Scaling Language models

He He

NEW YORK UNIVERSITY

Nov 1, 2023

What do language models do?

- Answer questions
- Summarize documents
- Write programs
- Prove theorems
- ...

User What is funny about this image? Describe it panel by panel.

Source: `https://www.reddit.com/r/hmmm/comments/ubab5v/hmmm/`

GPT-4 The image shows a package for a "Lightning Cable" adapter with three panels.

Panel 1: A smartphone with a VGA connector (a large, blue, 15-pin connector typically used for computer monitors) plugged into its charging port.

Panel 2: The package for the "Lightning Cable" adapter with a picture of a VGA connector on it.

Panel 3: A close-up of the VGA connector with a small Lightning connector (used for charging iPhones and other Apple devices) at the end.

The humor in this image comes from the absurdity of plugging a large, outdated VGA connector into a small, modern smartphone charging port.

Dial back ten years

Which output is more likely?

- Speech recognition

  the *tail* of a dog
  the *tale* of a dog

  It's not easy to *wreck a nice beach*.
  It's not easy to *recognize speech*.
  It's not easy to *wreck an ice beach*.

- Machine translation

  He sat on the *table*.
  He sat on the *figure*.

  Such a Europe would *the rejection of any* ethnic nationalism.
  Such a Europe would *mark the refusal of all* ethnic nationalism.

Outline

Plan for today:

From n-gram language models to GPT

Problem formulation

Goal: Assign probabilities to a sequence of tokens, e.g.,

- $p(\text{the red fox jumped})$ $\quad$ $p(\text{the green fox jumped})$

Problem formulation

Goal: Assign probabilities to a sequence of tokens, e.g.,

- $p(\text{the red fox jumped}) \gg p(\text{the green fox jumped})$

Problem formulation

Goal: Assign probabilities to a sequence of tokens, e.g.,

- $p(\text{the red fox jumped}) \gg p(\text{the green fox jumped})$
- $p(\text{colorless green ideas sleep furiously})$
  $p(\text{furiously sleep ideas green colorless})$

Problem formulation

Goal: Assign probabilities to a sequence of tokens, e.g.,

- $p(\text{the red fox jumped}) \gg p(\text{the green fox jumped})$
- $p(\text{colorless green ideas sleep furiously}) \gg$
  $p(\text{furiously sleep ideas green colorless})$

Formulation:

Problem formulation

Goal: Assign probabilities to a sequence of tokens, e.g.,

- p(the red fox jumped) $\gg$ p(the green fox jumped)
- p(colorless green ideas sleep furiously) $\gg$ p(furiously sleep ideas green colorless)

Formulation:

- **Vocabulary**: a set of symbols $\mathcal{V}$, e.g. $\{\text{fox}, \text{green}, \text{red}, \text{jumped}, \text{a}, \text{the}\}$

Problem formulation

Goal: Assign probabilities to a sequence of tokens, e.g.,

- $p(\text{the red fox jumped}) \gg p(\text{the green fox jumped})$
- $p(\text{colorless green ideas sleep furiously}) \gg$
  $p(\text{furiously sleep ideas green colorless})$

Formulation:

- **Vocabulary**: a set of symbols $\mathcal{V}$, e.g.
  $\{\text{fox}, \text{green}, \text{red}, \text{jumped}, \text{a}, \text{the}\}$
- **Sentence**: a finite sequence over the vocabulary $x_1 x_2 \ldots x_n \in \mathcal{V}^n$ where $n \geq 0$

Problem formulation

Goal: Assign probabilities to a sequence of tokens, e.g.,

- $p(\text{the red fox jumped}) \gg p(\text{the green fox jumped})$
- $p(\text{colorless green ideas sleep furiously}) \gg$
  $p(\text{furiously sleep ideas green colorless})$

Formulation:

- **Vocabulary**: a set of symbols $\mathcal{V}$, e.g.
  $\{\text{fox}, \text{green}, \text{red}, \text{jumped}, \text{a}, \text{the}\}$
- **Sentence**: a finite sequence over the vocabulary $x_1 x_2 \ldots x_n \in \mathcal{V}^n$ where $n \geq 0$
- The set of all sentences (of varying lengths): $\mathcal{V}^*$

Problem formulation

Goal: Assign probabilities to a sequence of tokens, e.g.,

- p(the red fox jumped) $\gg$ p(the green fox jumped)
- p(colorless green ideas sleep furiously) $\gg$ p(furiously sleep ideas green colorless)

Formulation:

- **Vocabulary**: a set of symbols $\mathcal{V}$, e.g. $\{$fox, green, red, jumped, a, the$\}$
- **Sentence**: a finite sequence over the vocabulary $x_1 x_2 \ldots x_n \in \mathcal{V}^n$ where $n \geq 0$
- The set of all sentences (of varying lengths): $\mathcal{V}^*$
- Assign a probability $p(x)$ to all sentences $x \in \mathcal{V}^*$.

A naive solution

- **Training data**: a set of N sentences

A naive solution

- **Training data**: a set of N sentences
- **Modeling**: use a multinomial distribution as our language model

$$p_s(x) = \frac{\text{count}(x)}{N} \ .$$

(Exercise: Check that $\sum_{x \in \mathcal{V}^*} p_s(x) = 1$.)

