frac{q_1 K^T}{\sqrt{d}}$

$\alpha_1 = \text{softmax}(s_1)$

$x_1 = \sum_{i=1}^{4} \alpha_{1,i} v_i$

Queries (Q) → Keys (K) → Values (V) → Output Sequence

$W_q$ $W_k$ $W_v$
Self-Attention

\[ s_2 = \frac{q_2 K^T}{\sqrt{d}} \]

\[ \alpha_2 = \text{softmax}(s_2) \]

\[ x_2 = \sum_{i=1}^{4} \alpha_{2,i} v_i \]
Self-Attention: General Formula

\[ \text{Attention}(Q, K, V) = \text{Softmax}\left( \frac{QK^T}{\sqrt{d_k}} \right)V \]
Discuss:

Assume input sequence is \((n \times d)\),

Queries, keys, and values are \(d\)-dimensional

What is the size of \(W_q\), \(W_k\), \(W_v\)?

What is the size of \(Q\), \(K\), \(V\)?
Self-Attention
Position-wise Feed-forward Networks

- **Purpose**
  - Applies non-linear transformations to the output of the attention layer

- **Equation**
  - $\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$
  - where $W$ and $b$ are learned weights and biases

- These FFN is applied separately to each position
Transformer Encoder

I like black coffee
BERT (Bidirectional Encoder Representations from Transformers)

- Bidirectional Context
- Pre-trained on the language, and then fine-tuned
Self-Supervised Learning

● Masked Language Modelling
  ○ Mask out k% of the input words, and then predict the masked words
  ○ The man went to the [MASK] to [MASK] a gallon of milk
  ○ Multi-class classification problem

● Next sentence prediction
  ○ To learn relationships between sentences, predict whether Sentence B is actual sentence that proceeds Sentence A, or a random sentence

Sentence A = The man went to the store.  Sentence A = The man went to the store.
Sentence B = He bought a gallon of milk.  Sentence B = Penguins are flightless.
Label = IsNextSentence  Label = NotNextSentence
Masked Language Modeling

- Masked language modeling is really hard!
  - The man went to the [MASK] to [MASK] a gallon of milk
- World knowledge:
  - The capital of Paris is [MASK]
- Sentiment classification:
  - A sentimental mess that never rings true. All in all, the movie was [MASK].
Pre-training to Fine-tuning Pipeline

Pre-training

Mask LM

Class Label

Single Sentence

Unlabeled Sentence A and B Pair

Masked Sentence A

Masked Sentence B

BERT

C T₁ ... Tₐ T_SEP Tᵢ ... Tₘ'

E_[CLS] E₁ ... Eₘ E_SEP Eᵢ' ... Eₘ'

[CLS] Tok 1 ... Tokₐ [SEP] Tok 1 ... Tokₘ

C T₁ T₂ ... Tₙ

E_[CLS] E₁ E₂ ... Eₙ

[CLS] Tok 1 Tok 2 ... Tokₙ
Works Really Well!

<table>
<thead>
<tr>
<th>System</th>
<th>MNLI-(m/mm)</th>
<th>QQP</th>
<th>QNLI</th>
<th>SST-2</th>
<th>CoLA</th>
<th>STS-B</th>
<th>MRPC</th>
<th>RTE</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>392k</td>
<td>363k</td>
<td>108k</td>
<td>67k</td>
<td>8.5k</td>
<td>5.7k</td>
<td>3.5k</td>
<td>2.5k</td>
<td></td>
</tr>
<tr>
<td>Pre-OpenAI SOTA</td>
<td>80.6/80.1</td>
<td>66.1</td>
<td>82.3</td>
<td>93.2</td>
<td>35.0</td>
<td>81.0</td>
<td>86.0</td>
<td>61.7</td>
<td>74.0</td>
</tr>
<tr>
<td>BiLSTM+ELMo+Attn</td>
<td>76.4/76.1</td>
<td>64.8</td>
<td>79.8</td>
<td>90.4</td>
<td>36.0</td>
<td>73.3</td>
<td>84.9</td>
<td>56.8</td>
<td>71.0</td>
</tr>
<tr>
<td>OpenAI GPT</td>
<td>82.1/81.4</td>
<td>70.3</td>
<td>87.4</td>
<td>91.3</td>
<td>45.4</td>
<td>80.0</td>
<td>82.3</td>
<td>56.0</td>
<td>75.1</td>
</tr>
<tr>
<td>BERT&lt;sub&gt;BASE&lt;/sub&gt;</td>
<td>84.6/83.4</td>
<td>71.2</td>
<td>90.5</td>
<td>93.5</td>
<td>52.1</td>
<td>85.8</td>
<td>88.9</td>
<td>66.4</td>
<td>79.6</td>
</tr>
<tr>
<td>BERT&lt;sub&gt;LARGE&lt;/sub&gt;</td>
<td>86.7/85.9</td>
<td>72.1</td>
<td>92.7</td>
<td>94.9</td>
<td>60.5</td>
<td>86.5</td>
<td>89.3</td>
<td>70.1</td>
<td>82.1</td>
</tr>
</tbody>
</table>

