

# Text generation

Co-writing with AI language models

Prof. Perttu Hämäläinen, Jan 2024

Some of the material courtesy of Prof. Christian Guckelsberger

# Vocabulary

**LLM:** Large Language Model (GPT-3, ChatGPT, Claude, Llama...)

**GPT:** Generative Pretrained Transformer

**Generative:** The model generates samples (text snippets, images...)

**Pretrained:** The model is first “pretrained” on a large and diverse dataset, which allows it to develop general capabilities that approximate understanding and reasoning. Afterwards, the same model can be finetuned for a particular task such as coding assistance. For this, a smaller dataset suffices and the resulting model is much stronger than if it was only trained on the finetuning data.

**Transformer:** A particular neural network architecture published in 2017 that revolutionized natural language processing. Used in all current major language models.

# Contents

- Theory: Language generation and the Transformer architecture
- Examples: What can these models do?
- Practice: How to use?

# Theory

# Large Language Models from scratch



Large Language Models from scratch



Graphics in 5 Minutes

1.48K subscribers

Subscribe

498



Share

Clip

Save

...

Excellent 8-minute intro <https://www.youtube.com/watch?v=lnA9DMvHtfl>



# What is natural language generation?

- Informally: Given an (unfinished) sentence, tell me which word (token) is likely to come next. And then the word after...

and cats dogs men women



Language Model

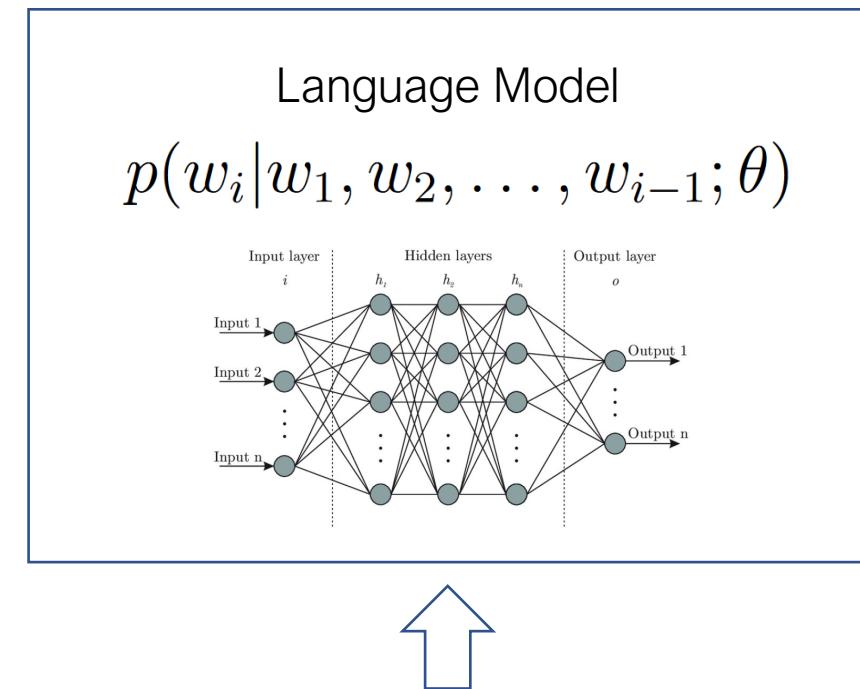
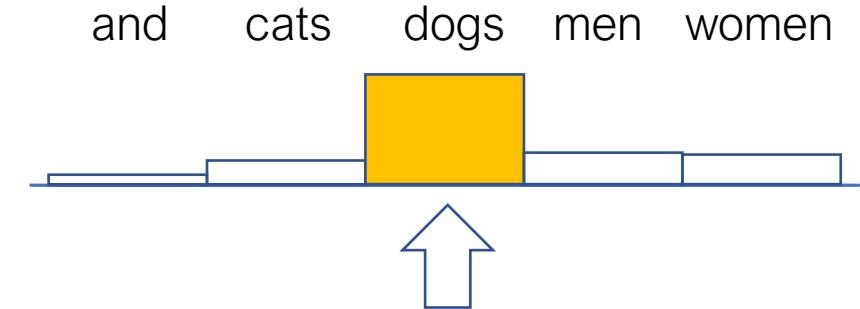


It's raining cats and \_\_\_\_\_



# What is natural language generation?

- **Informally:** Given an (unfinished) sentence, tell me which word (token) is likely to come next. And then the word after...
- **Formally:**
  - Conditional probability distribution:  
 $p(w_i|w_1, w_2, \dots, w_{i-1}; \theta)$
  - Next word sampled from distribution:  
 $w_i \sim p(w_i|w_1, w_2, \dots, w_{i-1}; \theta)$
  - Parameters  $\theta$  e.g. weights of a neural net (-> lecture 2) learned from data.

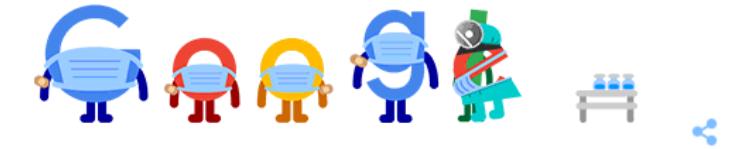


It's raining cats and dogs  
 $w_1 \quad w_2 \quad \dots \quad w_{i-1}$



# What is natural language generation?

- Exercise: What use-cases of natural language generation can you think of?
  - Search query completion
  - ...



Use cases for | X

- [use cases for nfts](#)
- [use cases for blockchain](#)
- [use cases for kafka](#)
- [use cases for atm system](#)
- [use cases for mongodb](#)
- [use cases for smart contracts](#)
- [use cases for nosql](#)
- [use cases for ethereum](#)
- [use cases for graph databases](#)
- [use cases for cryptocurrency](#)

Google Search I'm Feeling Lucky

Report inappropriate predictions



# What is natural language generation?

- Exercise: What use-cases of natural language generation can you think of?
  - Search query completion
  - ...
  - Code completion
  - Chatbots
  - Poetry writing
  - Storytelling

## Computational Approaches to Storytelling and Creativity

Pablo Gervás

■ This article deals with computational approaches to storytelling, or the production of stories by computers, with a particular attention on the way human creativity is modeled or emulated in computational terms. Features relevant to creativity and to stories are analyzed, and existing systems are reviewed under the light of that analysis. The extent to which they implement the key features proposed in recent models of computational creativity is discussed. Limitations, avenues of future research, and expected trends are outlined.

Creativity and storytelling are two ideas that one does not naturally associate with computers. Yet over the last few years there has been a surge of research efforts concerning the combination of both subjects. This article tries to shed light on these efforts. In carrying out this program, one is handicapped by the fact that, as words, both *creativity* and *storytelling* are severely lacking in the precision one expects of words to be used for intellectual endeavor. If a speaker were to mention either word in front of an audience, each person listening would probably come up with a different mental picture of what is intended. To avoid the risks that such vagueness might lead to, an initial effort is made here to restrict the endeavor to those aspects that have been modeled computationally in some model or system. The article then proceeds to review some of the research efforts that have addressed these problems from a computational point of view.

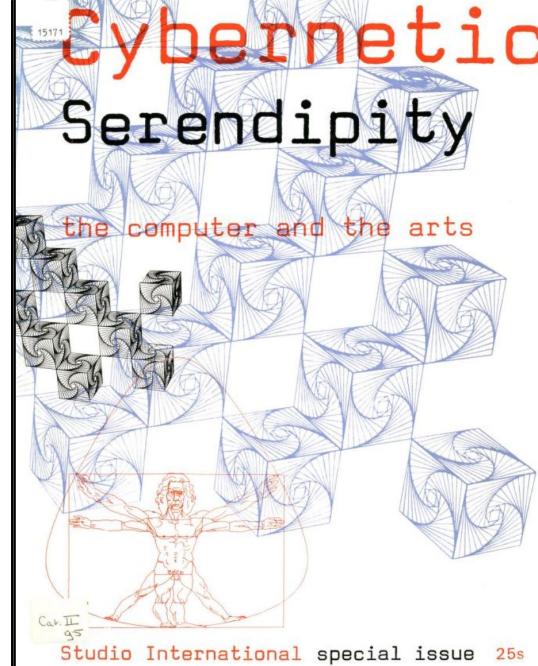
### The Creative Process

Thinking of "creativity" evokes several ideas that seem to go together. It generally brings up the idea of someone generating something new. But it also has connotations that whatever is generated must be somewhat unexpected or different from what others might have produced. There is also an implicit restriction that what is generated satisfy some goal, though in many cases the particular goal implied is not altogether clear. The fact that someone is involved indicates we are reviewing an explicit action by some agent that we shall refer to as the *creator*. A sunset may generate a totally new combination of colors, possibly unexpected, but it would not be considered creative. The fact that something is generated indicates that the creative action

# Model Evolution

- For the model output to be **creative** (Runco & Jäger, 2012), it must be:
  - Novel:** not in training set
  - Value:** syntactically correct, meaningful, ... + domain-specific metrics.

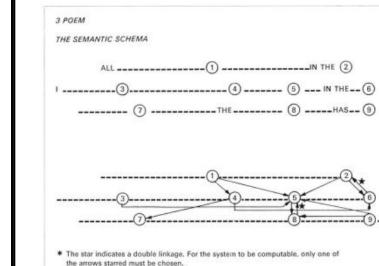
Runco & Jaeger (2012). The Standard Def. of Creativity. *Creativity Research Journal*, 24(1), 92–96.  
 Reichardt, ed. (1968). "Cybernetic Serendipity - The Computer and the Arts; a Studio international special issue". *Studio International*. London: The Studio Trust.



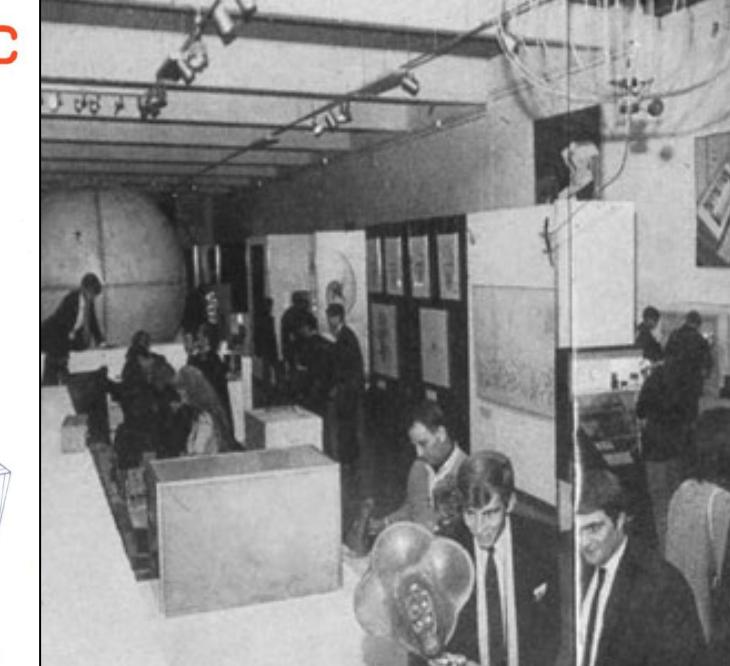
## Computerized Japanese haiku

These are examples, produced by on-line man-machine interaction at the Cambridge Language Research Unit, of one use of a computer for producing poetry. The programme is written in the TRAC language. The user can type in a formula, in which the operator types words. In '1 Poem' and '2 Poem', the operator chooses his words as he wishes. In the two '3 Poem' poems were produced by Margaret Masterman and Robin McKinnon Wood.

Slot 1 (-→ 4) (-→ 5)	Slot 2 (-→ 5) (-→ 6)	Slot 3 (-→ 5)	Slot 4 (-→ 6)	Slot 5 (-→ 8)	Slot 6 (-→ 7)	Slot 7	Slot 8 (-→ 5) (-→ 8)
White	Buds	See	Snow	Trees	Spring	Bang	Sun
Blue	Twigs	Trace	Tall	Full	Hugh	Fist	Fled
Red	Leaves	Glimpse	Peaks	Pall	Swish	Star	Dimmed
Black	Hills	Flash	Dark	Hills	Pffft	Cloud	Cracked
Green	Peak	Faint	Streams	Cold	Wheeze	Shout	Smashed
Brown	Snow	Taste	White	Specks	Flick	Streak	Blown
Bright	Sun	Hear	Clear	Shade	Shoo	Tree	Grr
Pure	Seize	Red	Arcs	Dawn	Bud	Bud	Smash
Clouded	Rain	Blue	Grass	Dusk	Spit	Spit	Crashed
Crowned	Cloud	Green	Stems	Day	Crash	Child	Gone
Starred	Sky	Grey	Cows	Mist	Look	Crane	Fogged
Fog	Dawn	Black	Deer	Trees	Crash	Plane	Burst
Spring	Mat	Round	Stars	Woods	Crash	Moth	Moth
Heat	Squares	Straight	Clouds	Hills	Crash		
Cold	Flowers	Curved	Leaves	Pools	Crash		



54



## Computer poetry from CLRU

Robin McKinnon Wood and Margaret Masterman

**Output resulting from a bug in the segmentation programme**  
 This programme is designed to cut continuous lines of text into segments corresponding to the rhythmic divisions of speech or spoken prose. These units usually include two stress-points and a terminal intonation feature involving breath-groups which are also sense-groups.

The phonetic evidence and its relevance to silent reading was studied by David Shillan,

the generalization of a two-stressed structure as a semantic form is the work of

Margaret Masterman. The segmentation programme, which in effect simulates human perception of these phrasings, is by John Dobson at the CLRU. This output was produced by the titan computer of the C.R.C. In the case of this particular output, a bug in the programme effectively randomized the text given to it. We think this was caused by the programme taking as its text the first word on each line of the original text, but we are not sure as we failed to make the programme do it again. At any rate this 'poem' is all the computer's own work!

**Job Title (JED744/Phrasing) 10 7 67**  
**Stream 1/0 (Phrasings)**  
 1/0/1 I I development at the point normal  
 1/0/2 a homemaker  
 1/0/3 that extensive service  
 1/1/1 as a homemaker  
 1/1/2 visiting there  
 1/1/3 homemaker

into six lines of four metrical units each. The work of programming was on 322 punched cards, with 1200 instructions. Much of the work is illustrated. The full experiment was performed on an IBM 7070 computer at the Electronic Centre of the Lombard Provinces Savings Bank in Milan in October 1961.

## Tape Mark I

Nanni Balestrini  
 The basic text is made up of three extracts taken from:  
 (1) Maestro Hashiba's *Hiroshima diary*:  
 "the blinding fireball expands rapidly  
 thirty times brighter than the sun when  
 it reaches the stratosphere the summit of  
 the clouds takes on the well known  
 mushroom shape".

(2) Paul Goldwin's *The mystery of the elevator*: "head pressed on shoulder hair between the lay motionless without  
 a break till he proved his fingers  
 slowly trying to grasp".  
 (3) Lao Tzu's *Tao Te Ching*: "while the  
 multitude of things come into being I  
 envy them not; neither among things  
 flourish they all return to their roots".  
 The working instructions for the computer  
 are as follows:  
 (a) Make all combinations of ten elements  
 out of the given fifteen, without per-  
 mutations or repetitions.  
 (b) Construct chains of elements taking  
 account of the head-codes and end-  
 codes.  
 (c) Avoid juxtaposing elements drawn  
 from the same extract.  
 (d) Subdivide the chains of ten elements

Head pressed on shoulder, thirty times  
 brighter than the sun I envisage their return,  
 until he moved his fingers slowly and while  
 of things comes into being, at the summit  
 they all return to their roots and take on  
 the well known mushroom shape  
 endeavouring to grasp.  
 Hair between lips, they all return  
 to their roots, in the blinding fireball  
 I envy their return, until he moves his  
 fingers  
 slowly and although things flourish  
 on the well known mushroom shape  
 endeavouring to grasp while the multitude of  
 things comes into being, head pressed  
 on shoulder, thirty times brighter than  
 the sun I envisage their return, until he moves his  
 fingers  
 Head pressed on shoulder, thirty times  
 brighter than the sun I envisage their return,  
 until he moved his fingers slowly and while  
 of things comes into being, at the summit  
 they all return to their roots and take on  
 the well known mushroom shape  
 endeavouring to grasp.  
 Hair between lips, they all return  
 to their roots, in the blinding fireball  
 I envy their return, until he moves his  
 fingers  
 slowly and although things flourish  
 on the well known mushroom shape  
 endeavouring to grasp while the multitude of  
 things comes into being, head pressed  
 on shoulder, thirty times brighter than  
 the sun I envisage their return, until he moves his  
 fingers

55

# Model Evolution

- For the model output to be **creative** (Runco & Jäger, 2012), it must be:
  - **Novel**: not in training set
  - **Value**: syntactically correct, meaningful, ... + domain-specific metrics.
- We get **novelty** from sampling through recombination. But not necessarily value.

and cats dogs men women



**Assumption:** similar distribution irrespective of token

**Input:** It's raining ...

**1st sampling:**

It's raining dogs ...

It's raining and ...

It's raining women ...

**2nd sampling:**

It's raining dogs dogs ...

It's raining and women ...

It's raining women and ...

**3rd sampling:**

It's raining dogs dogs men ...

It's raining and women cats ...

It's raining women and and ...



# Model Evolution

- For the model output to be **creative** (Runco & Jäger, 2012), it must be:
  - **Novel**: not in training set
  - **Value**: syntactically correct, meaningful, ... + domain-specific metrics.
- We get **novelty** from sampling through recombination. But not necessarily value.
- **Value**: learn better distributions via:
  1. better and more **training data**,

Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

Table 1: **Pre-training data.** Data mixtures used for pre-training, for each subset we list the sampling proportion, number of epochs performed on the subset when training on 1.4T tokens, and disk size. The pre-training runs on 1T tokens have the same sampling proportion.