A naive solution

- **Training data**: a set of N sentences
- **Modeling**: use a multinomial distribution as our language model

$$p_s(x) = \frac{\text{count}(x)}{N} \ .$$

(Exercise: Check that $\sum_{x \in \mathcal{V}^*} p_s(x) = 1$.)

- Is p_s a good LM?

A naive solution

- **Training data**: a set of N sentences
- **Modeling**: use a multinomial distribution as our language model

$$p_s(x) = \frac{\text{count}(x)}{N} .$$

(Exercise: Check that $\sum_{x \in \mathcal{V}^*} p_s(x) = 1$.)

- Is p_s a good LM?

A naive solution

- **Training data**: a set of N sentences
- **Modeling**: use a multinomial distribution as our language model

$$p_s(x) = \frac{\text{count}(x)}{N} .$$

(Exercise: Check that $\sum_{x \in \mathcal{V}^*} p_s(x) = 1$.)

- Is p_s a good LM?
  - Most sentences only occur once. *sparsity issue*

A naive solution

- **Training data**: a set of N sentences
- **Modeling**: use a multinomial distribution as our language model

$$p_s(x) = \frac{\text{count}(x)}{N} .$$

(Exercise: Check that $\sum_{x \in \mathcal{V}^*} p_s(x) = 1$.)

- Is p_s a good LM?
  - Most sentences only occur once. *sparsity issue*
  - Need to restrict the model.

Simplification 1: sentence to tokens

Solve a smaller problem: model probability of each token

Decompose the joint probability using the **probability chain rule**:

$$
\begin{aligned}
p(x) &= p(x_1, \dots, x_n) \\
&= p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2) \dots p(x_n \mid x_1, \dots, x_{n-1})
\end{aligned}
$$

Simplification 1: sentence to tokens

Solve a smaller problem: model probability of each token

Decompose the joint probability using the **probability chain rule**:

$$
\begin{aligned}
p(x) &= p(x_1, \ldots, x_n) \\
&= p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2) \ldots p(x_n \mid x_1, \ldots, x_{n-1}) \\
&\text{(Doesn't have to go from left to right)} \\
&= p(x_n)p(x_{n-1} \mid x_n) \ldots p(x_1 \mid x_2, \ldots, x_n)
\end{aligned}
$$

Simplification 1: sentence to tokens

Solve a smaller problem: model probability of each token

Decompose the joint probability using the **probability chain rule**:

$$
\begin{aligned}
p(x) &= p(x_1, \ldots, x_n) \\
&= p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2) \ldots p(x_n \mid x_1, \ldots, x_{n-1}) \\
&\text{(Doesn't have to go from left to right)} \\
&= p(x_n)p(x_{n-1} \mid x_n) \ldots p(x_1 \mid x_2, \ldots, x_n)
\end{aligned}
$$

- Problem reduced to modeling conditional token probabilities
  the red fox $\rightarrow$ jumped

Simplification 1: sentence to tokens

Solve a smaller problem: model probability of each token

Decompose the joint probability using the **probability chain rule**:

$$
\begin{aligned}
p(x) &= p(x_1, \ldots, x_n) \\
&= p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2) \ldots p(x_n \mid x_1, \ldots, x_{n-1}) \\
&\text{(Doesn't have to go from left to right)} \\
&= p(x_n)p(x_{n-1} \mid x_n) \ldots p(x_1 \mid x_2, \ldots, x_n)
\end{aligned}
$$

- Problem reduced to modeling conditional token probabilities
  the red fox $\rightarrow$ jumped
- The left-to-right decomposition is also called an **autoregressive model**

Simplification 1: sentence to tokens

Solve a smaller problem: model probability of each token

Decompose the joint probability using the **probability chain rule**:

$$
\begin{aligned}
p(x) &= p(x_1, \ldots, x_n) \\
&= p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2) \ldots p(x_n \mid x_1, \ldots, x_{n-1}) \\
&\text{(Doesn't have to go from left to right)} \\
&= p(x_n)p(x_{n-1} \mid x_n) \ldots p(x_1 \mid x_2, \ldots, x_n)
\end{aligned}
$$

- Problem reduced to modeling conditional token probabilities
  the red fox $\rightarrow$ jumped
- The left-to-right decomposition is also called an **autoregressive model**
- This is a classification problem we have seen

Simplification 1: sentence to tokens

Solve a smaller problem: model probability of each token

Decompose the joint probability using the **probability chain rule**:

$$
\begin{aligned}
p(x) &= p(x_1, \ldots, x_n) \\
&= p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2) \ldots p(x_n \mid x_1, \ldots, x_{n-1}) \\
&\quad \text{(Doesn't have to go from left to right)} \\
&= p(x_n)p(x_{n-1} \mid x_n) \ldots p(x_1 \mid x_2, \ldots, x_n)
\end{aligned}
$$

- Problem reduced to modeling conditional token probabilities
  the red fox $\rightarrow$ jumped
- The left-to-right decomposition is also called an **autoregressive model**
- This is a classification problem we have seen
- But there is still a large number of contexts!

Simplification 2: limited context

Reduce dependence on context by the **Markov assumption**:

- First-order Markov model

$$p(x_i \mid x_1, \ldots, x_{i-1}) = p(x_i \mid x_{i-1})$$
$$p(x) = \prod_{i=1}^{n} p(x_i \mid x_{i-1})$$

- Number of contexts?