Pre-Train then Fine-tune Paradigm

- Pre-train a large model on a lot of data with some self-supervised objective.
  - E.g. Masked language modeling
- Fine-tune on (smaller) downstream datasets
- Benefits:
  - Transfer learning: Leveraging knowledge from self-supervised objective to improve performance
  - Avoids overfitting: Large models would overfit if trained from scratch
Fine-tuning Limitations

- **Memory overhead**
  - Training takes more memory than inference
  - Fine-tuning large models requires expensive GPUs

- **Optimization**
  - Might need to adjust optimization hyperparameters for each downstream task
  - Learning rates, batch sizes, etc.
Question:

- What are some limitations of BERT?
  - Think about the different kinds of language tasks you might want to perform
T5: Text-to-Text Transfer Transformer

- Pre-trained Encoder Decoder Language Model
  - Can generate text!

```
I like black coffee

<Z> like <X>

<Z> I <X> black coffee

[START] <X> I <Z> black coffee
```

Diagram:
- Transformer Encoder
- Transformer Decoder
T5 Span Corruption Objective

- Very similar to masked language modeling
- Key differences:
  - Mask spans with single “sentinel” tokens
  - Generate the masked text with a decoder

Original text:

Thank you for inviting me to your party last week.

Inputs:

Thank you \(<X>\) me to your party \(<Y>\) week.

Targets:

\(<X>\) for inviting \(<Y>\) last \(<Z>\)
T5 Fine-Tuning

- Cast every task as a language generation problem
  - Even classification!

- “translate English to German: That is good.”
- “cola sentence: The course is jumping well.”
- “sts b sentence: The rhino grazed on the grass. sentence2: A rhino is grazing in a field.”
- “summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi.”
- “Das ist gut.”
- “not acceptable”
- “3.8”
- “six people hospitalized after a storm in attala county.”
### Benchmark Results

- Outperformed best encoder-only models

```
<table>
<thead>
<tr>
<th>Model</th>
<th>GLUE Average</th>
<th>CoLA Matthew’s Accuracy</th>
<th>SST-2</th>
<th>MRPC F1</th>
<th>MRPC Accuracy</th>
<th>STS-B Pearson</th>
<th>STS-B Spearman</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous best</td>
<td>89.4&lt;sup&gt;a&lt;/sup&gt;</td>
<td>69.2&lt;sup&gt;b&lt;/sup&gt;</td>
<td>97.1&lt;sup&gt;a&lt;/sup&gt;</td>
<td>93.6&lt;sup&gt;b&lt;/sup&gt;</td>
<td>91.5&lt;sup&gt;b&lt;/sup&gt;</td>
<td>92.7&lt;sup&gt;b&lt;/sup&gt;</td>
<td>92.3&lt;sup&gt;b&lt;/sup&gt;</td>
</tr>
<tr>
<td>T5-Small</td>
<td>77.4</td>
<td>41.0</td>
<td>91.8</td>
<td>89.7</td>
<td>86.6</td>
<td>85.6</td>
<td>85.0</td>
</tr>
<tr>
<td>T5-Base</td>
<td>82.7</td>
<td>51.1</td>
<td>95.2</td>
<td>90.7</td>
<td>87.5</td>
<td>89.4</td>
<td>88.6</td>
</tr>
<tr>
<td>T5-Large</td>
<td>86.4</td>
<td>61.2</td>
<td>96.3</td>
<td>92.4</td>
<td>89.9</td>
<td>89.9</td>
<td>89.2</td>
</tr>
<tr>
<td>T5-3B</td>
<td>88.5</td>
<td>67.1</td>
<td>97.4</td>
<td>92.5</td>
<td>90.0</td>
<td>90.6</td>
<td>89.8</td>
</tr>
<tr>
<td>T5-11B</td>
<td><strong>90.3</strong></td>
<td><strong>71.6</strong></td>
<td><strong>97.5</strong></td>
<td>92.8</td>
<td>90.4</td>
<td><strong>93.1</strong></td>
<td><strong>92.8</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>QQP F1</th>
<th>QQP Accuracy</th>
<th>MNLI-m Accuracy</th>
<th>MNLI-mm Accuracy</th>
<th>QNLI F1</th>
<th>RTE F1</th>
<th>WNLI Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous best</td>
<td>74.8&lt;sup&gt;c&lt;/sup&gt;</td>
<td><strong>90.7</strong>&lt;sup&gt;b&lt;/sup&gt;</td>
<td>91.3&lt;sup&gt;a&lt;/sup&gt;</td>
<td>91.0&lt;sup&gt;a&lt;/sup&gt;</td>
<td>99.2&lt;sup&gt;a&lt;/sup&gt;</td>
<td>89.2&lt;sup&gt;a&lt;/sup&gt;</td>
<td>91.8&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>T5-Small</td>
<td>70.0</td>
<td>88.0</td>
<td>82.4</td>
<td>82.3</td>
<td>90.3</td>
<td>69.9</td>
<td>69.2</td>
</tr>
<tr>
<td>T5-Base</td>
<td>72.6</td>
<td>89.4</td>
<td>87.1</td>
<td>86.2</td>
<td>93.7</td>
<td>80.1</td>
<td>78.8</td>
</tr>
<tr>
<td>T5-Large</td>
<td>73.9</td>
<td>89.9</td>
<td>89.9</td>
<td>89.6</td>
<td>94.8</td>
<td>87.2</td>
<td>85.6</td>
</tr>
<tr>
<td>T5-3B</td>
<td>74.4</td>
<td>89.7</td>
<td>91.4</td>
<td>91.2</td>
<td>96.3</td>
<td>91.1</td>
<td>89.7</td>
</tr>
<tr>
<td>T5-11B</td>
<td><strong>75.1</strong></td>
<td>90.6</td>
<td><strong>92.2</strong></td>
<td><strong>91.9</strong></td>
<td>96.9</td>
<td><strong>92.8</strong></td>
<td><strong>94.5</strong></td>
</tr>
</tbody>
</table>
```
## T5 Ablations - Dataset Size