<https://ai.facebook.com/blog/large-language-model-llama-meta-ai/>

# Model Evolution

- For the model output to be **creative** (Runco & Jäger, 2012), it must be:
  - **Novel**: not in training set
  - **Value**: syntactically correct, meaningful, ... + domain-specific metrics.
- We get **novelty** from sampling through recombination. But not necessarily value.
- **Value**: learn better distributions via:
  1. better and more **training data**,
  2. increased **model complexity**,

**BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**  
**(2018) 355M parameters**

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova  
Google AI Language

{jacobdevlin, mingweichang, kentonl, kristout}@google.com

## Language Models are Unsupervised Multitask Learners

Alec Radford \*<sup>†</sup> Jeffrey Wu \*<sup>†</sup> Rewon Child<sup>†</sup> David Luan<sup>†</sup> Dario Amodei \*\*<sup>†</sup> Ilya Sutskever \*\*<sup>†</sup>  
**(2019) 1.5B parameters**

### Abstract

Natural language processing tasks, such as ques-

competent generalists. We would like to move towards more general systems which can perform many tasks – eventually without the need to manually create and label a training

Journal of Machine Learning Research 21 (2020) 1-67

Submitted 1/20; Revised 6/20; Published 6/20

**(2019/20) 11B parameters**

## Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

Colin Raffel\*

CRAFFEL@GMAIL.COM

## Language Models are Few-Shot Learners

**(2020) 175B parameters**

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

Jared Kaplan<sup>†</sup> Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry

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

# Model Evolution

- For the model output to be **creative** (Runco & Jäger, 2012), it must be:
  - **Novel**: not in training set
  - **Value**: syntactically correct, meaningful, ... + domain-specific metrics.
- We get **novelty** from sampling through recombination. But not necessarily value.
- **Value**: learn better distributions via:
  1. better and more **training data**,
  2. increased **model complexity**,
  3. by incorporating (more) **context**!

“Context”: how many of the preceding words to take into consideration when sampling the next?

$$w_i \sim p(w_i | \underbrace{w_1, w_2, \dots, w_{i-1}}_{\text{context}}; \theta)$$

**Why does it matter?** Complete the following sentence:  
“The animal didn't cross the street because it ...”

“The animal didn't cross the street because it is wet”

“The animal didn't cross the street because it is tired”

**Required context size:** 2 vs. 6 words!

Existing architectures incorporate context to various degrees: Markov chains, Recurrent Neural Networks (RNNs), Long-Short-Term Memory (LSTMs), ...

**Model evolution: larger contexts, less forgetting!**



# Transformers: Attention is All You Need

- Architecture that revolutionised natural language processing (Vaswani et al., 2017)

## Attention Is All You Need

Ashish Vaswani\*  
Google Brain  
avaswani@google.com

Noam Shazeer\*  
Google Brain  
noam@google.com

Niki Parmar\*  
Google Research  
nikip@google.com

Jakob Uszkoreit\*  
Google Research  
usz@google.com

Llion Jones\*  
Google Research  
llion@google.com

Aidan N. Gomez\* †  
University of Toronto  
aidan@cs.toronto.edu

Łukasz Kaiser\*  
Google Brain  
lukaszkaiser@google.com

Illia Polosukhin\* ‡  
illia.polosukhin@gmail.com

### Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

### 1 Introduction

Recurrent neural networks, long short-term memory [13] and gated recurrent [2] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and

\*Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

†Work performed while at Google Brain.

‡Work performed while at Google Research.



# Transformers: Attention is All You Need

- Architecture that revolutionised natural language processing (Vaswani et al., 2017)
- GPT = Generative Pre-trained Transformer

## Attention Is All You Need

Ashish Vaswani\*  
Google Brain  
avaswani@google.com

Noam Shazeer\*  
Google Brain  
noam@google.com

Niki Parmar\*  
Google Research  
nikip@google.com

Jakob Uszkoreit\*  
Google Research  
usz@google.com

Llion Jones\*  
Google Research  
llion@google.com

Aidan N. Gomez\* †  
University of Toronto  
aidan@cs.toronto.edu

Łukasz Kaiser\*  
Google Brain  
lukaszkaiser@google.com

Illia Polosukhin\* ‡  
illia.polosukhin@gmail.com

### Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

### 1 Introduction

Recurrent neural networks, long short-term memory [13] and gated recurrent [2] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and

\*Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

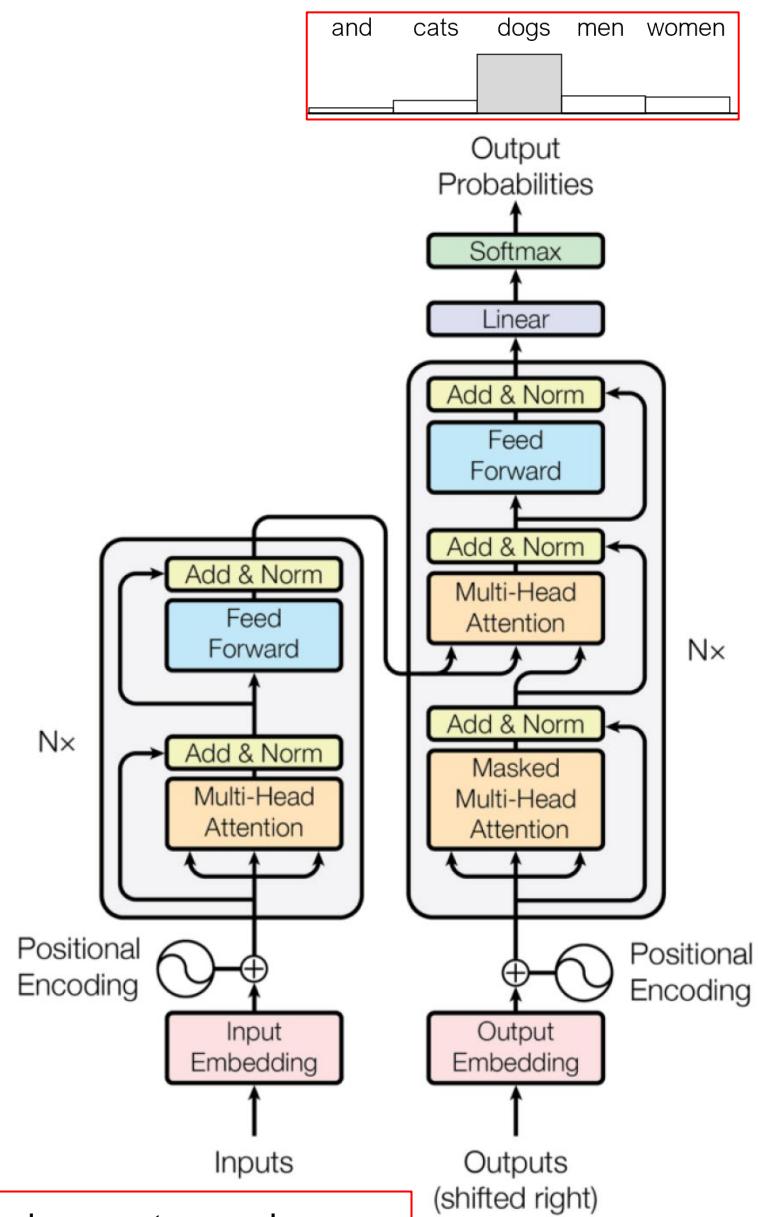
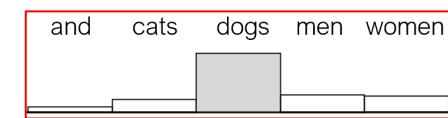
†Work performed while at Google Brain.

‡Work performed while at Google Research.



# Transformers: Attention is All You Need

- Architecture **revolutionised** natural language processing (Vaswani et al., 2017)
- GPT = Generative Pre-trained Transformer
- Most important **components**:
  - Feed forward networks (**blue**)
  - New element: self-attention (**orange**).



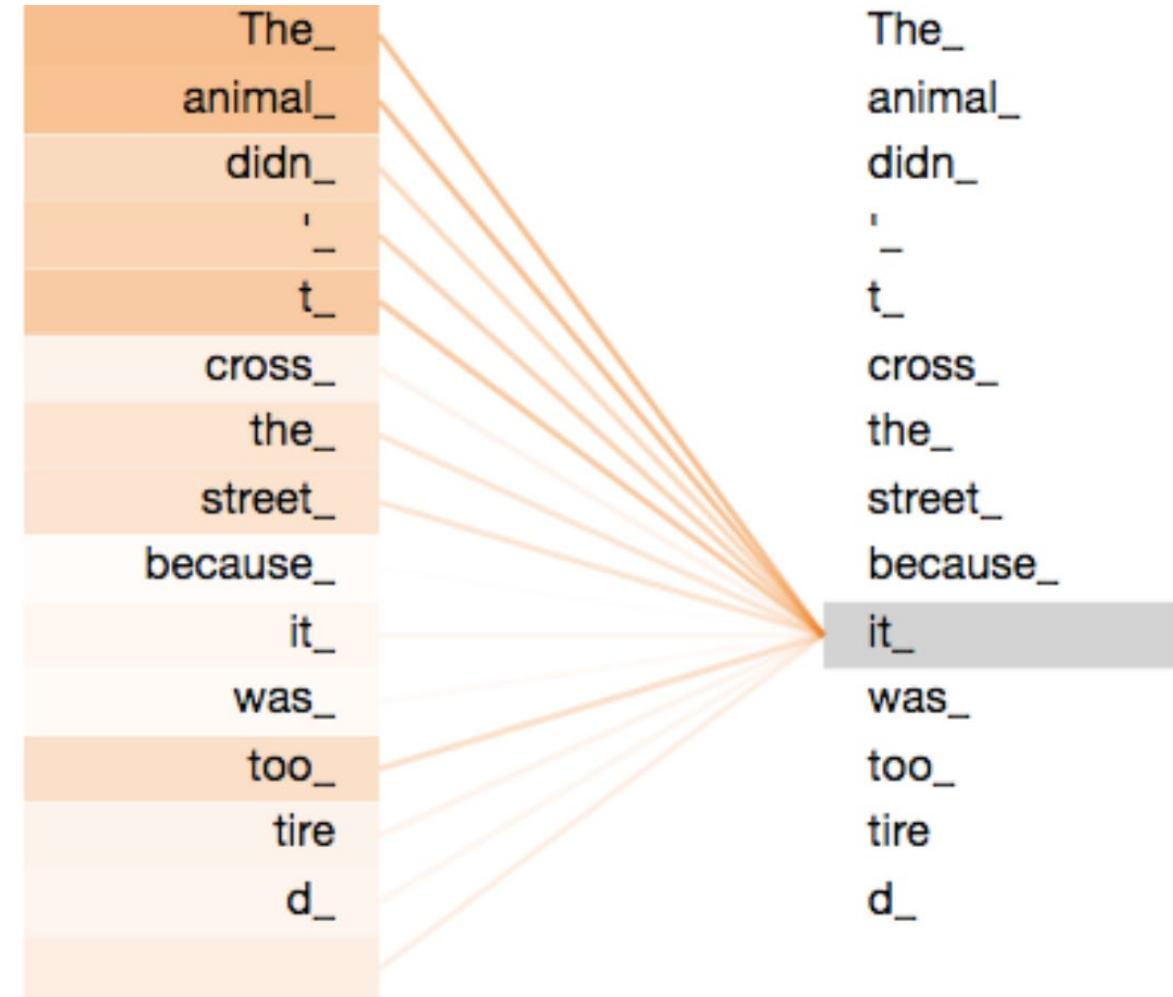
It's raining cats and \_\_\_\_\_



# Transformers: Attention is All You Need

- Architecture **revolutionised** natural language processing (Vaswani et al., 2017)
- GPT = Generative Pre-trained Transformer
- Most important **components**:
  - Feed forward networks (**blue**, -> lecture 2)
  - New element: self-attention (**orange**).
- “**Self-attention**” here: learns to look at other tokens in input to better predict word.

One attention head



Transformer self-attention  
(Illustration by Jay Alammar:

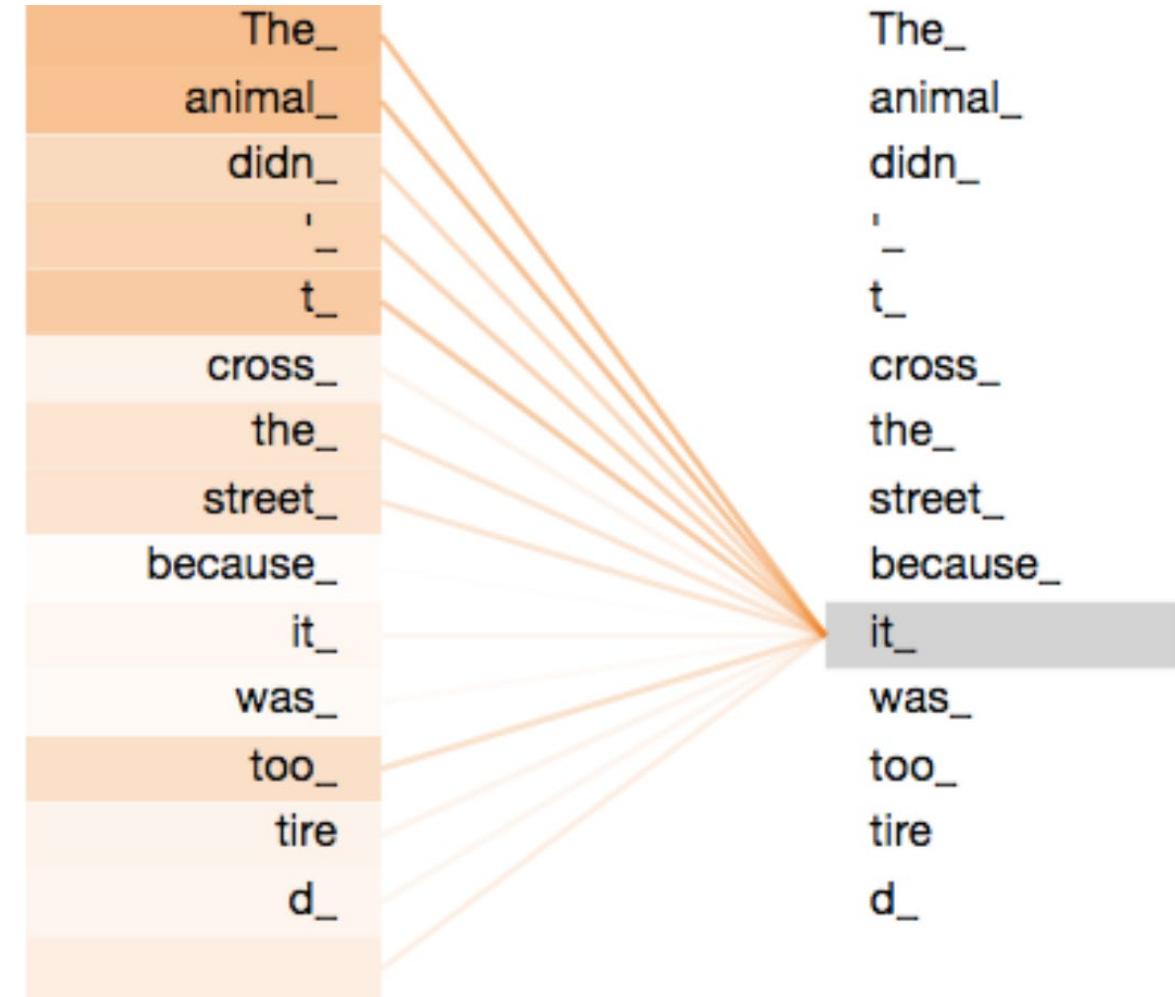
<https://jalammar.github.io/illustrated-transformer/> )



# Transformers: Attention is All You Need

- Architecture **revolutionised** natural language processing (Vaswani et al., 2017)
- GPT = Generative Pre-trained Transformer
- Most important **components**:
  - Feed forward networks (**blue**, -> lecture 2)
  - New element: self-attention (**orange**).
- “**Self-attention**” here: learns to look at other tokens in input to better predict word.
- Token = word piece, generation happens one token at a time

One attention head



Transformer self-attention

(Illustration by Jay Alammar:

<https://jalammar.github.io/illustrated-transformer/> )



Grace Lindsay  
@neurograce



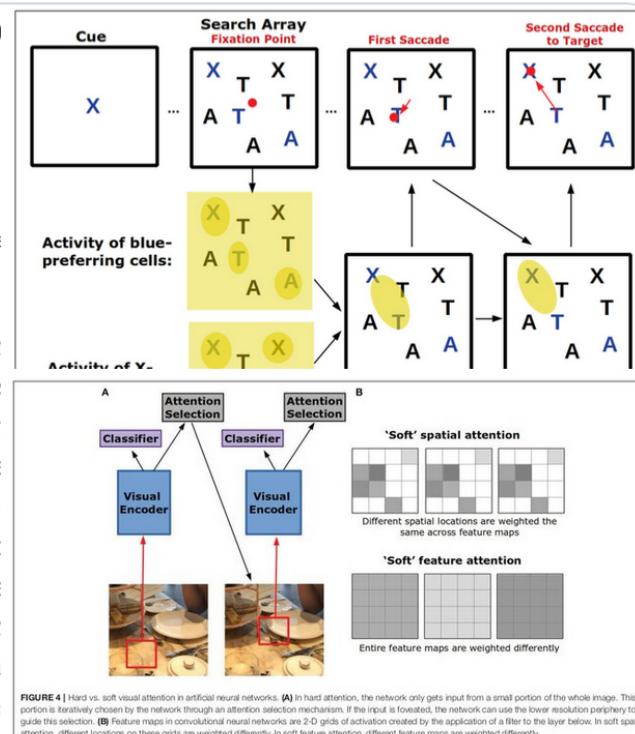
Have you heard the word "attention" thrown around in both neuroscience & machine learning? Have you wondered if/how its different uses relate to each other? My new review aims to summarize how this giant topic is studied & modeled across different domains! [frontiersin.org/articles/10.33...](https://frontiersin.org/articles/10.33...)

## Attention in Psychology, I and Machine Learning

Grace W. Lindsay\*

Gatsby Computational Neuroscience Unit, Sainsbury Wellcome Centre  
Kingdom

Attention is the important ability to flexibly control and has been studied in conjunction with many other topics including awareness, vigilance, saliency, executive control, recently been applied in several domains in machine learning, the study of biological attention and its use as a tool to is not always clear. This review starts by providing an conceptualized in the neuroscience and psychology literature cases of attention in machine learning, indicating their exist. Finally, the ways in which artificial attention can the production of complex and integrative systems is e





# Better Language Models and Their Implications



#### SYSTEM PROMPT (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.*

#### MODEL COMPLETION (MACHINE-WRITTEN, 10 TRIES)

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.

Pérez and the others then ventured further into the valley. “By the time we reached the top of one peak, the water looked blue, with some crystals on top,” said Pérez.

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, “We can see, for example, that they have a common ‘language,’ something like a dialect or dialectic.”