Simplification 2: limited context

Reduce dependence on context by the **Markov assumption**:

- First-order Markov model

$$p(x_i \mid x_1, \ldots, x_{i-1}) = p(x_i \mid x_{i-1})$$

$$p(x) = \prod_{i=1}^{n} p(x_i \mid x_{i-1})$$

- Number of contexts? $|\mathcal{V}|$
- Number of parameters?

Simplification 2: limited context

Reduce dependence on context by the **Markov assumption**:

- First-order Markov model

$$p(x_i \mid x_1, \ldots, x_{i-1}) = p(x_i \mid x_{i-1})$$

$$p(x) = \prod_{i=1}^{n} p(x_i \mid x_{i-1})$$

- Number of contexts? $|\mathcal{V}|$
- Number of parameters? $|\mathcal{V}|^2$

Model sequences of variable lengths

Assume each sequence starts with a special **start symbol**: $x_0 = *$.

Assume that all sequences end with a **stop symbol** $\texttt{STOP}$, e.g.

$$
\begin{aligned}
& p(\text{the}, \text{fox}, \text{jumped}, \texttt{STOP}) \\
= \; & p(\text{the} \mid *)p(\text{fox} \mid \text{the})p(\text{jumped} \mid \text{fox})p(\texttt{STOP} \mid \text{jumped})
\end{aligned}
$$

Model sequences of variable lengths

Assume each sequence starts with a special **start symbol**: $x_0 = *$.

Assume that all sequences end with a **stop symbol** $\texttt{STOP}$, e.g.

$$\begin{aligned} & p(\text{the}, \text{fox}, \text{jumped}, \texttt{STOP}) \\ = \; & p(\text{the} \mid *) p(\text{fox} \mid \text{the}) p(\text{jumped} \mid \text{fox}) p(\texttt{STOP} \mid \text{jumped}) \end{aligned}$$

What if we don't have the stop symbol?

- Which one is larger: $p(\text{the fox})$ or $p(\text{the fox jumped})$?

N-gram LM

- Unigram language model (no context):

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i) \ .$$

- Bigram language model ($x_0 = *$):

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i \mid x_{i-1}) \ .$$

- n-gram language model:

$$p(x_1, \ldots, x_m) = \prod_{i=1}^{m} p(x_i \mid \underbrace{x_{i-n+1}, \ldots, x_{i-1}}_{\text{previous } n-1 \text{ words}}) \ .$$

Parameter estimation

Maximum likelihood estimation over a corpus (a set of sentences):

- Unigram LM

$$p_{\text{MLE}}(x) = \frac{\text{count}(w)}{\sum_{w \in \mathcal{V}} \text{count}(w)}$$

- Bigram LM

$$p_{\text{MLE}}(w \mid w') = \frac{\text{count}(w, w')}{\sum_{w \in \mathcal{V}} \text{count}(w, w')}$$

- In general, for n-gram LM,

$$p_{\text{MLE}}(w \mid c) = \frac{\text{count}(w, c)}{\sum_{w \in \mathcal{V}} \text{count}(w, c)}$$

  where $c \in \mathcal{V}^{n-1}$.

Example

- Training corpus (after tokenization)

  $$\{\text{The fox is red, The red fox jumped, I saw a red fox}\}$$

- Collect counts

  $$\text{count}(\text{fox}) = 3$$
  $$\text{count}(\text{red}) = 3$$
  $$\text{count}(\text{red}, \text{fox}) = 2$$
  $$\ldots$$

- Parameter estimates

  $$\hat{p}(\text{red} \mid \text{fox}) =$$

Example

- Training corpus (after tokenization)

  $$\{\text{The fox is red, The red fox jumped, I saw a red fox}\}$$

- Collect counts

  $$\text{count}(\text{fox}) = 3$$
  $$\text{count}(\text{red}) = 3$$
  $$\text{count}(\text{red}, \text{fox}) = 2$$
  $$\ldots$$

- Parameter estimates

  $$\hat{p}(\text{red} \mid \text{fox}) = 2/3$$
  $$\hat{p}(\text{saw} \mid \text{i}) =$$

Example

- Training corpus (after tokenization)

  $$\{\text{The fox is red, The red fox jumped, I saw a red fox}\}$$

- Collect counts

  $$\begin{aligned}
  &\text{count}(\text{fox}) = 3 \\
  &\text{count}(\text{red}) = 3 \\
  &\text{count}(\text{red}, \text{fox}) = 2 \\
  &\ldots
  \end{aligned}$$

- Parameter estimates

  $$\begin{aligned}
  &\hat{p}(\text{red} \mid \text{fox}) = 2/3 \\
  &\hat{p}(\text{saw} \mid \text{i}) = 1/1
  \end{aligned}$$

Example

- Training corpus (after tokenization)

  $$\{\text{The fox is red, The red fox jumped, I saw a red fox}\}$$

- Collect counts

  $$\begin{aligned} \text{count}(\text{fox}) &= 3 \\ \text{count}(\text{red}) &= 3 \\ \text{count}(\text{red}, \text{fox}) &= 2 \\ \ldots \end{aligned}$$

- Parameter estimates

  $$\begin{aligned} \hat{p}(\text{red} \mid \text{fox}) &= 2/3 \\ \hat{p}(\text{saw} \mid \text{i}) &= 1/1 \end{aligned}$$

- What is the probability of “The fox saw I jumped”?