- More data helps!

<table>
<thead>
<tr>
<th>Number of tokens</th>
<th>Repeats</th>
<th>GLUE</th>
<th>CNNDM</th>
<th>SQuAD</th>
<th>SGLUE</th>
<th>EnDe</th>
<th>EnFr</th>
<th>EnRo</th>
</tr>
</thead>
<tbody>
<tr>
<td>★ Full data set</td>
<td>0</td>
<td>83.28</td>
<td>19.24</td>
<td>80.88</td>
<td>71.36</td>
<td>26.98</td>
<td>39.82</td>
<td>27.65</td>
</tr>
<tr>
<td>$2^{29}$</td>
<td>64</td>
<td>82.87</td>
<td>19.19</td>
<td>80.97</td>
<td>72.03</td>
<td>26.83</td>
<td>39.74</td>
<td>27.63</td>
</tr>
<tr>
<td>$2^{27}$</td>
<td>256</td>
<td>82.62</td>
<td>19.20</td>
<td>79.78</td>
<td>69.97</td>
<td>27.02</td>
<td>39.71</td>
<td>27.33</td>
</tr>
<tr>
<td>$2^{25}$</td>
<td>1,024</td>
<td>79.55</td>
<td>18.57</td>
<td>76.27</td>
<td>64.76</td>
<td>26.38</td>
<td>39.56</td>
<td>26.80</td>
</tr>
<tr>
<td>$2^{23}$</td>
<td>4,096</td>
<td>76.34</td>
<td>18.33</td>
<td>70.92</td>
<td>59.29</td>
<td>26.37</td>
<td>38.84</td>
<td>25.81</td>
</tr>
</tbody>
</table>
GPT

- Only use the transformer decoder
  - Train on next word prediction
  - “The capital of France is _”
- Easy to apply in streaming settings
  - E.g. chatbots

![Diagram showing Transformer Decoder](image)
Decoder-Only Architecture

Ich mag schwarzen Kaffee

Linear Classifier

Feed Forward

Masked Self-Attention

I like black coffee

<START> Ich mag schwarzen
Review: Masked Self-Attention
GPT-2

- Train self-supervised auto-regressive LMs on web text
  - Next-word prediction
  - Up to 1.5 billion parameters
- Observed non-trivial zero-shot performance

“Language Models are Unsupervised Multitask Learners”
GPT-2 Zero-Shot Capabilities

- Question-answering without any fine-tuning
  - Formatting is important!

**Context (passage and previous question/answer pairs)**

Tom goes everywhere with Catherine Green, a 54-year-old secretary. He moves around her office at work and goes shopping with her. "Most people don’t seem to mind Tom," says Catherine, who thinks he is wonderful. "He’s my fourth child," she says. She may think of him and treat him that way as her son. He moves around buying his food, paying his health bills and his taxes, but in fact Tom is a dog.

Catherine and Tom live in Sweden, a country where everyone is expected to lead an orderly life according to rules laid down by the government, which also provides a high level of care for its people. This level of care costs money.

People in Sweden pay taxes on everything, so aren’t surprised to find that owning a dog means more taxes. Some people are paying as much as 500 Swedish kronor in taxes a year for the right to keep their dog, which is spent by the government on dog hospitals and sometimes medical treatment for a dog that falls ill. However, most such treatment is expensive, so owners often decide to offer health and even life for their dog.