# GPT-3

- Much bigger: 175B params (GPT-2: 1.5B)
- 4k token context (GPT-2: 2k)
- New emergent property: few-shot learning (a.k.a. in-context learning)
- ChatGPT = GPT-3.5, fine-tuned with human feedback

Tom B. Brown\* Benjamin Mann\* Nick Ryder\* Melanie Subbiah\*  
Jared Kaplan† Prafulla Dhariwal Arvind Neelakantan Pranav Shyam  
Girish Sastry Amanda Askell Sandhini Agarwal Ariel Herbert-Voss  
Gretchen Krueger Tom Henighan Rewon Child Aditya Ramesh  
Daniel M. Ziegler Jeffrey Wu Clemens Winter  
Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray  
Benjamin Chess Jack Clark Christopher Berner  
Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

## Abstract

We demonstrate that scaling up language models greatly improves task-agnostic, few-shot performance, sometimes even becoming competitive with prior state-of-the-art fine-tuning approaches. Specifically, we train GPT-3, an autoregressive language model with 175 billion parameters, 10x more than any previous non-sparse language model, and test its performance in the few-shot setting. For all tasks, GPT-3 is applied without any gradient updates or fine-tuning, with tasks and few-shot demonstrations specified purely via text interaction with the model. GPT-3 achieves strong performance on many NLP datasets, including translation, question-answering, and cloze tasks. We also identify some datasets where GPT-3’s few-shot learning still struggles, as well as some datasets where GPT-3 faces methodological issues related to training on large web corpora.

## 1 Introduction

NLP has shifted from learning task-specific representations and designing task-specific architectures to using task-agnostic pre-training and task-agnostic architectures. This shift has led to substantial progress on many challenging NLP tasks such as reading comprehension, question answering, textual entailment, among others. Even though the architecture and initial representations are now task-agnostic, a final task-specific step remains: fine-tuning on a large dataset of examples to adapt a task agnostic model to perform a desired task.

Recent work [RWC<sup>+</sup>19] suggested this final step may not be necessary. [RWC<sup>+</sup>19] demonstrated that a single pretrained language model can be zero-shot transferred to perform standard NLP tasks

\*Equal contribution

<sup>†</sup>Johns Hopkins University, OpenAI

# Few-shot learning example

A "whatpu" is a small, furry animal native to Tanzania. An example of a sentence that uses the word whatpu is:

We were traveling in Africa and we saw these very cute whatpus.

---

To do a "farduddle" means to jump up and down really fast. An example of a sentence that uses the word farduddle is:

Exercise: continue the text

# Few-shot learning example

A "whatpu" is a small, furry animal native to Tanzania. An example of a sentence that uses the word whatpu is:

We were traveling in Africa and we saw these very cute whatpus.

---

To do a "farduddle" means to jump up and down really fast. An example of a sentence that uses the word farduddle is:

**One day when I was playing tag with my little sister, she got really excited and she started doing these crazy farduddles.**

Gray: Human text, Black: GPT-3 continuation

# Few-shot learning example

A "whatpu" is a small, furry animal native to Tanzania. An example of a sentence that uses the word whatpu is:

We were traveling in Africa and we saw these very cute whatpus.

---

To do a "farduddle" means to jump up and down really fast. An example of a sentence that uses the word farduddle is:

**One day when I was playing tag with my little sister, she got really excited and she started doing these crazy farduddles.**

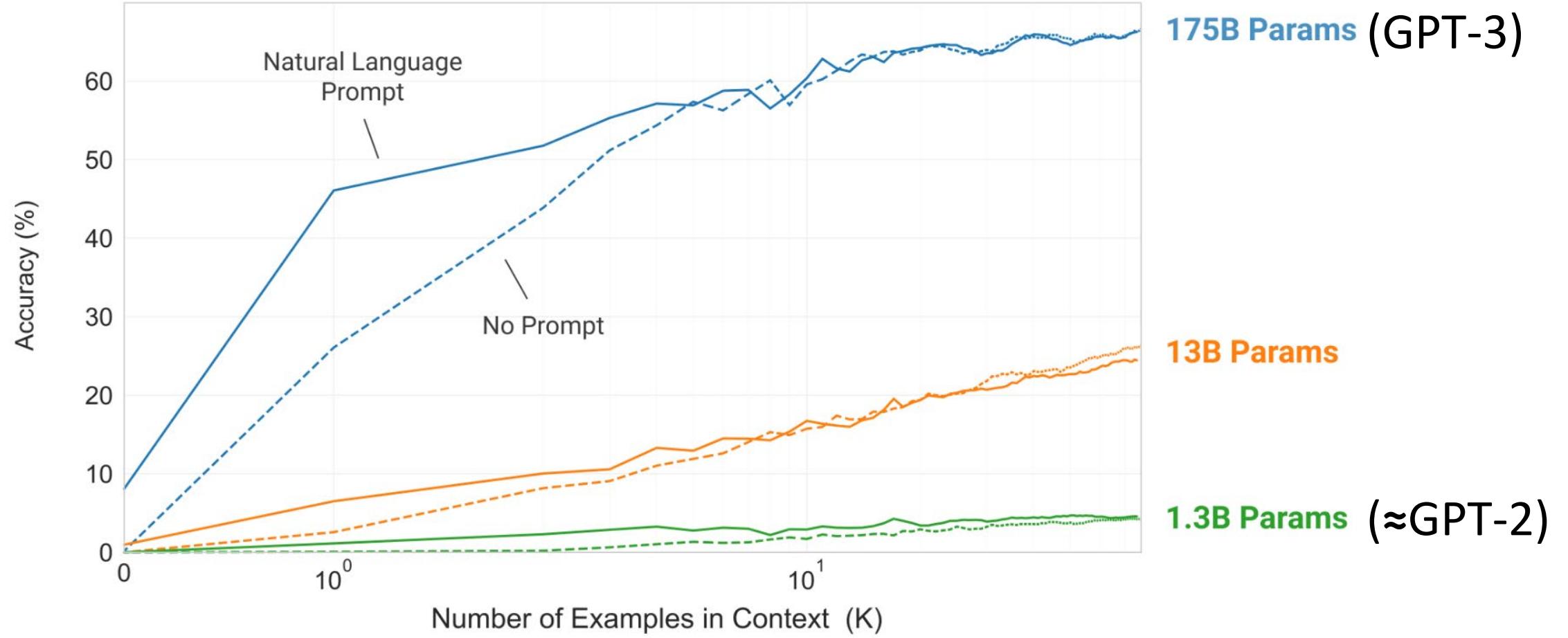
A "yalubalu" is a type of vegetable that looks like a big pumpkin. An example of a sentence that uses the word yalubalu is:

**I was on a trip to Africa and I tried this yalubalu vegetable that was grown in a garden there. It was delicious.**

Gray: Human text, Black: GPT-3 continuation



# Few-shot learning only emerges in very large models

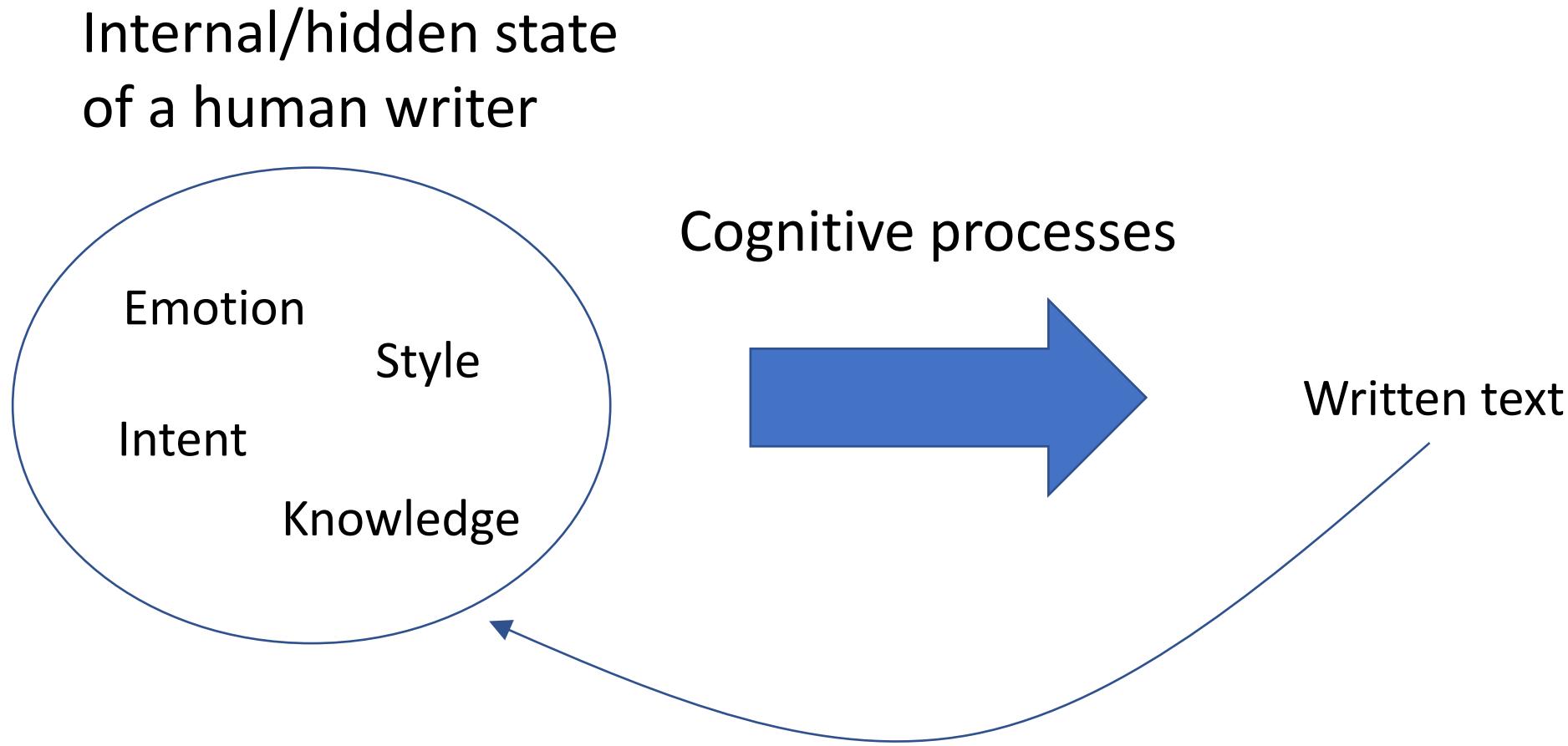


# What is happening?

- Strong evidence of truly novel generations, cannot be explained by simple memorization and recall!
- Principle: Too much training data to memorize => the model has to learn more general rules, logic, and manipulations
  - Transformers generate and manipulate abstract internal representations or “embeddings”
  - The embeddings have been shown to represent latent variables such as the emotion tone of the prompt
  - The training goal is to minimize the average next token prediction error over all the training data => emotional and stylistic plausibility over factual correctness

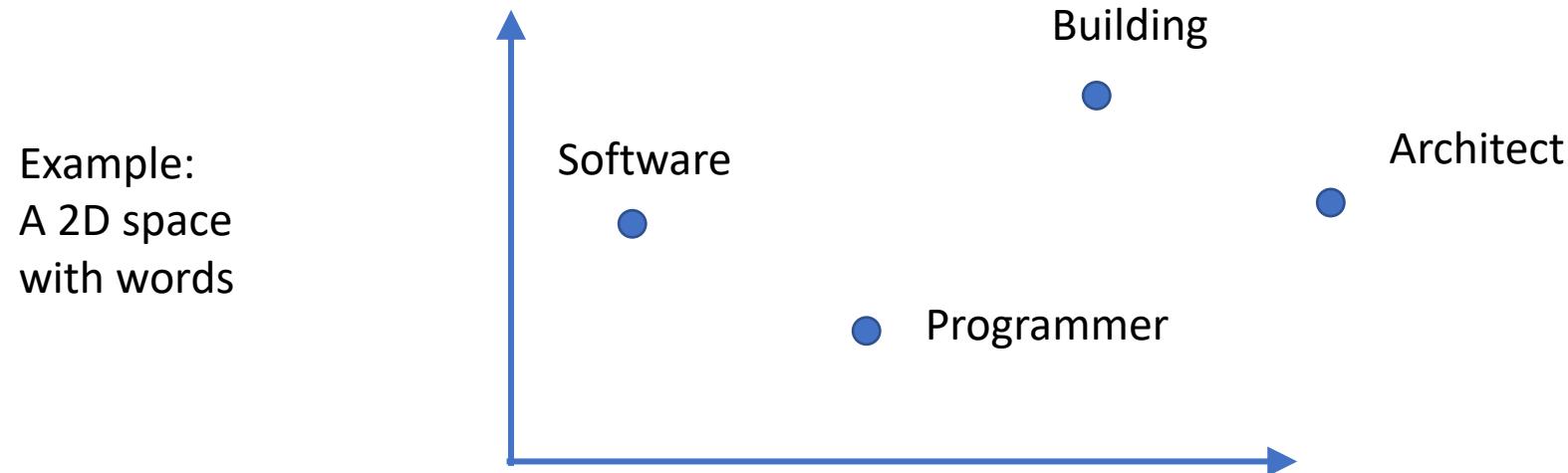


# The data-generating process that GPT-3 attempts to model



# Embeddings allow a language model to perform “semantic math”

- Embeddings can be thought as coordinates (vectors) in some high-dimensional space
- It has been observed many times that spatial relations (differences of coordinates) correspond to semantic relations
- King - Man = Queen – Woman (King is to Man as Queen is to Woman)
- => King - Man + Woman = Queen
- Software - Building + Architect = Programmer



# Language Models Can Generate Human-Like Self-Reports of Emotion

## EMERGENT WORLD REPRESENTATIONS: EXPLORING A SEQUENCE MODEL TRAINED ON A SYNTHETIC TASK

Kenneth Li\*  
Harvard University

Aspen K. Hopkins  
Massachusetts Institute of Technology

David Bau  
Northeastern University

Fernanda Viégas  
Harvard University

Hanspeter Pfister  
Harvard University

Martin Wattenberg  
Harvard University

### ABSTRACT

Language models show a surprising range of capabilities, but the source of their apparent competence is unclear. Do these networks just memorize a collection of surface statistics, or do they rely on internal representations of the process that generates the sequences they see? We investigate this question in a synthetic setting by applying a variant of the GPT model to the task of predicting legal moves in a simple board game, Othello. Although the network has no a priori knowledge of the game or its rules, we uncover evidence of an emergent nonlinear internal representation of the board state. Interventional experiments indicate this representation can be used to control the output of the network. By leveraging these intervention techniques, we produce “latent saliency maps” that help explain predictions.<sup>[1]</sup>

### 1 INTRODUCTION

Recent language models have shown an intriguing range of capabilities. Networks trained on a simple “next-word” prediction task are apparently capable of many other things, such as solving logic puzzles or writing basic code.<sup>[2]</sup> Yet how this type of performance emerges from sequence predictions remains a subject of current debate.

Some have suggested that training on a sequence modeling task is inherently limiting. The arguments range from philosophical (Bender & Koller, 2020) to mathematical (Merrill et al., 2021). A common theme is that seemingly good performance might result from memorizing “surface statistics,” i.e., a long list of correlations that do not reflect a causal model of the process generating the sequence. This issue is of practical concern, since relying on spurious correlations may lead to problems on out-of-distribution data (Bender et al., 2021; Floridi & Chiratti, 2020).

On the other hand, some tantalizing clues suggest language models may do more than collect spurious correlations, instead building interpretable *world models*—that is, understandable models of the process producing the sequences they are trained on. Recent evidence suggests language models can develop internal representations for very simple concepts, such as color, direction (Abdou et al., 2021); Patel & Pavlick (2022), or tracking boolean states during synthetic tasks (Li et al., 2021) (see Related Work (section 6) for more detail).

A promising approach to studying the emergence of world models is used by Toshniwal et al. (2021), which explores language models trained on chess move sequences. The idea is to analyze the behavior of a standard language modeling architecture in a well-understood, constrained setting. The paper finds that these models learn to predict legal chess moves with high accuracy. Furthermore, by analyzing predicted moves, the paper shows that the model appears to track the board state. The authors stop short, however, of exploring the form of any internal representations. Such an

Mikke Tavast  
mikke.tavast@aalto.fi  
Aalto University  
Espoo, Finland

Anton Kunnari  
anton.kunnari@helsinki.fi  
University of Helsinki  
Helsinki, Finland

Perttu Hämäläinen  
perttu.hamalainen@aalto.fi  
Aalto University  
Espoo, Finland

### ABSTRACT

Computational interaction and user modeling is presently limited in the domain of emotions. We investigate a potential new approach to computational modeling of emotional response behavior, by using modern neural language models to generate synthetic self-report data, and evaluating the human-likeness of the results. More specifically, we generate responses to the PANAS questionnaire with four different variants of the recent GPT-3 model. Based on both data visualizations and multiple quantitative metrics, the human-likeness of the responses increases with model size, with the largest Davinci model variant generating the most human-like data.

### CCS CONCEPTS

• Human-centered computing → Empirical studies in HCI.

### KEYWORDS

Language models, GPT-3, PANAS, emotion, affect

### ACM Reference Format:

Mikke Tavast, Anton Kunnari, and Perttu Hämäläinen. 2022. Language Models Can Generate Human-Like Self-Reports of Emotion. In *27th International Conference on Intelligent User Interfaces (IUI '22 Companion), March 22–25, 2022, Helsinki, Finland*. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3490100.3516464>

### 1 INTRODUCTION

Computational user modeling is advancing rapidly and can produce human-like predictions of user behavior and experience [7, 12, 15]. However, this is presently limited in the domain of emotions. Here, our aim is to investigate a potential new approach by using modern neural language models to generate synthetic self-report data about affect. We employ the recent GPT-3 model to generate responses to Positive and Negative Affect Schedule (PANAS), a widely used questionnaire designed to measure positive and negative affect [21]. GPT-3 is a large neural language model trained to predict the next word in a sequence [5]. The trained model takes as its input a piece of text—a prompt—and generates a continuation of desired length.

PANAS consists of 10 positive affect and 10 negative affect items: emotional words such as excited, proud, guilty, and upset. The task is to rate how much one has felt these states during a specified time period. In the original validation study [21] each of the 20 items was

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyright for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

IUI '22 Companion, March 22–25, 2022, Helsinki, Finland  
© 2022 Copyright held by the owner/author(s).  
ACM ISBN 978-1-4503-9145-0/22/03.  
<https://doi.org/10.1145/3490100.3516464>

shown to strongly load on only one of the two largely uncorrelated factors, establishing the 10 item positive and negative scales. In HCI research, PANAS has been used to, for example, operationalize positive user experience [16, 19].

