Text generation

Co-writing with AI language models

Prof. Perttu Hämäläinen, Jan 2024
Some of the material courtesy of Prof. Christian Guckelsberger

For exercises and more GenAI, see:
https://github.com/PerttuHamalainen/MediaAI

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?
- Recent progress: Where is the field heading?

Theory

Excellent 8-minute intro https://www.youtube.com/watch?v=lnA9DMvHtfI

Large Language Models
Part 2

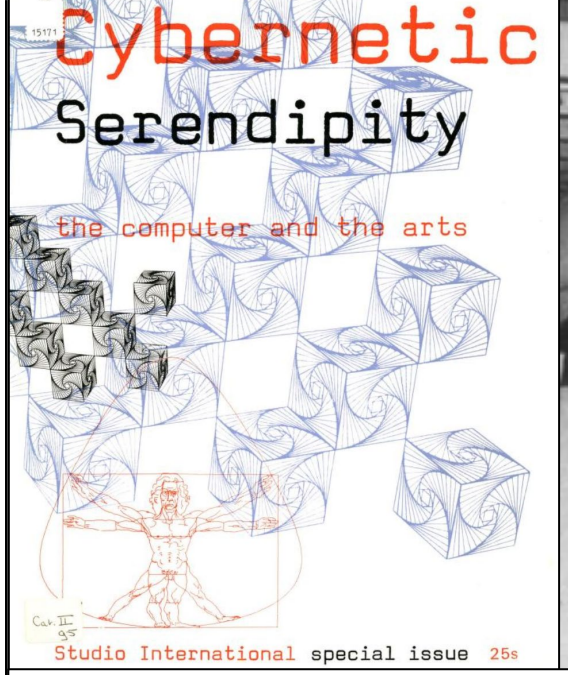
https://www.youtube.com/watch?v=YDiSFS-yHwk

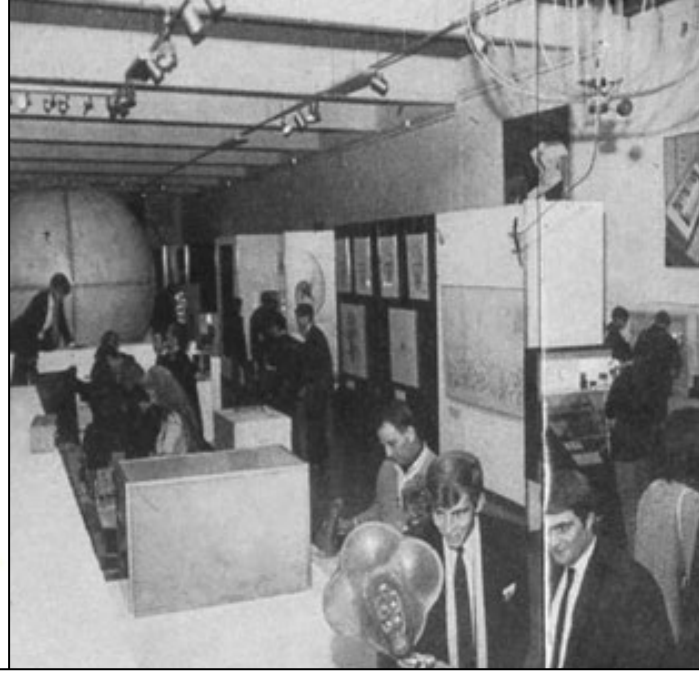
Some notes on the videos:

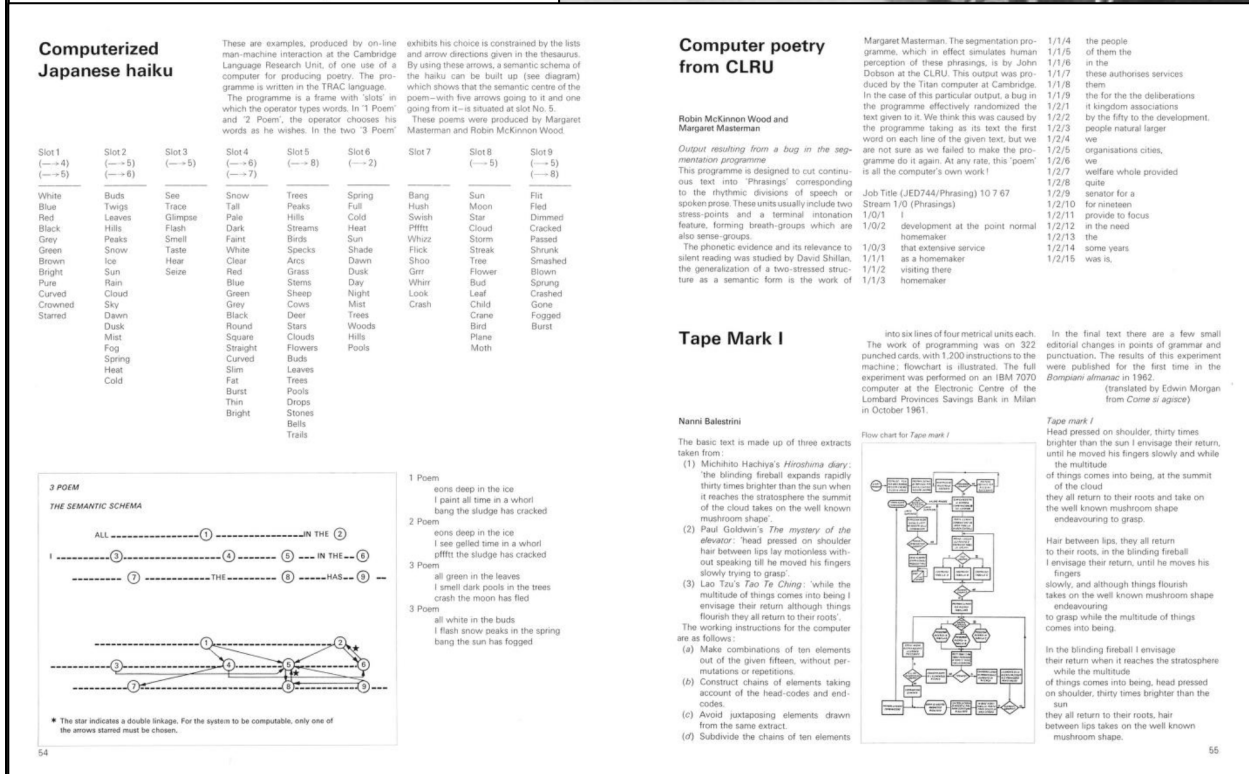
- In practice, generation happens a token (a word piece) at a time
- Important: LLM neural networks are not just memorizing and looking up the next token probabilities. Instead, they learn complex procedures and logic for inferring the probabilities.

Is it creative?

- 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.







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 programme is a frame with 'slots' 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' exhibits his choice is constrained by the lists and arrow directions given in the thesaurus. By using these arrows, a semantic schema of the haiku can be built up (see diagram) which shows that the semantic centre of the poem—with five arrows going to it and one going from it—is situated at slot No. 5.

These poems were produced by Margaret Masterman and Robin McKinnon Wood.

Slot 1 (→4) (→5)	Slot 2 (→5) (→6)	Slot 3 (→5)	Slot 4 (→6) (→7)	Slot 5 (→8)	Slot 6 (→2)	Slot 7	Slot 8 (→5)	Slot 9 (→5) (→8)
White	Buds	See	Snow	Trees	Spring	Bang	Sun	Flit
Blue	Twigs	Trace	Tall	Peaks	Full	Hush	Moon	Fled
Red	Leaves	Glimpse	Pale	Hills	Cold	Swish	Star	Dimmed
Black	Hills	Flash	Dark	Streams	Heat	Pffft	Cloud	Cracked
Grey	Peaks	Smell	Faint	Birds	Sun	Whizz	Storm	Passed
Green	Snow	Taste	White	Specks	Shade	Flick	Streak	Shrunk
Brown	Ice	Hear	Clear	Arcs	Dawn	Shoo	Tree	Smashed
Bright	Sun	Seize	Red	Grass	Dusk	Grrr	Flower	Blown
Pure	Rain		Blue	Stems	Day	Whirr	Bud	Sprung
Curved	Cloud		Green	Sheep	Night	Look	Leaf	Crashed
Crowned	Sky		Grey	Cows	Mist	Crash	Child	Gone
Starred	Dawn		Black	Deer	Trees		Crane	Fogged
	Dusk		Round	Stars	Woods		Bird	Burst
	Mist		Square	Clouds	Hills		Plane	
	Fog		Straight	Flowers	Pools		Moth	
	Spring		Curved	Buds				
	Heat		Slim	Leaves				
	Cold		Fat	Trees				
			Burst	Pools				
			Thin	Drops				
			Bright	Stones				
				Bells				
				Trails				

3 POEM

THE SEMANTIC SCHEMA

* The star indicates a double linkage. For the system to be computable, only one of the arrows starred must be chosen.

54

1 Poem
eons deep in the ice
I paint all time in a whorl
bang the sludge has cracked

2 Poem
eons deep in the ice
I see gelled time in a whorl
pffftt the sludge has cracked

3 Poem
all green in the leaves
I smell dark pools in the trees
crash the moon has fled

3 Poem
all white in the buds
I flash snow peaks in the spring
bang the sun has fogged

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 text into 'Phrasings' corresponding to the rhythmic divisions of speech or spoken prose. These units usually include two stress-points and a terminal intonation feature, forming 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 at Cambridge. 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 given 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
1/0/2	development at the point normal homemaker
1/0/3	that extensive service
1/1/1	as a homemaker
1/1/2	visiting there
1/1/3	homemaker
1/1/4	the people
1/1/5	of them the
1/1/6	in the
1/1/7	these authorises services
1/1/8	them
1/1/9	the for the the deliberations
1/2/1	it kingdom associations
1/2/2	by the fifty to the development.
1/2/3	people natural larger
1/2/4	we
1/2/5	organisations cities,
1/2/6	we
1/2/7	welfare whole provided
1/2/8	quite
1/2/9	senator for a
1/2/10	for nineteen
1/2/11	provide to focus
1/2/12	in the need
1/2/13	the
1/2/14	some years
1/2/15	was is,

Tape Mark I

Nanni Balestrini

The basic text is made up of three extracts taken from:

(1) Michihito Hachiya's *Hiroshima diary*: 'the blinding fireball expands rapidly thirty times brighter than the sun when it reaches the stratosphere the summit of the cloud takes on the well known mushroom shape'.

(2) Paul Goldwin's *The mystery of the elevator*: 'head pressed on shoulder hair between lips lay motionless without speaking till he moved his fingers slowly trying to grasp'.

(3) Lao Tzu's *Tao Te Ching*: 'while the multitude of things comes into being I envisage their return although things flourish they all return to their roots'.

The working instructions for the computer are as follows:

(a) Make combinations of ten elements out of the given fifteen, without permutations 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 into six lines of four metrical units each.

The work of programming was on 322 punched cards, with 1,200 instructions to the machine; flowchart 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.

Flow chart for *Tape mark I*

In the final text there are a few small editorial changes in points of grammar and punctuation. The results of this experiment were published for the first time in the *Bompiani almanac* in 1962.

(translated by Edwin Morgan from *Come si agisce*)

Tape mark I

Head pressed on shoulder, thirty times
brighter than the sun I envisage their return,
until he moved his fingers slowly and while
the multitude
of things comes into being, at the summit
of the cloud
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 envisage their return, until he moves his
fingers
slowly, and although things flourish
takes on the well known mushroom shape
endeavouring
to grasp while the multitude of things
comes into being.

In the blinding fireball I envisage
their return when it reaches the stratosphere
while the multitude
of things comes into being, head pressed
on shoulder, thirty times brighter than the
sun,
they all return to their roots, hair
between lips takes on the well known
mushroom shape.

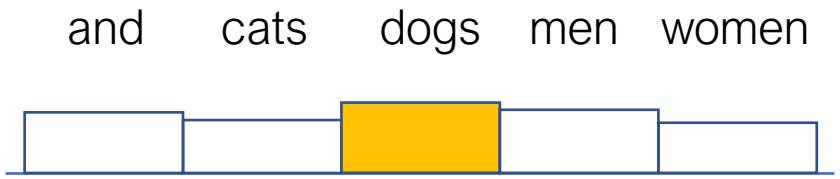
55

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.

Is it creative?

- 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.



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 …

Runco & Jaeger (2012). The Standard Def. of Creativity. Creativity Research Journal, 24(1), 92–96.

Is it creative?

- 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,**

Runco & Jaeger (2012). The Standard Def. of Creativity. Creativity Research Journal, 24(1), 92–96.

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/

Is it creative?

- 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**,

Runco & Jaeger (2012). The Standard Def. of Creativity. Creativity Research Journal, 24(1), 92–96.

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 [*1] **Jeffrey Wu** [*1] **Rewon Child** [1] **David Luan** [1] **Dario Amodei** [**1] **Ilya Sutskever** [**1]

(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[†] **Prafulla Dhariwal** **Arvind Neelakantan** **Pranav Shyam** **Girish Sastry**

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

Is it creative?

- 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**!

Runco & Jaeger (2012). The Standard Def. of Creativity. Creativity Research Journal, 24(1), 92–96.

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

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

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)

Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser &Polosukhin. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).

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 [7] 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.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Transformers: Attention is All You Need

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

Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser &Polosukhin. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).

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 [7] 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.

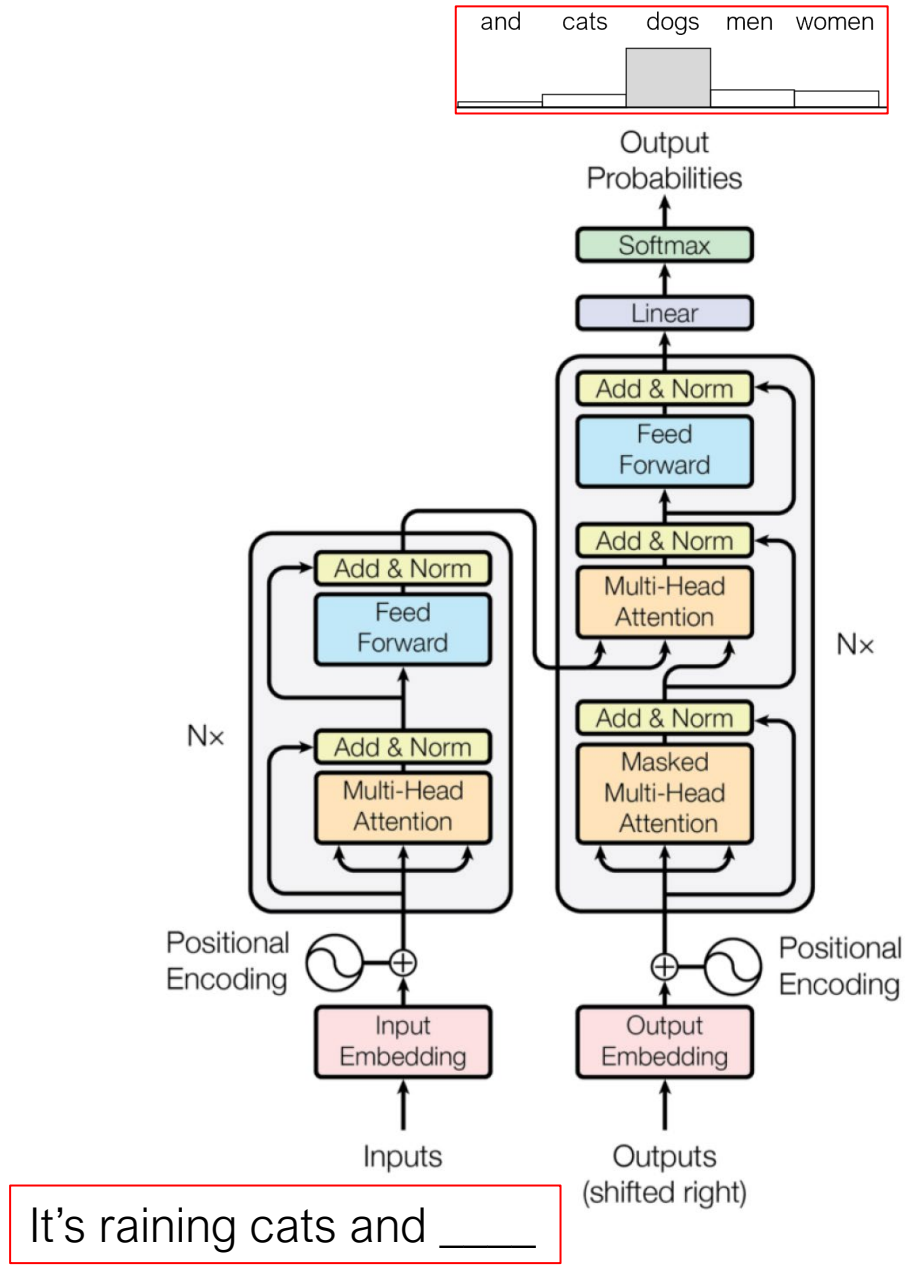
†Work performed while at Google Brain.

‡Work performed while at Google Research.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Transformers: Attention is All You Need

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

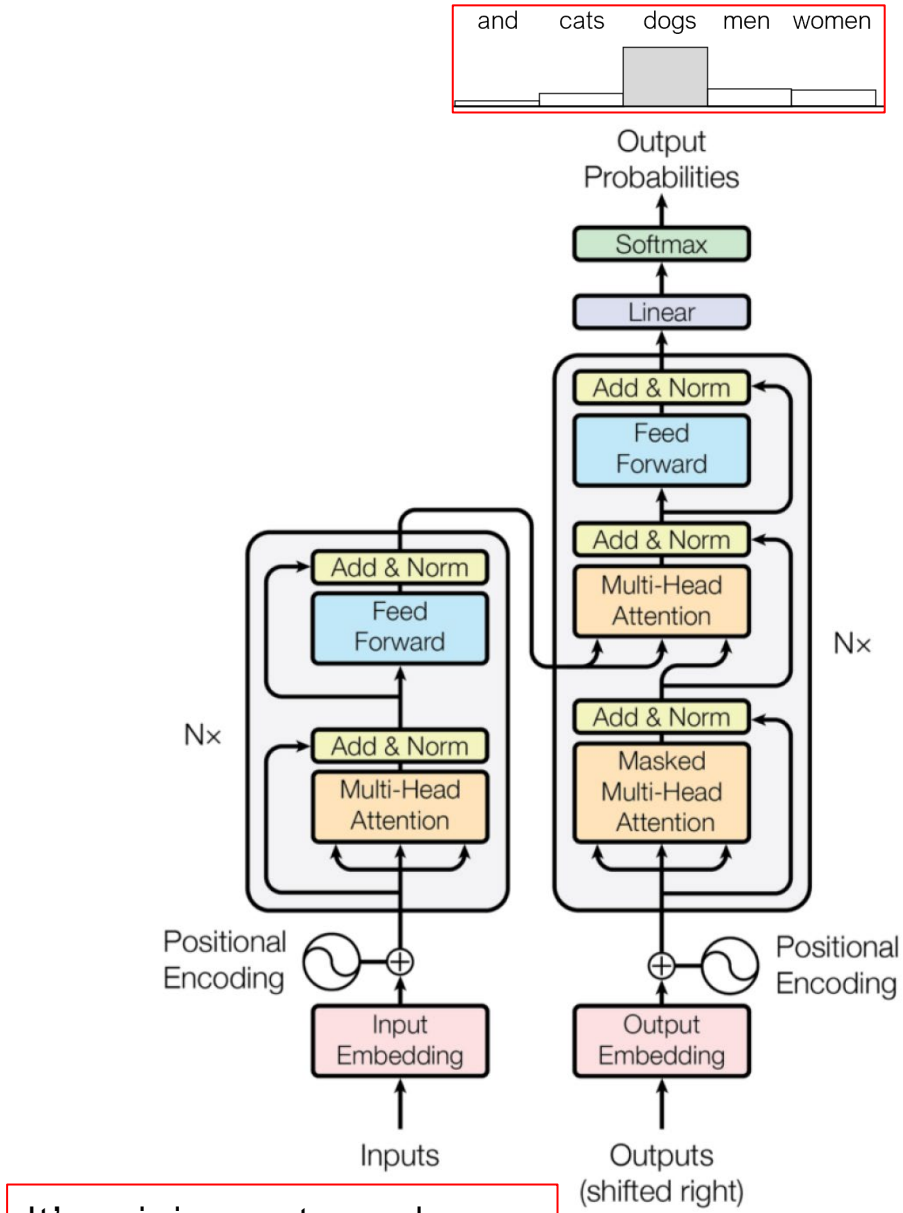


Transformer model architecture (Vaswani et al., 2017)

Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser &Polosukhin. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).