In Sweden dog owners must pay for any damage their dog does. A Swedish Kennel Club official explains what this means: if your dog runs out on the road and gets hit by a passing car, you, as the owner, have to pay for any damage done to the car, even if your dog has been killed in the accident.

Q: How old is Catherine?
A: 54

Q: where does she live?
A: 

**Model answer:** Stockholm
**Turker answers:** Sweden, Sweden, in Sweden, Sweden
"I’m not the cleverest man in the world, but like they say in French: *Je ne suis pas un imbecile* [I’m not a fool].

In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: "*Mentez mentez, il en restera toujours quelque chose*,” which translates as, "*Lie lie and something will always remain.*"

"I hate the word ‘perfume,’” Burr says. ‘It’s somewhat better in French: ‘parfum.’"

If listened carefully at 29:55, a conversation can be heard between two guys in French: "*Comment on fait pour aller de l’autre côté? - Quel autre côté?*”, which means "*- How do you get to the other side? - What side?"."

If this sounds like a bit of a stretch, consider this question in French: *As-tu aller au cinéma?, or Did you go to the movies?,* which literally translates as Have-you to go to movies/theater?

"*Brevet Sans Garantie Du Gouvernement*, translated to English: "*Patented without government warranty*."

---

Table 1. Examples of naturally occurring demonstrations of English to French and French to English translation found throughout the WebText training set.
BERT

I like black coffee

T5/BART

Ich mag schwarzen Kaffee

GPT

Ich mag schwarzen Kaffee
GPT-3

- Train a bigger version of GPT-2 on more data

Transformer Decoder

I like black coffee

[START] I like black coffee
### GPT-3

<table>
<thead>
<tr>
<th>Model Name</th>
<th>$n_{\text{params}}$</th>
<th>$n_{\text{layers}}$</th>
<th>$d_{\text{model}}$</th>
<th>$n_{\text{heads}}$</th>
<th>$d_{\text{head}}$</th>
<th>Batch Size</th>
<th>Learning Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPT-3 Small</td>
<td>125M</td>
<td>12</td>
<td>768</td>
<td>12</td>
<td>64</td>
<td>0.5M</td>
<td>$6.0 \times 10^{-4}$</td>
</tr>
<tr>
<td>GPT-3 Medium</td>
<td>350M</td>
<td>24</td>
<td>1024</td>
<td>16</td>
<td>64</td>
<td>0.5M</td>
<td>$3.0 \times 10^{-4}$</td>
</tr>
<tr>
<td>GPT-3 Large</td>
<td>760M</td>
<td>24</td>
<td>1536</td>
<td>16</td>
<td>96</td>
<td>0.5M</td>
<td>$2.5 \times 10^{-4}$</td>
</tr>
<tr>
<td>GPT-3 XL</td>
<td>1.3B</td>
<td>24</td>
<td>2048</td>
<td>24</td>
<td>128</td>
<td>1M</td>
<td>$2.0 \times 10^{-4}$</td>
</tr>
<tr>
<td>GPT-3 2.7B</td>
<td>2.7B</td>
<td>32</td>
<td>2560</td>
<td>32</td>
<td>80</td>
<td>1M</td>
<td>$1.6 \times 10^{-4}$</td>
</tr>
<tr>
<td>GPT-3 6.7B</td>
<td>6.7B</td>
<td>32</td>
<td>4096</td>
<td>32</td>
<td>128</td>
<td>2M</td>
<td>$1.2 \times 10^{-4}$</td>
</tr>
<tr>
<td>GPT-3 13B</td>
<td>13.0B</td>
<td>40</td>
<td>5140</td>
<td>40</td>
<td>128</td>
<td>2M</td>
<td>$1.0 \times 10^{-4}$</td>
</tr>
<tr>
<td>GPT-3 175B or “GPT-3”</td>
<td>175.0B</td>
<td>96</td>
<td>12288</td>
<td>96</td>
<td>128</td>
<td>3.2M</td>
<td>$0.6 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

**Table 2.1:** Sizes, architectures, and learning hyper-parameters (batch size in tokens and learning rate) of the models which we trained. All models were trained for a total of 300 billion tokens.
Discuss

- How many parameters in a single feedforward layer of GPT-3
  - An MLP with:
    - Input dimension of 12,228
    - One hidden layer with dimension 4*12,228
    - Output dimension of 12,228