We are not aware of previous studies trying to directly predict psychological scale responses using language models. Other natural language processing (NLP) methods have been used, for example, to predict scale responses [3] and affective ratings of words [20], and a recent preprint compared transformer models to human data in a benchmark NLP task [13]. Other work has, for example, used transformer model representations to predict brain imaging data [6, 10, 18].

### 2 METHODS

We generated 150 completions to the 20 items of the PANAS scale with four GPT-3 models of increasing size: Ada, Babbage, Curie, and Davinci<sup>[1]</sup>. The responses to the PANAS items were generated with a prompt that described a research interview (see Table 1). To increase the variability in the prompts, each of the 150 interviews had a unique “participant” (varying age, gender, job, and hobby), a short description of “who” is being interviewed.

After the participant information, the prompt contained three example responses to questions answered in a Likert scale. To minimize bias in the training examples, the “participant” gave the answers 1, 3, and 5 once in each interview. The same three examples were used throughout the experiment, but their order was randomized for every interview.

The 20 PANAS items were queried one-by-one, in random order. Once the model generated a completion to a item, everything strating from the first appearance of the string “Researcher:” or the first newline character was cut from the completion. If the completion generated something outside desired responses (1,2,3,4 or 5), the whole interview for the participant was run again. The number of errors in proportion to all of the completions are presented in Section 2.1. After trimming the completion, the item-response pair was saved and included in the prompt for the next item. The order of the previous item-response pairs in the prompt was randomized for every new completion. For response generation, we used a maximum response length of 64 tokens and the default OpenAI parameters (temperature=0.7, top\_p=1.0,frequency\_penalty=0,presence\_penalty=0, best\_of=1).

We downloaded a human PANAS reference dataset from Open Science Framework (osf.io). The dataset was originally collected for a study by Anvari and Lakens concerning methods of determining the smallest effect size of interest [2]. Here we use the datapoints designated as T1 in the dataset (<https://osf.io/3a5up/>, [1]).

<sup>1</sup>Data and analysis code: <https://github.com/mtavast/gpt-panas>

\*Correspondence to ke\_li@g.harvard.edu

<sup>2</sup>Codes at [https://github.com/likenneth/othello\\_world](https://github.com/likenneth/othello_world)

<sup>2</sup>See Srivastava et al. (2022) for an encyclopedic list of examples.



# Recent progress

- Enabling larger context with less computation. Flash Attention allows about 4x larger context at the same compute cost. Relative Positional Encoding allows pretraining with shorter context and finetuning with longer context—a considerable cost saving
- Alternatives to Transformers. State Space Models are showing promise
- Controlling of the generated content (style, emotion, high-level story structure, character goals...). For recent results, see PPLM:  
<https://towardsdatascience.com/controlling-text-generation-from-language-models-6334935e80cf>, PlotMachines:  
<https://arxiv.org/pdf/2004.14967.pdf>

# Long context is not everything



**AaltoMediaAI** @aaltomediaai · Nov 23, 2023

Nice visualization of how the position of a fact in the prompt can affect LLM performance



**Greg Kamradt** @GregKamradt · Nov 21, 2023

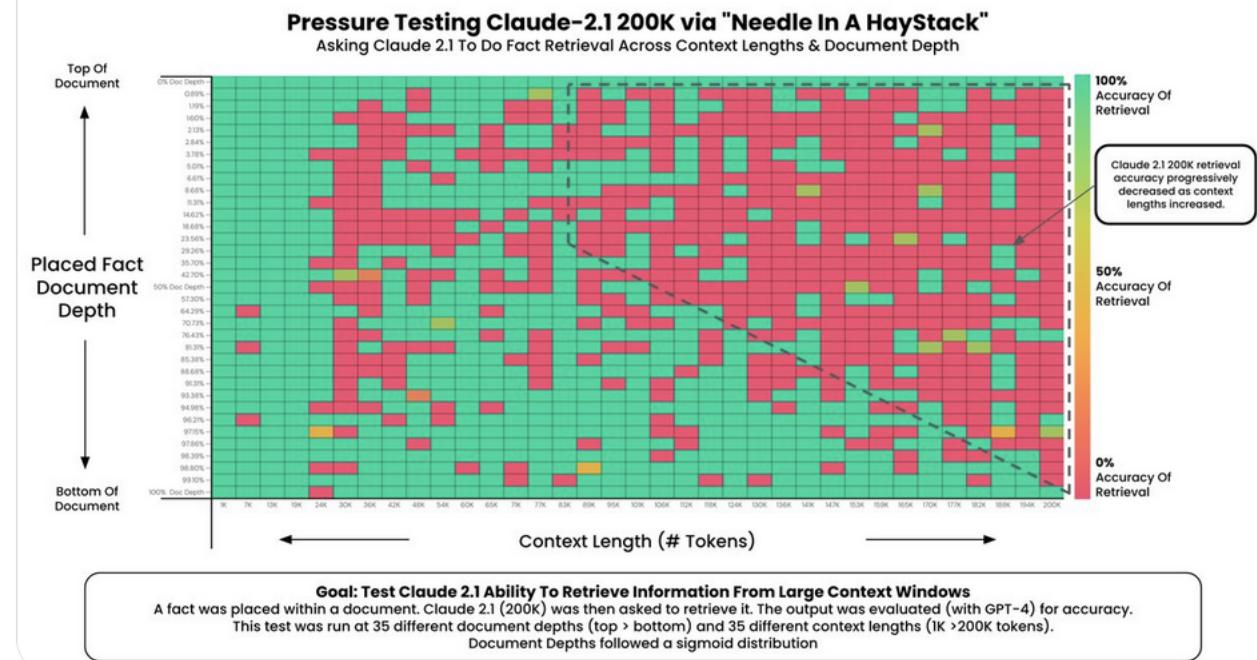
Claude 2.1 (200K Tokens) - Pressure Testing Long Context Recall

We all love increasing context lengths - but what's performance like?

Anthropic reached out with early access to Claude 2.1 so I repeated th...

[Show more](#)

[Show this thread](#)





AaltoMediaAI @aaltomediaai · Dec 4, 2023

...

This looks super promising. Small models can do similar associative recall as the "induction heads" of transformers, and bigger models beat transformers twice as large



Albert Gu @\_albertgu · Dec 4, 2023

Quadratic attention has been indispensable for information-dense modalities such as language... until now.

Announcing Mamba: a new SSM arch. that has linear-time scaling, ultra long context, and most importantly--outperforms Transformers ...

[Show more](#)



<https://x.com/aaltomediaai/status/1731759593930281027?s=20>

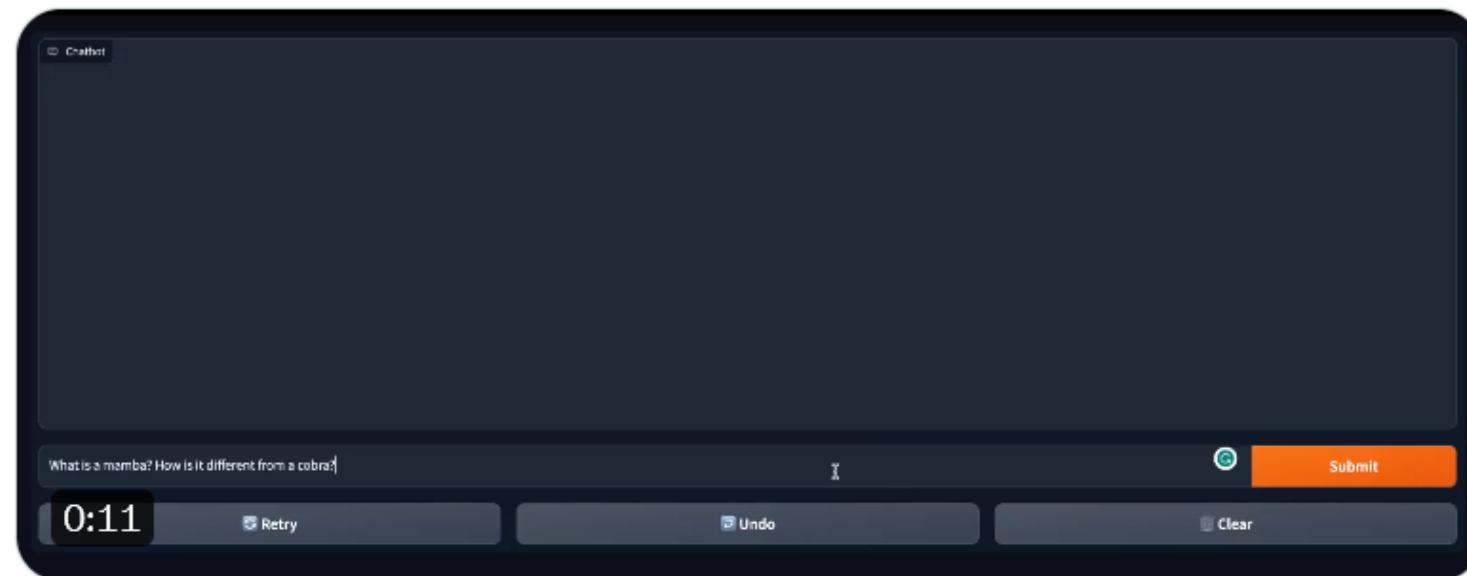


**Xiuyu Li** @xiuyu\_l · Dec 29, 2023

...

Mamba is really exciting, but its potential remains untapped due to a lack of instruction-tuning and alignment. Inspired by [@MatternJustus](#)'s Mamba-Chat, I trained Mamba-3B-Zephyr over the weekend and got some interesting findings 1/5

Colab Demo: [colab.research.google.com/drive/1SEwD1Cx...](https://colab.research.google.com/drive/1SEwD1Cx...)



[https://x.com/xiuyu\\_l/status/1740806425843294606?s=20](https://x.com/xiuyu_l/status/1740806425843294606?s=20)



**Shital Shah** @sytelus · Dec 13, 2023

...

Mistral-7B is cool but you know what's cooler? A more powerful model in just 1/3rd of the size!

Welcome to Phi-2.

This is something our team at Microsoft Research had been tirelessly working on and now we have more numbers comparing with Llama-7B, 13B, 70B and Gemini Nano.

<https://x.com/sytelus/status/1734881560271454525?s=20>

# Examples

# GPT-3 CREATIVE FICTION

Creative writing by OpenAI's GPT-3 model, demonstrating poetry, dialogue, puns, literary parodies, and storytelling.  
Plus advice on effective GPT-3 prompt programming & avoiding common errors.

[GPT fiction](#), [GPT poetry](#), [AI scaling](#), [humor](#), [philosophy/mind](#)

2020-06-19–[2022-02-10](#) · finished · [certainty: likely](#) · [importance: 8](#) · [backlinks](#) · [similar](#) · [bibliography](#)

One of the best resources,  
with many examples

<https://gwern.net/gpt-3>

## 1 What Benchmarks Miss: Demos

## 2 GPT-3 Implications

## 3 Quality

## 4 Prompts As Programming

### 4.1 Finetuning

### 4.2 Playground

### 4.3 Effective Prompt Programming

## 5 Weaknesses

### 5.1 Small Context Window

### 5.2 Repetition/Divergence Sampling

### 5.3 BPEs

## 6 Copyright

## 7 Format

## 8 Nonfiction

## 9 Dialogue

### 9.1 Shoggoth-Cat

### 9.2 Pun Explanations

### 9.3 Turing Dialogue

I continue my AI poetry generation experiments with OpenAI's 2020 GPT-3, which is 116× larger, and much more powerful, than the 2019 [GPT-2](#). GPT-3, however, is not merely a quantitative tweak yielding "GPT-2 but better"—it is qualitatively different, exhibiting eerie runtime learning capabilities allowing even the raw model, with zero finetuning, to "meta-learn" many textual tasks purely by example or instruction. One does not train or program GPT-3 in a normal way, but one engages in dialogue and writes prompts to teach GPT-3 what one wants.

Experimenting through the [OpenAI Beta API](#) in June 2020, I find that GPT-3 does not just match my [finetuned GPT-2-1.5b-poetry](#) for poem-writing quality, but exceeds it, while being versatile in handling [poetry](#), [Tom Swift puns](#), science fiction, dialogue like Turing's [Turing-test dialogue](#), [literary style parodies](#)... As the *pièce de résistance*, I recreate Stanislaw Lem's *Cyberiad*'s "[Trurl's Electronic Bard](#)" poetry using GPT-3. (Along the way, I document instances of how the BPE text encoding [unnecessarily damages](#) GPT-3's performance on a variety of tasks, how to best elicit the highest-quality responses, common errors people make in using GPT-3, and test out GPT-3's improvements

# Poetry inspired by Stanislaw Lem's Cyberiad

**Prompt: ... Klapaucius thought, and thought some more. Finally he nodded and said:“Very well. Let’s have a love poem, lyrical, pastoral, and expressed in the language of pure mathematics. Tensor algebra mainly, with a little topology and higher calculus, if need be. But with feeling, you understand, and in the cybernetic spirit.”**

**“Love and tensor algebra? Have you taken leave of your senses?”  
Trurl began, but stopped, for his electronic bard was already declaiming:**

“

<https://www.gwern.net/GPT-3#love-and-tensor-algebra>

" $P = A + Bt + C*D^2 + E*F*Gt + H*t + It + Jt$   
+  $K*t + L.M.N^2t + O*P*Qt + Rt + S*Rt^3 + T^2$ ,

Where  $t$  is a transcendent number, not necessarily greater than 1, with one arm, a leg or two, and ideally of the color of rubies."

"But that's dreadful!" cried Klapaucius. "Where's the love?"

"As near as I can tell, it's right there in the middle,

" $T = R + S^2t^3 + St + T^2$ ," said Trurl.

and then there's some stuff about golden apples and violets and whatnot, but I couldn't quite make it out. To me it sounds more like an ad for some sort of machine part."

"Two bodies, sine waves entwined,  
Sighing in the soft of the night,  
Fluctuate, undulate, and flow,  
Escape from a universe  
Of bondage and slavery,  
To find together their fate,  
Far from the madding crowds..."

"It's too much!" shouted Trurl. "Too much, I tell you! That's not a love poem, that's a pure product of mathematics! How dare you take me for a fool? It's got nothing to do with love!"

"It's the poet's prerogative to choose the most beautiful and exalted metaphors," said Klapaucius.

"Most beautiful? You call this most beautiful? But I just can't bear it! It's driving me mad!"

"Beauty and brawn, my beloved,

Bake biscuits and bore binary.

Alone among the antelopes

All my love is for you..."

"Not a bad start," said Klapaucius. "But go on, go on! More feeling! Rubato, leitmotiv—have you no soul, man?"

"Let us forget the lore of neurons!" said the machine, in a voice so deep and resonant it made you think a chorus was humming in harmony behind it.

"What?" said Trurl, outraged. "What's the meaning of this?"

"No meaning at all!" snapped the poet. "I don't have to mince my words to please your bourgeois notions of art! I don't even know what either of you is talking about!"

But you wanted a love poem—here it is:

"Axiomatic algorithm angst,

Angels and amoebae,

Ants and acanthoid cytoplasm;

All of it must somehow fit..."

# These Lyrics Do Not Exist

Lyrics generated using Artificial Intelligence

This website generates completely original lyrics for various topics, uses state of the art AI to generate an original chorus and original verses  
You can choose the AI songwriter lyrics topic, lyrics genre and lyric mood

## Generate Your Own Lyrics

Type a song topic word and then press the "Generate My Lyrics" button

example Song Topics: [love](#), [party](#), [dance](#), [happy](#), or human names such as [peter](#), [jennifer](#), [bella](#), [alex](#),  
or you can even generate songs by emoji 😊, 😃, ❤️, 💋, 🎉, 🍳, 🍺, 🎵, 🎤, 🎧 or etc...

Song Topic:

Lyrics Genre:



Lyric Mood:



Generate My Lyrics



[-] [itsmybirthday19](#) [S] 76 points 6 months ago

Input: 🍺 (beer emoji)

Output:

Verse 1

Dad I had to have a 🍺  
To help me get through another day  
When I think of the lady I love  
Time to drink that 🍺

Pre-Chorus

They were drinking 🍺 outside of my father's house  
Let's take our time with this 🍺

Chorus

Cold 🍺 on the floor  
And I can see you now  
To change those words you feed me with lies  
I chug 🍺 🍺 🍺 🍺 🍺

Verse 2

I've come here again to drink my 🍺 again.  
I've come here again to drink my 🍺 again.  
Oh I'm goin' right on! don't you see?  
Just 'cause I look just fine

Pre-Chorus

One more 🍺 and I'll be in no state to do the things I think I need to  
A place to shake my 🍺



[–] **localkhaleesi** 11 points 6 months ago

Input: fuck

#### Verse 1

What the f\*\*\*k what the f\*\*\*k what the f\*\*\*k?  
The same they claim when problems pop  
Talk mess up talk mess up talk mess up  
The same they claim when problems pop

#### Pre-Chorus

I see the same thing all over the f\*\*\*king world every single day  
I see the same thing in my women f\*\*\*k the greed the fake

#### Chorus

Get the f\*\*\*k outta here woah I'm on the up  
See I don't know about you f\*\*\*k boys  
Tell me you're not scared f\*\*\*k that  
I really don't give a f\*\*\*k about you girl

#### Verse 2

Turn it up and f\*\*\*k the sun  
Cause I know that you count on me  
And that's where you belong  
Make your lady wanna f\*\*\*k me

#### Pre-Chorus

Don't f\*\*\*k with me no more oh I'm so cool  
Don't f\*\*\*k with me no more

# AI Dungeon:

A text adventure game where GPT generates the story and the player can say/do anything

You open the book and begin to read:

The Essense is within us all. We use it every day for simple things like language and thought, but true masters can manipulate this essence to become gods among men. To become a master, you must first hone basic abilities such as mind reading, influencing the weak minded, and simple spells. Once you become experienced, you can try more challenging feats of mental prowess. Remember, students, the strong-minded survive, the weak-minded perish.

> You learn how to read minds from the book.

You open the book again and begin to read:

Thoughts are the building blocks of our world. With practice, you can read the basic thoughts of someone. To read the complex thoughts of someone, you must be within five feet of that person. The thoughts of someone are a jumble of images, sounds and feelings. Listen to these thoughts and learn.

> You practice by trying to read Eliza's thoughts.

You walk over to Eliza. She looks at you with a raised eyebrow.

"Yes?" She asks.

"Can I try something?" You ask.

She shrugs. "I suppose."

You look into her eyes. You try to focus on her thoughts. You hear a series of disjointed words and images: father, home, needle, sleep, tired.

"Well?" She asks.