Example

- Training corpus (after tokenization)

  $$\{\text{The fox is red}, \text{The red fox jumped}, \text{I saw a red fox}\}$$

- Collect counts

  $$\text{count}(\text{fox}) = 3$$
  $$\text{count}(\text{red}) = 3$$
  $$\text{count}(\text{red}, \text{fox}) = 2$$
  $$\ldots$$

- Parameter estimates

  $$\hat{p}(\text{red} \mid \text{fox}) = 2/3$$
  $$\hat{p}(\text{saw} \mid \text{i}) = 1/1$$

- What is the probability of "The fox saw I jumped"? Zero probability on unseen n-grams

Generating text from an n-gram model

1. Initial condition: context $= *$
2. Iterate until next_word is `STOP`:
   2.1 next_word $\sim p(\cdot \mid \text{context}[:-(n-1)])$
   2.2 context $\leftarrow$ context $+$ next_word

Generating text from an n-gram model

1. Initial condition: context $= *$
2. Iterate until next_word is `STOP`:
   2.1 next_word $\sim p(\cdot \mid \text{context}[: -(n-1)])$
   2.2 context $\leftarrow$ context $+$ next_word

1 gram	–To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have –Hill he late speaks; or! a more to leg less first you enter
2 gram	–Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow. –What means, sir. I confess she? then all sorts, he is trim, captain.
3 gram	–Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, 'tis done. –This shall forbid it should be branded, if renown made it empty.
4 gram	–King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv'd in; –It cannot be but so.

What is the training data?

Perplexity

What is the loss function for learning language models?

Perplexity

What is the loss function for learning language models?

Held-out likelihood on test data D (negative test loss):

$$\ell(D) = \sum_{i=1}^{|D|} \log p_\theta(x_i \mid x_{1:i-1}) \ ,$$

Perplexity

What is the loss function for learning language models?

Held-out likelihood on test data D (negative test loss):

$$\ell(D) = \sum_{i=1}^{|D|} \log p_\theta(x_i \mid x_{1:i-1}) \; ,$$

Perplexity:

$$\mathrm{PPL}(D) = 2^{-\frac{\ell(D)}{|D|}} \; .$$

- Base of log and exponentiation should match
- Exponent is cross entropy: $H(p_{data}, p_\theta) = -\mathbb{E}_{x \sim p} \log p_\theta(x)$.
- Interpretation: a model of perplexity k predicts the next word by throwing a fair k-sided die.

Summary

Language models: assign probabilities to sentences

Summary

Language models: assign probabilities to sentences

N-gram language models:

- Assume each word only conditions on the previous $n - 1$ words
- MLE estimate: counting n-grams in the training corpus

Summary

Language models: assign probabilities to sentences

N-gram language models:

- Assume each word only conditions on the previous $n - 1$ words
- MLE estimate: counting n-grams in the training corpus

Evaluation by **held-out perplexity**: how much probability mass does the model assign to unseen text

Summary

Language models: assign probabilities to sentences

N-gram language models:

- Assume each word only conditions on the previous $n - 1$ words
- MLE estimate: counting n-grams in the training corpus

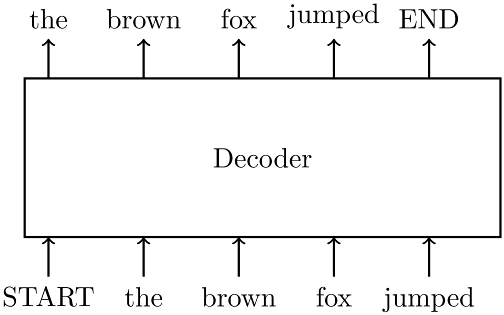
Evaluation by **held-out perplexity**: how much probability mass does the model assign to unseen text

Challenges:

- **Generalization**: sentences containing unseen n-grams have zero probability
- Much research in n-gram LM is dedicated to **smoothing** methods that allocate probability mass to unseen n-grams

Neural language models

Neural networks solve the generalization problem in n-gram LMs.



- A decoder-only autoregressive neural language model
- Decoder can be an RNN or a transformer (with causal masking)
- What’s the context size?

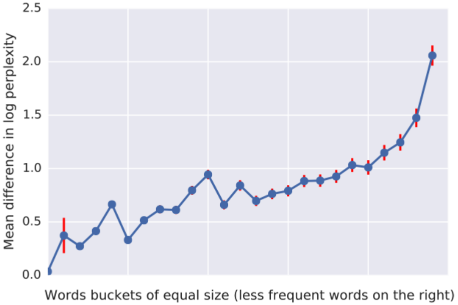
Early efforts on scaling neural language models