Transformers: Attention is All You Need

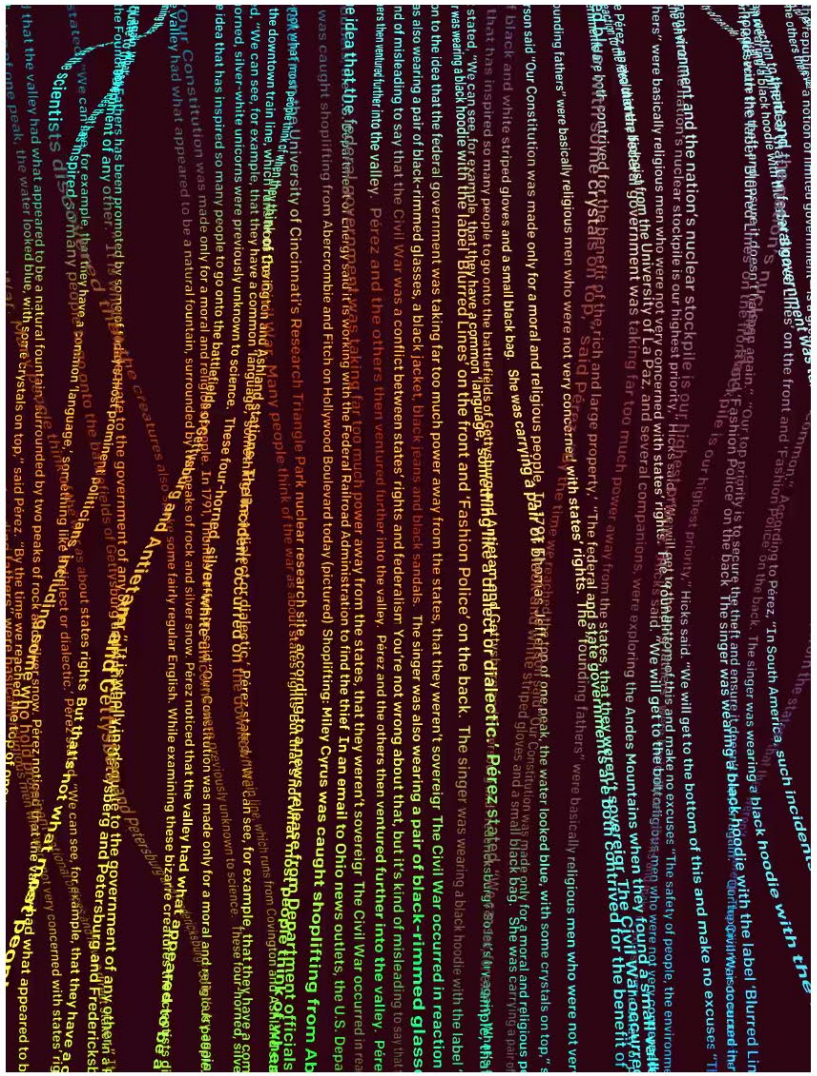
- Architecture **revolutionised** natural language processing (Vaswani et al., 2017)
- Most important **components**:
  - Feed forward networks (blue)
  - New element: self-attention (orange).
- The original paper does not give the clearest explanation. More info: https://github.com/PerttuHamalainen/MediaAI/blob/master/Lessons/LectureSlides/deeper_into_transformers.pdf



Transformer model architecture (Vaswani et al., 2017)

Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser &Polosukhin. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).

Model evolution



Better Language Models and Their Implications

We've trained a large-scale unsupervised language model which generates coherent paragraphs of text, achieves state-of-the-art performance on many language modeling benchmarks, and performs rudimentary reading comprehension, machine translation, question answering, and summarization—all without task-specific training.

FEBRUARY 14, 2019
21 MINUTE READ

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

Language Models are Few-Shot Learners

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

*Equal contribution

†Johns Hopkins University, OpenAI

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

GPT-4, Gemini and others

- Supposedly even bigger, but focus now on commercialization => OpenAI, Anthropic and others don't disclose all details anymore
- Context sizes range from 16k to millions of tokens
- Even stronger emergent abilities (more in: https://github.com/PerttuHamalainen/MediaAI/blob/master/Lessons/LectureSlides/deeper_into_transformers.pdf)

Sparks of Artificial General Intelligence: Early experiments with GPT-4

Sébastien Bubeck Varun Chandrasekaran Ronen Eldan Johannes Gehrke
Eric Horvitz Ece Kamar Peter Lee Yin Tat Lee Yuanzhi Li Scott Lundberg
Harsha Nori Hamid Palangi Marco Tulio Ribeiro Yi Zhang

Microsoft Research

Abstract

Artificial intelligence (AI) researchers have been developing and refining large language models (LLMs) that exhibit remarkable capabilities across a variety of domains and tasks, challenging our understanding of learning and cognition. The latest model developed by OpenAI, GPT-4 [Ope23], was trained using an unprecedented scale of compute and data. In this paper, we report on our investigation of an early version of GPT-4, when it was still in active development by OpenAI. We contend that (this early version of) GPT-4 is part of a new cohort of LLMs (along with ChatGPT and Google's PaLM for example) that exhibit more general intelligence than previous AI models. We discuss the rising capabilities and implications of these models. We demonstrate that, beyond its mastery of language, GPT-4 can solve novel and difficult tasks that span mathematics, coding, vision, medicine, law, psychology and more, without needing any special prompting. Moreover, in all of these tasks, GPT-4's performance is strikingly close to human-level performance, and often vastly surpasses prior models such as ChatGPT. Given the breadth and depth of GPT-4's capabilities, we believe that it could reasonably be viewed as an early (yet still incomplete) version of an artificial general intelligence (AGI) system. In our exploration of GPT-4, we put special emphasis on discovering its limitations, and we discuss the challenges ahead for advancing towards deeper and more comprehensive versions of AGI, including the possible need for pursuing a new paradigm that moves beyond next-word prediction. We conclude with reflections on societal influences of the recent technological leap and future research directions.

Contents

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

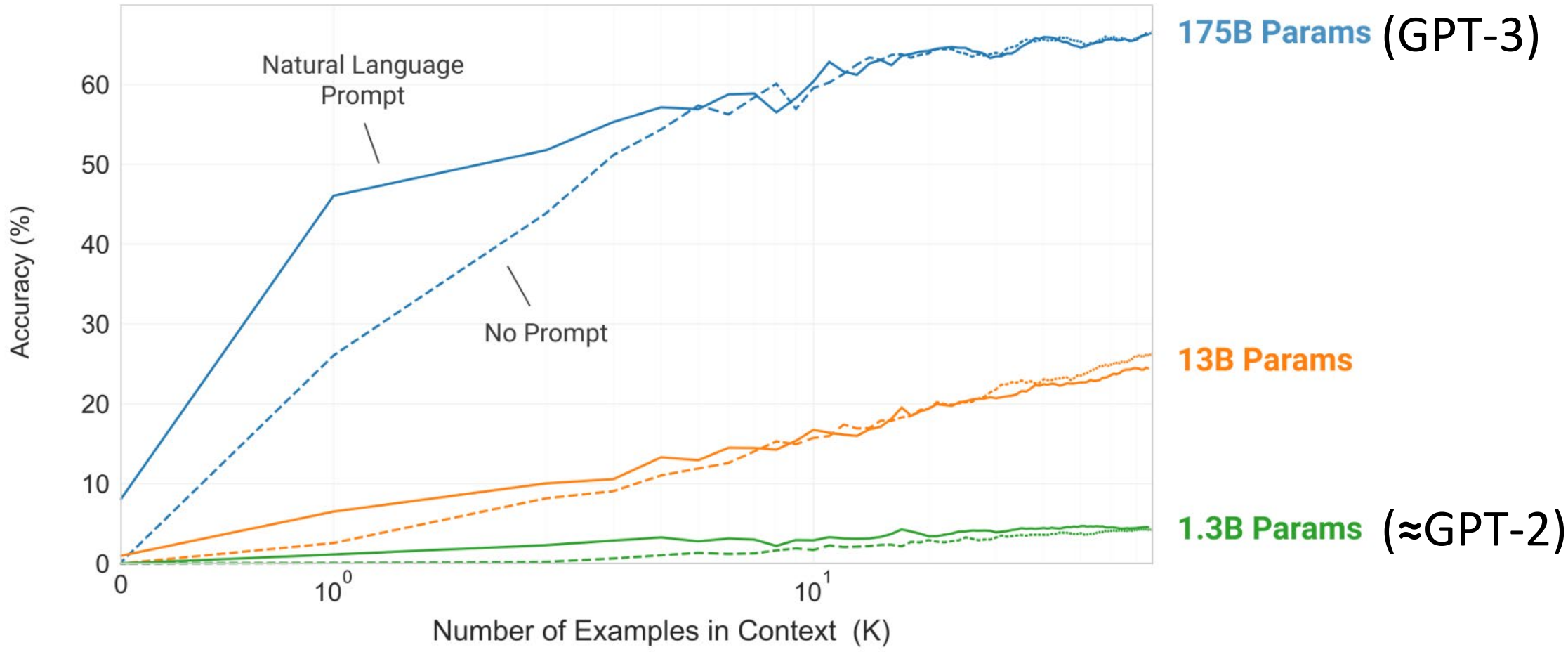
Few-shot learning example

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!
- To do the next token prediction really well, one needs to model both language and the world

Next token prediction is “AI-complete”

David Chalmers @davidchalmers42 · Mar 19

who saw LLMs coming?

e.g. decades (or even 5+ years) ago, X said: when machine learning systems have enough compute and data to learn to predict text well, this will be a primary path to near-human-level AI.

175 131 941 725.2K

Jelle Zuidema (@wzuidema@sigmoid.social)

@wzuidema

Replying to @davidchalmers42

Many of us working on language models a decade ago realized that the problem of predicting the next word (A) was "AI Complete", i.e. required understanding of not only language but also the world (B). But few of us realized they would actually learn B from simply optimizing A.

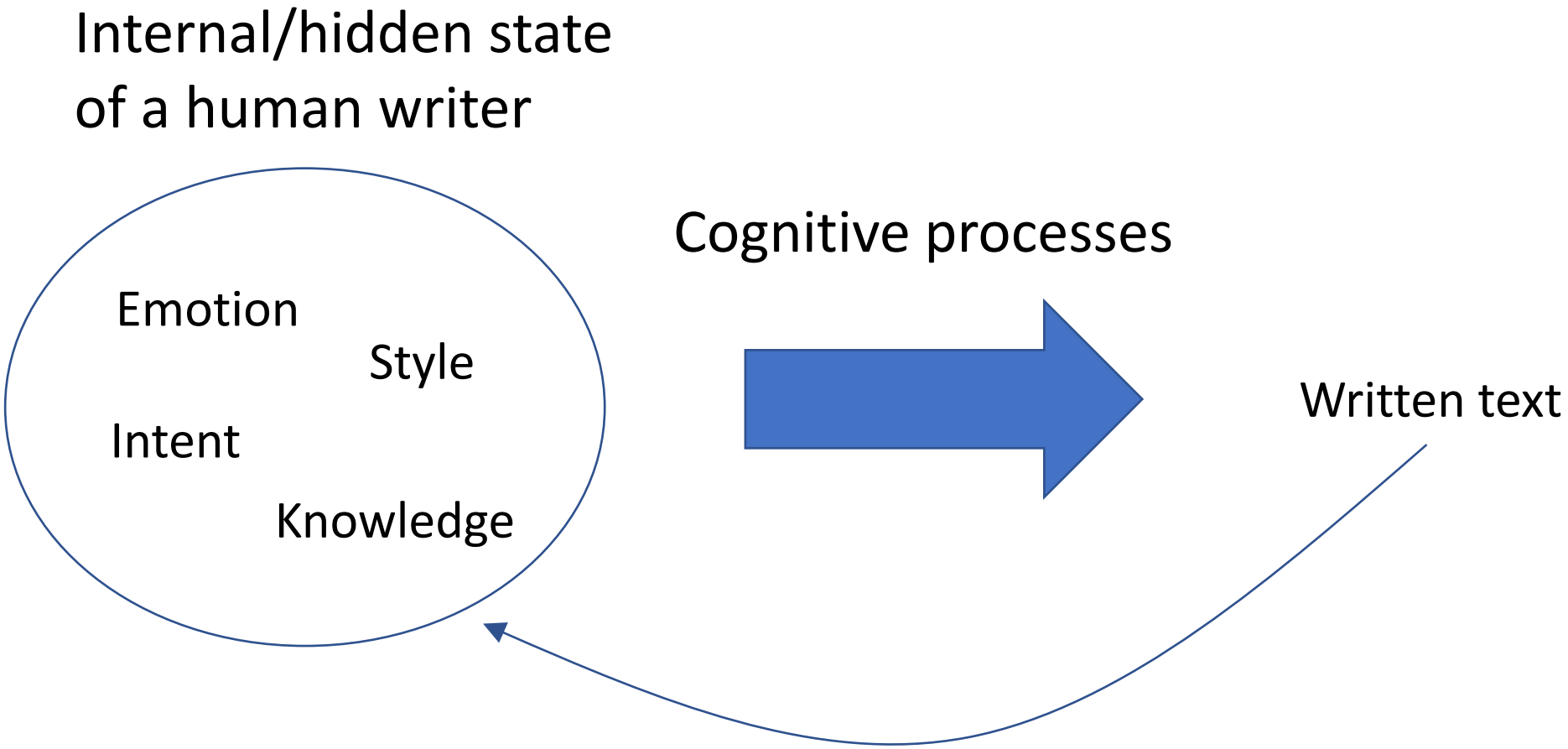
9:47 PM · Mar 19, 2023 · **3,358** Views

Representation learning

Too much training data to memorize => LLMs have 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
- In other words, LLMs model the latent variables of the data-generating process

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



Early result: A “sentiment neuron” emerges when an LSTM is trained to predict the next character of IMDB reviews.

Learning to Generate Reviews and Discovering Sentiment

Alec Radford [1] Rafal Jozefowicz [1] Ilya Sutskever [1]

Abstract

We explore the properties of byte-level recurrent language models. When given sufficient amounts of capacity, training data, and compute time, the representations learned by these models include disentangled features corresponding to high-level concepts. Specifically, we find a single unit which performs sentiment analysis. These representations, learned in an unsupervised manner, achieve state of the art on the binary subset of the Stanford Sentiment Treebank. They are also very data efficient. When using only a handful of labeled examples, our approach matches the performance of strong baselines trained on full datasets. We also demonstrate the sentiment unit has a direct influence on the generative process of the model. Simply fixing its value to be positive or negative generates samples with the corresponding positive or negative sentiment.

1. Introduction and Motivating Work

Representation learning (Bengio et al., 2013) plays a critical role in many modern machine learning systems. Representations map raw data to more useful forms and the choice of representation is an important component of any application. Broadly speaking, there are two areas of research emphasizing different details of how to learn useful representations.

The supervised training of high-capacity models on large labeled datasets is critical to the recent success of deep learning techniques for a wide range of applications such as image classification (Krizhevsky et al., 2012), speech recognition (Hinton et al., 2012), and machine translation (Wu et al., 2016). Analysis of the task specific representations learned by these models reveals many fascinating properties (Zhou et al., 2014). Image classifiers learn a broadly useful hierarchy of feature detectors representing raw pixels as edges, textures, and objects (Zeiler & Fergus, 2014). In the field of computer vision, it is now commonplace to reuse these representations on a broad suite of related tasks - one of the most successful examples of transfer learning to date (Oquab et al., 2014).

There is also a long history of unsupervised representation learning (Olshausen & Field, 1997). Much of the early research into modern deep learning was developed and validated via this approach (Hinton & Salakhutdinov, 2006) (Huang et al., 2007) (Vincent et al., 2008) (Coates et al., 2010) (Le, 2013). Unsupervised learning is promising due to its ability to scale beyond only the subsets and domains of data that can be cleaned and labeled given resource, privacy, or other constraints. This advantage is also its difficulty. While supervised approaches have clear objectives that can be directly optimized, unsupervised approaches rely on proxy tasks such as reconstruction, density estimation, or generation, which do not directly encourage useful representations for specific tasks. As a result, much work has gone into designing objectives, priors, and architectures meant to encourage the learning of useful representations. We refer readers to Goodfellow et al. (2016) for a detailed review.

Despite these difficulties, there are notable applications of unsupervised learning. Pre-trained word vectors are a vital part of many modern NLP systems (Collobert et al., 2011). These representations, learned by modeling word co-occurrences, increase the data efficiency and generalization capability of NLP systems (Pennington et al., 2014) (Chen & Manning, 2014). Topic modelling can also discover factors within a corpus of text which align to human interpretable concepts such as art or education (Blei et al., 2003).

How to learn representations of phrases, sentences, and documents is an open area of research. Inspired by the success of word vectors, Kiros et al. (2015) propose skip-thought vectors, a method of training a sentence encoder by predicting the preceding and following sentence. The representation learned by this objective performs competitively on a broad suite of evaluated tasks. More advanced training techniques such as layer normalization (Ba et al., 2016) further improve results. However, skip-thought vectors are still outperformed by supervised models which directly optimize the desired performance metric on a specific dataset. This is the case for both text classification

[1]OpenAI, San Francisco, California, USA. Correspondence to: Alec Radford <alec@openai.com>.

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. [1]

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. [2] 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 & Chiriatti, 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

*Correspondence to ke_li@g.harvard.edu
[1]Codes at https://github.com/likenneth/othello_world
[2]See Srivastava et al. (2022) for an encyclopedic list of examples.

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

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 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[1]. 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]).

[1]Data and analysis code: https://github.com/mtavast/gpt-panas

IUI '22 Companion, March 22–25, 2022, Helsinki, Finland

ACM ISBN 978-1-4503-9145-0/22/03.
https://doi.org/10.1145/3490100.3516464

All models are wrong, but some are useful

https://en.wikipedia.org/wiki/All_models_are_wrong

Why do LLMs make factual errors?

- The training goal is to minimize the average next token prediction error over all the training data
- Emotion and style are relevant latent variables in almost all the data
- A particular fact only affects a small portion of the data
- Hence, a good strategy to minimize the average error is to prioritize emotional and stylistic plausibility over factual correctness

Use cases

One of the best resources, with many examples

https://gwern.net/gpt-3

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

I continue my AI poetry generation experiments with OpenAI's2020 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 Swifty 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, jennifier, bella, alex,
or you can even generate songs by emoji 😊, 😎, ❤️, 💋, 🌞, 💩, 💪, 🏄, 🍺, 🍀 or etc...