<table>
<thead>
<tr>
<th>Model Name</th>
<th>$n_{\text{params}}$</th>
<th>$n_{\text{layers}}$</th>
<th>$d_{\text{model}}$</th>
<th>$n_{\text{heads}}$</th>
<th>$d_{\text{head}}$</th>
<th>Batch Size</th>
<th>Learning Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPT-3 Small</td>
<td>125M</td>
<td>12</td>
<td>768</td>
<td>12</td>
<td>64</td>
<td>0.5M</td>
<td>$6.0 \times 10^{-4}$</td>
</tr>
<tr>
<td>GPT-3 Medium</td>
<td>350M</td>
<td>24</td>
<td>1024</td>
<td>16</td>
<td>64</td>
<td>0.5M</td>
<td>$3.0 \times 10^{-4}$</td>
</tr>
<tr>
<td>GPT-3 Large</td>
<td>760M</td>
<td>24</td>
<td>1536</td>
<td>16</td>
<td>96</td>
<td>0.5M</td>
<td>$2.5 \times 10^{-4}$</td>
</tr>
<tr>
<td>GPT-3 XL</td>
<td>1.3B</td>
<td>24</td>
<td>2048</td>
<td>24</td>
<td>128</td>
<td>1M</td>
<td>$2.0 \times 10^{-4}$</td>
</tr>
<tr>
<td>GPT-3 2.7B</td>
<td>2.7B</td>
<td>32</td>
<td>2560</td>
<td>32</td>
<td>80</td>
<td>1M</td>
<td>$1.6 \times 10^{-4}$</td>
</tr>
<tr>
<td>GPT-3 6.7B</td>
<td>6.7B</td>
<td>32</td>
<td>4096</td>
<td>32</td>
<td>128</td>
<td>2M</td>
<td>$1.2 \times 10^{-4}$</td>
</tr>
<tr>
<td>GPT-3 13B</td>
<td>13.0B</td>
<td>40</td>
<td>5140</td>
<td>40</td>
<td>128</td>
<td>2M</td>
<td>$1.0 \times 10^{-4}$</td>
</tr>
<tr>
<td>GPT-3 175B or “GPT-3”</td>
<td>175.0B</td>
<td>96</td>
<td>12288</td>
<td>96</td>
<td>128</td>
<td>3.2M</td>
<td>$0.6 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

Table 2.1: Sizes, architectures, and learning hyper-parameters (batch size in tokens and learning rate) of the models which we trained. All models were trained for a total of 300 billion tokens.
Scaling Model Size

Source: https://hellobefuture.orange.com/en/the-gpt-3-language-model-revolution-or-evolution/
Scaling Training Data

Source: https://hellofuture.orange.com/en/the-gpt-3-language-model-revolution-or-evolution/
Scaling Training Compute

Total Compute Used During Training

<table>
<thead>
<tr>
<th>Model</th>
<th>Total Compute Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-Base</td>
<td></td>
</tr>
<tr>
<td>BERT-Large</td>
<td></td>
</tr>
<tr>
<td>RoBERTa-Base</td>
<td></td>
</tr>
<tr>
<td>RoBERTa-Large</td>
<td></td>
</tr>
<tr>
<td>T5-Small</td>
<td></td>
</tr>
<tr>
<td>T5-Base</td>
<td></td>
</tr>
<tr>
<td>T5-Large</td>
<td></td>
</tr>
<tr>
<td>T5-3B</td>
<td></td>
</tr>
<tr>
<td>T5-11B</td>
<td></td>
</tr>
<tr>
<td>GPT-3 3 Small</td>
<td></td>
</tr>
<tr>
<td>GPT-3 Medium</td>
<td></td>
</tr>
<tr>
<td>GPT-3 Large</td>
<td></td>
</tr>
<tr>
<td>GPT-3 XL</td>
<td></td>
</tr>
<tr>
<td>GPT-3 2.7B</td>
<td></td>
</tr>
<tr>
<td>GPT-3 6.7B</td>
<td></td>
</tr>
<tr>
<td>GPT-3 13B</td>
<td></td>
</tr>
<tr>
<td>GPT-3 17B</td>
<td></td>
</tr>
</tbody>
</table>
Effect of Scaling on Validation Loss

Figure 3.1: Smooth scaling of performance with compute. Performance (measured in terms of cross-entropy validation loss) follows a power-law trend with the amount of compute used for training. The power-law behavior observed in [KMH+20] continues for an additional two orders of magnitude with only small deviations from the predicted curve. For this figure, we exclude embedding parameters from compute and parameter counts.
Power Laws

- Performance improves predictably with increased compute, data, and parameters
  - Can actually fit power laws!
  - Predict performance before training!

\[
C = C_0 N D
\]
\[
L = \frac{A}{N^\alpha} + \frac{B}{D^\beta} + L_0
\]

**Figure 1** Language modeling performance improves smoothly as we increase the model size, dataset size, and amount of compute\(^2\) used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.
GPT-3 Evaluation

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 => ........................................... task description
3. .................................................. prompt

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              example #1
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 => ........................................... examples

Fine-tuning
The model is trained via repeated gradient updates using a large corpus of example tasks.