Model	Test Perplexity	Number of Params [billions]
Sigmoid-RNN-2048 (Ji et al., 2015a)	68.3	4.1
Interpolated KN 5-gram, 1.1B n-grams (Chelba et al., 2013)	67.6	1.76
Sparse Non-Negative Matrix LM (Shazeer et al., 2015)	52.9	33
RNN-1024 + MaxEnt 9-gram features (Chelba et al., 2013)	51.3	20
LSTM-512-512	54.1	0.82
LSTM-1024-512	48.2	0.82
LSTM-2048-512	43.7	0.83
LSTM-8192-2048 (No Dropout)	37.9	3.3
LSTM-8192-2048 (50% Dropout)	32.2	3.3
2-Layer LSTM-8192-1024 (BIG LSTM)	30.6	1.8
BIG LSTM+CNN Inputs	**30.0**	**1.04**
BIG LSTM+CNN Inputs + CNN Softmax	39.8	**0.29**
BIG LSTM+CNN Inputs + CNN Softmax + 128-dim correction	35.8	**0.39**
BIG LSTM+CNN Inputs + Char LSTM predictions	47.9	**0.23**

Figure: From Exploring the Limits of Language Modeling

Significant improvement in held-out perplexity given similar model sizes ($\sim$1B)

Improvement from neural language models



< S > With even more new technologies coming onto the market quickly during the past three years , an increasing number of companies now must tackle the ever-changing and ever-changing environmental challenges online . < S > Check back for updates on this breaking news story . < S > About 800 people gathered at Hever Castle on Long Beach from noon to 2pm , three to four times that of the funeral cortège . < S > We are aware of written instructions from the copyright holder not to , in any way , mention Rosenberg 's negative comments if they are relevant as indicated in the documents , " eBay said in a statement . < S > It is now known that coffee and cacao products can do no harm on the body . < S > Yuri Zhirkov was in attendance at the Stamford Bridge at the start of the second half but neither Drogba nor Malouda was able to push on through the Barcelona defence .

Figure: From Exploring the Limits of Language Modeling

LSTM vs KN5: improved perplexity on tail words

Recap: language modeling as pretraining

What can we do with a very large language model?

- The cats that are raised by my sister ________ sleeping. *syntax*
- Jane is happy that John invited ________ friends to his birthday party. *coreference*
- ________ is the capital of Tanzania. *knowledge*
- The boy is ________ because he lost his keys. *commonsense*
- John took 100 bucks to Vegas. He won 50 and then lost 100. Now he only has ________ to go home. *numerical reasoning*

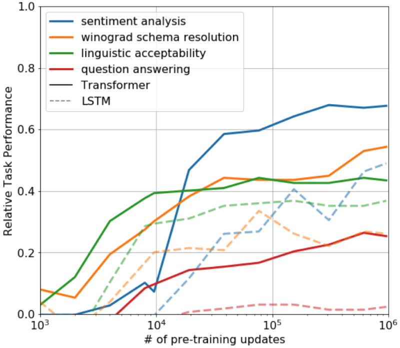
Predicting the next word entails many natural language understanding tasks

Recap: Zero-shot behaviors from GPT

Key insight: if the model has learned to understand language through predicting next words, it should be able to perform these tasks *without finetuning*

Recap: Zero-shot behaviors from GPT

Key insight: if the model has learned to understand language through predicting next words, it should be able to perform these tasks *without finetuning*



Heuristics for zero-shot prediction:

- Sentiment classification: [example] + very + {positive, negative} *prompting*
- Linguistic acceptability: thresholding on log probabilities
- Multiple choice: predicting the answer with the highest log probabilities

Learning dynamics: zero-shot performance increases during pretraining

GPT-2: going beyond finetuning

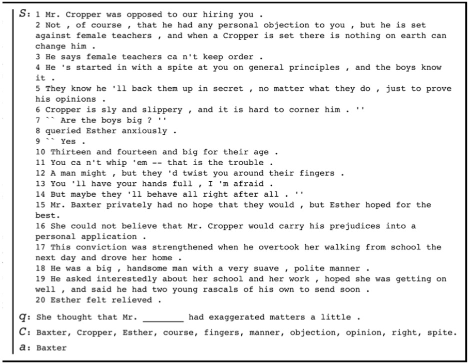
Language Models are Unsupervised Multitask Learners [Radford et al., 2019]

- Supervised learning: models must be trained (finetuned) on a curated task dataset.
- They fail to generalize to out-of-distribution data (adversarial examples, robustness issues etc.)
- A generalist model must be trained on *many* tasks—but how do we get the datasets?
- **Hypothesis**: a (large enough) LM should be able to infer and learn tasks demonstrated in natural language, effectively performing unsupervised multitask learning

GPT-2 details

- Similar to GPT-1 but scaled up (1.5B parameters)
- Data (WebText): $\sim$40GB of web pages scraped from the internet that was curated to include high-quality text
- Tokenization: BPE over byte sequences for universal text processing.
  - Small base vocabulary (256)
  - Can process any text data regardless of pre-processing, tokenization, or vocab size.
- Larger context size (1024 tokens)