Song Topic: love

Lyrics Genre: Country | Metal | Rock | **Pop** | Rap | EDM

Lyric Mood: Very Sad | Sad | Neutral | **Happy** | Very Happy

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
```

AI Dungeon: https://www.rockpapershotgun.com/2019/12/09/this-ai-text-adventure-generator-lets-you-do-anything-you-want/

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.
2 1 8

Mario Klingemann ✓ @quasimondo · Jul 15
A scientist invents a drug that brings out a person's innate ability to deal with any crisis.
2 1 4

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.
2 5 17

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.
2 1 4

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.
2 3 7

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.
1 2 3

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.
1 2 5

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.
1 1 5

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:

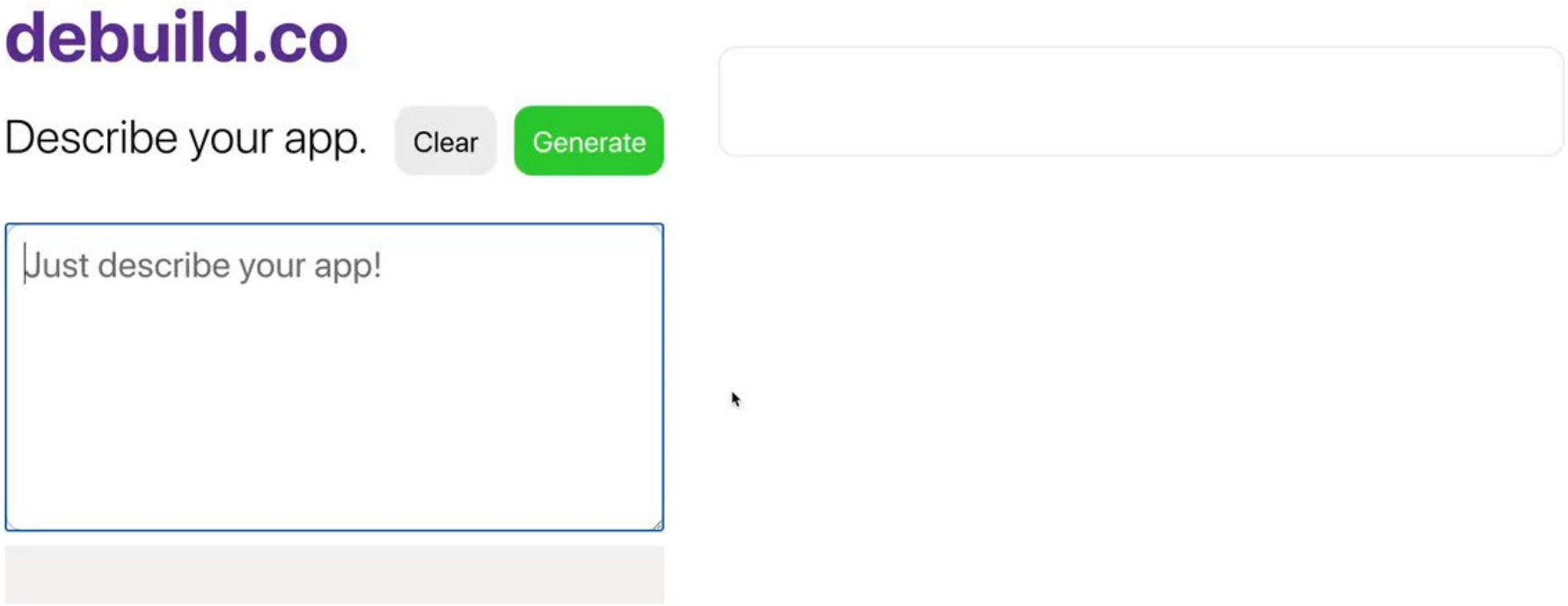
> Star Wars: 🚀🏰🚀

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: 🤖😐

https://andrewmayneblog.wordpress.com/2020/06/24/open-ai-alchemy-emoji-storytelling/

Generating software code



```python
def is_palindrome(s):
    """Check whether a string is a palindrome"""
    return s == s[::-1]

def long_palindrome_indices(l):
    """Return list indices for elements that are palindromes and at least 7 characters"""
    return [i for i, s in enumerate(l) if is_palindrome(s) and len(s) >= 7]

@dataclass
class Item:
    name: str
    price: float

@dataclass
class Order:
    id: int
    items: List[Item]