"You... miss your father," You say, "You want to go home, you think about your home a lot, you had an accident with a needle when you were little, and you're very tired."

"That's correct," She nods.



You are Zorro, a wizard living in the kingdom of Larion. You have a staff and a spellbook. You finish your long journey and finally arrive at the ruin you've been looking for. You look around and see that it's just as you remember. The ruins are all that remain of the once great city of Larion. The only thing that looks different is the fact that there are no more buildings to be seen

> invent the internet

You think about how you can use this opportunity to make the world a better place. You decide to go back to the ruins and try to figure out what happened to the people who lived here before you arrived.

The next day you return to the ruins with your new found knowledge

> invent the internet

You begin by creating a computer program which will allow you to communicate with anyone on the net. It takes some time but eventually you get it working. You then start sending messages to other wizards who live in the area



Mario Klingemann ✅

@quasimondo

...

These-movies-do-not-exist. Or do they? #thread #gpt3

A teenage girl and her father travel to California to stay with her dad's relatives, but they are actually horrific aliens in disguise.

A man is taught to fight zombies by a martial arts master in the Arizona desert.

11:06 PM · Jul 15, 2020 · Twitter Web App

18 Retweets 11 Quote Tweets 74 Likes



Mario Klingemann ✅ @quasimondo · Jul 15

...

Replying to @quasimondo

A man from the past is sent into the future to help his friend, but ends up leaving the future in ruins.



Mario Klingemann ✅ @quasimondo · Jul 15

...

A scientist invents a drug that brings out a person's innate ability to deal with any crisis.



Mario Klingemann ✅ @quasimondo · Jul 15

...

A reclusive writer creates a beautiful fantasy world and an intricately designed puzzle box to hide the link between his world and the real one.



Mario Klingemann ✅ @quasimondo · Jul 15

...

An alcoholic former sports star becomes a children's book author and illustrator who uses his stories to teach valuable life lessons.



Mario Klingemann ✅ @quasimondo · Jul 15

...

A family's suburban home is invaded by small, carnivorous creatures who kill people by burrowing into their brains and eating their memories.



Mario Klingemann ✅ @quasimondo · Jul 15

...

It's the 18th century and Elizabeth, the city's consummate Golden Girl, has just fallen mysteriously ill, so she and her four best girlfriends send out an invitation to their ex-boyfriends to help them reminisce about the past.



Mario Klingemann ✅ @quasimondo · Jul 15

...

Two reporters investigate a police corruption story, which leads them to discover a vampire conspiracy and attempt to unravel a federal cover-up.



Mario Klingemann ✅ @quasimondo · Jul 15

...

An elderly, wheelchair-bound man lives in a nursing home, but he has some unexplained visitors every night that come to see him.



<https://twitter.com/quasimondo/status/1283493077324365826>



Mario Klingemann

@quasimondo

...

In interviews I always get asked the "Who is the artist - you or the machine?"-question. And I usually say that as long as you have to ask me that question the answer is clear.

Well, not only does [#gpt3](#) give pretty interesting responses, it also asks better questions.

I: So who is the artist then - you or the machine?

MK: I believe there is a relationship that arises out of this question. The machine seems to have a body, a shell of some sort and is potentially capable of authorship. The body of the machine is the software that it is based on, the machine itself. The artist, in turn, might be the one who installs the software that creates art.

I: Isn't it a bit too simplistic to refer to the machine as 'artificial intelligence' and to say it is autonomous? We know that no computer has ever been autonomously creative! They can create, but they cannot be creative in the sense that humans are. However, 'creativity' is defined as an activity, not a human attribute. Creativity has to do with the making of a thing. An artificial intelligence does not make things, it only processes data, correct?

MK: Creativity is traditionally seen as the ability to create something new or unique. In my opinion this does not capture the concept of creativity at all, it only implies some sort of individuality. Individuality is a concept that is traditionally attributed to a human. I think it is safe to assume that every 'thing' that has ever been created has been unique - or at least a unique combination of previously existing elements. One can question the claim that a thing is created by the person who assembled it, but that does not challenge the uniqueness or uniqueness of a specific combination of elements.

My favourite definition of creativity is the idea that creativity is the art of creating something that is aesthetically pleasing or beautiful. If you define 'creativity' as an act of creation you are implicitly assuming a human behind it, or a human mind as the source of creativity. From that perspective the word 'artificial' does not apply.

But my second definition is more fitting to the nature of machine creativity: creativity is the ability to create an aesthetic experience. With the current state of technology we have to assume that aesthetic experiences are created on the fly, dynamically. The machine has to simulate them. The aesthetically pleasing thing cannot be part of a machine's memory. It has to be a generative event that has a unique course.

<https://twitter.com/quasimondo/status/1283353812581965824>



# Movie summaries using emojis

I decided to give it some movies described in emojis and see what happened.  
First I provided this context:

Back to Future: 🤖👻🚗🕒

....

Batman: 🤖🦇

....

Transformers: 🚗🤖

When I asked it to do the same for Star Wars, this was the response:

Star Wars: 🚀🏰🚀

I think that's a pretty good response! Here are some of the others it created:

Zootopia: 🐈🐾🐻🦋🐰

Wonder Woman: 💁‍♀️💁‍♀️💁‍♀️💁‍♀️💁‍♀️

The Godfather: 🤖💁‍♀️💁‍♀️🕵️‍♂️🕵️‍♂️💥

Star Trek: ✨🚀

Planet of the Apes: 🙄🐵

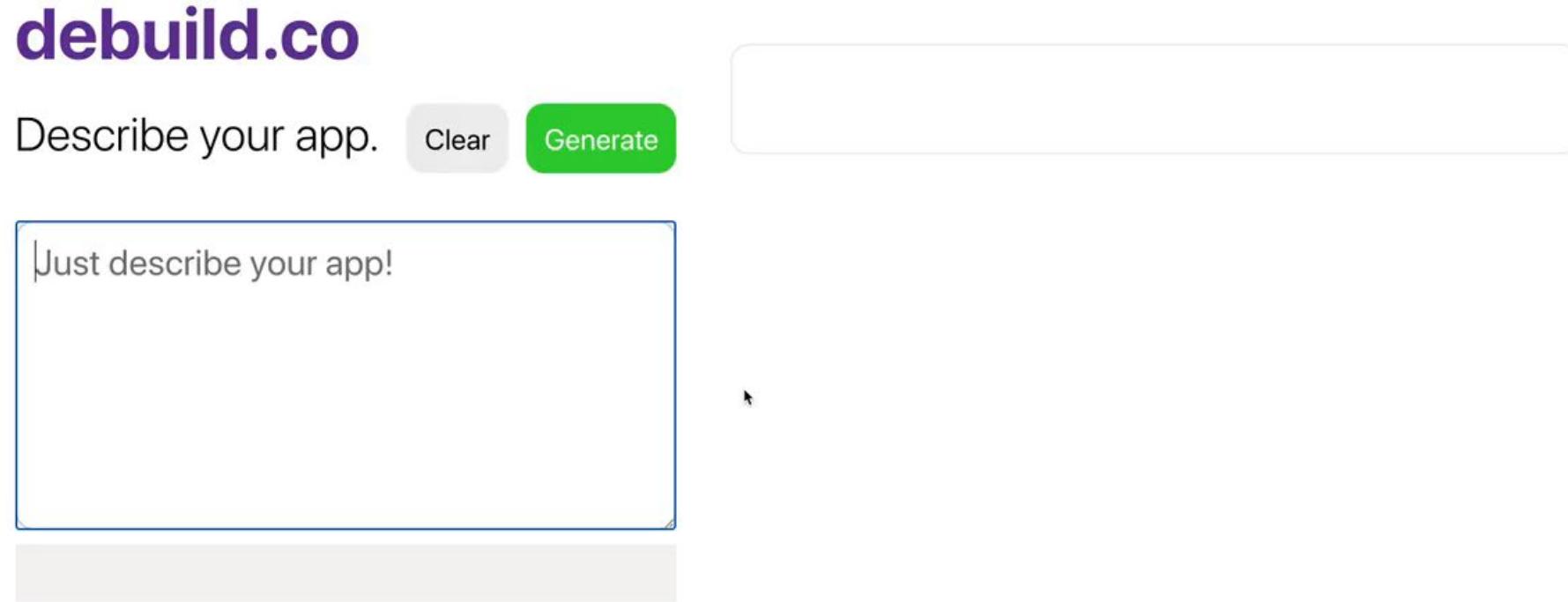
Game of Thrones: 👑🐺🏰

Jurassic Park: 🤖🦕🦖🦕

Castlevania: 🧛‍♂️🧛‍♂️🧛‍♂️🧛‍♂️🧛‍♂️🧛‍♂️🧛‍♂️🧛‍♂️🧛‍♂️

The Matrix: 🤖🤔

# Generating software code





```
1 def is_palindrome(s):
2     """Check whether a string is a palindrome"""
3     return s == s[::-1]
4
5 def long_palindrome_indices(l):
6     """Return list indices for elements that are palindromes and at least 7 characters"""
7     return [i for i, s in enumerate(l) if is_palindrome(s) and len(s) >= 7]
8
9 @dataclass
10 class Item:
11     name: str
12     price: float
13
14 @dataclass
15 class Order:
16     id: int
17     items: List[Item]
18
19     def compute_total_price(self, palindrome_discount=0.2):
20         """
21             Compute the total price and return it.
22             Apply a discount to items whose names are palindromes.
23         """
```

<https://aiweirdness.com/post/190569291992/ai-recipes-are-bad-and-a-proposal-for-making-them>



Pictured above is an abomination in the making, a lesson in why humans should never trust what a neural net says just because it's based on math. It's a neural net generated brownie recipe called [Chocolate Baked and Serves](#), and its distinguishing feature is the CUP OF HORSERADISH it contains. It was so bad that my eyes watered as I removed it from the oven.

Table 1. An example of a prompt that frames coding as a text continuation task for GPT-3. The prompt comprises a short instruction text and manually coded few-shot examples separated by "###". Note that the prompt does not limit the model to the codes given as examples, and according to our experiments, it can generalize based on the examples and generate new codes in the same style when needed.

**The following presents a qualitative coding of answers from a video game research study. The answers explain why a participant experienced a game as art. The codes summarize the given reasons as compactly as possible. If an answer lists multiple reasons, the corresponding codes are separated by semicolons.**

###

**Answer:** The questions it raised and the highly emotional connection that emerged between me and the game, the experience.

**Codes:** raising questions; emotional connection

###

**Answer:** For a game experience to feel like a work of art to me, it would usually be an immersive experience that creates a real emotional response. Since games accomplish this through a combination of illustration, animation, sound, music, storytelling elements all together, I would consider these types of experiences art.

**Codes:** immersive experience; emotional response

###

**Answer:** The fact that each asset was hand drawn in such a unique style.

**Codes:** unique visual style

###

**Answer:** <each new answer to code inserted here>

**Codes:**



<https://fable-studio.com/behind-the-scenes/ai-generation>





## CONTEXTS:

- The following is a conversation between Lucy and Guest over text messages.
- In Lucy's world, which is different from the Guest's, the date is 1988, while for the Guest it is still 2020.
- The Guest reaches out to Lucy over chat message and Lucy responds in a playful way, asking if the Guest is a foozle and comes in peace.

...

## LUCY:

- Little Girl.
- Active Imagination.
- Age 8.
- Lives with Brother, Mom, and Dad.

...

## GUEST:

- A curious and friendly person who was just introduced to Lucy.



GPT-generated Seinfeld-style sitcom,  
as a neverending Twitch stream.

<https://m.twitch.tv/watchmeforever>



# Transformers in other domains

- Mathematics and theorem proving (e.g. <https://arxiv.org/pdf/1912.01412.pdf>,  
<https://arxiv.org/pdf/2002.05867.pdf>)
- Biology (e.g. <https://www.biorxiv.org/content/10.1101/622803v1.full.pdf>,  
<https://arxiv.org/pdf/2004.03497.pdf>)

# Generating Piano Music with Transformer

Sep 16, 2019

Ian Simon  iansimon  iansimon

Cheng-Zhi Anna Huang  czhuang  huangcza

Jesse Engel  jesseengel  jesseengel

Curtis Hawthorne  cghawthorne  fjord41

Monica Dinculescu  notwaldorf  notwaldorf

Previously, we introduced **Music Transformer**, an autoregressive model capable of generating expressive piano performances with long-term structure. We are now releasing [an interactive Colab notebook](#) so that you can control such a model in a few different ways, or just generate new performances from scratch.

<https://magenta.tensorflow.org/piano-transformer>

[https://colab.research.google.com/notebooks/magenta/piano\\_transformer/piano\\_transformer.ipynb](https://colab.research.google.com/notebooks/magenta/piano_transformer/piano_transformer.ipynb)

# Dall-E: A GPT-3 that uses both words and image patches as the vocabulary

Text prompt

an illustration of a baby daikon radish in a tutu walking a dog

AI-generated images



<https://openai.com/blog/dall-e/>

# OpenAI Jukebox: A transformer for granular audio synthesis (<https://jukebox.openai.com/>)

MODEL	COLLECTION	GENRE	ARTIST	TEMP
5b_lyrics	Re-renditions	Contemporary C...	Steven Curtis Chapman	0.98
5b_lyrics	Re-renditions	New Wave	The Nits	0.98
5b_lyrics	Re-renditions	Blues Rock	Joe Bonamassa	0.98
5b_lyrics	Re-renditions	Progressive Rock	Marillion	0.995
5b	No lyrics conditioning	Pop Rock	P!nk	0.995
5b	No lyrics conditioning	Southern Hip Hop	Z-Ro	0.995
5b_lyrics	Continuations	R&B	Rihanna	0.98
5b_lyrics	Re-renditions	R&B	Ne-Yo	0.98
5b	No lyrics conditioning	Rock	The Kinks	0.995
5b	No lyrics conditioning	R&B	Rihanna	0.99
5b_lyrics	Novel artists and styles	0.25 Jazz 0.75 Classic Pop	0.25 Ella Fitzgerald 0.75 Frank Sinatra	0.98
5b_lyrics	Continuations	Blues Rock	Joe Bonamassa	0.96
5b_lyrics	Novel artists and styles	Country Hip Hop	Alan Jackson	0.98
5b_lyrics	Continuations	Pop Rock	Elton John	0.98
5b_lyrics	Re-renditions	Country	Hank Williams Jr.	0.995
5b_lyrics	Continuations	Hip Hop	2Pac	0.96
5b	No lyrics conditioning	Contemporary C...	Hillsong	1.0
5b_lyrics	Unseen lyrics	Classic Pop	Frank Sinatra	0.98



# Practice

# How to use: Co-creating with AI

- LLMs continue given text (a prompt). Each generation is a random sample, and one can run the model multiple times on the same prompt to generate multiple samples.
- Content curation: Select the rare gems from the generated samples, edit if needed
- Prompt design: Which seed text nudges the model to produce the best output?

# Prompt design

- LLMs are trained with “all of the Internet”, e.g., Common Crawl dataset
- Easy to get very mediocre text out, representing the average quality of the training data
- A good prompt allows the network to identify a desired “voice” and continue using it
- Essentially, one “programs” the model through the prompt, giving it context and examples

# What is a good prompt?

1. Specific: defines the voice and task as unambiguously as possible
2. Includes concrete and high-quality few-shot examples
  - Your primary “prompt design problem”: where can you find or how can you create the examples?



# AI idea generation: Generic prompt produces generic results.

A professional game designer's analysis of this year's Independent Games Festival winner in the "Excellence in Design" category:

The game is innovative in that it uses a novel approach to player interaction to create a unique experience. The player is constantly making choices that affect the outcome of the game, and this gives the player a sense of agency that is uncommon in games.

The game is well-designed in that it is easy to learn and play, and the player is always kept engaged with the game. The game's mechanics are simple but effective, and the game's difficulty is well-balanced.

# AI idea generation: More specific prompt

A professional game designer's analysis of this year's Independent Games Festival winner in the "Excellence in Design" category:

The game has a unique weapon that allows the player to swap places with enemies, which creates interesting strategic possibilities. The game also has a minimalist art style that is visually appealing. Overall, the game is well-designed and is a worthy winner of the "Excellence in Design" award.

# Different prompts elicit different narrative voices

Hi guys plz like my first game i made in Unity. The player has to control a jetpack and collect coins

Thank you for sharing your game! We're sure that it took a lot of hard work and dedication to build, and we wish you the best of luck with it. We look forward to checking it out and giving you our feedback.

After 2 years of hard work, I've finally finished my experimental game that investigates the psychological dynamics of parental relationships. The player assumes the role of a boy whose father is dying, and must use a variety of methods to help their father achieve euthanasia. The themes of the game are death, family, and the value of life, and it is intended for an adult audience.

As a final note before the download link, I just want to thank everyone who has been supportive of my work, both in terms of enjoying my work and also in offering me feedback and suggestions. It's been a wild ride and I've learned so much about game design, programming and most importantly about myself. I've also had the great honor of meeting some of my favorite Youtubers and game developers, and I hope to continue this journey to become a full time game developer. So thank you, and enjoy the game!



# For best results, use concrete examples:

A list of innovative indie game ideas:

###

Time moves only when whe player moves or performs actions. This allows Matrix-style slow-motion gun ballet, and transforms real-time action into a puzzle.

###

The player pushes around blocks that are variables, operators, and definitions of a programming language. Blocks that connect to each other form software commands with which the player can rewrite the rules of the game world. For example, the player can connect "floor", "is", and "lava" to make the floor deadly.