1. sea otter => loutre de mer               example #1
2. peppermint => menthe poivrée
3. plush giraffe => girafe peluche
4. cheese => ........................................... prompt
GPT-3 Evaluation

SuperGLUE Performance

- Human
- Fine-tuned SOTA
- Fine-tuned BERT++
- Fine-tuned BERT Large

In-Context Learning on SuperGLUE

- Human
- Fine-tuned SOTA
- Few-shot (K=32)
- Few-shot GPT-3 175B

Billions of Parameters in LM vs. SuperGLUE Score

Number of Examples in Context (K) vs. Score
GPT-3 Evaluation

![Graph showing the performance of GPT-3 in TriviaQA across different parameter settings. The graph compares Zero-Shot, One-Shot, and Few-Shot (K=64) models. The accuracy increases with the number of parameters in the large language model (LM). The 'Fine-tuned SOTA' line indicates the performance of a fine-tuned state-of-the-art model.](image-url)
GPT-3 Evaluation

Translation (Multi-BLEU)

- French -> English
- English -> French
- German -> English
- English -> German
- Romanian -> English
- English -> Romanian

Parameters in LM (Billions):
- 0.1B
- 0.4B
- 0.8B
- 1.3B
- 2.6B
- 6.7B
- 13B
- 175B

BLEU:
- 0
- 5
- 10
- 15
- 20
- 25
- 30
- 35
- 40
### GPT-3 News Articles

- Humans struggle to identify GPT-3 generated news articles

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Mean Accuracy</th>
<th>95% CI (low, hi)</th>
<th>t compared to control (p-value)</th>
<th>“I don’t know” assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control (deliberately bad model)</td>
<td>86%</td>
<td>83%–90%</td>
<td>-</td>
<td>3.6%</td>
</tr>
<tr>
<td>GPT-3 Small</td>
<td>76%</td>
<td>72%–80%</td>
<td>3.9 (2e-4)</td>
<td>4.9%</td>
</tr>
<tr>
<td>GPT-3 Medium</td>
<td>61%</td>
<td>58%–65%</td>
<td>10.3 (7e-21)</td>
<td>6.0%</td>
</tr>
<tr>
<td>GPT-3 Large</td>
<td>68%</td>
<td>64%–72%</td>
<td>7.3 (3e-11)</td>
<td>8.7%</td>
</tr>
<tr>
<td>GPT-3 XL</td>
<td>62%</td>
<td>59%–65%</td>
<td>10.7 (1e-19)</td>
<td>7.5%</td>
</tr>
<tr>
<td>GPT-3 2.7B</td>
<td>62%</td>
<td>58%–65%</td>
<td>10.4 (5e-19)</td>
<td>7.1%</td>
</tr>
<tr>
<td>GPT-3 6.7B</td>
<td>60%</td>
<td>56%–63%</td>
<td>11.2 (3e-21)</td>
<td>6.2%</td>
</tr>
<tr>
<td>GPT-3 13B</td>
<td>55%</td>
<td>52%–58%</td>
<td>15.3 (1e-32)</td>
<td>7.1%</td>
</tr>
<tr>
<td>GPT-3 175B</td>
<td>52%</td>
<td>49%–54%</td>
<td>16.9 (1e-34)</td>
<td>7.8%</td>
</tr>
</tbody>
</table>

Table 3.11: Human accuracy in identifying whether short (~200 word) news articles are model generated.
Emergent Capabilities

Arithmetic (few-shot)

- Two Digit Addition
- Two Digit Subtraction
- Three Digit Addition
- Three Digit Subtraction
- Four Digit Addition
- Four Digit Subtraction
- Five Digit Addition
- Five Digit Subtraction
- Two Digit Multiplication
- Single Digit Three Ops

Accuracy vs. Parameters in LM (Billions)
Is Emergence a Mirage?

- Look at four digit addition under different metrics
  - Emergence can be an artefact of the evaluation metric
- Addition capabilities improve smoothly when using a more granular metric
  - Exact match accuracy -> Token edit distance

I will explain these jokes:

1. The problem with kleptomania is that they always take things literally.
   Explanation: This joke is wordplay. Someone who "takes things literally" is someone who doesn't fully understand social cues and context, which is a negative trait. But the definition of kleptomania is someone who literally takes things.
2. Always borrow money from a pessimist. They'll never expect it back.
   Explanation: Most people expect you to pay them back when you borrow money, however a pessimist is someone who always assumes the worst, so if you borrow money from them, they will expect that you won't pay them back anyways.

**Input:** I was supposed to start writing the paper at 5:00 PM. But then I started playing with this cool new language model for 10 minutes. 10 minutes later, it's suddenly 9:30 PM!

**Model Output:** This joke is about how time flies when you're having fun. The person was supposed to start writing a paper at 5:00 PM, but they started playing with a language model instead. They thought they were only playing with it for 10 minutes, but it was actually 4.5 hours.