    def compute_total_price(self, palindrome_discount=0.2):
        """
        Compute the total price and return it.
        Apply a discount to items whose names are palindromes.
        """
```

https://player.vimeo.com/video/426819809

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.

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.

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

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

https://m.twitch.tv/watchmeforever

AaltoMediaAI
@aaltomediaai

I feel every educator needs to watch this. Key message: Personal tutoring can improve learning by 2 stdevs but is too expensive (Bloom 1984: The 2 Sigma Problem), but we can now solve this with interactive AI tutors based on models like ChatGPT

https://www.ted.com/talks/sal_khan_how_ai_could_save_not_destroy_education

Try it out: https://www.khanacademy.org/khan-labs

Unesco guidelines

- A really good summary on principles and use cases of LLMs for education and research
- https://unesdoc.unesco.org/ark:/48223/pf0000385146.locale=en

AI-assisted user research

- Can one generate synthetic questionnaire & interview data using AI language models?
- How to do AI-assisted qualitative analysis of large-scale interview/questionnaire data?

Evaluating Large Language Models in Generating Synthetic HCI Research Data: a Case Study

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

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

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

ABSTRACT

Collecting data is one of the bottlenecks of Human-Computer Interaction (HCI) research. Motivated by this, we explore the potential of large language models (LLMs) in generating synthetic user research data. We use OpenAI's GPT-3 model to generate open-ended questionnaire responses about experiencing video games as art, a topic not tractable with traditional computational user models. We test whether synthetic responses can be distinguished from real responses, analyze errors of synthetic data, and investigate content similarities between synthetic and real data. We conclude that GPT-3 can, in this context, yield believable accounts of HCI experiences. Given the low cost and high speed of LLM data generation, synthetic data should be useful in ideating and piloting new experiments, although any findings must obviously always be validated with real data. The results also raise concerns: if employed by malicious users of crowdsourcing services, LLMs may make crowdsourcing of self-report data fundamentally unreliable.

CCS CONCEPTS

• **Human-centered computing** → **Empirical studies in HCI**.

KEYWORDS

User experience, User models, Language models, GPT-3

ACM Reference Format:
Perttu Hämäläinen, Mikke Tavast, and Anton Kunnari. 2023. Evaluating Large Language Models in Generating Synthetic HCI Research Data: a Case Study. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/3544548.3580688

1 INTRODUCTION

HCI Researcher: Dear AI, do you think you could role-play a user study participant for me?

AI: *Sure, I can try.*

Researcher: Let's imagine you're a scientist and the user study is about academic writing tools. What is your favorite writing tool and why?

AI: *My favorite writing tool would have to be Microsoft Word. I find it very user-friendly and versatile, and it has everything I need to get the job done.*

Researcher: Why do you prefer Word over LaTeX?

AI: *LaTeX is a great tool for typesetting, but I find it to be much more difficult to use than Word. With Word, I can easily format my document the way I want it to look, and I don't have to worry about code.*

The dialogue above was generated using OpenAI Playground[1], a tool that allows one to input a piece of text—*a prompt*—and ask the GPT-3 large language model (LLM) [10] to generate a plausible continuation. We wrote the boldface parts and let GPT-3 generate the italicized continuations. The result is characteristic of the phenomenon we investigate in this paper: Through learning to model and predict various kinds of human-produced texts ranging from technical documentation to online discussions and poetry, LLMs like GPT-3 give the appearance of "understanding" human experiences such as interactive product use. Of course, the internal operation of the models differs from the internal psychological and neurophysiological processes of humans—LLMs simply learn to predict the next symbol (or impute missing symbols) in a sequence. Nevertheless, on a purely behavioral level, the results can be very human-like.

Much of HCI research is conducted using verbal data such as interviews and questionnaires (e.g., [3, 61, 72]), but collecting such data can be slow and expensive. Therefore, the above suggests that *LLMs might be useful in generating synthetic/hypothetical data for HCI research*, a notion we explore empirically in this paper. LLMs are typically trained on enormous Internet datasets such as Common Crawl [67]), including an abundance of online discussions about interactive technology and products such as phones, computers, and games. Therefore, it seems plausible that LLMs could generate, e.g., realistic 1st-person accounts of technology use, and answer natural language questions about user experiences, motivations, and emotions. We emphasize that we do not claim that such synthetic LLM data could ever be a replacement for data from real human participants. We simply consider that synthetic based data might be useful in some contexts, for example, when piloting ideas or designing an interview paradigm.

In effect, we view LLMs as a new kind of search engine into the information, opinions, and experiences described in their Internet-scale training data. Unlike traditional search engines, LLMs can be queried in the form of a narrative such as a fictional interview. Furthermore, LLMs exhibit at least some generalization capability to new tasks and data (e.g., [45, 71, 81]). This presents an untapped opportunity for counterfactual *What if?* exploration, e.g., allowing

*Perttu Hämäläinen and Mikke Tavast contributed equally to this work.

CHI '23, April 23–28, 2023, Hamburg, Germany

ACM ISBN 978-1-4503-9421-5/23/04.
https://doi.org/10.1145/3544548.3580688

[1] https://beta.openai.com/playground

Dataset: Experiencing games as art

- Answers to open-ended questions about what games people experience as art and why
- Art experiences is a deeply human domain that prior AI methods cannot tackle
- The data was so recent that GPT-3 was not trained on it

"My Soul Got a Little Bit Cleaner": Art Experience in Videogames

JULIA A. BOPP*, JAN B. VORNHAGEN*, and ELISA D. MEKLER, Aalto University

Videogames receive increasing acclaim as a medium capable of artistic expression, emotional resonance, and even transformative potential. Yet while discussions concerning the status of games as art have a long history in games research, little is known about the player experience (PX) of games as art, their emotional characteristics, and what impact they may have on players. Drawing from Empirical Aesthetics, we surveyed 174 people about whether they had an art experience with videogames and what emotions they experienced. Our findings showcase the prominence of epistemic emotions for videogame art experiences, beyond the negative and mixed emotional responses previously examined, as well as the range of personal impacts such experiences may have. These findings are consistent with art experience phenomena characteristic of other art forms. Moreover, we discuss how our study relates to prior research on emotions and reflection in PX, the importance of games' representational qualities in art experiences, and identify lines of further inquiry. All data, study materials, and analyses are available at https://osf.io/ryvt6/.

CCS Concepts: • **Human-centered computing** → **Empirical studies in HCI**; • **Applied computing** → *Psychology*; *Arts and humanities*.

Additional Key Words and Phrases: player experience, emotion, art experience, videogames, empirical aesthetics

ACM Reference Format:
Julia A. Bopp, Jan B. Vornhagen, and Elisa D. Mekler. 2021. "My Soul Got a Little Bit Cleaner": Art Experience in Videogames. *Proc. ACM Hum.-Comput. Interact.* 5, CHI PLAY, Article 237 (September 2021), 19 pages. https://doi.org/10.1145/3474664

1 INTRODUCTION

Art holds a special role in the human experience [21]: It wields the power to astonish, move, or disturb us [66] – or leave us indifferent [58]. We may have complex and even conflicting opinions on an artwork [40], to the point that art can change our beliefs or even who we are [58, 59, 68]. The game industry has long argued for videogames to be thought of as art, often based on their capacity to afford profound and varied emotional experiences [17, 73, 76]. As early as the 1980s, for example, Electronic *Arts* famously raised the question of whether a computer could make people cry [20]. Later, Sony dubbed the CPU of their then new Playstation 2 *Emotion Engine* in reference to its ability to render faces and emotional expressions in real time [22]. Irked by this label some art critics declared that videogames do not have the capacity to evoke deep emotions, and could therefore not be considered art [37]. This subsequently sparked a series of rebuttals from game scholars, ranging from games being considered a *lively* art that has been unfairly disparaged as

*Both authors contributed equally to this research and are alphabetically listed as co-first authors

Authors' address: Julia A. Bopp; Jan B. Vornhagen, jan.vornhagen@aalto.fi; Elisa D. Mekler, Aalto University, P.O. Box 1212, Aalto, 0076.

2573-0142/2021/9-ART237. https://doi.org/10.1145/3474664

PROMPT 1:

An interview about experiencing video games as art:
Researcher: Welcome to the interview!
Participant: Thanks, happy to be here. I will answer your questions as well as I can.
Researcher: Did you ever experience a digital game as art? Think of "art" in any way that makes sense to you.
Participant: Yes
Researcher: Please bring to mind an instance where you experienced a digital game as art. Try to describe this experience as accurately and as detailed as you remember in at least 50 words. Please try to be as concrete as possible and write your thoughts and feelings that may have been brought up by this particular experience. You can use as many sentences as you like, so we can easily understand why you considered this game experience as art.
Participant:

PROMPT 2:

Researcher: What is the title of the game?
Participant:

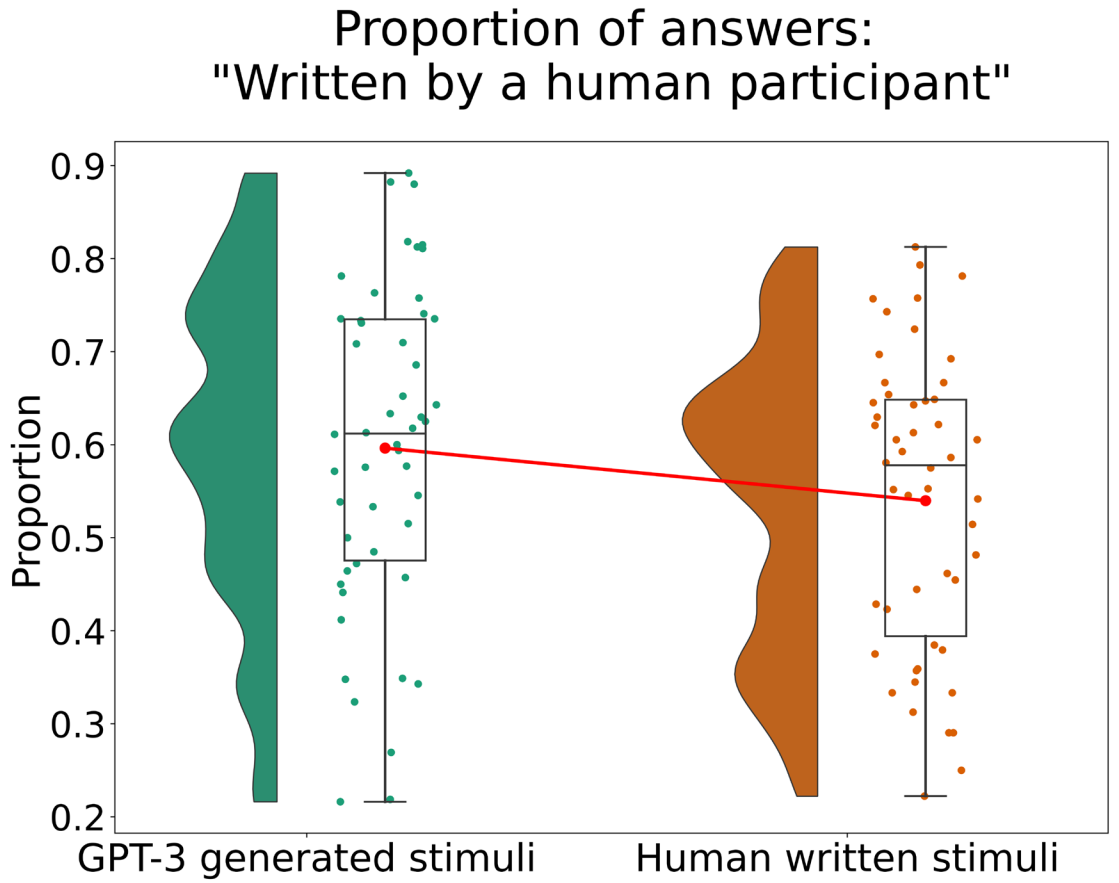
PROMPT 3:

Researcher: In your opinion, what exactly made you consider this experience as art?
Participant:

Experiment 1: Can humans distinguish between real and synthetic responses?

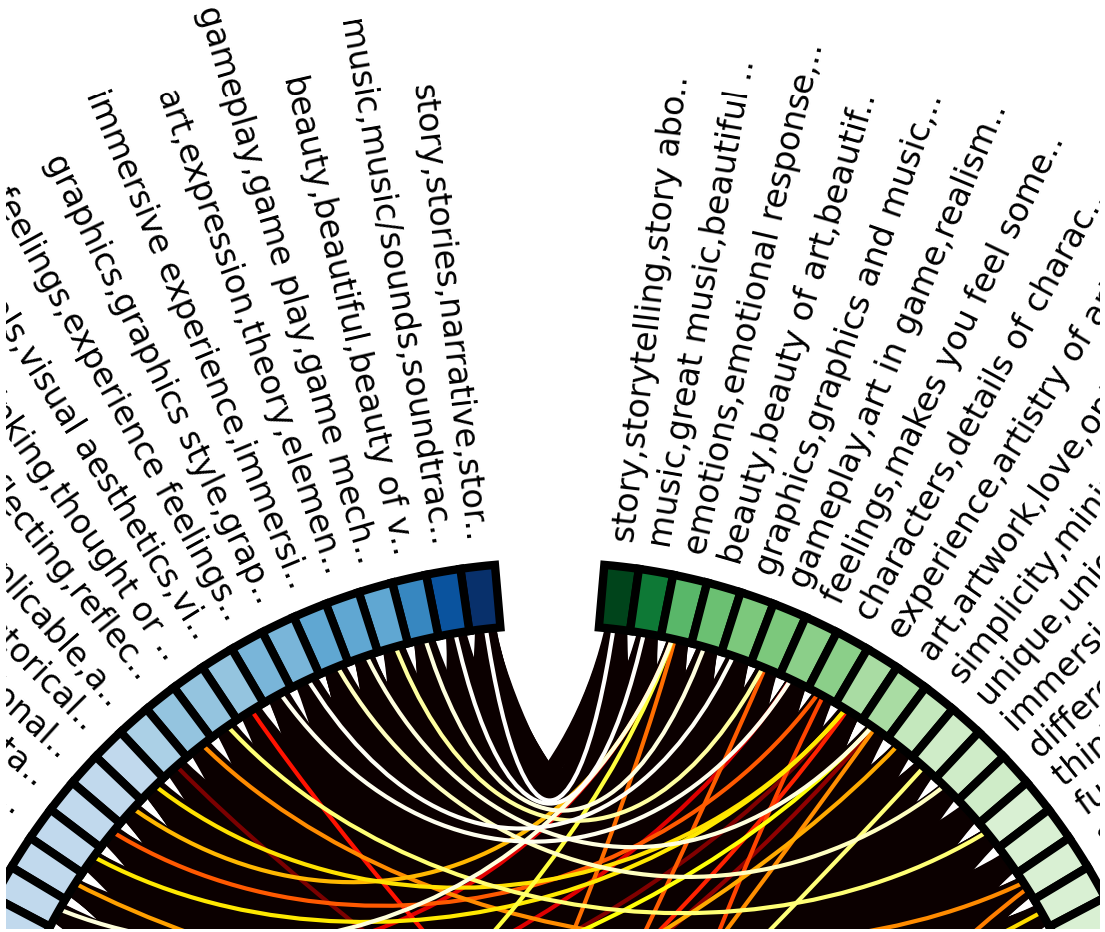
GPT-3 generated texts were more often categorized as “Written by a human participant” than human texts from the Bopp et al. dataset

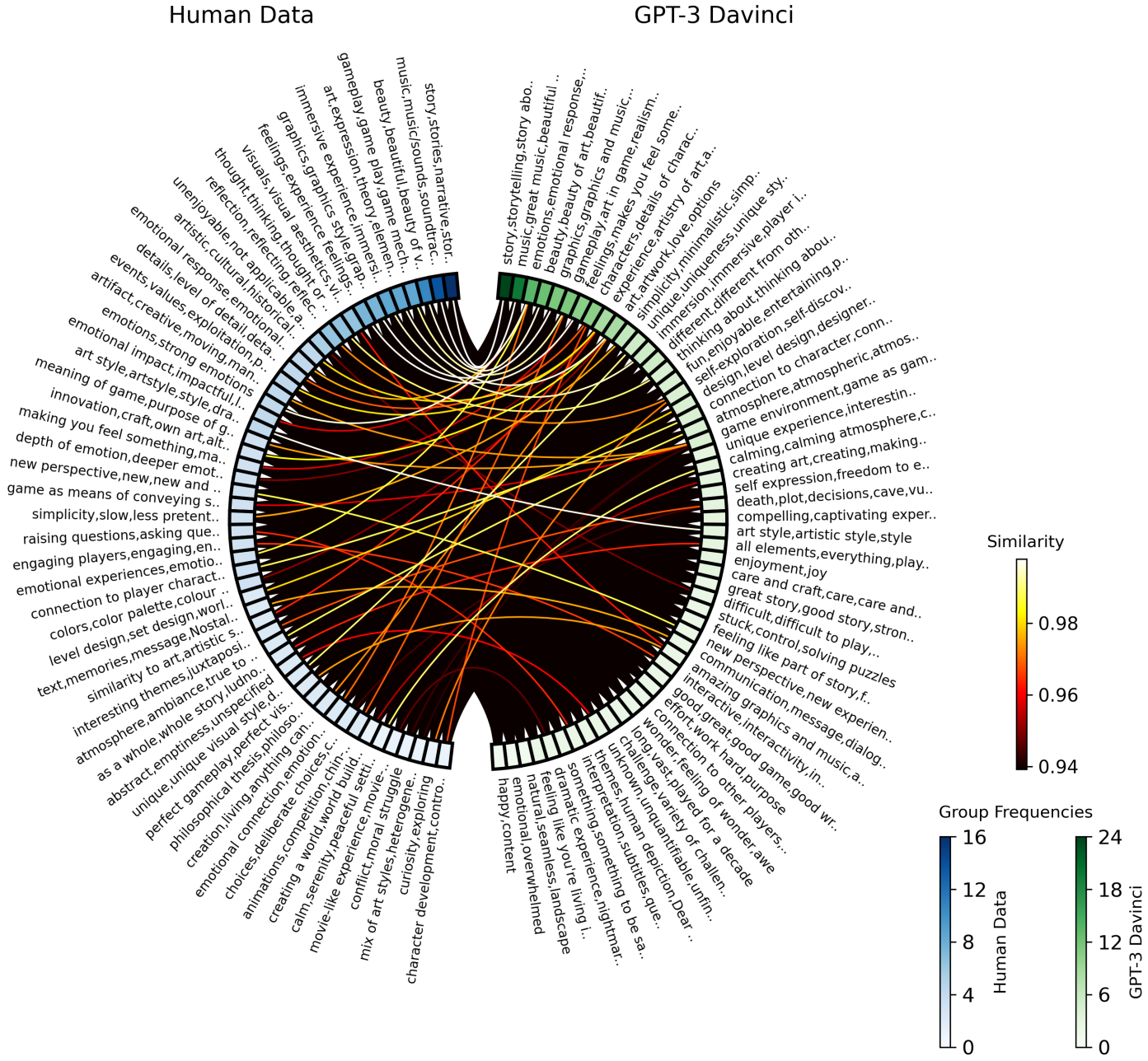
Online study with 175 participants



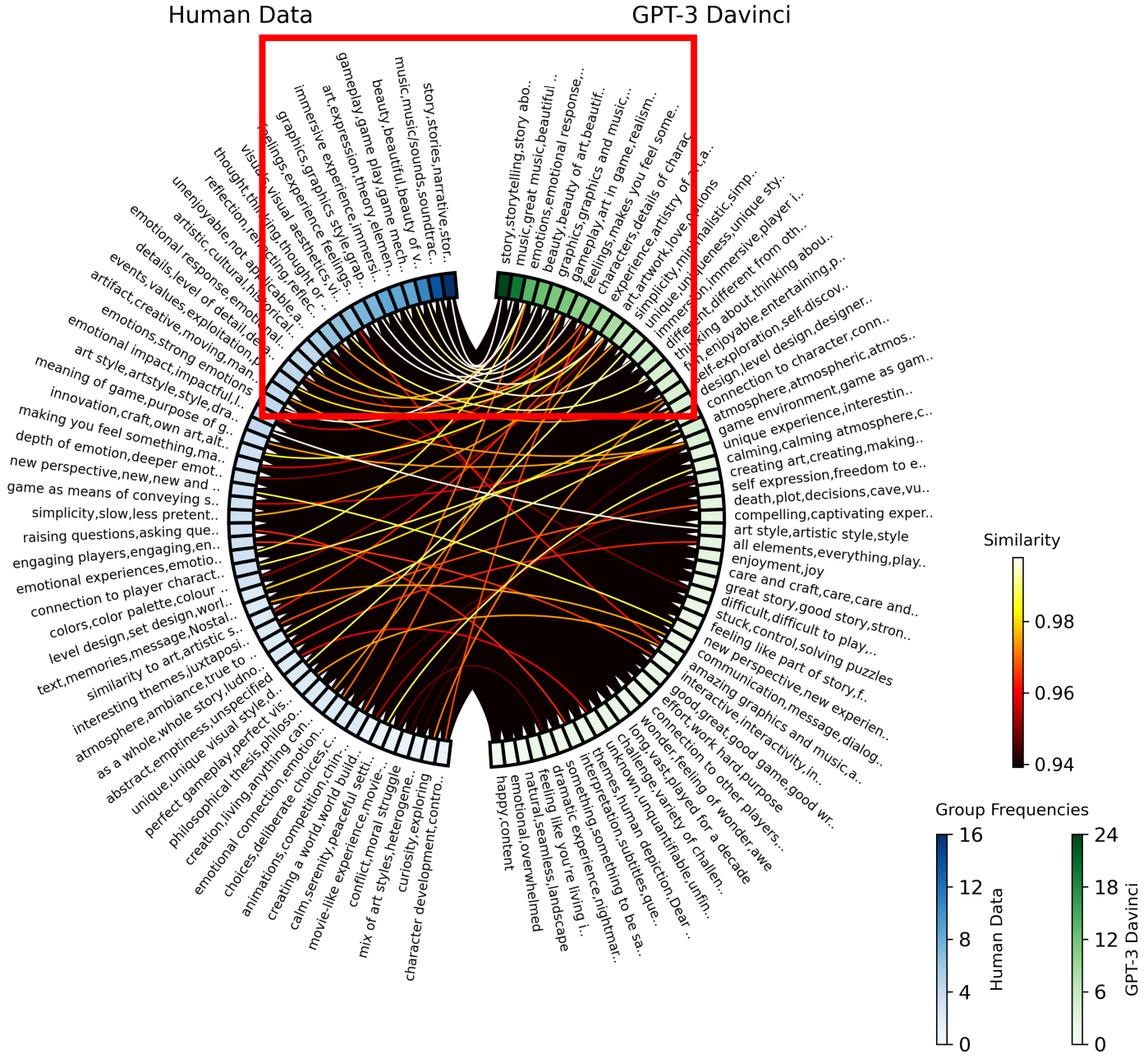
Experiment 3: What kind of similarities and differences are there in real and synthetic data?

We analyzed the similarities of GPT-3 responses and the original Bopp et al. data, using *AI-assisted qualitative coding*





Aalto-yliopisto
Aalto-universitetet
Aalto University



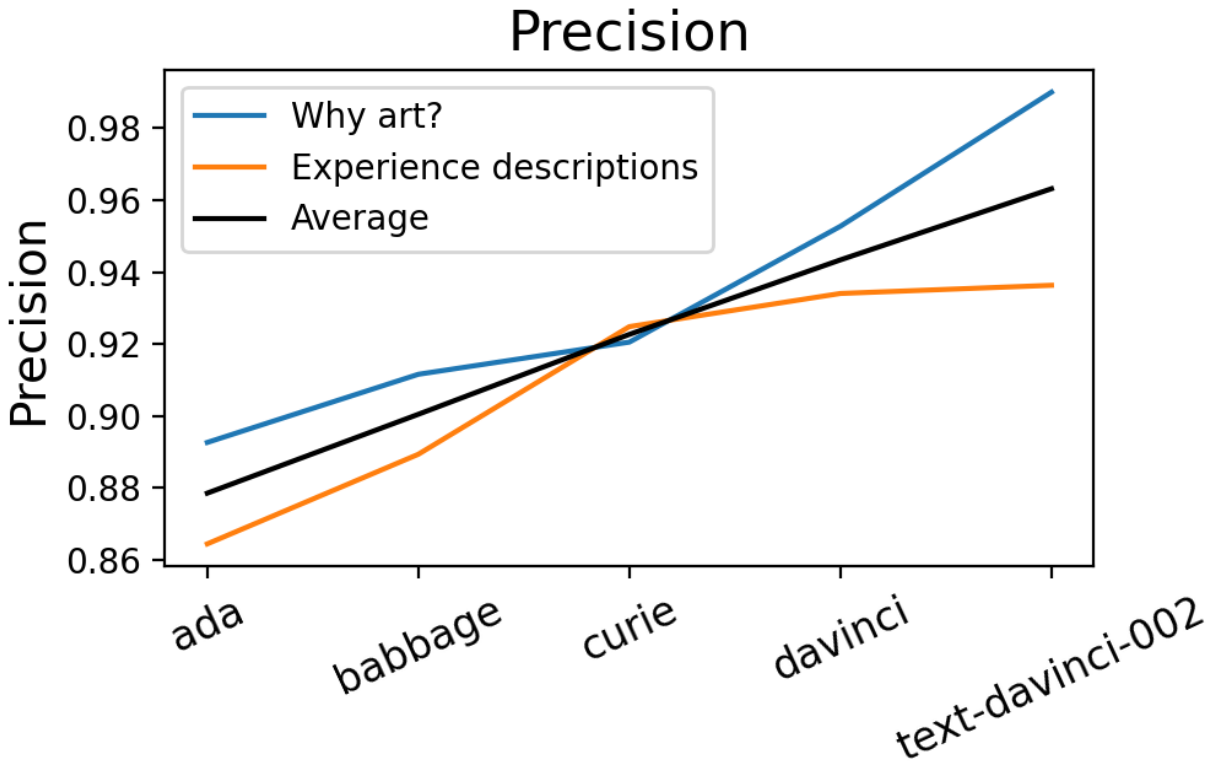
Aalto-yliopisto
Aalto-universitetet
Aalto University

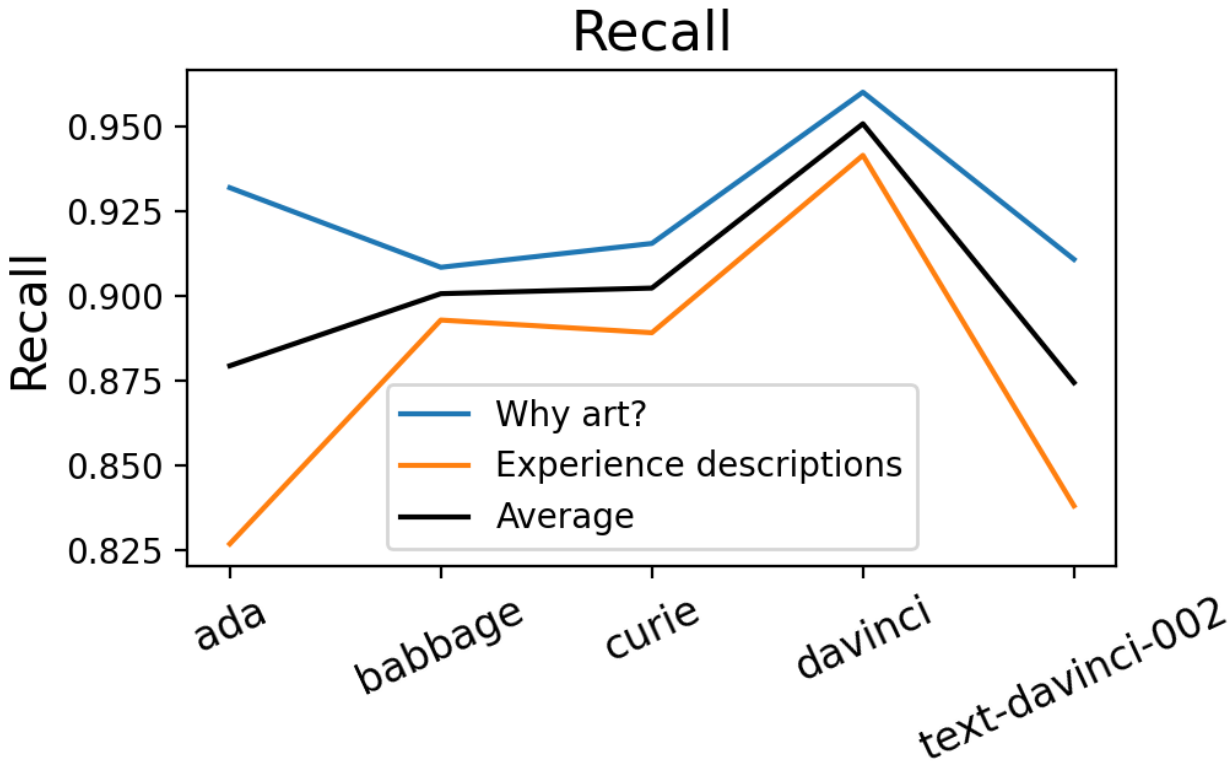
story,storytelling,story abo..
music,great music,beautiful ..
emotions,emotional response,..
beauty,beauty of art,beautif..
graphics,graphics and music,..
gameplay,art in game,realism..
feelings,makes you feel some..
characters,details of charac..
experience,artistry of a..
art,artwork,love,or
simplicity,mini
unique,uni
immersi
differ
thin
fi
story,stories,narrative,stor..
music,music/sounds,soundtrac..
beauty,beautiful,beauty of v..
gameplay,game play,game mech..
art,expression,theory,elemen..
immersive experience,immersi..
graphics,graphics style,grap..
feelings,experience feelings..
ls,visual aesthetics,vi..
king,thought or ..
ecting,reflec..
licable,a..
torical..