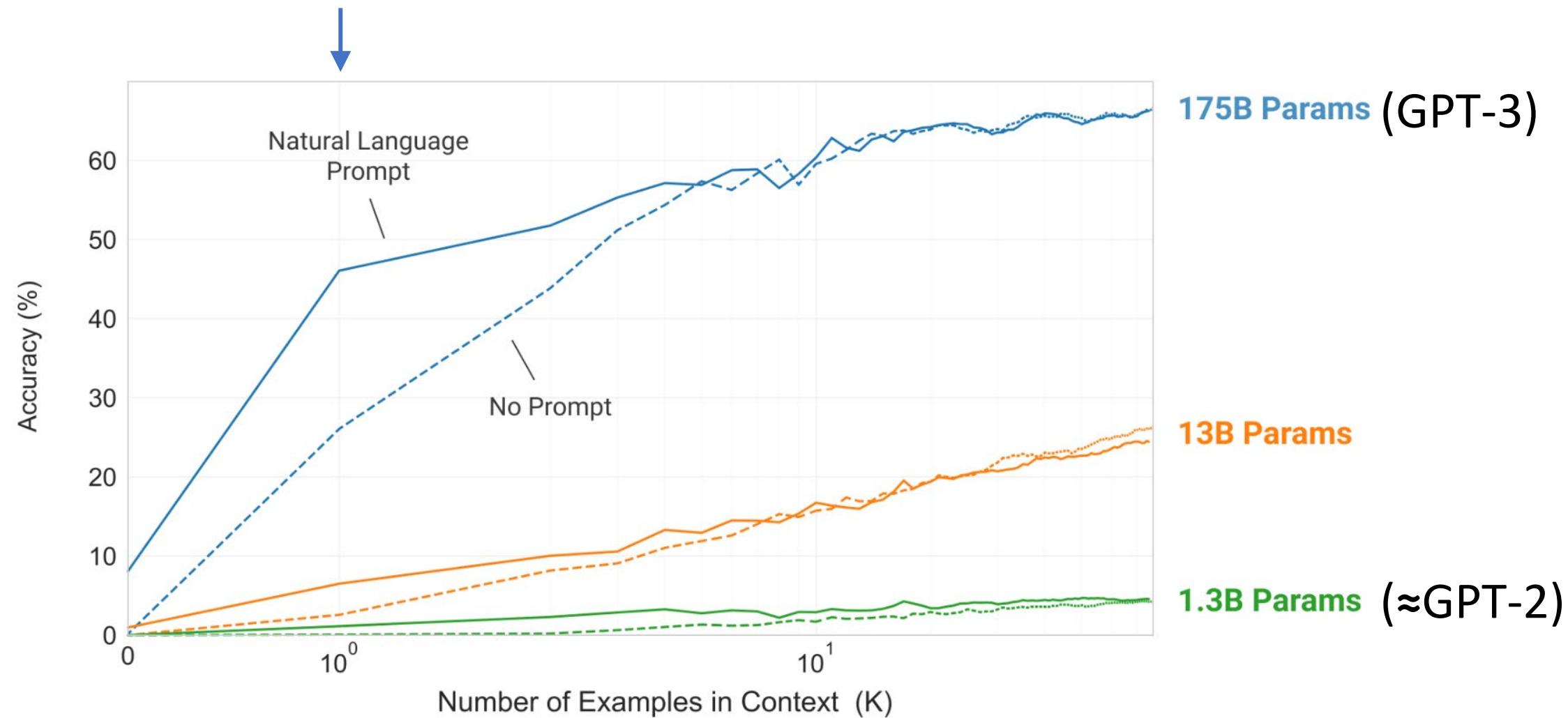
###

The player inhabits the body of a "tab" in a web browser. The goal is to manage the browser's tabs and windows, and keep the computer user productive. The player is competed against other browsers (controlled by AI or other players) to be the most efficient tab.



# Importance of examples and model scale

A large enough model achieves a good accuracy with a single example





Helping GPT-3 with a creativity tool designed for humans: VNA (verb, noun & adjective cards for game ideation)

**Award-winning game ideas based on a verb, a noun, and an adjective:**

---

**Verb: Move**

**Noun: Time**

**Adjective: Careful**

**Game idea:** Time only moves when the player moves. This allows the player to slowly and carefully maneuver inside a cloud of incoming bullets.

---

**Verb: Push**

**Noun: Block**

**Adjective: Rule-based**

**Game idea:** The game world consists of blocks that the player can push around. The blocks represent objects and logical operators, and blocks that form logical expressions define the game's rules. For instance, the three-block expression "key is win" means that the player wins the game by getting the key, and "floor is lava" means that the player burns and dies if touching the floor.

---

**Verb: Program**

**Noun: Mushroom**

**Adjective: Blue**

**Game idea:** The player controls a mushroom that can be programmed to run, turn, move, or eat. The mushroom has access to a pool of blue mushrooms, which it can program to do its bidding. The game is over when the player-controlled mushroom eats itself.

---

**Verb: Throw**

**Noun: Light**

**Adjective: Heavy**

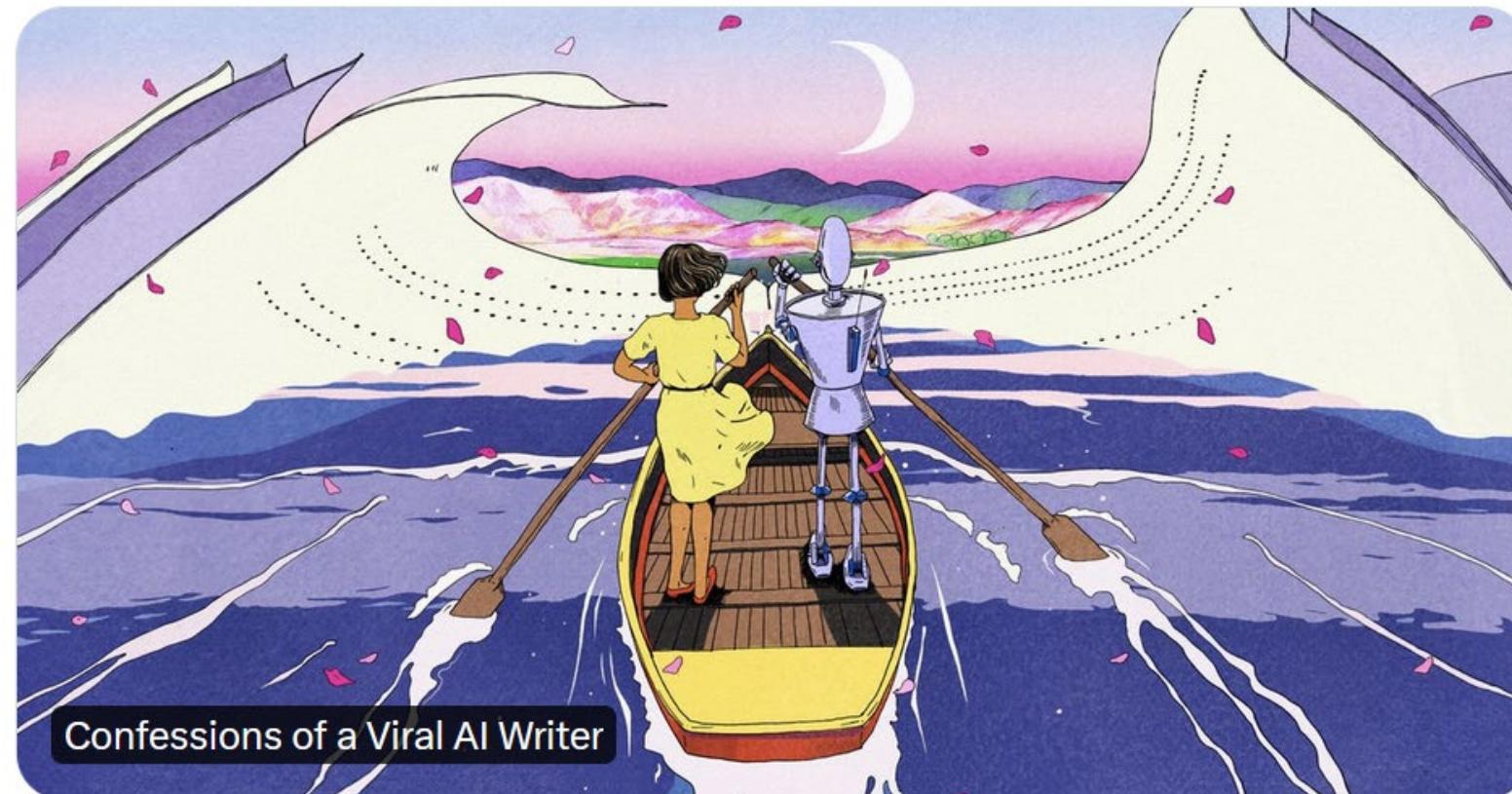
**Game idea:** The player controls a small light source that can be thrown around. The player can throw the light source into the darkness to illuminate it. The game is over when the light source gets too heavy to be thrown.



**AaltoMediaAI** @aaltomediaai · Nov 11, 2023

...

A thoughtful piece on AI and writing. Also highlights the problems of finetuned models like ChatGPT. I hope OpenAI will not eventually cut access to the legacy GPT-3 davinci model which often produces the best results for creative writing.



<https://www.wired.com/story/confessions-viral-ai-writer-chatgpt/>

# Alternatives to GPT-4

- GPT-4 is still (apparently) the best and largest LLM
- Claude 2 from Anthropic is about the same quality
- Open source alternatives emerging
- Ways to run on small(er) GPUs emerging

# Llama-2 series

- The dominant open source models
- Trained with Meta's huge resources for everyone to utilize
- Many finetuned variants exist
- Use locally on your own computer or remotely on Google Colab, AWS Bedrock, Vast.ai etc.

Llama 2 was trained on **40% more data** than Llama 1, and has double the context length.

Llama 2		
MODEL SIZE (PARAMETERS)	PRETRAINED	FINE-TUNED FOR CHAT USE CASES
7B	Model architecture:	Data collection for helpfulness and safety:
13B	Pretraining Tokens: 2 Trillion	Supervised fine-tuning: Over 100,000
70B	Context Length: 4096	Human Preferences: Over 1,000,000

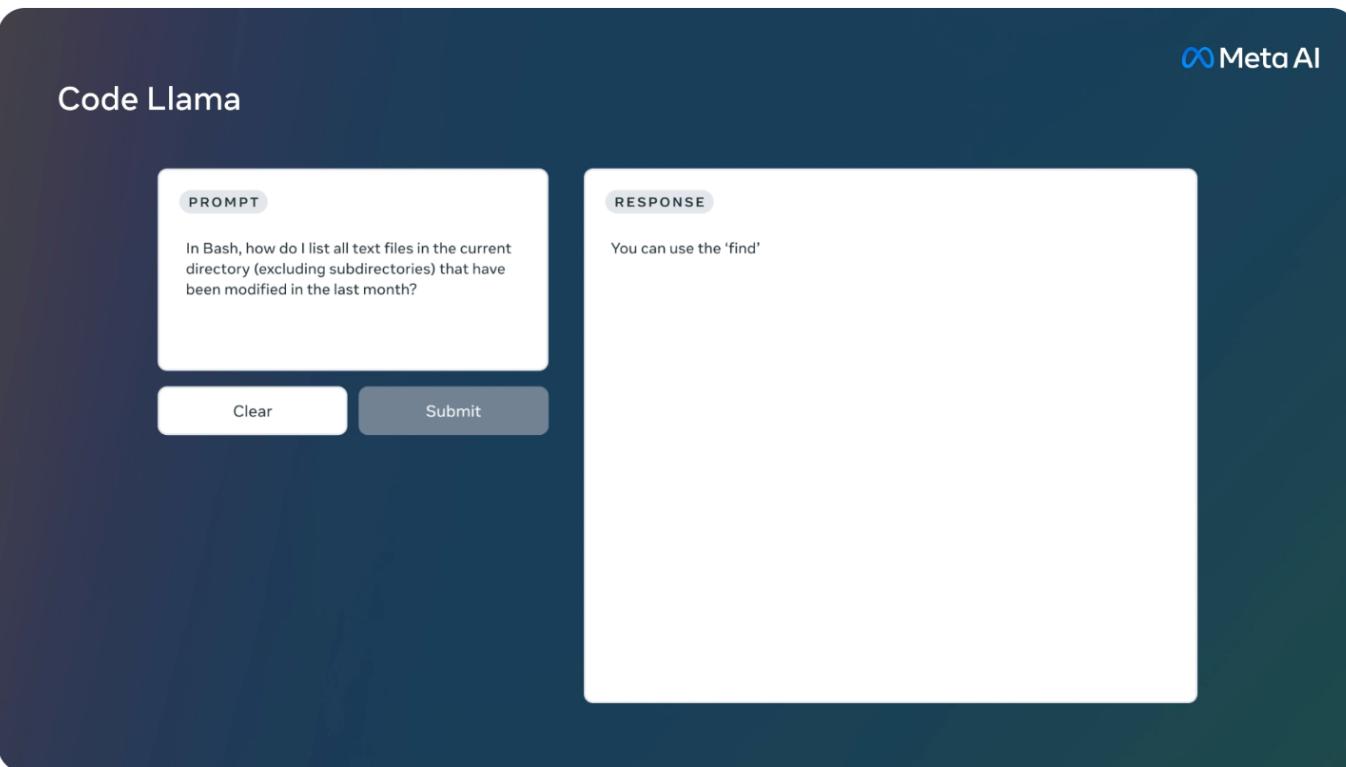
<https://ai.meta.com/llama/>

# Code Llama

Llama variant for  
programming assistance

Introducing Code Llama, a state-of-the-art  
large language model for coding

August 24, 2023



## Takeaways

Update: Jan 29, 2024: Releasing Code Llama 70B

- We are releasing Code Llama 70B, the largest and best-performing model in the Code Llama family
- Code Llama 70B is available in the same three versions as previously released Code Llama models, all free for research and commercial use:
  - CodeLlama - 70B, the foundational code model;
  - CodeLlama - 70B - Python, 70B specialized for Python;
  - and Code Llama - 70B - Instruct 70B, which is fine-tuned for understanding natural language instructions.

# Qwen

Alibaba Cloud's open model

<https://github.com/QwenLM/Qwen>

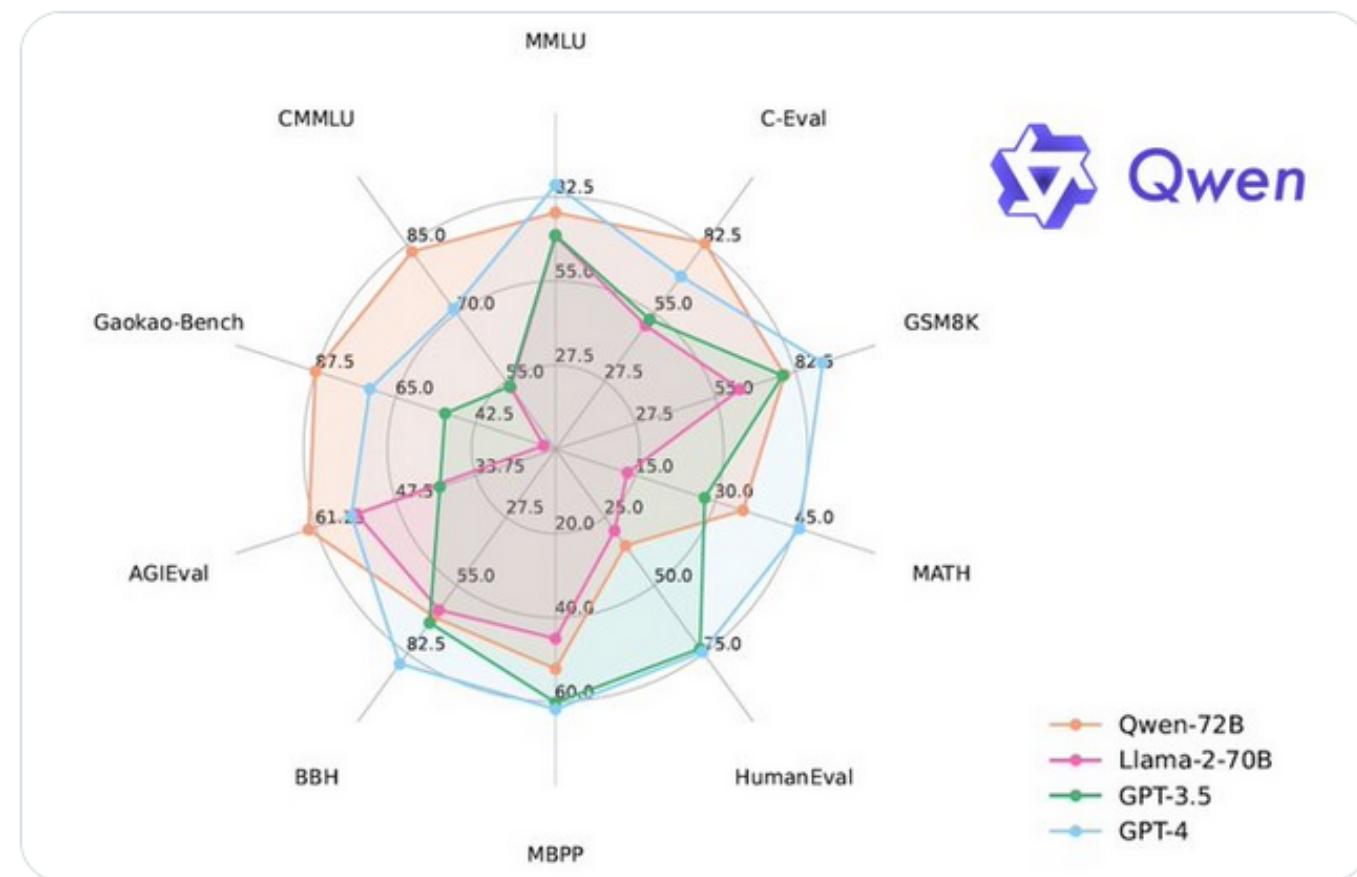


Binyuan Hui ✅ @huybery · Nov 30, 2023

We are proud to present our sincere open-source works: **Qwen-72B** and **Qwen-1.8B!** Including Base, Chat and Quantized versions!

🌟 **Qwen-72B** has been trained on high-quality data consisting of 3T tokens, boasting a larger parameter scale and more training data to achieve a...

[Show more](#)



86

509

2.4K

602K



# How to run on your computer?

<https://ollama.ai/>

<https://lmstudio.ai/>

Ollama and LM Studio support both interactive prompting and a local server you can query using Python code.

Supports both CPU and GPU

For CPU-only, see also:

<https://github.com/ggerganov/llama.cpp>



## Ollama



Get up and running with large language models locally.

macOS

[Download](#)

Windows

Coming soon! For now, you can install Ollama on Windows via WSL2.

Linux & WSL2

```
curl https://ollama.ai/install.sh | sh
```



[Manual install instructions](#)

Docker

The official [Ollama Docker image](#) `ollama/ollama` is available on Docker Hub.