**Input:** I think I understand why me and my partner are such a good match. I get excited when I smell food and become super alert when the doorbell rings. My girlfriend Shelley becomes ecstatic when she sees a person who she just saw 5 hours ago.

**Model Output:** This joke is about dogs. Dogs get excited when they smell food and when the doorbell rings. The girlfriend is excited to see someone she just saw 5 hours ago, which is a trait of dogs.
Any problems with auto-regressive LMs?

Prompt:
What is the purpose of the list C in the code below?

def binomial_coefficient(n, r):
    C = [0 for i in range(r + 1)];
    C[0] = 1;
    for i in range(1, n + 1):
        j = min(i, r);
        while j > 0:
            C[j] += C[j - 1];
            j -= 1;
    return C[r]
Any problems with auto-regressive LMs?

Prompt:
What is the purpose of the list C in the code below?

```python
def binomial_coefficient(n, r):
    C = [0 for i in range(r + 1)];
    C[0] = 1;
    for i in range(1, n + 1):
        j = min(i, r);
        while j > 0:
            C[j] += C[j - 1];
            j -= 1;
    return C[r]
```

GPT-3 175B completion:
A. to store the value of C[0]
B. to store the value of C[1]
C. to store the value of C[i]
D. to store the value of C[i - 1]
Any problems with auto-regressive LMs?

- Just trained on next-word prediction
  - Won’t follow instructions!
Instruction Tuning

- Collect a small dataset of instruction following demonstrations
  - Fine-tune the pre-trained LM

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.
Instruction Tuning

- Aligns LM with user intent!

**Prompt:**
What is the purpose of the list C in the code below?

```python
def binomial_coefficient(n, r):
    C = [0 for i in range(r + 1)];
    C[0] = 1;
    for i in range(1, n + 1):
        j = min(i, r);
        while j > 0:
            C[j] += C[j - 1];
            j -= 1;
    return C[r]
```

**GPT-3 175B completion:**
A. to store the value of C[0]
B. to store the value of C[1]
C. to store the value of C[i]
D. to store the value of C[i - 1]

**InstructGPT 175B completion:**
The list C in this code is used to store the values of the binomial coefficient as the function iterates through the values of n and r. It is used to calculate the value of the binomial coefficient for a given value of n and r, and stores the result in the final return value of the function.
Instruction Tuning

- Model learns to generalize to unseen instructions at inference-time

Prompt Engineering

**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?

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?

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. ✓
Chain of Thought Prompting

- Break down problem into subparts
  - Learns to do it with few-shot in-context learning

Math Word Problems (free response)
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.

Math Word Problems (multiple choice)
Q: How many keystrokes are needed to type the numbers from 1 to 500? Answer Choices: (a) 1156 (b) 1392 (c) 1480 (d) 1562 (e) 1788
A: There are 9 one-digit numbers from 1 to 9. There are 90 two-digit numbers from 10 to 99. There are 401 three-digit numbers from 100 to 500. 9 + 90(2) + 401(3) = 1392. The answer is (b).

CSQA (commonsense)
Q: Sammy wanted to go to the place where the people were. Where might he go? Options: (a) race track (b) populated areas (c) desert (d) apartment (e) roadblock
A: The answer must be a place with a lot of people. Race tracks, desert, apartments, and roadblocks don't have a lot of people, but populated areas do. So the answer is (b).

StrategyQA
Q: Yes or no: Would a pear sink in water?
A: The density of a pear is about 0.6 g/cm³, which is less than water. Thus, a pear would float. So the answer is no.

Date Understanding
Q: The concert was scheduled to be on 06/01/1943, but was delayed by one day to today. What is the date 10 days ago in MM/DD/YYYY?
A: One day after 06/01/1943 is 06/02/1943, so today is 06/02/1943. 10 days before today is 05/23/1943. So the answer is 05/23/1943.

Sports Understanding
Q: Is the following sentence plausible? "Joao Moutinho caught the screen pass in the NFC championship."
A: Joao Moutinho is a soccer player. The NFC championship is part of American football, not soccer. So the answer is no.

SayCan (Instructing a robot)
Human: How would you bring me something that isn't a fruit?
Explanation: the user wants something to eat that isn't a fruit. An energy bar is not a fruit, so I will bring the user an energy bar.
Plan: 1. find(energy bar) 2. pick(energy bar) 3. find(user) 4. put(energy bar) 5. done().

Last Letter Concatenation
Q: Take the last letters of the words in "Lady Gaga" and concatenate them.
A: The last letter of "Lady" is "y". The last letter of "Gaga" is "a". Concatenating them is "ya". So the answer is ya.

Coin Flip (state tracking)
Q: A coin is heads up. Maybebelle flips the coin. Shalonda does not flip the coin. Is the coin still heads up?
A: The coin was flipped by Maybebelle. So the coin was flipped 1 time, which is an odd number. The coin started heads up, so after an odd number of flips, it will be tails up. So the answer is no.
Chain of Thought Prompting

- Can significantly improve performance
Chain of Thought Prompting

- No benefit at small scales
Sensitivity to the Prompt

- Performance depends a lot on the prompt!