Zero-shot performance: cloze test

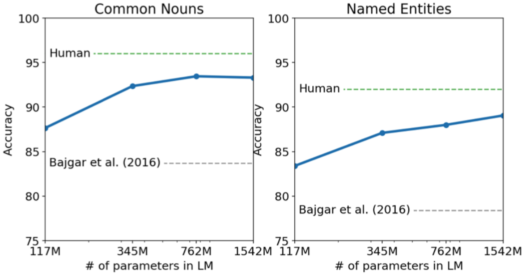


S: 1 Mr. Cropper was opposed to our hiring you .
2 Not , of course , that he had any personal objection to you , but he is set against female teachers , and when a Cropper is set there is nothing on earth can change him .
3 He says female teachers ca n't keep order .
4 He 's started in with a spite at you on general principles , and the boys know it .
5 They know he 'll back them up in secret , no matter what they do , just to prove his opinions .
6 Cropper is sly and slippery , and it is hard to corner him . ''
7 `` Are the boys big ? ''
8 queried Esther anxiously .
9 `` Yes .
10 Thirteen and fourteen and big for their age .
11 You ca n't whip 'em -- that is the trouble .
12 A man might , but they 'd twist you around their fingers .
13 You 'll have your hands full , I 'm afraid .
14 But maybe they 'll behave all right after all . ''
15 Mr. Baxter privately had no hope that they would , but Esther hoped for the best.
16 She could not believe that Mr. Cropper would carry his prejudices into a personal application .
17 This conviction was strengthened when he overtook her walking from school the next day and drove her home .
18 He was a big , handsome man with a very suave , polite manner .
19 He asked interestedly about her school and her work , hoped she was getting on well , and said he had two young rascals of his own to send soon .
20 Esther felt relieved .

q: She thought that Mr. ________ had exaggerated matters a little .

C: Baxter, Cropper, Esther, course, fingers, manner, objection, opinion, right, spite.

a: Baxter



Larger models quickly closes the gap with human performance

Zero-shot performance: generative QA

Question	Generated Answer	Correct	Probability
Who wrote the book the origin of species?	Charles Darwin	✓	83.4%
Who is the founder of the ubuntu project?	Mark Shuttleworth	✓	82.0%
Who is the quarterback for the green bay packers?	Aaron Rodgers	✓	81.1%
Panda is a national animal of which country?	China	✓	76.8%
Who came up with the theory of relativity?	Albert Einstein	✓	76.4%
When was the first star wars film released?	1977	✓	71.4%
What is the most common blood type in sweden?	A	✗	70.6%
Who is regarded as the founder of psychoanalysis?	Sigmund Freud	✓	69.3%
Who took the first steps on the moon in 1969?	Neil Armstrong	✓	66.8%
Who is the largest supermarket chain in the uk?	Tesco	✓	65.3%
What is the meaning of shalom in english?	peace	✓	64.0%
Who was the author of the art of war?	Sun Tzu	✓	59.6%
Largest state in the us by land mass?	California	✗	59.2%
Green algae is an example of which type of reproduction?	parthenogenesis	✗	56.5%
Vikram samvat calender is official in which country?	India	✓	55.6%
Who is mostly responsible for writing the declaration of independence?	Thomas Jefferson	✓	53.3%
What us state forms the western boundary of montana?	Montana	✗	52.3%
Who plays ser davos in game of thrones?	Peter Dinklage	✗	52.1%
Who appoints the chair of the federal reserve system?	Janet Yellen	✗	51.5%
State the process that divides one nucleus into two genetically identical nuclei?	mitosis	✓	50.7%

Zero-shot performance: summarization

	R-1	R-2	R-L	R-AVG
Bottom-Up Sum	**41.22**	**18.68**	**38.34**	**32.75**
Lede-3	40.38	17.66	36.62	31.55
Seq2Seq + Attn	31.33	11.81	28.83	23.99
GPT-2 `TL;DR:`	29.34	8.27	26.58	21.40
Random-3	28.78	8.63	25.52	20.98
GPT-2 no hint	21.58	4.03	19.47	15.03

- Challenge: not a “native” LM task
- Induce the task: [document] + [TL;DR]
- Not much better than copying 3 random sentences from the document
- Key question in the zero-shot paradigm: how do we tell the model what the intended task is?

Zero-shot performance: machine translation

- Induce the task through a demonstration example:

  translation $\sim p(\cdot \mid$ [french sentence] = [english sentence]; [french sentence] =$)$

- WMT-14 French-English test set: 11.5 BLEU (worse than unsupervised MT)
- But, there's only 10MB french data in the 40GB training data!
  - Typical unsupervised MT methods require crosslingual embeddings or monolingual corpora

Has the model memorized everything?

Is there data contamination (test data in training set)?

- Approach: check percentage of 8-grams that occur in both training and test data (using Bloom filters)

Has the model memorized everything?

Is there data contamination (test data in training set)?

- Approach: check percentage of 8-grams that occur in both training and test data (using Bloom filters)
- Overlap is not higher than existing overlap on train and test in datasets

Has the model memorized everything?

Is there data contamination (test data in training set)?

- Approach: check percentage of 8-grams that occur in both training and test data (using Bloom filters)
- Overlap is not higher than existing overlap on train and test in datasets
  - Overlap between test and WebText: 3.2%

Has the model memorized everything?

Is there data contamination (test data in training set)?

- Approach: check percentage of 8-grams that occur in both training and test data (using Bloom filters)
- Overlap is not higher than existing overlap on train and test in datasets
    - Overlap between test and WebText: 3.2%
    - Overlap between test and their own training split: 5.9%

Has the model memorized everything?

Is there data contamination (test data in training set)?

- Approach: check percentage of 8-grams that occur in both training and test data (using Bloom filters)
- Overlap is not higher than existing overlap on train and test in datasets
  - Overlap between test and WebText: 3.2%
  - Overlap between test and their own training split: 5.9%
- Model performance does get better when the test data is in pretraining

Has the model memorized everything?