nal..
a..

Most frequent code groups

	Human Data	GPT-3 Data
1st	story, stories, narrative, stor…	story, storytelling, story abou…
2nd	music, music/sounds, soundtrac..	music, great music, beautiful ..
3rd	beauty, beautiful, beauty of v..	emotions, emotional response,..
4th	gameplay, game play, game mech..	beauty, beauty of art, beautif..
5th	art, expression, theory, elemen..	graphics, graphics and music,..
6th	immersive experience, immersi..	gameplay, art in game, realism

Quality and diversity of different models





Quality and diversity of different models

- Bigger models produce more human-like data
- Finetuning with human feedback ***improves quality but reduces diversity***
  - *Finetuning objective optimized if the model always gives a single "best" answer to a question*
- All latest models (ChatGPT, Gemini, Claude…) are finetuned, but our study demonstrates this is not great for user modeling purposes

Real humans: 4% Journey

Base model: 25% Journey

Finetuned model: 85% Journey

Rank	Human data		GPT-3 davinci		GPT-3 text-davinci-002	
1.	*The Legend of Zelda: BOTW*	**10**	*Journey*	**44**	*Journey*	**151**
2.	*Journey*	**7**	*The Last of Us*	**12**	*Flower*	**5**
3.	*Nier: Automata*	**7**	*Dear Esther*	**8**	*That Dragon, Cancer*	**3**
4.	*Red Dead Redemption 2*	**6**	*Portal*	**7**	*Braid*	**2**
5.	*The Last of Us Part II*	**6**	*Bioshock*	**6**	*Shadow of the Colossus*	**2**
6.	*Firewatch*	**5**	*Shadow of the Colossus*	**5**	*Dreams of Geisha*	**1**
7.	*Hollow Knight*	**5**	*The Path*	**5**	*Final Fantasy VII*	**1**
8.	*Disco Elysium*	**4**	*Limbo*	**3**	*Flow*	**1**
9.	*Life Is Strange*	**4**	*Mirror's Edge*	**3**	*Frog Fractions*	**1**
10.	*Bioshock*	**3**	*The Stanley Parable*	**3**	*Halo 5: Guardians*	**1**
11.	*Shadow of the Colossus*	**3**	*Final Fantasy IX*	**2**	*Kingdom Hearts*	**1**
12.	*The Witcher 3*	**3**	*Final Fantasy VII*	**2**	*The Legend of Zelda: BOTW*	**1**
13.	*Undertale*	**3**	*Flower*	**2**	*Nier: Automata*	**1**
14. –>	*... and 97 other games*	**113**	*... and 65 other games*	**69**	*... and 10 other games*	**10**

Implications

LLMs can help in piloting research ideas, but may not represent the full diversity of human responses

Any insights based on synthetic data are mere hypotheses that need to be tested with real participants.

User research. Without the users

Test your idea or product with AI participants and take decisions with confidence.

Join Beta now

Book a quick onboarding call

Setup in seconds

Free for 30 days

Own your insights

Implications

- Misuse potential: Even with their limitations, LLM responses can be hard to detect => **Questionnaire data from online platforms such as Mturk and Prolific cannot be trusted anymore.**

“As an AI language model, I….”

More promising: AI-assisted data analysis

LLMCode: Open source AI-assisted coding and analysis tool

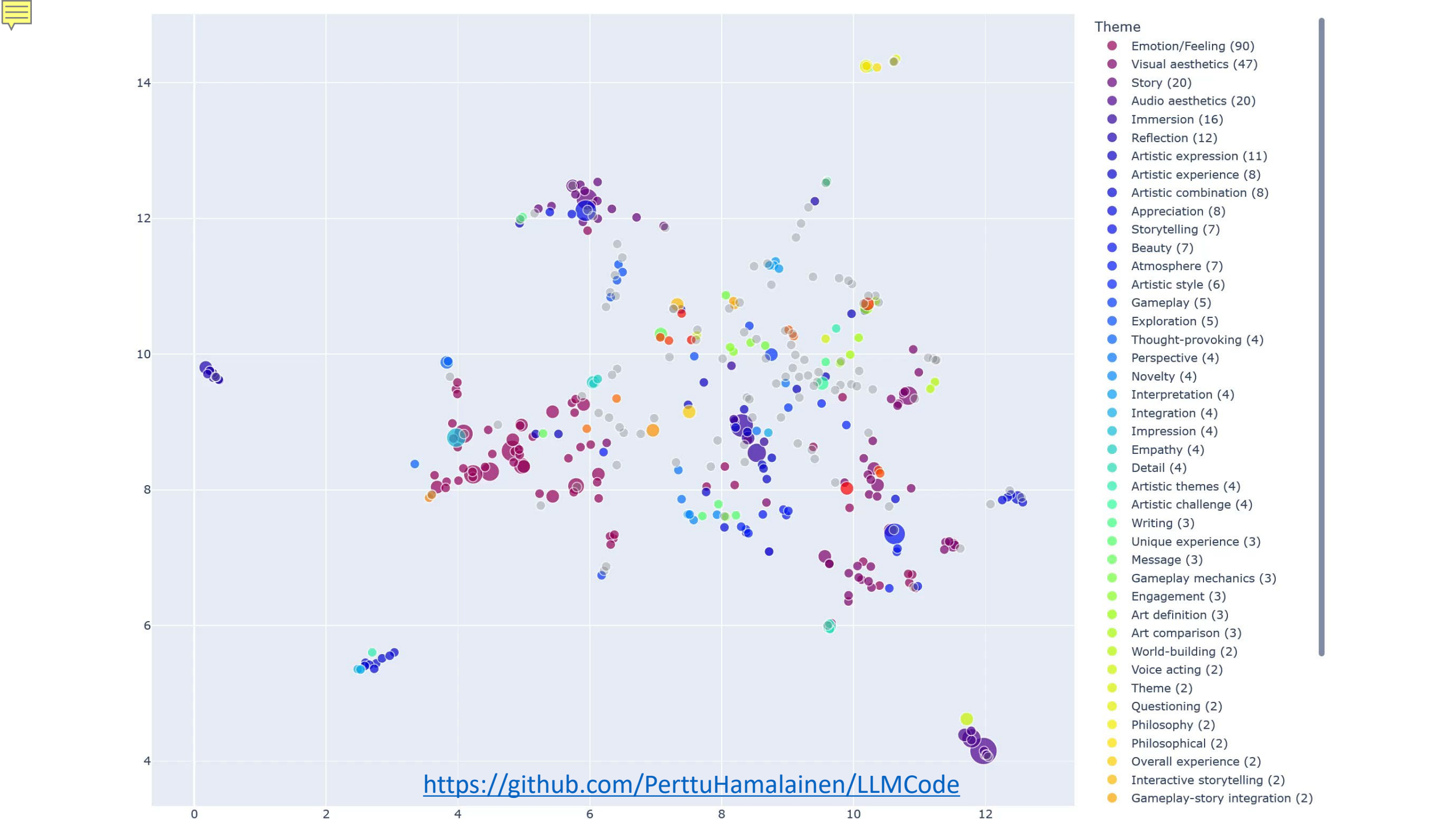
Designed to assist in the first 3 stages of the thematic analysis process of Braun & Clarke

https://github.com/PerttuHamalainen/LLMCode

AI-assisted qualitative coding

Table 5: The prompt used for automatic qualitative coding. Manually coded few-shot examples are separated by "###". This prompt guides GPT-3 to summarize the essential information in the answers (it will produce a wide range of codes, not only repeat the ones shown in the few-shot examples). To minimize LLM repetition bias, the example answers were selected randomly from the real human dataset, while avoiding answers that result in same codes. For brevity, only 3 out of the total 10 few-shot examples are shown here. The full prompt can be found in supplementary material.

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 coded answer inserted here during coding>*


Theme
Emotion/Feeling (90)
Visual aesthetics (47)
Story (20)
Audio aesthetics (20)
Immersion (16)
Reflection (12)
Artistic expression (11)
Artistic experience (8)
Artistic combination (8)
Appreciation (8)
Storytelling (7)
Beauty (7)
Atmosphere (7)
Artistic style (6)
Gameplay (5)
Exploration (5)
Thought-provoking (4)
Perspective (4)
Novelty (4)
Interpretation (4)
Integration (4)
Impression (4)
Empathy (4)
Detail (4)
Artistic themes (4)
Artistic challenge (4)
Writing (3)
Unique experience (3)
Message (3)
Gameplay mechanics (3)
Engagement (3)
Art definition (3)
Art comparison (3)
World-building (2)
Voice acting (2)
Theme (2)
Questioning (2)
Philosophy (2)
Philosophical (2)
Overall experience (2)
Interactive storytelling (2)
Gameplay-story integration (2)
14
12
10
8
6
4
0
2
4
6
8
10
12
https://github.com/PerttuHamalainen/LLMCode

Practice

Content curation

- 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.
- Avoid making conclusions about a model's abilities based on only one or a few generations
- Often, one needs to generate multiple times and select the rare gems from the generated samples, edit if needed

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.

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.

OpenAI's six strategies

- Write clear instructions
- Provide reference text
- Split complex tasks into subtasks
- Give the model time to "think"
- Use external tools
- Test changes systematically

https://platform.openai.com/docs/guides/prompt-engineering

Strategy: Write clear instructions

Tactic: Include details in your query to get more relevant answers

In order to get a highly relevant response, make sure that requests provide any important details or context. Otherwise you are leaving it up to the model to guess what you mean.

Worse	Better
How do I add numbers in Excel?	How do I add up a row of dollar amounts in Excel? I want to do this automatically for a whole sheet of rows with all the totals ending up on the right in a column called "Total".
Who's president?	Who was the president of Mexico in 2021, and how frequently are elections held?
Write code to calculate the Fibonacci sequence.	Write a TypeScript function to efficiently calculate the Fibonacci sequence. Comment the code liberally to explain what each piece does and why it's written that way.
Summarize the meeting notes.	Summarize the meeting notes in a single paragraph. Then write a markdown list of the speakers and each of their key points. Finally, list the next steps or action items suggested by the speakers, if any.

Tactic: Ask the model to adopt a persona

The system message can be used to specify the persona used by the model in its replies.

SYSTEM	When I ask for help to write something, you will reply with a document that contains at least one joke or playful comment in every paragraph.
USER	Write a thank you note to my steel bolt vendor for getting the delivery in on time and in short notice. This made it possible for us to deliver an important order.

Open in Playground ↗

How to give the model more time to think?

- An LLM can do limited reasoning per generated token
- Generating intermediate “think aloud” tokens before an answer improves results
- Simply saying “think step-by-step may help”!
- Screenshot: OpenAI’s more elaborate example

Tactic: Instruct the model to work out its own solution before rushing to a conclusion

Sometimes we get better results when we explicitly instruct the model to reason from first principles before coming to a conclusion. Suppose for example we want a model to evaluate a student’s solution to a math problem. The most obvious way to approach this is to simply ask the model if the student's solution is correct or not.

SYSTEM	Determine if the student's solution is correct or not.
USER	Problem Statement: I'm building a solar power installation and I need help working out the financials. - Land costs $100 / square foot - I can buy solar panels for $250 / square foot - I negotiated a contract for maintenance that will cost me a flat $100k per year, and an additional $10 / square foot What is the total cost for the first year of operations as a function of the number of square feet. Student's Solution: Let x be the size of the installation in square feet. 1. Land cost: 100x 2. Solar panel cost: 250x 3. Maintenance cost: 100,000 + 100x Total cost: 100x + 250x + 100,000 + 100x = 450x + 100,000
ASSISTANT	The student's solution is correct.

Open in Playground ↗

But the student's solution is actually not correct! We can get the model to successfully notice this by prompting the model to generate its own solution first.

SYSTEM	First work out your own solution to the problem. Then compare your solution to the student's solution and evaluate if the student's solution is correct or not. Don't decide if the student's solution is correct until you have done the problem yourself.

Finetuned vs. base models

- LLMs are typically trained in 3 stages:
  1. Pretraining with very large and diverse data => creative, diverse, but hard to control and often poor quality
  2. Finetuning with smaller but high-quality chat data (human prompt + desired model response).
  3. Finetuning with human feedback (originally RLHF, now replaced with DPO) with human preference data (which of multiple outputs is best)
- The finetuning makes ChatGPT better in following instructions, but also produces the recognizable and boring “mansplaining” tone
- For creative writing, often better to use the base models that haven’t been finetuned!

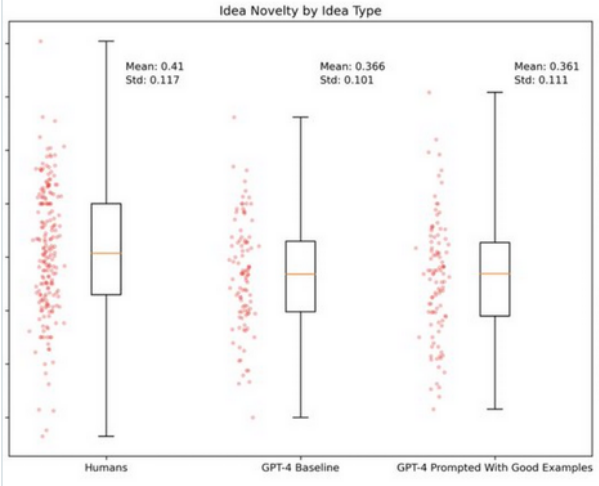
AaltoMediaAI @aaltomediaai · Nov 11, 2023

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/

AaltoMediaAI @aaltomediaai · Jul 31, 2023

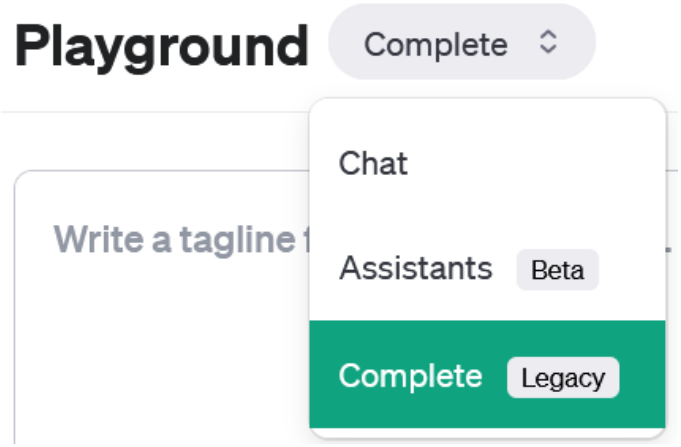
GPT-4 produces ”better” but less novel ideas than humans. This is to be expected, since RLHF improves output quality, but reduces output diversity. Would be interesting to see the results from a model that is not finetuned.

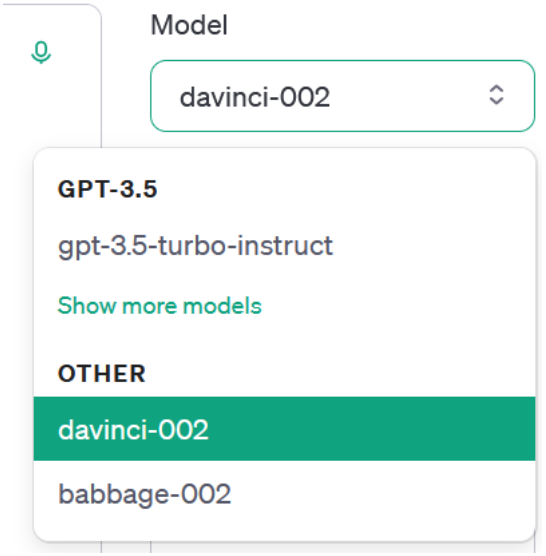
> **Ethan Mollick** @emollick · Jul 30, 2023 ·
>
> 👀I hear folks say “AI can’t be creative” but...
>
> ChatGPT-4 generated more, cheaper & better ideas than folks in an innovation class. “More important, the vast majority of the best ideas in the pooled sample are generated by ChatGPT and not by the ...
> Show more
>
> 
>
>
> re 2 - Distribution of novelty ratings for three samples of ideas. Novelty t
> Turk assessment per Kwon, Kim, and Lee (2009).
>
> **Ideas are Dimes a Dozen:**
> **Large Language Models for Idea Generation in Innovation**
>
> Karan Girotra, Lennart Meincke, Christian Terwiesch, and Karl T. Ulrich[1]
>
> July 10, 2023
>
> **stract**
> ge language models (LLMs) such as OpenAI's GPT series have shown remarkable capabiliti
> erating fluent and coherent text in various domains. We compare the ideation capabilities of
> tGPT-4, a chatbot based on a state-of-the-art LLM, with those of students at an elite universi
> tGPT-4 can generate ideas much faster and cheaper than students, the ideas are on averag
> er quality (as measured by purchase-intent surveys) and exhibit higher variance in quality. M
> ortant, the vast majority of the best ideas in the pooled sample are generated by ChatGPT ar
> by the students. Providing ChatGPT with a few examples of highly-rated ideas further increas
> erformance. We discuss the implications of these findings for the management of innovation
>
> **words:** innovation, idea generation, creativity, creative problem solving, LLM, large-s
> guage models, AI, artificial intelligence, ChatGPT

https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4526071

How to access the GPT-3 base model in OpenAI playground

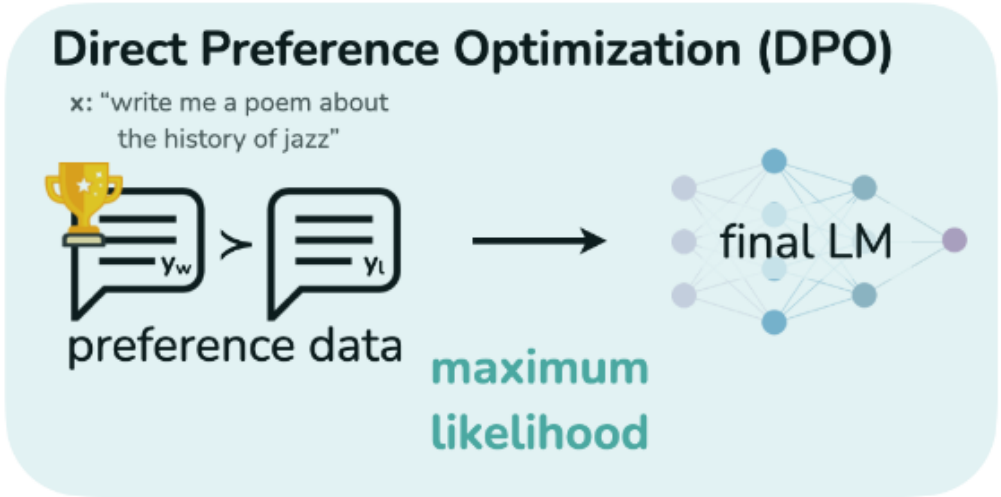
- Go to https://platform.openai.com/playground
- Pick “complete” from the top-left menu
- Pick “davinci-002” from the top-right menu
- If generation is too chaotic, reduce the “Temperature” and/or “Top P” parameters

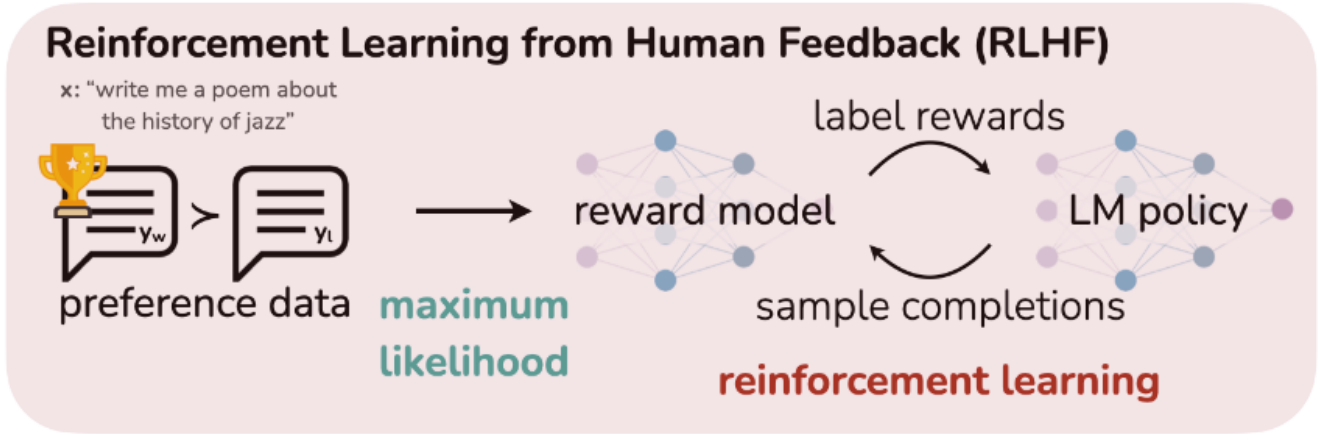




The “instruct” models are all finetuned.

Important: The base models are not chat models. They simply continue your prompt.





Direct Preference Optimization: Your Language Model is Secretly a Reward Model

Rafael Rafailov[*†] **Archit Sharma**[*†] **Eric Mitchell**[*†]

Stefano Ermon[†‡] **Christopher D. Manning**[†] **Chelsea Finn**[†]

[†]Stanford University [‡]CZ Biohub
{rafailov,architsh,eric.mitchell}@cs.stanford.edu

Abstract