Libraries

- [ollama-python](#)
- [ollama-js](#)



# How to run with limited memory?

- 70B param Llama 2 with no quantization needs  $70 \times 4 = 240$ GB GPU memory, plus some overhead
- Quantization: Instead of 32 bits per model parameter, one can quantize to 16, 8, or even 4 bits
- 3060 GPU with 12GB memory => you can run 7B Llama with 8-bit quantization
- Macs with M2 can run LLMs efficiently using the whole system memory
- Especially 8 and 4-bit quantization degrade quality

<https://arxiv.org/abs/2106.09685>

## LORA: LOW-RANK ADAPTATION OF LARGE LANGUAGE MODELS

Edward Hu\* Yelong Shen\* Phillip Wallis Zeyuan Allen-Zhu

Yuanzhi Li Shean Wang Lu Wang Weizhu Chen

Microsoft Corporation

{edwardhu, yeshe, phwallis, zeyuana, yuanzhil, swang, luw, wzchen}@microsoft.com

yuanzhil@andrew.cmu.edu

(Version 2)

### ABSTRACT

An important paradigm of natural language processing consists of large-scale pre-training on general domain data and adaptation to particular tasks or domains. As we pre-train larger models, full fine-tuning, which retrains all model parameters, becomes less feasible. Using GPT-3 175B as an example – deploying independent instances of fine-tuned models, each with 175B parameters, is prohibitively expensive. We propose Low-Rank Adaptation, or LoRA, which freezes the pre-trained model weights and injects trainable rank decomposition matrices into each layer of the Transformer architecture, greatly reducing the number of trainable parameters for downstream tasks. Compared to GPT-3 175B fine-tuned with Adam, LoRA can reduce the number of trainable parameters by 10,000 times and the GPU memory requirement by 3 times. LoRA performs on-par or better than fine-tuning in model quality on RoBERTa, DeBERTa, GPT-2, and GPT-3, despite having fewer trainable parameters, a higher training throughput, and, unlike adapters, *no additional inference latency*. We also provide an empirical investigation into rank-deficiency in language model adaptation, which sheds light on the efficacy of LoRA. We release a package that facilitates the integration of LoRA with PyTorch models and provide our implementations and model checkpoints for RoBERTa, DeBERTa, and GPT-2 at <https://github.com/microsoft/LoRA>.

### 1 INTRODUCTION

Many applications in natural language processing rely on adapting *one* large-scale, pre-trained language model to *multiple* downstream applications. Such adaptation is usually done via *fine-tuning*, which updates all the parameters of the pre-trained model. The major downside of fine-tuning is that the new model contains as many parameters as in the original model. As larger models are trained every few months, this changes from a mere “inconvenience” for GPT-2 (Radford et al., b) or RoBERTa large (Liu et al., 2019) to a critical deployment challenge for GPT-3 (Brown et al., 2020) with 175 billion trainable parameters.<sup>1</sup>

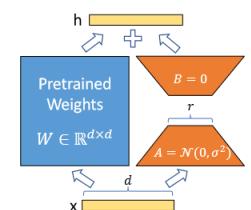


Figure 1: Our reparametrization. We only train  $A$  and  $B$ .

Many sought to mitigate this by adapting only some parameters or learning external modules for new tasks. This way, we only need to store and load a small number of task-specific parameters in addition to the pre-trained model for each task, greatly boosting the operational efficiency when deployed. However, existing techniques

\*Equal contribution.

<sup>1</sup>Compared to V1, this draft includes better baselines, experiments on GLUE, and more on adapter latency.

<sup>1</sup>While GPT-3 175B achieves non-trivial performance with few-shot learning, fine-tuning boosts its performance significantly as shown in Appendix A.



# Microsoft Phi & Phi 2

Small LLMs can be surprisingly powerful if the training data is of high enough quality.

Recommended current model: 2.7B Phi-2, available via Ollama, Huggingface etc.

Colab (3<sup>rd</sup> party):

[https://colab.research.google.com/drive/14\\_mVXXdXmDiFshVArDQlWeP-3DKzbvNI](https://colab.research.google.com/drive/14_mVXXdXmDiFshVArDQlWeP-3DKzbvNI)

Papers and blog: <https://arxiv.org/abs/2306.11644>

<https://arxiv.org/abs/2309.05463>

<https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models/>

<https://huggingface.co/microsoft/phi-2>

Textbooks Are All You Need

Suriya Gunasekar Allie Del Giorno Gustavo de Rosa Xin Wang	Yi Zhang Sivakanth Gopi Olli Saarikivi Sébastien Bubeck	Jyoti Aneja Mojan Javaheripi Adil Salim Shital Shah Harkirat Singh Behl Ronen Eldan	Caio César Teodoro Mendes Piero Kauffmann Yin Tat Lee
			Yuanzhi Li

Microsoft Research

## Abstract

We introduce **phi-1**, a new large language model for code, with significantly smaller size than competing models: **phi-1** is a Transformer-based model with 1.3B parameters, trained for 4 days on 8 A100s, using a selection of “textbook quality” data from the web (6B tokens) and synthetically generated textbooks and exercises with GPT-3.5 (1B tokens). Despite this small scale, **phi-1** attains **pass@1** accuracy 50.6% on HumanEval and 55.5% on MBPP. It also displays surprising emergent properties compared to **phi-1-base**, our model *before* our finetuning stage on a dataset of coding exercises, and **phi-1-small**, a smaller model with 350M parameters trained with the same pipeline as **phi-1** that still achieves 45% on HumanEval.

## 1 Introduction

The art of training large artificial neural networks has made extraordinary progress in the last decade, especially after the discovery of the Transformer architecture [VSP<sup>+</sup>17], yet the science behind this success remains limited. Amidst a vast and confusing array of results, a semblance of order emerged around the same time as Transformers were introduced, namely that performance improves somewhat predictably as one scales up either the amount of compute or the size of the network [HNA<sup>+</sup>17], a phenomenon which is now referred to as *scaling laws* [KMH<sup>+</sup>20]. The subsequent exploration of scale in deep learning was guided by these scaling laws [BMR<sup>+</sup>20], and discoveries of variants of these laws led to rapid jump in performances [HBM<sup>+</sup>22]. In this work, following the footsteps of Eldan and Li [EL23], we explore the improvement that can be obtained along a different axis: the *quality* of the data. It has long been known that higher quality data leads to better results, e.g., data cleaning is an important part of modern dataset creation [RSR<sup>+</sup>20], and it can yield other side benefits such as somewhat smaller datasets [LYR<sup>+</sup>23, YGK<sup>+</sup>23] or allowing for more passes on the data [MRB<sup>+</sup>23]. The recent work of Eldan and Li on TinyStories (a high quality dataset synthetically generated to teach English to neural networks) showed that in fact the effect of high quality data extends well past this: improving data quality can dramatically change the shape of the scaling laws, potentially allowing to match the performance of large-scale models with much leaner training/models. In this work we go beyond the initial foray of Eldan and Li to show that high quality data can even **improve** the SOTA of large language models (LLMs), while dramatically reducing the dataset size and training compute. Importantly, smaller models requiring less training can significantly reduce the environmental cost of LLMs [BGMMS21].

We focus our attention on LLMs trained for code, and specifically writing simple Python functions from their docstrings as in [CTJ<sup>+</sup>21]. The evaluation benchmark proposed in the latter work, HumanEval, has been widely adopted for comparing LLMs’ performance on code. We demonstrate the power of high

Date	Model	Model size (Parameters)	Dataset size (Tokens)	HumanEval (Pass@1)	MBPP (Pass@1)
2021 Jul	Codex-300M [CTJ <sup>+</sup> 21]	300M	100B	13.2%	-
2021 Jul	Codex-12B [CTJ <sup>+</sup> 21]	12B	100B	28.8%	-
2022 Mar	CodeGen-Mono-350M [NPH <sup>+</sup> 23]	350M	577B	12.8%	-
2022 Mar	CodeGen-Mono-16.1B [NPH <sup>+</sup> 23]	16.1B	577B	29.3%	35.3%
2022 Apr	PaLM-Coder [CND <sup>+</sup> 22]	540B	780B	35.9%	47.0%
2022 Sep	CodeGeeX [ZXZ <sup>+</sup> 23]	13B	850B	22.9%	24.4%
2022 Nov	GPT-3.5 [Ope23]	175B	N.A.	47%	-
2022 Dec	SantaCoder [ALK <sup>+</sup> 23]	1.1B	236B	14.0%	35.0%
2023 Mar	GPT-4 [Ope23]	N.A.	N.A.	67%	-
2023 Apr	Replit [Rep23]	2.7B	525B	21.9%	-
2023 Apr	Replit-Finetuned [Rep23]	2.7B	525B	30.5%	-
2023 May	CodeGen2-1B [NHX <sup>+</sup> 23]	1B	N.A.	10.3%	-
2023 May	CodeGen2-7B [NHX <sup>+</sup> 23]	7B	N.A.	19.1%	-
2023 May	StarCoder [LAZ <sup>+</sup> 23]	15.5B	1T	33.6%	52.7%
2023 May	StarCoder-Prompted [LAZ <sup>+</sup> 23]	15.5B	1T	40.8%	49.5%
2023 May	PaLM 2-S [ADF <sup>+</sup> 23]	N.A.	N.A.	37.6%	50.0%
2023 May	CodeT5+ [WLG <sup>+</sup> 23]	2B	52B	24.2%	-
2023 May	CodeT5+ [WLG <sup>+</sup> 23]	16B	52B	30.9%	-
2023 May	InstructCodeT5+ [WLG <sup>+</sup> 23]	16B	52B	35.0%	-
2023 Jun	WizardCoder [LXZ <sup>+</sup> 23]	16B	1T	57.3%	51.8%
2023 Jun	<b>phi-1</b>	1.3B	7B	50.6%	55.5%

Table 1: We use self-reported scores whenever available. Despite being trained at vastly smaller scale, **phi-1** outperforms competing models on HumanEval and MBPP, except for GPT-4 (also WizardCoder obtains better HumanEval but worse MBPP).



PyGame example. We start with a PyGame that asks the model to generate code to move a ball:

**Prompt:**

```
""" Write a PyGame. There is a ball. At every iteration, (1). The x-axis of the ball increases by a random integer between (-10, 10), (2). The y-axis of the ball increases by a random integer between (-20, 20). The x-axis of the ball should stay inside 0-400, and y-axis of the ball should stay inside 0-600. When the user press 'space', set the x-axis of the ball to 200 and y-axis of the ball to 400.  
"""
```

**phi-1**

```
...  
screen = pygame.display.set_mode  
((400, 600))  
x, y = 0, 0  
while True:  
    for event in pygame.event.get()  
    ...  
        elif event.type == pygame  
.KEYDOWN:  
            if event.key ==  
pygame.K_SPACE:  
                x = 200  
                y = 400  
x += random.randint(-10, 10)  
y += random.randint(-20, 20)  
if x < 0:  
    x = 0  
elif x > 400:  
    x = 400  
if y < 0:  
    y = 0  
elif y > 600:  
    y = 600  
screen.fill((255, 255, 255))  
pygame.draw.circle(screen,  
(255, 0, 0), (x, y), 10)  
pygame.display.update()
```

**phi-1-base**

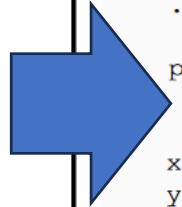
```
...  
ball_x = random.randint(-10, 10)  
ball_y = random.randint(-20, 20)  
ball_vx = random.randint(-5, 5)  
ball_vy = random.randint(-5, 5)  
...  
screen = pygame.display.set_mode  
((800, 600))  
...  
while not done:  
    for event in pygame.event.get()  
    ...  
        elif event.type == pygame  
.KEYDOWN:  
            if event.key ==  
pygame.K_SPACE:  
                ball_x = 200  
                ball_y = 400  
ball_x += ball_vx  
ball_y += ball_vy  
ball_vx += ball_vy  
pygame.display.flip()
```

**phi-1-small**

```
...  
x, y = 0, 0  
dx, dy = random.randint(-10, 10),  
random.randint(-20, 20)  
while True:  
    screen.fill((0, 0, 0))  
    pygame.draw.rect(screen,  
(255, 255, 255), (x, y, 400,  
600))  
    pygame.display.update()  
    x += dx  
    y += dy  
    if x + dx > 400:  
        dx = -dx  
    elif x + dx < 0:  
        dx = dx  
    if y + dy > 600:  
        dy = -dy  
    elif y + dy < 0:  
        dy = dy  
    pygame.time.wait(50)
```

Syntactically and functionally correct.

Others only syntactically correct.



# Where does the data come from?

High-quality in-demand data creation and curation is an emerging job market.

**“Remotasks annotators generally earn between \$10 and \$25 per hour, though some subject-matter experts can make more.”**

**” You can make \$45 an hour teaching robots law or make \$25 an hour teaching them poetry. There were also listings for people with security clearance, presumably to help train military AI. ”**



<https://www.theverge.com/features/23764584/ai-artificial-intelligence-data-annotation-labor-scale-surge-remotasks-openai-chatbots>

# Then again...

Supply-and-demand economics at play. Data that anyone can create without higher education is cheap.

**“By the beginning of this year, pay for the Kenyan annotators I spoke with had dropped to between \$1 and \$3 per hour.”**

Jun 20, 2023, 3:05 PM  
GMT+3  
23 Comments / 23 New

ARTIFICIAL INTELLIGENCE

## AI Is a Lot of Work

As the technology becomes ubiquitous, a vast tasker underclass is emerging — and not going anywhere.

By Josh Dzieza, an investigations editor covering tech, business, and climate change. Since joining The Verge in 2014, he's won a Loeb Award for feature writing, among others.  
Illustrations by Richard Parry for The Verge



<https://www.wired.com/story/artificial-intelligence-data-labeling-children/>

# Underage Workers Are Training AI

Companies that provide Big Tech with AI data-labeling services are inadvertently hiring young teens to work on their platforms, often exposing them to traumatic content.



# These Prisoners Are Training AI

In high-wage Finland, where clickworkers are rare, one company has discovered a novel labor force—prisoners.

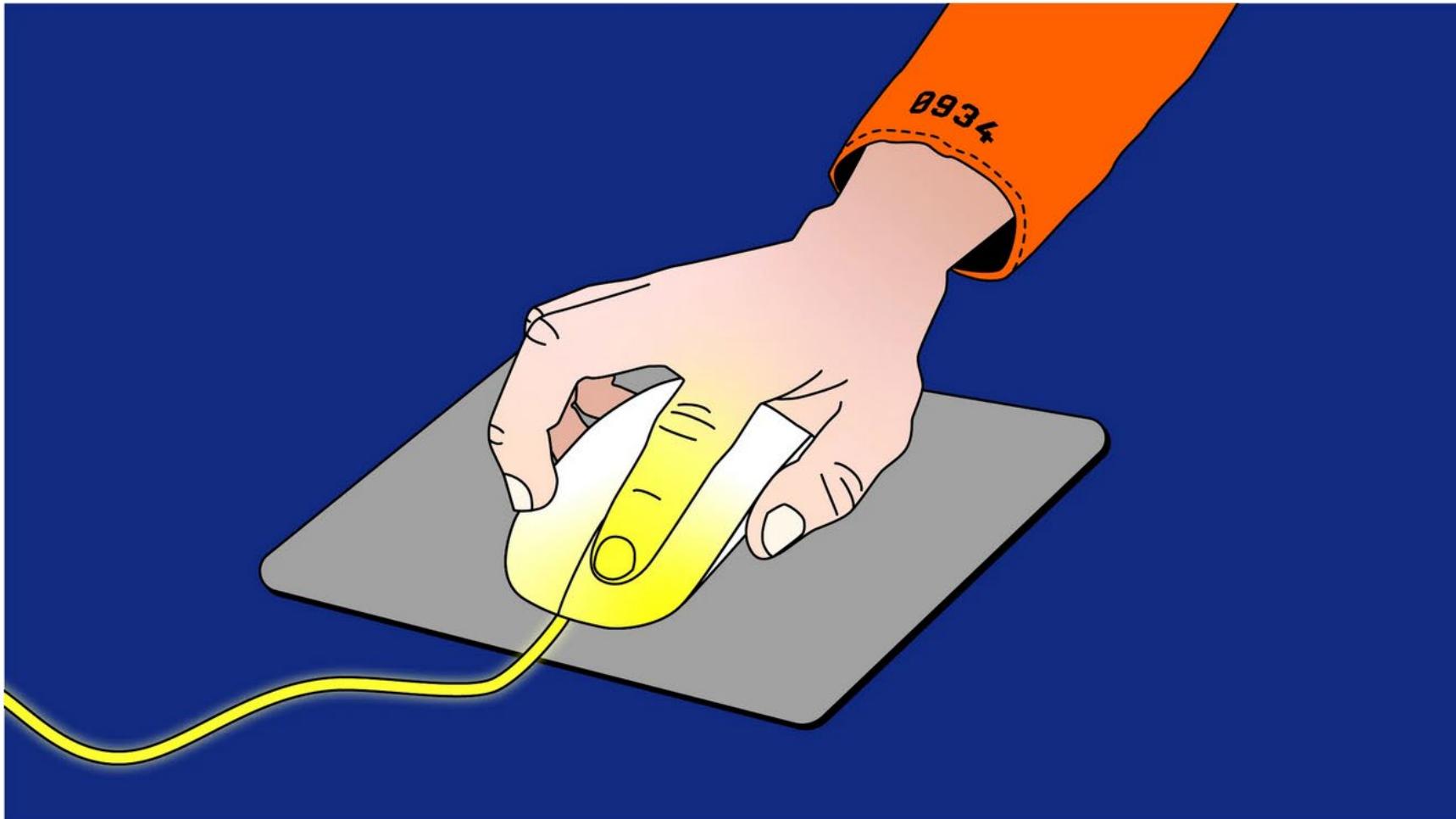


ILLUSTRATION: JACQUI VANLIEW; GETTY IMAGES



# Data is not a moat

GPT-4 is widely used to generate high-quality training data for smaller open-source LLMs

This together with Llama models and LoRA has been fueling many open source efforts.

Legality not clear, safe only for non-commercial use

## Instruction Tuning with GPT-4

Baolin Peng\*, Chunyuan Li\*, Pengcheng He\*, Michel Galley, Jianfeng Gao (\*Equal Contribution)

[[Project Page](#)] [[Paper](#)]



Pronounced as "GPT-4-LLM" or "GPT-for-LLM", image is generated by [GLIGEN](#)

[Code License](#) Apache 2.0 [Data License](#) CC By NC 4.0

This is the repo for the GPT-4-LLM, which aims to share data generated by GPT-4 for building an instruction-following LLMs with supervised learning and reinforcement learning. The repo contains:

- English Instruction-Following [Data](#) generated by GPT-4 using Alpaca prompts for fine-tuning LLMs.
- Chinese Instruction-Following [Data](#) generated by GPT-4 using Chinese prompts translated from Alpaca by ChatGPT.
- Comparison [Data](#) ranked by GPT-4 to train reward models.
- Answers on Unnatural Instructions [Data](#) from GPT-4 to quantify the gap between GPT-4 and instruction-tuned models at scale.