Is there data contamination (test data in training set)?

- Approach: check percentage of 8-grams that occur in both training and test data (using Bloom filters)
- Overlap is not higher than existing overlap on train and test in datasets
  - Overlap between test and WebText: 3.2%
  - Overlap between test and their own training split: 5.9%
- Model performance does get better when the test data is in pretraining
  - E.g., on CoQA, 3 F1 better on leaked documents

Has the model memorized everything?

Is there data contamination (test data in training set)?

- Approach: check percentage of 8-grams that occur in both training and test data (using Bloom filters)
- Overlap is not higher than existing overlap on train and test in datasets
  - Overlap between test and WebText: 3.2%
  - Overlap between test and their own training split: 5.9%
- Model performance does get better when the test data is in pretraining
  - E.g., on CoQA, 3 F1 better on leaked documents
- Verifying data contamination is an active research area!

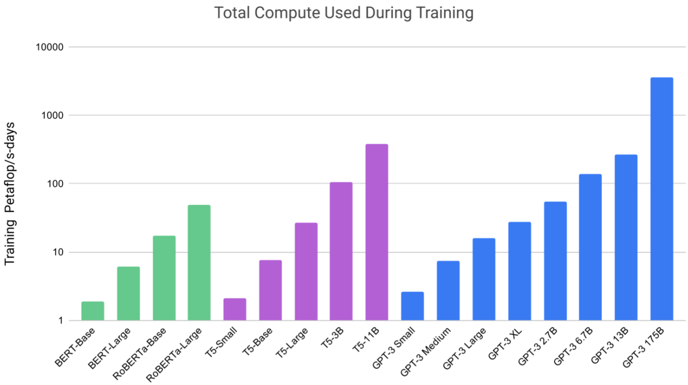
Has the model memorized everything?

Test the model on novel tasks:

Context (human-written): In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.
GPT-2: The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science. Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved. Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

GPT-3: scaling up

- GPT-2 shows promise for zero-shot learning, but performance is still unsatisfying
- GPT-3 scales up model size, data size and diversity, and number of training steps
- Notable improvement in zero-shot and few-shot performance
- Inducing a task through natural language task descriptions



What does scaling mean?

α x Mode l size x Training tokens = Training compute

	Model size (# parameters)	Training data (# tokens)	Training compute (FLOPs)	*Resources*
BERT-base (2018)	109M	250B	1.6e20	64 TPU v2 for 4 days (16 V100 GPU for 33 hrs)
GPT-3 (2020)	175B	300B	3.1e23	*~1,000x BERT-base*
PaLM (2022)	540B	780B	2.5e24	*6k TPU v4 for 2 months*