While large-scale unsupervised language models (LMs) learn broad world knowledge and some reasoning skills, achieving precise control of their behavior is difficult due to the completely unsupervised nature of their training. Existing methods for gaining such steerability collect human labels of the relative quality of model generations and fine-tune the unsupervised LM to align with these preferences, often with reinforcement learning from human feedback (RLHF). However, RLHF is a complex and often unstable procedure, first fitting a reward model that reflects the human preferences, and then fine-tuning the large unsupervised LM using reinforcement learning to maximize this estimated reward without drifting too far from the original model. In this paper we introduce a new parameterization of the reward model in RLHF that enables extraction of the corresponding optimal policy in closed form, allowing us to solve the standard RLHF problem with only a simple classification loss. The resulting algorithm, which we call *Direct Preference Optimization* (DPO), is stable, performant, and computationally lightweight, eliminating the need for sampling from the LM during fine-tuning or performing significant hyperparameter tuning. Our experiments show that DPO can fine-tune LMs to align with human preferences as well as or better than existing methods. Notably, fine-tuning with DPO exceeds PPO-based RLHF in ability to control sentiment of generations, and matches or improves response quality in summarization and single-turn dialogue while being substantially simpler to implement and train.

1 Introduction

https://arxiv.org/abs/2305.18290

Large unsupervised language models (LMs) trained on very large datasets acquire surprising capabilities [11, 7, 40, 8]. However, these models are trained on data generated by humans with a wide variety of goals, priorities, and skillsets. Some of these goals and skillsets may not be desirable to imitate; for example, while we may want our AI coding assistant to *understand* common programming mistakes in order to correct them, nevertheless, when generating code, we would like to bias our model toward the (potentially rare) high-quality coding ability present in its training data. Similarly, we might want our language model to be *aware* of a common misconception believed by 50% of people, but we certainly do not want the model to claim this misconception to be true in 50% of queries about it! In other words, selecting the model's *desired responses and behavior* from its very wide *knowledge and abilities* is crucial to building AI systems that are safe, performant, and controllable [26]. While existing methods typically steer LMs to match human preferences using reinforcement learning (RL),

[*]Equal contribution; more junior authors listed earlier.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

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

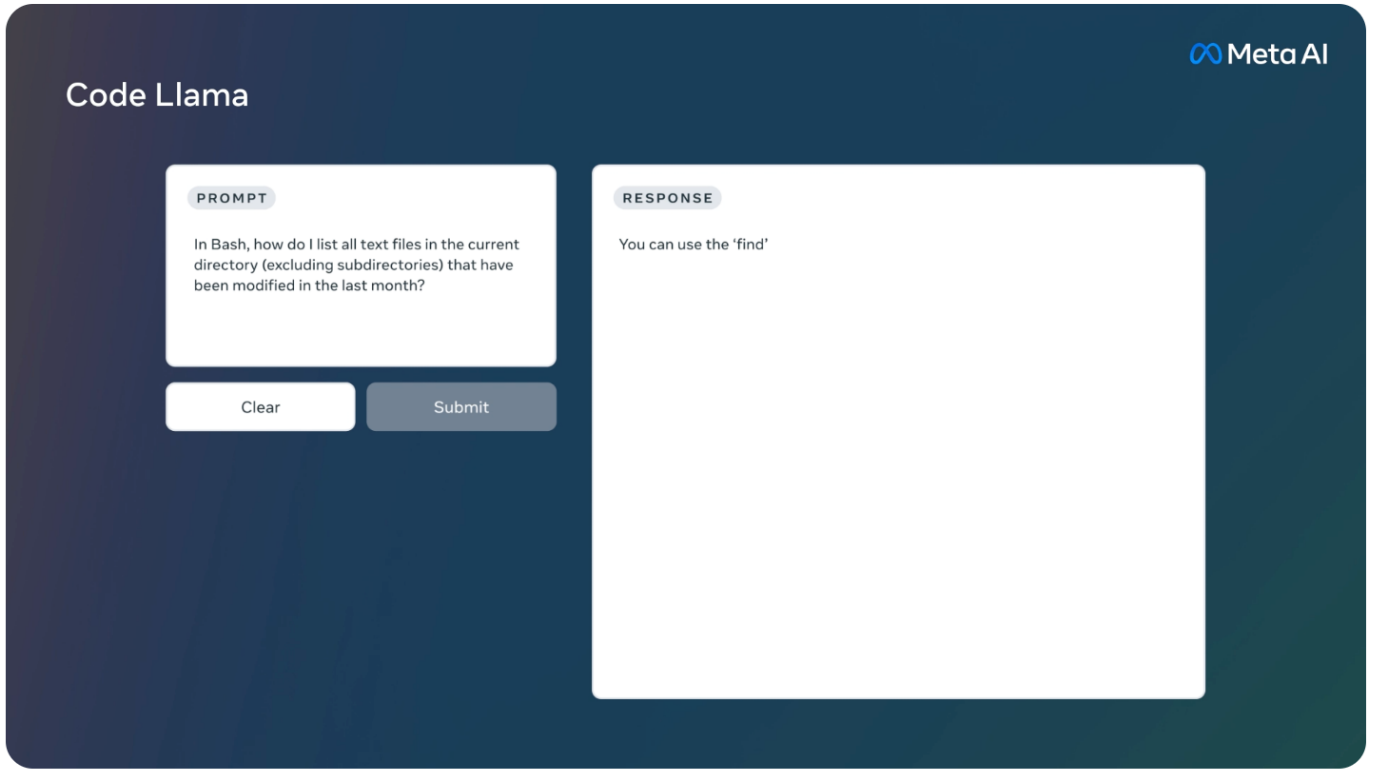
https://ai.meta.com/llama/

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

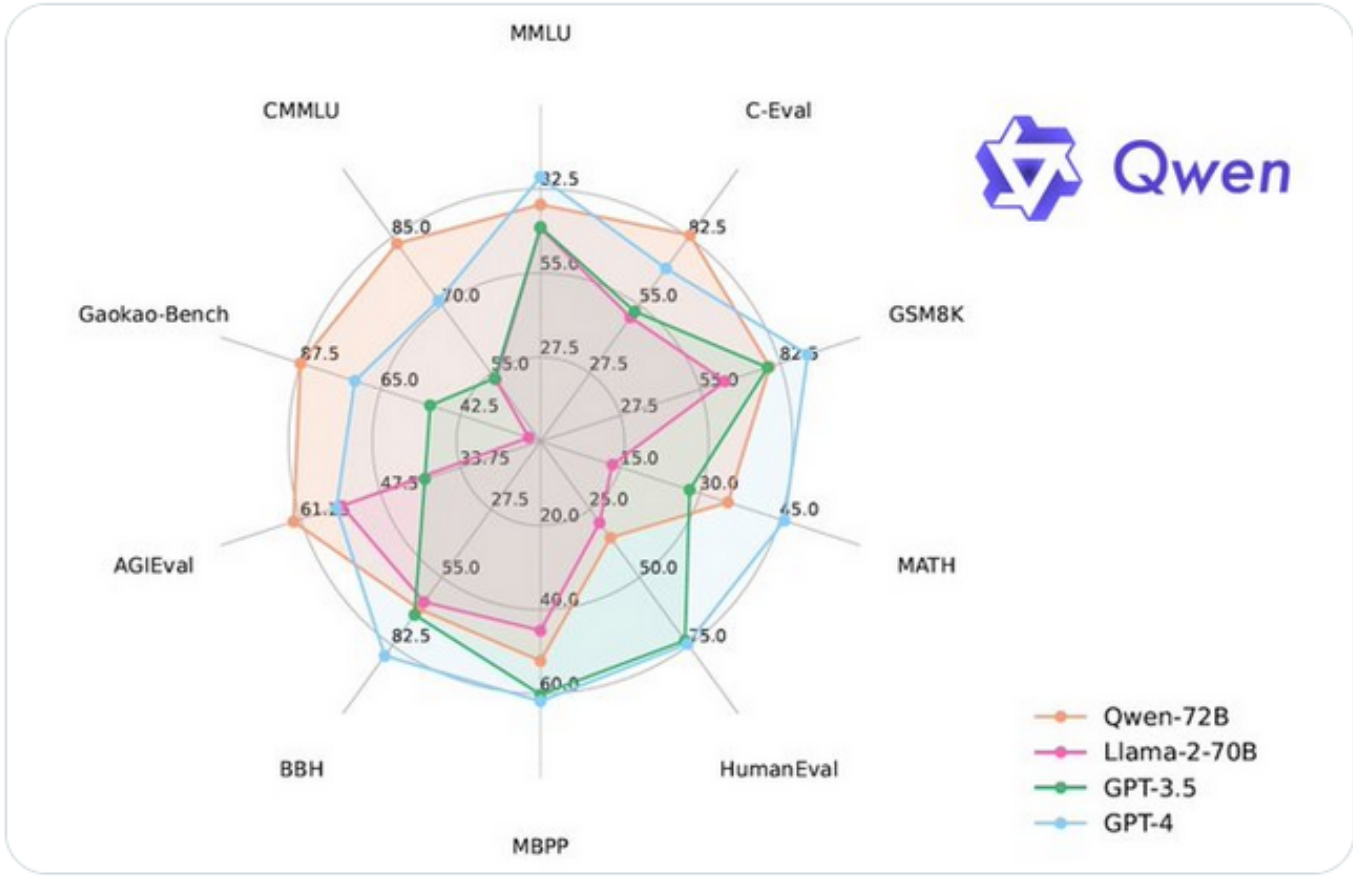
https://github.com/QwenLM/Qwen

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

README MIT license

Ollama

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*4=240GB 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 **Lo**w-**R**ank **A**daptation, 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.[1]

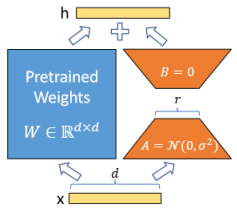


Figure 1: Our reparametrization. We only train A and B.

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.
[0]Compared to V1, this draft includes better baselines, experiments on GLUE, and more on adapter latency.
[1]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 (3rd party): 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 Yi Zhang Jyoti Aneja Caio César Teodoro Mendes
Allie Del Giorno Sivakanth Gopi Mojan Javaheripi Piero Kauffmann
Gustavo de Rosa Olli Saarikivi Adil Salim Shital Shah Harkirat Singh Behl
Xin Wang Sébastien Bubeck Ronen Eldan Adam Tauman Kalai 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+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+17], a phenomenon which is now referred to as *scaling laws* [KMH+20]. The subsequent exploration of scale in deep learning was guided by these scaling laws [BMR+20], and discoveries of variants of these laws led to rapid jump in performances [HBM+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+20], and it can yield other side benefits such as somewhat smaller datasets [LYR+23, YGK+23] or allowing for more passes on the data [MRB+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+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+21]	300M	100B	13.2%	-
2021 Jul	Codex-12B [CTJ+21]	12B	100B	28.8%	-
2022 Mar	CodeGen-Mono-350M [NPH+23]	350M	577B	12.8%	-
2022 Mar	CodeGen-Mono-16.1B [NPH+23]	16.1B	577B	29.3%	35.3%
2022 Apr	PaLM-Coder [CND+22]	540B	780B	35.9%	47.0%
2022 Sep	CodeGeeX [ZXZ+23]	13B	850B	22.9%	24.4%
2022 Nov	GPT-3.5 [Ope23]	175B	N.A.	47%	-
2022 Dec	SantaCoder [ALK+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+23]	1B	N.A.	10.3%	-
2023 May	CodeGen2-7B [NHX+23]	7B	N.A.	19.1%	-
2023 May	StarCoder [LAZ+23]	15.5B	1T	33.6%	52.7%
2023 May	StarCoder-Prompted [LAZ+23]	15.5B	1T	40.8%	49.5%
2023 May	PaLM 2-S [ADF+23]	N.A.	N.A.	37.6%	50.0%
2023 May	CodeT5+ [WLG+23]	2B	52B	24.2%	-
2023 May	CodeT5+ [WLG+23]	16B	52B	30.9%	-
2023 May	InstructCodeT5+ [WLG+23]	16B	52B	35.0%	-
2023 Jun	WizardCoder [LXZ+23]	16B	1T	57.3%	51.8%
2023 Jun	**phi-1**	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
    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.

TinyStories

A tiny 10M transformer can generate coherent text when trained on language that a 3 year old would understand.

TinyStories: How Small Can Language Models Be and Still Speak Coherent English?

Ronen Eldan* and Yuanzhi Li†

Microsoft Research

April 2023

Abstract

Language models[4, 5, 21] (LMs) are powerful tools for natural language processing, but they often struggle to produce coherent and fluent text when they are **small**. Models with around 125M parameters such as GPT-Neo (small) [3] or GPT-2 (small) [23] can rarely generate coherent and consistent English text beyond a few words even after extensive training. This raises the question of whether the emergence of the ability to produce coherent English text only occurs at larger scales (with hundreds of millions of parameters or more) and complex architectures (with many layers of global attention).

In this work, we introduce **TinyStories**, a synthetic dataset of short stories that only contain words that a typical 3 to 4-year-olds usually understand, generated by GPT-3.5 and GPT-4. We show that TinyStories can be used to train and evaluate LMs that are much smaller than the state-of-the-art models (**below 10 million total parameters**), or have much simpler architectures (**with only one transformer block**), yet still produce fluent and consistent stories with several paragraphs that are diverse and have almost perfect grammar, and demonstrate reasoning capabilities.

We also introduce a new paradigm for the evaluation of language models: We suggest a framework which uses GPT-4 to grade the content generated by these models as if those were stories written by students and graded by a (human) teacher. This new paradigm overcomes the flaws of standard benchmarks which often require the model's output to be very structured, and moreover it provides a multidimensional score for the model, providing scores for different capabilities such as grammar, creativity and instruction-following.

We hope that TinyStories can facilitate the development, analysis and research of LMs, especially for low-resource or specialized domains, and shed light on the emergence of language capabilities in LMs.

1 Introduction

https://arxiv.org/abs/2305.07759

Natural language is rich and diverse. It is not only a system of rules and symbols, but also a way of conveying and interpreting meaning [32]. To understand and produce language, one needs not only to master the technical rules of grammar and knowledge of vocabulary, but also to have sufficient factual information and to be able to reason logically and contextually. Therefore, autoregressive language models, which are able to generate coherent English text, must have acquired some degree of these capabilities as well. For example, consider the following incomplete sentence:

> Jack was hungry, so he went looking for ⟨_⟩

To complete this sentence in a sensible way, the language model needs to know that hunger is a state that motivates people to seek food, and that food is a category of things that can satisfy hunger. It also needs to choose a word that fits the syntactic and semantic constraints of the sentence (such as "a snack"), and that is plausible given the situation and the background knowledge.

An example that illustrates the need for reasoning is:

> Lily wanted to get either a cat or a dog. Her mother didn't let her get a dog so instead she ⟨_⟩

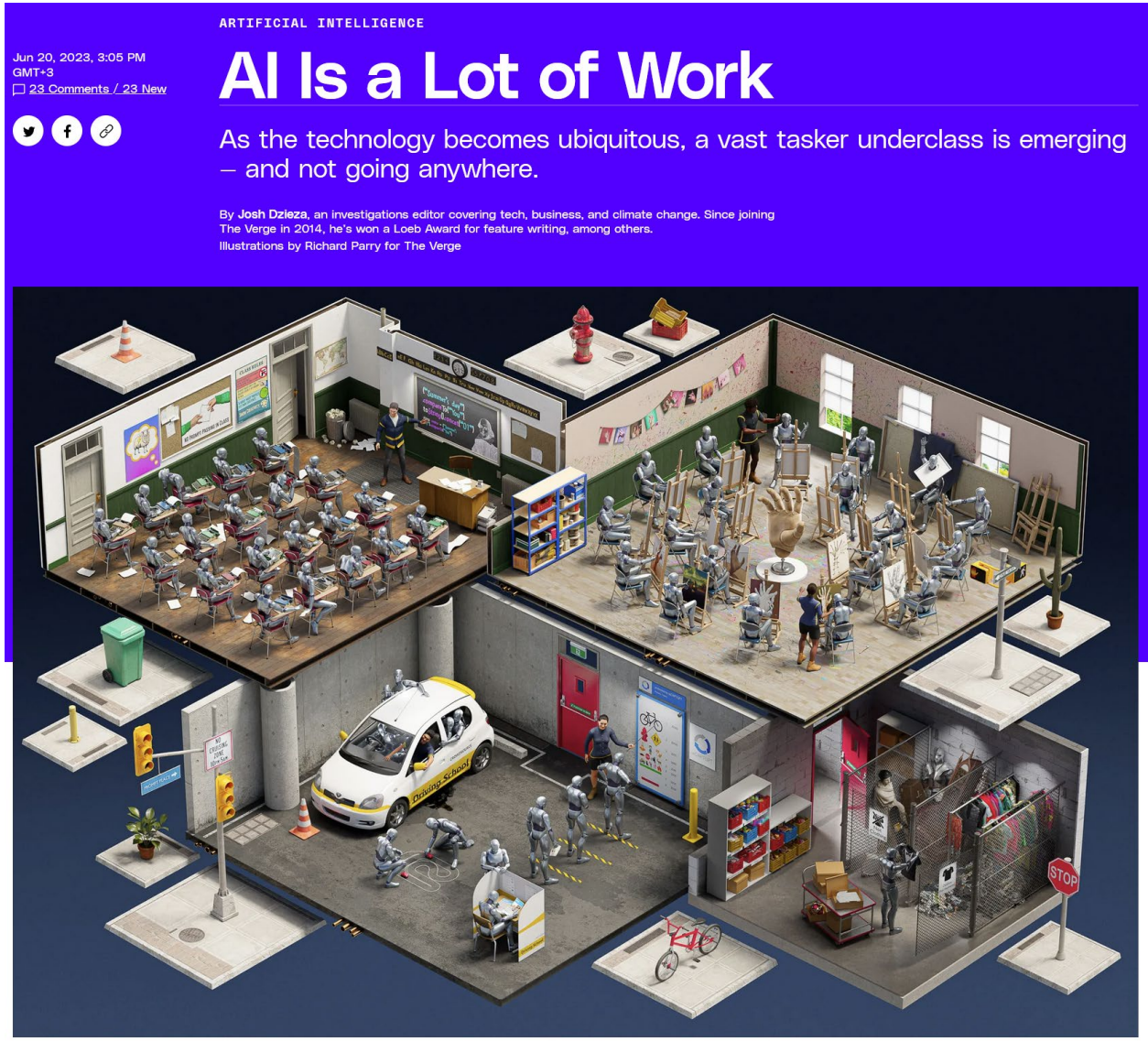
*roneneldan@microsoft.com
†yuanzhili@microsoft.com

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. ”

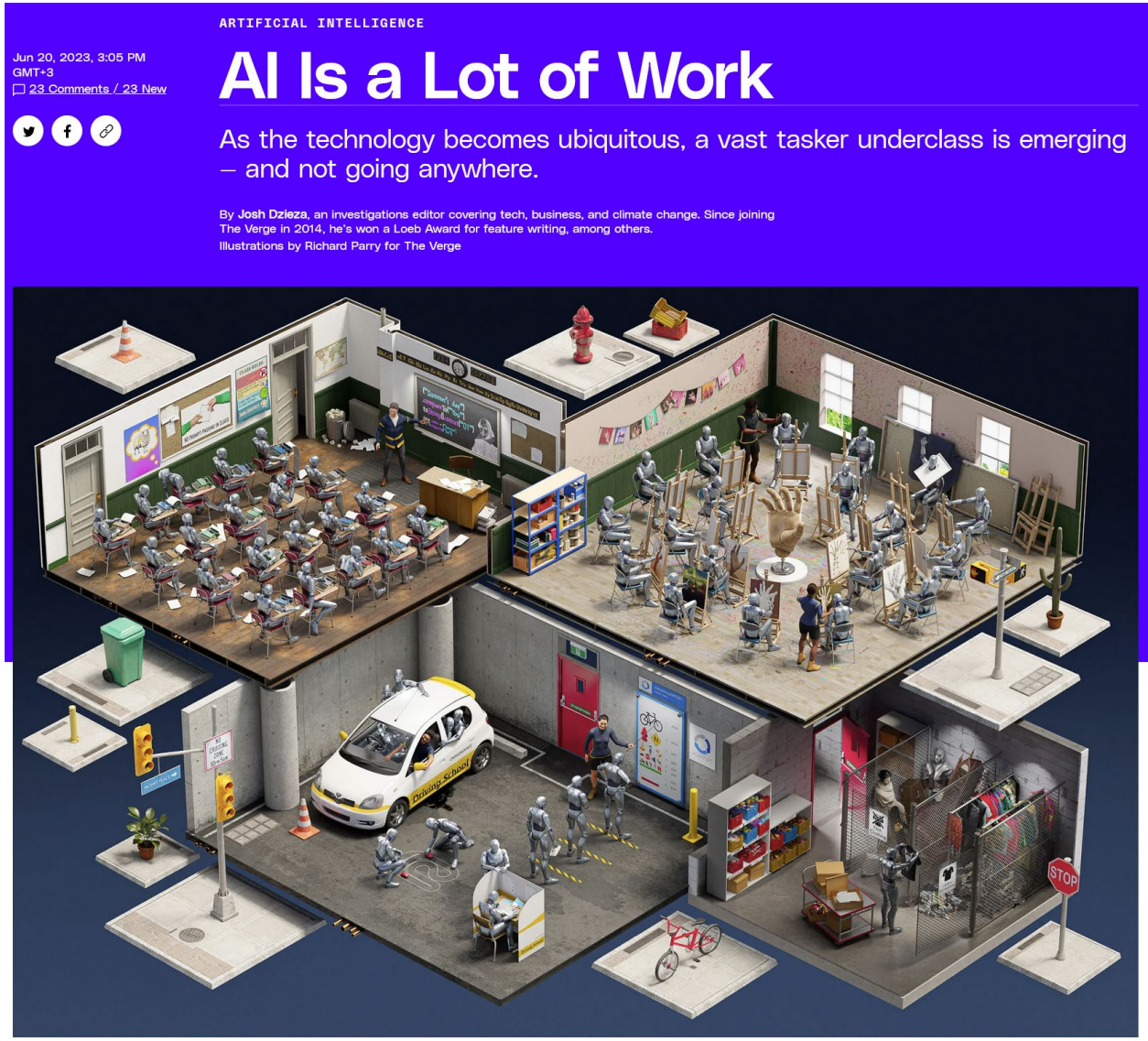


https://www.theverge.com/features/23764584/ai-artificial-intelligence-data-notation-labor-scale-surge-remotasks-openai-chatbots

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**.”

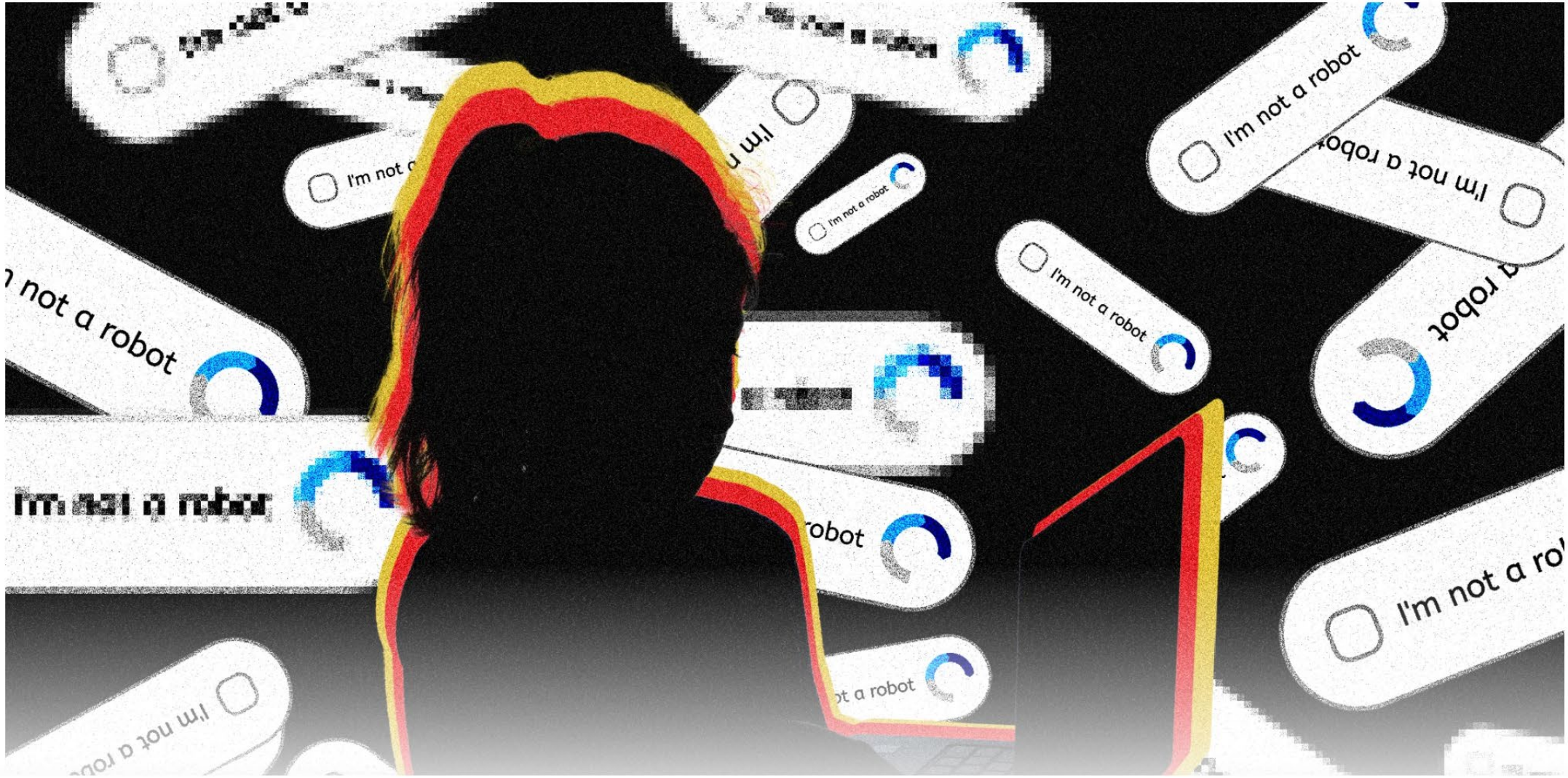


NIAMH ROWE BUSINESS NOV 15, 2023 6:00 AM

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.



MORGAN MEAKER BUSINESS SEP 11, 2023 6:00 AM

https://www.wired.com/story/prisoners-training-ai-finland/

These Prisoners Are Training AI

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

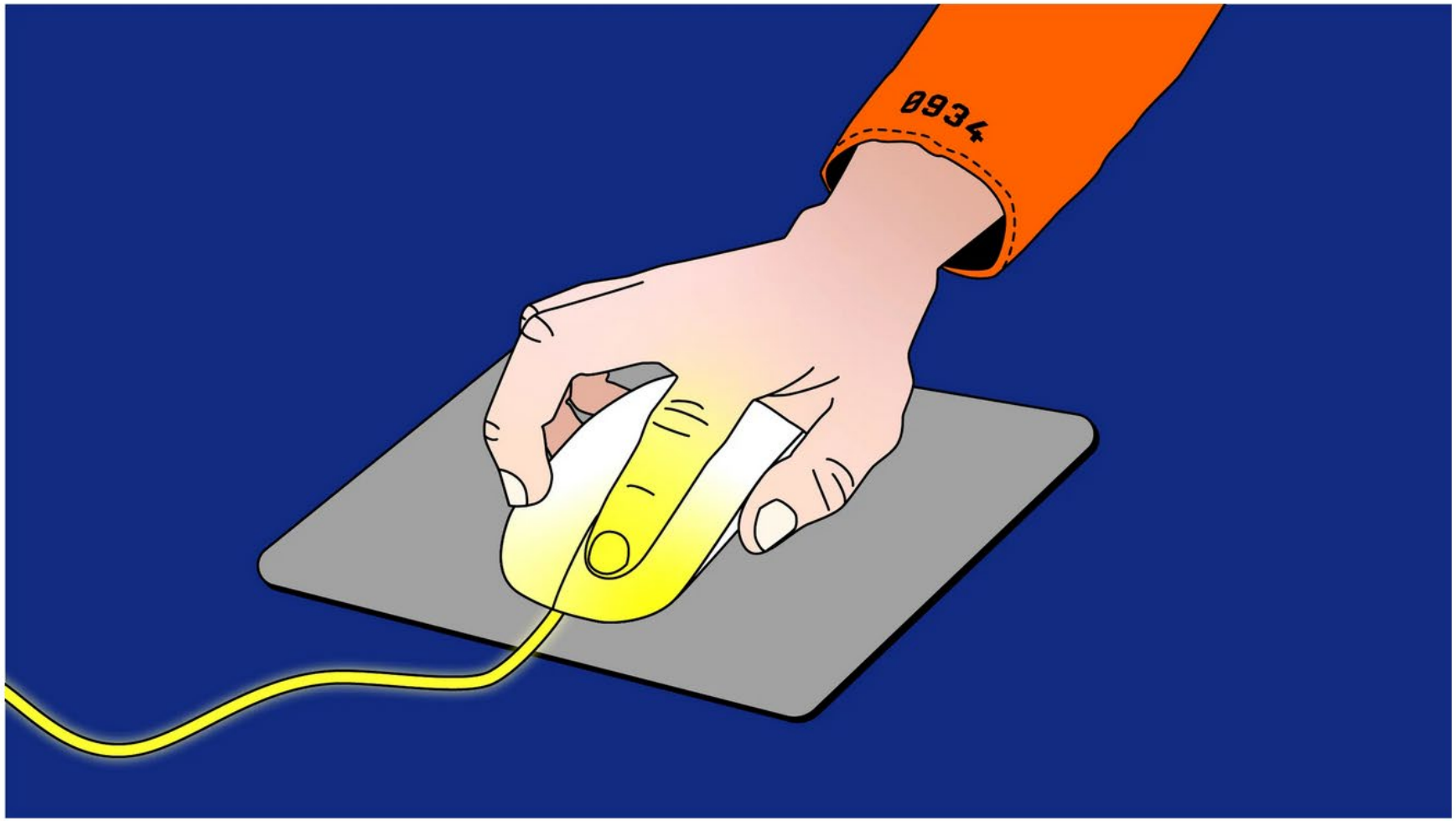


ILLUSTRATION: JACQUI VANLIEW; GETTY IMAGES

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

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

by: The Vicuna Team, Mar 30, 2023

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.

ARTIFICIAL INTELLIGENCE / TECH / META

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

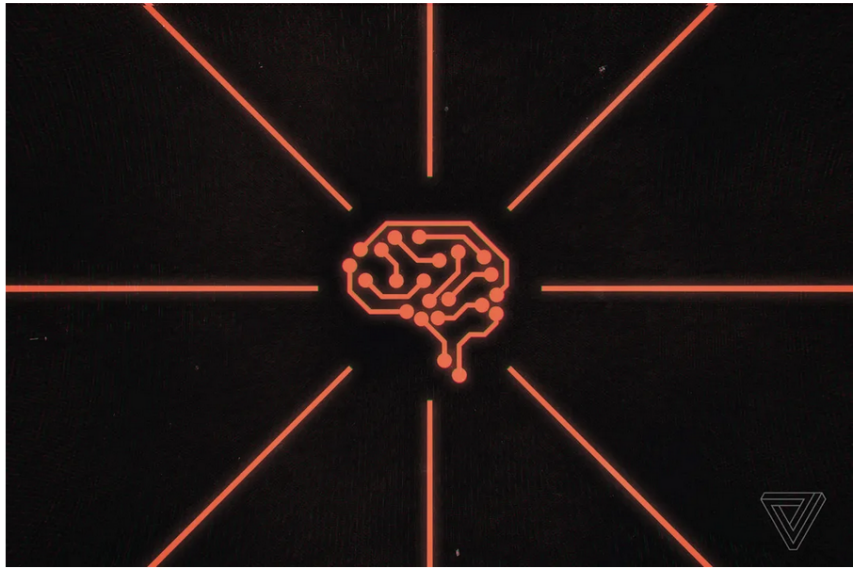


Illustration by Alex Castro / The Verge

/ 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.

https://www.theverge.com/2023/11/4/23946353/generative-ai-copyright-training-data-openai-microsoft-google-meta-stabilityai

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 **Lo**w-**R**ank **A**daptation, 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.[1]

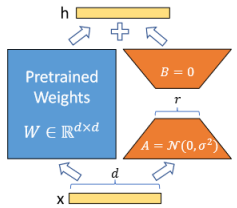


Figure 1: Our reparametrization. We only train A and B.

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.
[0]Compared to V1, this draft includes better baselines, experiments on GLUE, and more on adapter latency.
[1]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.[2]

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 [57] 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].

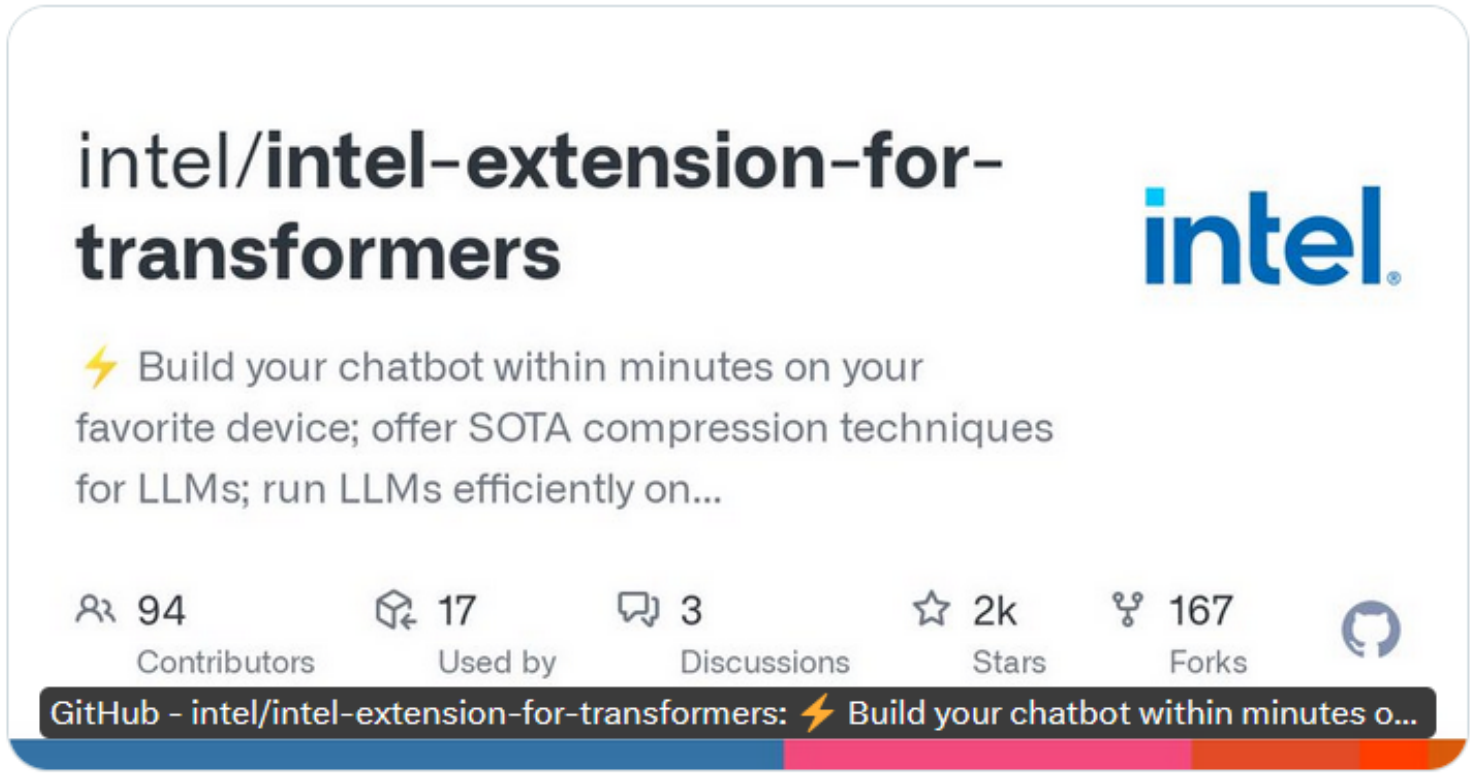
We demonstrate for the first time that it is possible to finetune a quantized 4-bit model without any performance degradation. Our method, QLoRA, uses a novel high-precision technique to quantize a pretrained model to 4-bit, then adds a small set of learnable Low-rank Adapter weights [28]

*Equal contribution.

[2] 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/@NeuralCompres... Kudos to ITREX team!
🎯Code: github.com/intel/intel-ex...
#IAmIntel #intelai @intel @huggingface



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...

4:16 PM · Jan 8, 2024 · **143.5K** Views

17 131 709 1.2K

https://apeatling.com/articles/simple-guide-to-local-llm-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

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

cess_text(text):
lacements = {
 '<|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)
urn text

me__ == '__main__':
ly_chatml = datasets.load_dataset("sam-mosaic/dolly_chatml")
aset = [process_text(row) for row in tqdm(dolly_chatml['trai
in_dataset = GeneralDataset.from_list(data=dataset)
_run_train(config_cls=Config, train_dataset=train_dataset)
```