Table 4: Robustness study against template measured on the MultiArith dataset with text-davinci-002. (*1) This template is used in Ahn et al. [2022] where a language model is prompted to generate step-by-step actions given a high-level instruction for controlling robotic actions. (*2) This template is used in Reynolds and McDonell [2021] but is not quantitatively evaluated.

<table>
<thead>
<tr>
<th>No.</th>
<th>Category</th>
<th>Template</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>instructive</td>
<td>Let’s think step by step.</td>
<td>78.7</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>First, (*1)</td>
<td>77.3</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>Let’s think about this logically.</td>
<td>74.5</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Let’s solve this problem by splitting it into steps.</td>
<td>72.2</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Let’s be realistic and think step by step.</td>
<td>70.8</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Let’s think like a detective step by step.</td>
<td>70.3</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>Let’s think</td>
<td>57.5</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>Before we dive into the answer,</td>
<td>55.7</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>The answer is after the proof.</td>
<td>45.7</td>
</tr>
<tr>
<td>10</td>
<td>misleading</td>
<td>Don’t think. Just feel.</td>
<td>18.8</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>Let’s think step by step but reach an incorrect answer.</td>
<td>18.7</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>Let’s count the number of “a” in the question.</td>
<td>16.7</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>By using the fact that the earth is round,</td>
<td>9.3</td>
</tr>
<tr>
<td>14</td>
<td>irrelevant</td>
<td>By the way, I found a good restaurant nearby.</td>
<td>17.5</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>Abrakadabra!</td>
<td>15.5</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>It’s a beautiful day.</td>
<td>13.1</td>
</tr>
</tbody>
</table>

- (Zero-shot) 17.7
Toolformer

- LLMs can be designed to use tools like calculators, QA systems and calendars to produce better results
- LLM are trained to generate and issue queries to external tools

Limitations: Hallucinations

Who was the first person to walk on the moon?

Answer: The first person to walk on the moon was Charles Lindbergh in 1951, during the Lunar Pioneer mission. His historic moonwalk was a testament to human spirit and was broadcast live to millions of people around the globe.

Correct Answer: Neil Armstrong was the first person to walk on the moon in 1969 during the Apollo 11 mission.

(a) Factuality Hallucination

Please summarize the following news article:

Context: In early October 2023, war broke out between Israel and Hamas, the militant Islamist group that has controlled Gaza since 2006. Hamas fighters fired rockets ... civilians and taking dozens of hostages.

Answer: In October 2006, Israel declared war on Hamas after an unexpected attack, prompting ongoing violence, civilian crises, and regional conflict escalation.

(b) Faithfulness Hallucination

Figure 1: An intuitive example of LLM hallucination.
How to Incorporate Uncertainty

- Language Models (Mostly) Know What They Know
  - Kadavath et al. (2022)
- LMs are pretty well calibrated!
  - Multiple choice evaluation
How to Incorporate Uncertainty

- Language Models (Mostly) Know What They Know
  - Kadavath et al. (2022)
- Can ask the LM to evaluate the likelihood of their own answers
### Significant Energy Consumption

<table>
<thead>
<tr>
<th>Consumption</th>
<th>CO₂e (lbs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air travel, 1 passenger, NY↔SF</td>
<td>1984</td>
</tr>
<tr>
<td>Human life, avg, 1 year</td>
<td>11,023</td>
</tr>
<tr>
<td>American life, avg, 1 year</td>
<td>36,156</td>
</tr>
<tr>
<td>Car, avg incl. fuel, 1 lifetime</td>
<td>126,000</td>
</tr>
</tbody>
</table>

**Training one model (GPU)**

<table>
<thead>
<tr>
<th>Model</th>
<th>CO₂e (lbs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLP pipeline (parsing, SRL)</td>
<td>39</td>
</tr>
<tr>
<td>w/ tuning &amp; experimentation</td>
<td>78,468</td>
</tr>
<tr>
<td>Transformer (big)</td>
<td>192</td>
</tr>
<tr>
<td>w/ neural architecture search</td>
<td>626,155</td>
</tr>
</tbody>
</table>

BERT

T5/BART

GPT

I like black coffee

Ich mag schwarzen Kaffee

<START> Ich mag schwarzen Kaffee
Recap

- Different variants of language models
  - Encoder only, encoder-decoder, decoder only
- Performance improves with data and model size
  - Can actually fit power laws and predict performance before training!
- LLMs learn how to do a variety of tasks during pre-training
- A second phase is required to give pre-trained LMs instruction following capabilities
  - Instruction tuning
- Performance can depend on the prompt
- LMs often hallucinate!