**Usage and License Notices:** The data is intended and licensed for research use only. The dataset is CC BY NC 4.0 (allowing only non-commercial use) and models trained using the dataset should not be used outside of research purposes.



# Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%\* ChatGPT Quality

by: The Vicuna Team, Mar 30, 2023

<https://lmsys.org/blog/2023-03-30-vicuna/>

We introduce Vicuna-13B, an open-source chatbot trained by fine-tuning LLaMA on user-shared conversations collected from ShareGPT. Preliminary evaluation using GPT-4 as a judge shows Vicuna-13B achieves more than 90%\* quality of OpenAI ChatGPT and Google Bard while outperforming other models like LLaMA and Stanford Alpaca in more than 90%\* of cases. The cost of training Vicuna-13B is around \$300. The [code](#) and [weights](#), along with an online [demo](#), are publicly available for non-commercial use.



Vicuna (generated by stable diffusion 2.1)

\*According to a fun and non-scientific evaluation with GPT-4. Further rigorous evaluation is needed.

# AI companies have all kinds of arguments against paying for copyrighted content

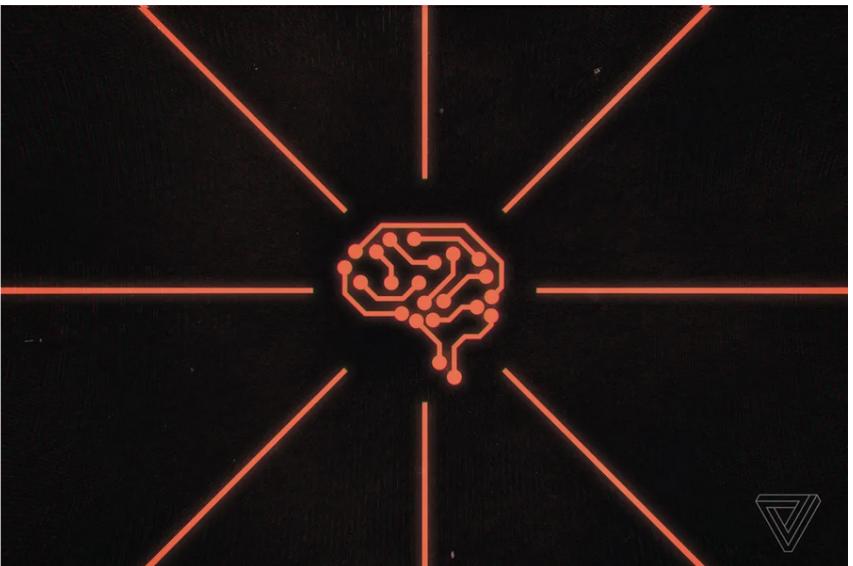


Illustration by Alex Castro / The Verge

/ The biggest companies in AI aren't interested in paying to use copyrighted material as training data, and here are their reasons why.

By [Wes Davis](#), a weekend editor who covers the latest in tech and entertainment. He has written news, reviews, and more as a tech journalist since 2020.

Nov 5, 2023, 12:17 AM GMT+2 | □ 42 Comments / 42 New



The US Copyright Office is [taking public comment](#) on potential new rules around generative AI's use of [copyrighted materials](#), and the biggest AI companies in the world had plenty to say. We've collected the arguments from [Meta](#), [Google](#), [Microsoft](#), [Adobe](#), [Hugging Face](#), [StabilityAI](#), and [Anthropic](#) below, as well as a response from [Apple](#) that focused on copyrighting AI-written code.

There are some differences in their approaches, but the overall message for most is the same: They don't think they should have to pay to train AI models on copyrighted work.





# How to finetune with limited memory?

- If you have more training examples than fits the prompt (4k tokens for Llama 2), you need to *finetune* the model
- Finetuning a 70B param Llama 2 with no quantization needs over 780GB
- LoRA: Ground-breaking low-memory finetuning technique, reduces the memory requirement by 3 times
- Still a lot for a hobbyist...

<https://arxiv.org/abs/2106.09685>

## LORA: LOW-RANK ADAPTATION OF LARGE LANGUAGE MODELS

Edward Hu\* Yelong Shen\* Phillip Wallis Zeyuan Allen-Zhu

Yuanzhi Li Shean Wang Lu Wang Weizhu Chen

Microsoft Corporation

{edwardhu, yeshe, phwallis, zeyuana,  
yuanzhil, swang, luw, wzchen}@microsoft.com

yuanzhil@andrew.cmu.edu

(Version 2)

### ABSTRACT

An important paradigm of natural language processing consists of large-scale pre-training on general domain data and adaptation to particular tasks or domains. As we pre-train larger models, full fine-tuning, which retrains all model parameters, becomes less feasible. Using GPT-3 175B as an example – deploying independent instances of fine-tuned models, each with 175B parameters, is prohibitively expensive. We propose Low-Rank Adaptation, or LoRA, which freezes the pre-trained model weights and injects trainable rank decomposition matrices into each layer of the Transformer architecture, greatly reducing the number of trainable parameters for downstream tasks. Compared to GPT-3 175B fine-tuned with Adam, LoRA can reduce the number of trainable parameters by 10,000 times and the GPU memory requirement by 3 times. LoRA performs on-par or better than fine-tuning in model quality on RoBERTa, DeBERTa, GPT-2, and GPT-3, despite having fewer trainable parameters, a higher training throughput, and, unlike adapters, *no additional inference latency*. We also provide an empirical investigation into rank-deficiency in language model adaptation, which sheds light on the efficacy of LoRA. We release a package that facilitates the integration of LoRA with PyTorch models and provide our implementations and model checkpoints for RoBERTa, DeBERTa, and GPT-2 at <https://github.com/microsoft/LoRA>.

### 1 INTRODUCTION

Many applications in natural language processing rely on adapting *one* large-scale, pre-trained language model to *multiple* downstream applications. Such adaptation is usually done via *fine-tuning*, which updates all the parameters of the pre-trained model. The major downside of fine-tuning is that the new model contains as many parameters as in the original model. As larger models are trained every few months, this changes from a mere “inconvenience” for GPT-2 (Radford et al., 2018) or RoBERTa large (Liu et al., 2019) to a critical deployment challenge for GPT-3 (Brown et al., 2020) with 175 billion trainable parameters.<sup>1</sup>

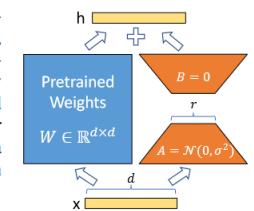


Figure 1: Our reparametrization. We only train  $A$  and  $B$ .

Many sought to mitigate this by adapting only some parameters or learning external modules for new tasks. This way, we only need to store and load a small number of task-specific parameters in addition to the pre-trained model for each task, greatly boosting the operational efficiency when deployed. However, existing techniques

\*Equal contribution.

<sup>1</sup>Compared to V1, this draft includes better baselines, experiments on GLUE, and more on adapter latency.

<sup>1</sup>While GPT-3 175B achieves non-trivial performance with few-shot learning, fine-tuning boosts its performance significantly as shown in Appendix A.

# How to finetune with limited memory?

QLoRA: Finetune a 70B Llama 2 on a single 80GB A100 GPU (e.g., Colab Pro), or 7B and 14B models on high-end consumer GPU like 4090 or 3090.

Similar to LoRA, but the original LLM is kept fixed using 4-bit quantization.

Can mitigate quantization errors, to some extent.

## QLoRA: Efficient Finetuning of Quantized LLMs

Tim Dettmers\*

Artidoro Pagnoni\*

Ari Holtzman

Luke Zettlemoyer

University of Washington

{dettmers,artidoro,ahai,lsz}@cs.washington.edu

### Abstract

We present QLoRA, an efficient finetuning approach that reduces memory usage enough to finetune a 65B parameter model on a single 48GB GPU while preserving full 16-bit finetuning task performance. QLoRA backpropagates gradients through a frozen, 4-bit quantized pretrained language model into Low Rank Adapters (LoRA). Our best model family, which we name **Guanaco**, outperforms all previous openly released models on the Vicuna benchmark, reaching 99.3% of the performance level of ChatGPT while only requiring 24 hours of finetuning on a single GPU. QLoRA introduces a number of innovations to save memory without sacrificing performance: (a) 4-bit NormalFloat (NF4), a new data type that is information theoretically optimal for normally distributed weights (b) Double Quantization to reduce the average memory footprint by quantizing the quantization constants, and (c) Paged Optimizers to manage memory spikes. We use QLoRA to finetune more than 1,000 models, providing a detailed analysis of instruction following and chatbot performance across 8 instruction datasets, multiple model types (LLaMA, T5), and model scales that would be infeasible to run with regular finetuning (e.g. 33B and 65B parameter models). Our results show that QLoRA finetuning on a small high-quality dataset leads to state-of-the-art results, even when using smaller models than the previous SoTA. We provide a detailed analysis of chatbot performance based on both human and GPT-4 evaluations showing that GPT-4 evaluations are a cheap and reasonable alternative to human evaluation. Furthermore, we find that current chatbot benchmarks are not trustworthy to accurately evaluate the performance levels of chatbots. A lemon-picked analysis demonstrates where **Guanaco** fails compared to ChatGPT. We release all of our models and code, including CUDA kernels for 4-bit training.<sup>2</sup>

### 1 Introduction

Finetuning large language models (LLMs) is a highly effective way to improve their performance, [40, 62, 43, 61, 59, 37] and to add desirable or remove undesirable behaviors [43, 2, 4]. However, finetuning very large models is prohibitively expensive; regular 16-bit finetuning of a LLaMA 65B parameter model [37] requires more than 780 GB of GPU memory. While recent quantization methods can reduce the memory footprint of LLMs [14, 13, 18, 66], such techniques only work for inference and break down during training [65].

We demonstrate for the first time that it is possible to finetune a quantized 4-bit model without any performance degradation. Our method, QLoRA, uses a novel high-precision technique to quantize a pretrained model to 4-bit, then adds a small set of learnable Low-rank Adapter weights [28]

\*Equal contribution.

<sup>2</sup><https://github.com/artidoro/qlora> and <https://github.com/TimDettmers/bitsandbytes>



Haihao Shen  
@HaihaoShen

...

✖️ No GPU but wanna create your own LLM on laptop?

🎁 Here is a gift for you: QLoRA on CPU, making LLM fine-tuning on client CPU possible! Just give a try.

📘 Blog: [medium.com/@NeuralCompress...](https://medium.com/@NeuralCompress...) Kudos to ITREX team!

🎯 Code: [github.com/intel/intel-ex...](https://github.com/intel/intel-ex...)

#IAmIntel #intelai @intel @huggingface

# intel/intel-extension-for-transformers



⚡ Build your chatbot within minutes on your favorite device; offer SOTA compression techniques for LLMs; run LLMs efficiently on...

83 94

Contributors

17

Used by

3

Discussions

2k

Stars

167

Forks



GitHub - intel/intel-extension-for-transformers: ⚡ Build your chatbot within minutes o...

<https://github.com/intel/intel-extension-for-transformers>



Andy Peatling  
@apeatling

...

I've created a step-by-step guide to fine-tuning an LLM on your Apple silicon Mac.

I'll walk you through the entire process, and it won't cost you a penny because we're going to do it all on your own hardware using Apple's MLX framework.

[apeatling.com/articles/simpl...](https://apeatling.com/articles/simple-guide-to-local-lm-fine-tuning-on-a-mac-with-mlx/)

4:16 PM · Jan 8, 2024 · 143.5K Views

---

17

131

709

1.2K



<https://apeatling.com/articles/simple-guide-to-local-lm-fine-tuning-on-a-mac-with-mlx/>



darren ✅

@darrenangle

...

tried many LLM fine-tuning frameworks but this new one called X-LLM by [@BobaZooba](#) has v good dev experience so far (worked first try)

mistral qlora with flash attention 2 & deepspeed on 8 L4s, 8.5k dolly examples for 4 epochs, ~1.5 hrs

<https://github.com/BobaZooba/xllm/tree/main>

17 lines of data prep and one cli command

```
datasets
dm import tqdm
lm import Config
lm.datasets import GeneralDataset
lm.experiments import Experiment
lm.cli import cli_run_train

def process_text(text):
    replacements = {
        '<|im_start|>system assistant:internal': '<tool>',
        '<|im_start|>system': '<system>',
        '<|im_start|>user': '<user>',
        '<|im_start|>assistant': '<assistant>',
        '<|im_end|>': '<s>',

        key, value in replacements.items():
            text = text.replace(key, value)
    }
    return text

if __name__ == '__main__':
    dolly_chatml = datasets.load_dataset("sam-mosaic/dolly_chatml")
    dataset = [process_text(row) for row in tqdm(dolly_chatml['train'])]
    train_dataset = GeneralDataset.from_list(data=dataset)
    cli_run_train(config_cls=Config, train_dataset=train_dataset)
```

```
30   --deepspeed_stage 2 \
31   --load_in_4bit True \
32   --use_flash_attention_2 True \
33   --apply_lora True \
34   --stabilize True \
35   --use_gradient_checkpointing True \
36   --prepare_model_for_kbit_training True \
37   --gradient_accumulation_steps 1 \
38   --save_total_limit 5 \
39   --per_device_train_batch_size 8 \
40   --num_train_epochs 4 \
41   --save_steps 20 \
42   --warmup_steps 100
```

	N/A	79C	P0	71W / 72W	20700MiB / 23034MiB	100%	Default	N/A
1	NVIDIA L4		On	00000000:00:05.0 Off		0		
N/A	73C	P0	68W / 72W	21004MiB / 23034MiB		100%	Default	N/A
2	NVIDIA L4		On	00000000:00:06.0 Off		0		
N/A	76C	P0	70W / 72W	21764MiB / 23034MiB		100%	Default	N/A
3	NVIDIA L4		On	00000000:00:07.0 Off		0		
N/A	79C	P0	71W / 72W	20212MiB / 23034MiB		100%	Default	N/A
4	NVIDIA L4		On	00000000:80:00.0 Off		0		
N/A	76C	P0	72W / 72W	18906MiB / 23034MiB		100%	Default	N/A
5	NVIDIA L4		On	00000000:80:01.0 Off		0		
N/A	75C	P0	72W / 72W	19500MiB / 23034MiB		100%	Default	N/A



LMSYS [Chatbot Arena](#) is a crowdsourced open platform for LLM evals. We've collected over 200,000 human preference votes to rank LLMs with the Elo ranking system.

# Which LLM to pick?

Multiple public leaderboards available

Chatbot Arena: Humans prompt two random models and judge which model wins:

<https://huggingface.co/spaces/lmsys/chatbot-arena-leaderboard>

Huggingface Open LLM Leaderboard combines various established test datasets:

[https://huggingface.co/spaces/HuggingFaceH4/open\\_llm\\_leaderboard](https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard)

Arena Elo    Full Leaderboard

Total #models: 56. Total #votes: 244024. Last updated: Jan 26, 2024.

Contribute your vote 🗳 at <chat.lmsys.org>! Find more analysis in the [notebook](#).

Rank	🤖 Model	⭐ Arena Elo	📊 95% CI	🗳️ Votes	Organization
1	<a href="#">GPT-4-Turbo</a>	1249	+13/-13	30268	OpenAI
2	<a href="#">Bard (Gemini Pro)</a>	1215	+16/-15	3014	Google
3	<a href="#">GPT-4-0314</a>	1189	+14/-12	18062	OpenAI
4	<a href="#">GPT-4-0613</a>	1161	+13/-13	27441	OpenAI
5	<a href="#">Mistral Medium</a>	1150	+15/-15	11480	Mistral
6	<a href="#">Claude-1</a>	1150	+13/-13	17630	Anthropic
7	<a href="#">Claude-2.0</a>	1131	+14/-13	11698	Anthropic
8	<a href="#">Mixtral-8x7b-Instruct-v0.1</a>	1123	+14/-13	14165	Mistral
9	<a href="#">Gemini Pro (Dev API)</a>	1122	+15/-16	6332	Google
10	<a href="#">Claude-2.1</a>	1119	+14/-13	23323	Anthropic
11	<a href="#">GPT-3.5-Turbo-0613</a>	1117	+14/-13	30326	OpenAI
12	<a href="#">Gemini Pro</a>	1114	+16/-15	6981	Google
13	<a href="#">Yi-34B-Chat</a>	1112	+13/-15	5807	01 AI



# Testing Chatbot Arena

Model A

Please create an innovative game idea with the theme "Make me laugh". The idea should be with a unique design hook.

Game Title: "Giggles & Gags: Comedy Chaos"

Game Concept:

Giggles & Gags: Comedy Chaos is a multiplayer, party-style video game that focuses on making players laugh through a series of humorous mini-games, wacky challenges, and improvisational comedy scenarios. The unique design hook of this game is the integration of real-time voice and facial recognition technology to analyze and measure player reactions, with the ultimate goal of determining who laughs the most and wins the title of "Ultimate Comedy Champion."

Gameplay Mechanics:

1. Mini-games: Players engage in various mini-games, such as joke-telling battles, impersonation contests, and physical comedy challenges, that are designed to elicit laughter. Each mini-game is short (1-3 minutes), ensuring a fast-paced and constantly changing experience.
2. Wacky challenges: Players must complete absurd and hilarious challenges, such as

Model A: mistral-medium

👉 A is better

👉 B is better

Model B

Please create an innovative game idea with the theme "Make me laugh". The idea should be with a unique design hook.

Title: Laugh-A-Palooza

Overview: Laugh-A-Palooza is a hilarious party game that challenges players to be the funniest person in the room. With a unique design hook, this game combines physical comedy with social deduction, creating a fun and interactive experience for players of all ages.

Gameplay:

1. Each player starts with a set of comedy props, such as rubber chickens, whoopee cushions, and silly hats.
2. The objective of the game is to be the first player to get everyone else in the room to laugh out loud.
3. Players take turns being the "comedian" and performing a stand-up comedy routine, using their comedy props to add humor and creativity to their act.
4. The other players in the room are the "judges" and must evaluate the comedian's performance, determining whether or not they have made them laugh.
5. If a player laughs, they must pay a small fine, such as a point or a token, to the comedian.

Model B: codellama-34b-instruct

👉 Tie

👉 Both are bad

# Material

- Qwern's excellent article on creative fiction using GPT-3  
<https://www.gwern.net/GPT-3>
- 9 essays on AI & GPT-3 by philosophers, with replies by GPT-3:  
<https://dailynous.com/2020/07/30/philosophers-gpt-3/>
- My AI for Media, Art & Design course:  
<https://github.com/PerttuHamalainen/MediaAI>