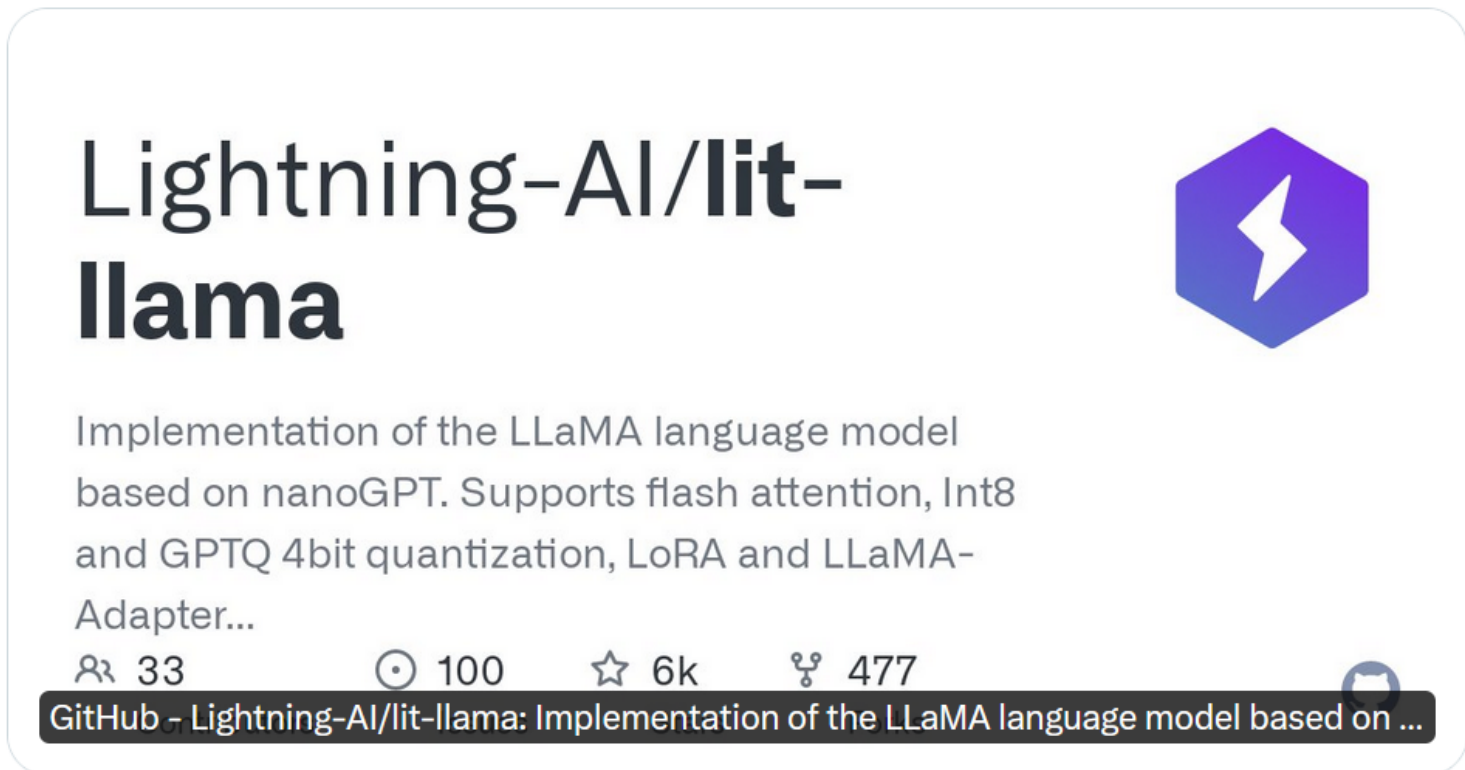
```
    --deepspeed_stage 2 \
    --load_in_4bit True \
    --use_flash_attention_2 True \
    --apply_lora True \
    --stabilize True \
    --use_gradient_checkpointing True \
    --prepare_model_for_kbit_training True \
    --gradient_accumulation_steps 1 \
    --save_total_limit 5 \
    --per_device_train_batch_size 8 \
    --num_train_epochs 4 \
    --save_steps 20 \
    --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 |
```