Figure: From Jason Wei's slides

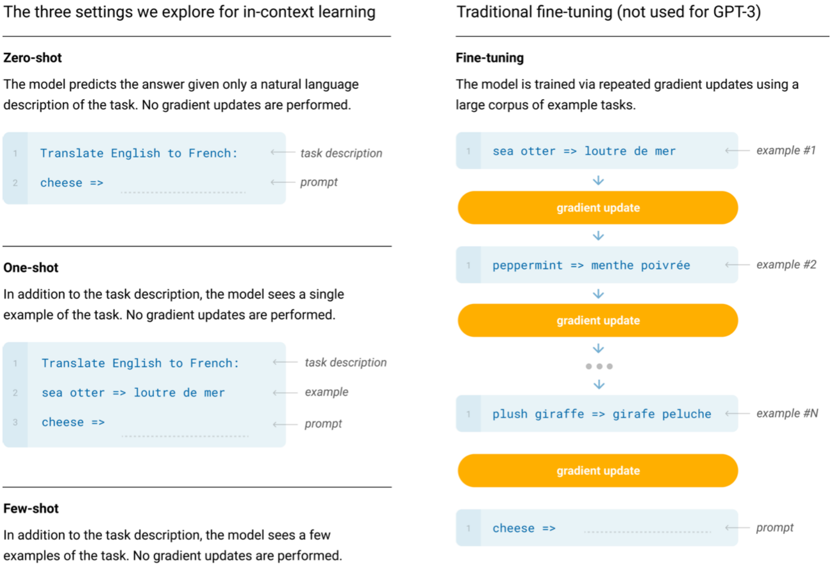
Training data

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

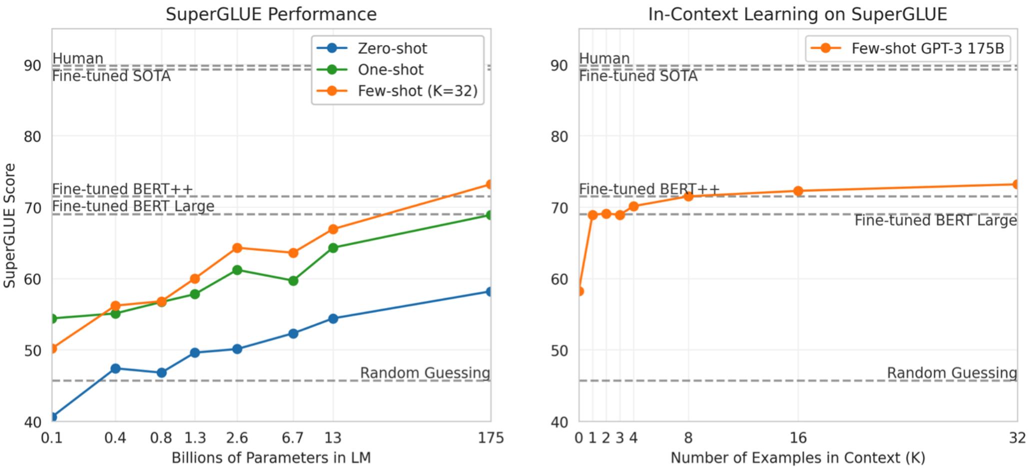
Key challenge: data quality control

- Filter CommonCrawl based on similarity to high-quality reference corpora
- Fuzzy deduplication: avoid redundancy and data contamination
- Mix in known high quality data
- Upsampling high quality data during training

Evaluation settings

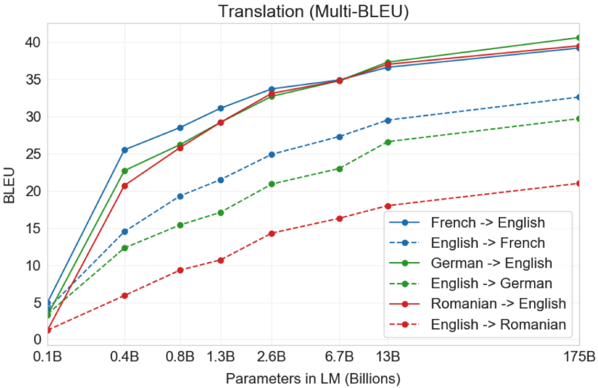


Results: natural language understanding



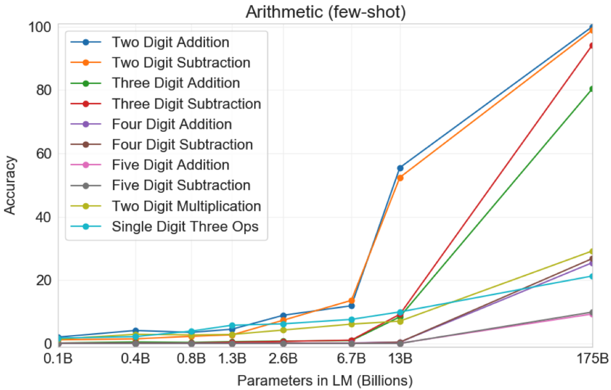
Comparable to supervised results

Results: few-shot machine translation



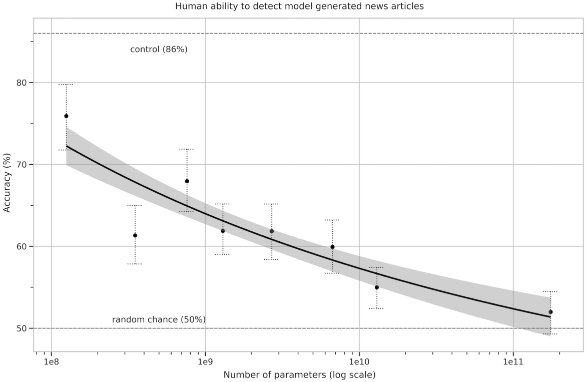
- Pretraining data: 93% English, 7% other languages
- Zero-shot is still worse than unsupervised MT
- But even giving one examples significantly boosts the result (+7 BLEU points)
- Results much better when translation into English
- Also see [Briakou et al., 2023] for the impact of bilingual data on MT performance

Results: arithmetic



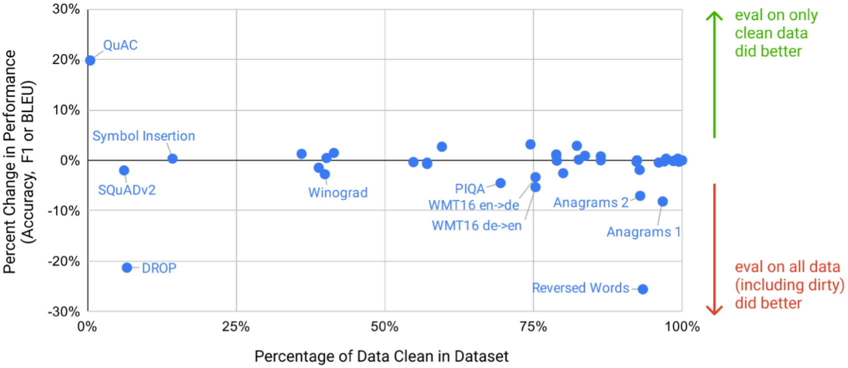
- ”Emergent” ability at certain model scale
- Not systemic: works better on frequent numbers [Razeghi et al., 2022]

Results: generation



Generated text is hard to detect from human-written text

Analysis: data contamination



- Overlap can be large (e.g., many reading comprehension articles come from wikipedia)
- Result on clean part of the benchmark doesn't change much

Conclusion

- A perfect language model on all human-written text can do all text-based tasks
- New behaviors that are not written in the training objective emerge (e.g., in-context learning)
- Open questions:
  - How much are they memorizing vs generalizing?
  - How do new abilities emerge?
  - How to mitigate harmful, toxic, biased responses?