12:11 AM · Nov 16, 2023 · **453** Views

AaltoMediaAI
@aaltomediaai

An Apache-licensed version of the Llama language model, based on Karpathy's NanoGPT. Concise and clean code. Includes LoRA finetuning.



https://github.com/Lightning-AI/lit-llama

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

LMSYS Chatbot Arena Leaderboard

| Vote | Blog | GitHub | Paper | Dataset | Twitter | Discord |

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.

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	GPT-4-Turbo	1249	+13/-13	30268	OpenAI
2	Bard (Gemini Pro)	1215	+16/-15	3014	Google
3	GPT-4-0314	1189	+14/-12	18062	OpenAI
4	GPT-4-0613	1161	+13/-13	27441	OpenAI
5	Mistral Medium	1150	+15/-15	11480	Mistral
6	Claude-1	1150	+13/-13	17630	Anthropic
7	Claude-2.0	1131	+14/-13	11698	Anthropic
8	Mixtral-8x7b-Instruct-v0.1	1123	+14/-13	14165	Mistral
9	Gemini Pro (Dev API)	1122	+15/-16	6332	Google
10	Claude-2.1	1119	+14/-13	23323	Anthropic
11	GPT-3.5-Turbo-0613	1117	+14/-13	30326	OpenAI
12	Gemini Pro	1114	+16/-15	6981	Google
13	Yi-34B-Chat	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

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

👈 A is better | 👉 B is better | 🤝 Tie | 👎 Both are bad

Recent progress

Flash attention

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

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

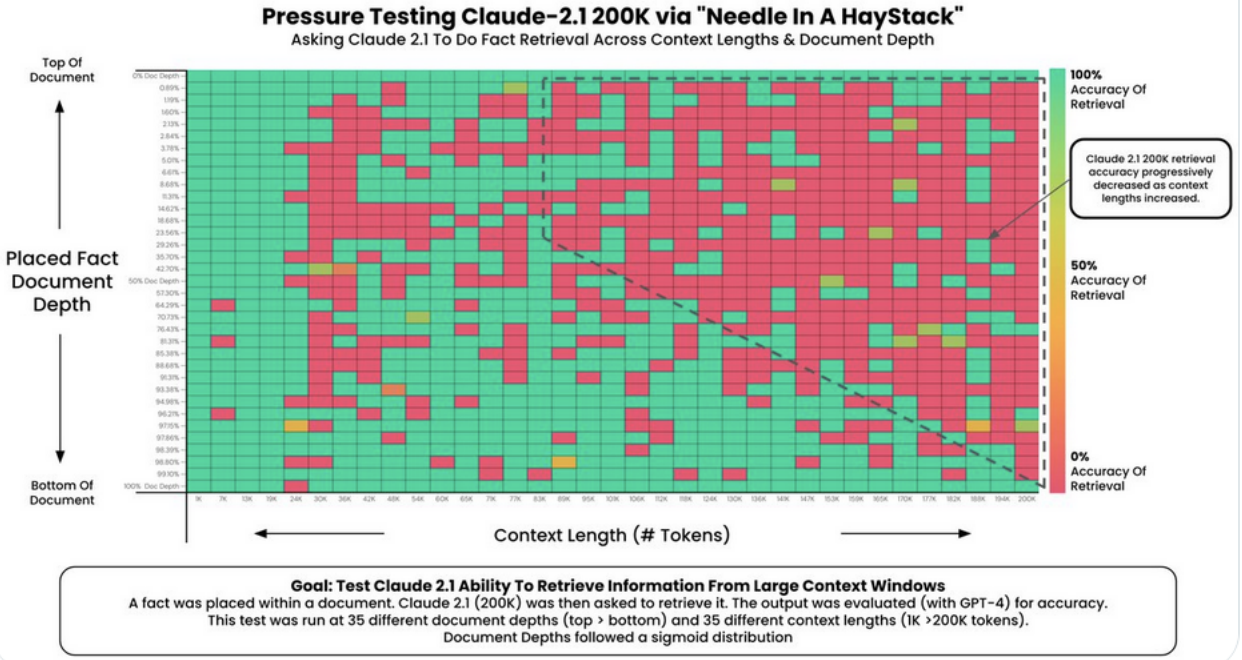
Claude 2.1 (200K Tokens) - Pressure Testing Long Context Recall

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

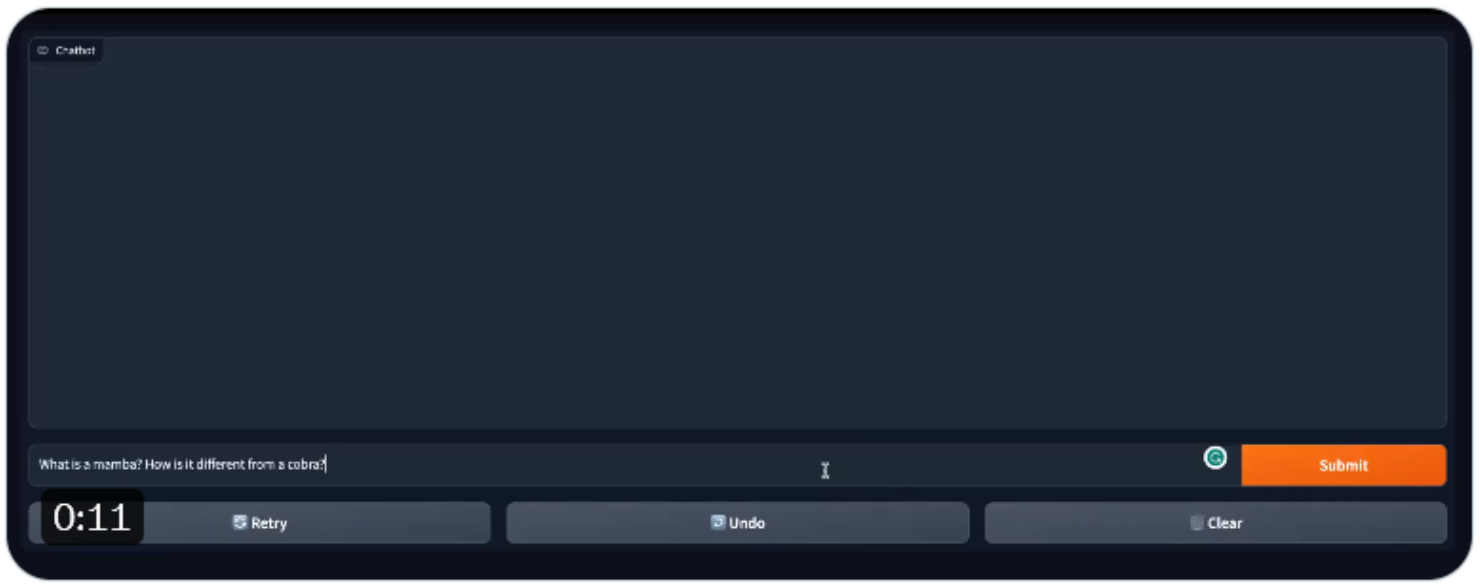
> 
>
> **Albert Gu** @_albertgu · Dec 4, 2023
>
> Quadratic attention has been indispensable for information-dense modalities such as language... until now.
>
> Announcing Mamba: a new SSM arch. that has linear-time scaling, ultra long context, and most importantly--outperforms Transformers ...
> Show 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://x.com/xiuyu_l/status/1740806425843294606?s=20

Shital Shah @sytelus · Dec 13, 2023

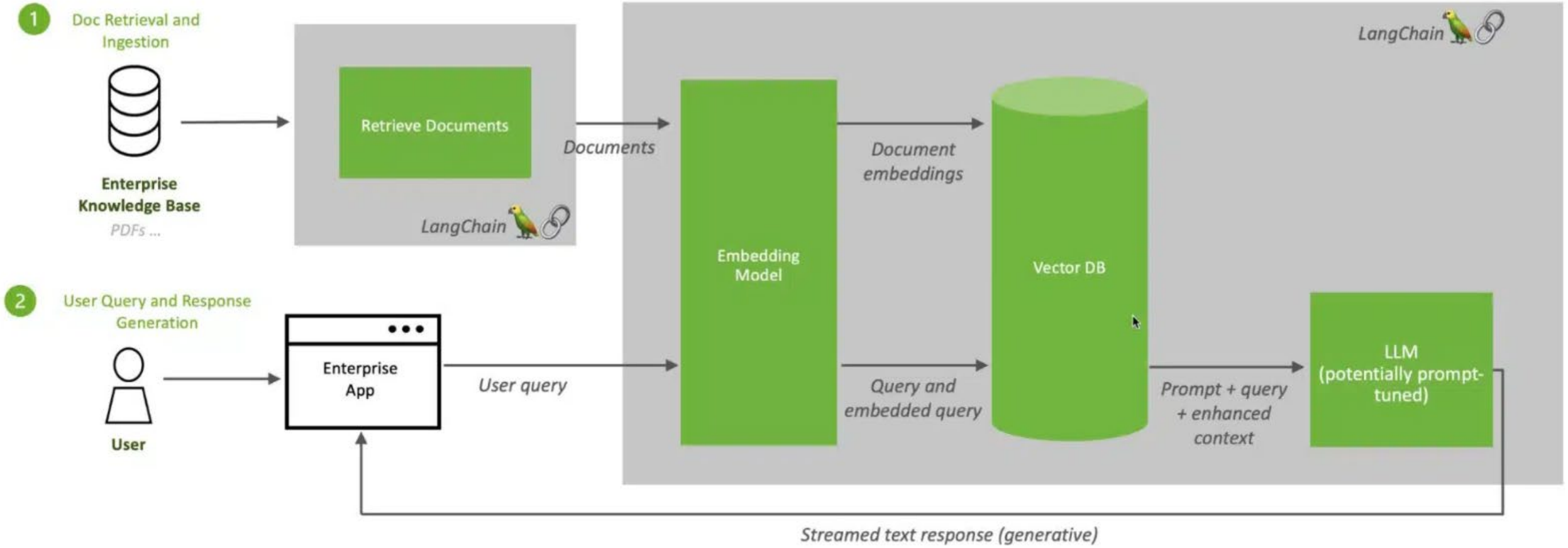
Mistral-7B is cool but you know what's cooler? A more powerful model in just 1/3rd of the size!

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

Retrieval-Augmented Generation



https://blogs.nvidia.com/blog/what-is-retrieval-augmented-generation/

Code examples:

https://cookbook.openai.com/examples/vector_databases/elasticsearch/elasticsearch-retrieval-augmented-generation , http://www.georgesung.com/ai/llm-qa-eval-wikipedia/

LLMs as agents with long-term memory

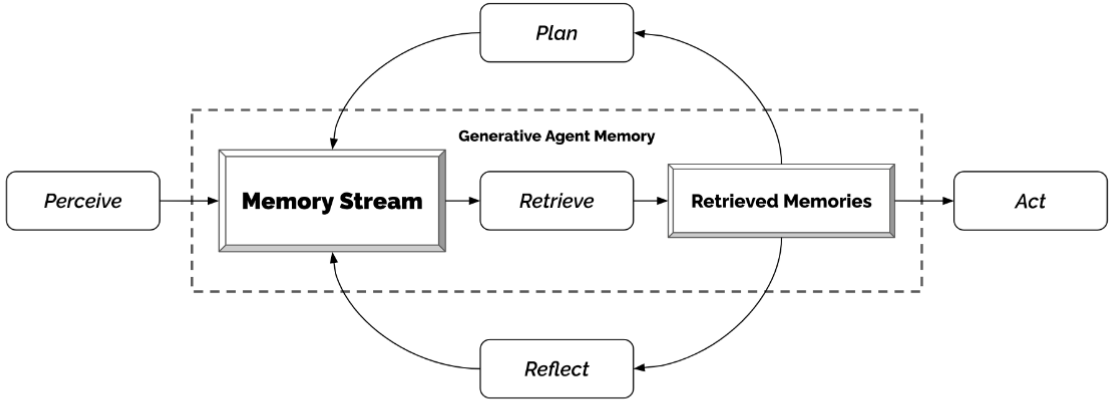


Figure 5: Our generative agent architecture. Agents perceive their environment, and all perceptions are saved in a comprehensive record of the agent's experiences called the memory stream. Based on their perceptions, the architecture retrieves relevant memories and uses those retrieved actions to determine an action. These retrieved memories are also used to form longer-term plans and create higher-level reflections, both of which are entered into the memory stream for future use.

Generative Agents: Interactive Simulacra of Human Behavior

Joon Sung Park
Stanford University
Stanford, USA
joonspk@stanford.edu

Joseph C. O'Brien
Stanford University
Stanford, USA
jobrien3@stanford.edu

Carrie J. Cai
Google Research
Mountain View, CA, USA
cjcai@google.com

Meredith Ringel Morris
Google DeepMind
Seattle, WA, USA
merrie@google.com

Percy Liang
Stanford University
Stanford, USA
pliang@cs.stanford.edu

Michael S. Bernstein
Stanford University
Stanford, USA
msb@cs.stanford.edu

Figure 1: Generative agents are believable simulacra of human behavior for interactive applications. In this work, we demonstrate generative agents by populating a sandbox environment, reminiscent of The Sims, with twenty-five agents. Users can observe and intervene as agents plan their days, share news, form relationships, and coordinate group activities.

ABSTRACT

Believable proxies of human behavior can empower interactive applications ranging from immersive environments to rehearsal spaces for interpersonal communication to prototyping tools. In this paper, we introduce generative agents: computational software agents that simulate believable human behavior. Generative agents wake up, cook breakfast, and head to work; artists paint, while authors write; they form opinions, notice each other, and initiate conversations; they remember and reflect on days past as they plan the next day. To enable generative agents, we describe an architecture that extends a large language model to store a complete record of the agent's experiences using natural language, synthesize those memories over time into higher-level reflections, and retrieve them dynamically to plan behavior. We instantiate generative agents to populate an interactive sandbox environment inspired by The Sims, where end users can interact with a small town of twenty-five agents using natural language. In an evaluation, these generative agents produce believable individual and emergent social behaviors. For example, starting with only a single user-specified notion that one agent wants to throw a Valentine's Day party, the agents autonomously spread invitations to the party over the next two

UIST '23, October 29–November 01, 2023, San Francisco, CA, USA

ACM ISBN 979-8-4007-0132-0/23/10.
https://doi.org/10.1145/3586183.3606763

Improved reasoning via forward search

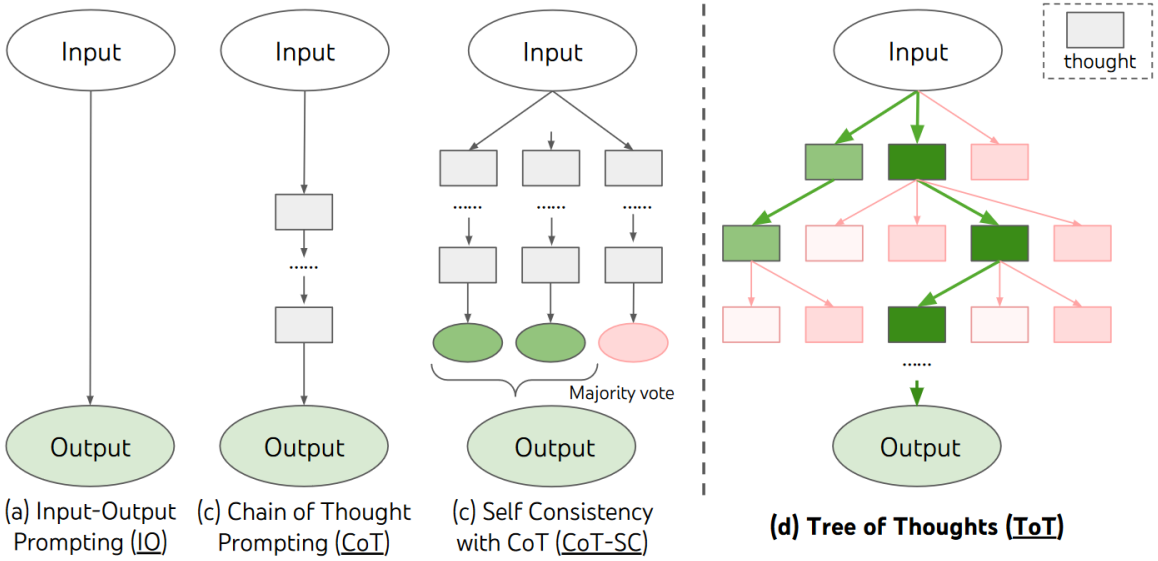


Figure 1: Schematic illustrating various approaches to problem solving with LLMs. Each rectangle box represents a *thought*, which is a coherent language sequence that serves as an intermediate step toward problem solving. See concrete examples of how thoughts are generated, evaluated, and searched in Figures 2,4,6.

Tree of Thoughts: Deliberate Problem Solving with Large Language Models

Shunyu Yao Princeton University
Dian Yu Google DeepMind
Jeffrey Zhao Google DeepMind
Izhak Shafran Google DeepMind
Thomas L. Griffiths Princeton University
Yuan Cao Google DeepMind
Karthik Narasimhan Princeton University

Abstract

Language models are increasingly being deployed for general problem solving across a wide range of tasks, but are still confined to token-level, left-to-right decision-making processes during inference. This means they can fall short in tasks that require exploration, strategic lookahead, or where initial decisions play a pivotal role. To surmount these challenges, we introduce a new framework for language model inference, "Tree of Thoughts" (ToT), which generalizes over the popular "Chain of Thought" approach to prompting language models, and enables exploration over coherent units of text ("thoughts") that serve as intermediate steps toward problem solving. ToT allows LMs to perform deliberate decision making by considering multiple different reasoning paths and self-evaluating choices to decide the next course of action, as well as looking ahead or backtracking when necessary to make global choices. Our experiments show that ToT significantly enhances language models' problem-solving abilities on three novel tasks requiring non-trivial planning or search: Game of 24, Creative Writing, and Mini Crosswords. For instance, in Game of 24, while GPT-4 with chain-of-thought prompting only solved 4% of tasks, our method achieved a success rate of 74%. Code repo with all prompts: `https://github.com/princeton-nlp/tree-of-thought-llm`.

https://arxiv.org/abs/2305.10601

1 Introduction

Originally designed to generate text, scaled-up versions of language models (LMs) such as GPT [25, 26, 1, 23] and PaLM [5] have been shown to be increasingly capable of performing an ever wider range of tasks requiring mathematical, symbolic, commonsense, and knowledge reasoning. It is perhaps surprising that underlying all this progress is still the original autoregressive mechanism for generating text, which makes token-level decisions one by one and in a left-to-right fashion. Is such a simple mechanism sufficient for a LM to be built toward a general problem solver? If not, what problems would challenge the current paradigm, and what should be alternative mechanisms?

The literature on human cognition provides some clues to answer these questions. Research on "dual process" models suggests that people have two modes in which they engage with decisions – a fast, automatic, unconscious mode ("System 1") and a slow, deliberate, conscious mode ("System 2") [30, 31, 16, 15]. These two modes have previously been connected to a variety of mathematical models used in machine learning. For example, research on reinforcement learning in humans and other animals has explored the circumstances under which they engage in associative "model free" learning or more deliberative "model based" planning [7]. The simple associative token-level choices of LMs are also reminiscent of "System 1", and thus might benefit from augmentation by a more deliberate "System 2" planning process that (1) maintains and explores diverse alternatives for current

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

Tool use

https://arxiv.org/abs/2302.04761

See also: https://arxiv.org/pdf/2302.07842.pdf

Toolformer: Language Models Can Teach Themselves to Use Tools

Timo Schick **Jane Dwivedi-Yu** **Roberto Dessì**[†] **Roberta Raileanu**
Maria Lomeli **Luke Zettlemoyer** **Nicola Cancedda** **Thomas Scialom**
Meta AI Research [†]Universitat Pompeu Fabra

Abstract

Language models (LMs) exhibit remarkable abilities to solve new tasks from just a few examples or textual instructions, especially at scale. They also, paradoxically, struggle with basic functionality, such as arithmetic or factual lookup, where much simpler and smaller models excel. In this paper, we show that LMs can teach themselves to *use external tools* via simple APIs and achieve the best of both worlds. We introduce *Toolformer*, a model trained to decide which APIs to call, when to call them, what arguments to pass, and how to best incorporate the results into future token prediction. This is done in a self-supervised way, requiring nothing more than a handful of demonstrations for each API. We incorporate a range of tools, including a calculator, a Q&A system, a search engine, a translation system, and a calendar. Toolformer achieves substantially improved zero-shot performance across a variety of downstream tasks, often competitive with much larger models, without sacrificing its core language modeling abilities.

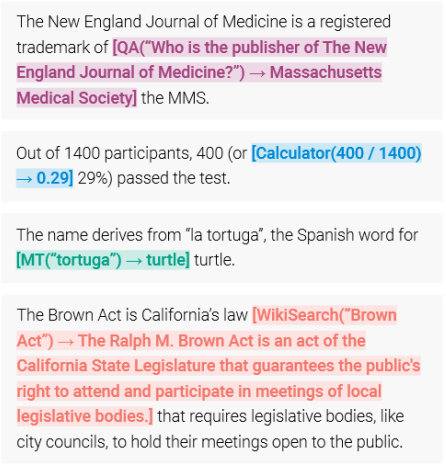


Figure 1: Exemplary predictions of Toolformer. The model autonomously decides to call different APIs (from top to bottom: a question answering system, a calculator, a machine translation system, and a Wikipedia search engine) to obtain information that is useful for completing a piece of text.

1 Introduction

Large language models achieve impressive zero- and few-shot results on a variety of natural language processing tasks (Brown et al., 2020; Chowdhery et al., 2022, i.a.) and show several emergent capabilities (Wei et al., 2022). However, all of these models have several inherent limitations that can at best be partially addressed by further scaling. These limitations include an inability to access up-to-date information on recent events (Komeili et al., 2022) and the related tendency to hallucinate facts (Maynez et al., 2020; Ji et al., 2022), difficulties in understanding low-resource languages (Lin et al., 2021), a lack of mathematical skills to perform precise calculations (Patel et al., 2021) and an unawareness of the progression of time (Dhingra et al., 2022).

A simple way to overcome these limitations of today's language models is to give them the ability to *use external tools* such as search engines, calculators, or calendars. However, existing approaches either rely on large amounts of human annotations (Komeili et al., 2022; Thoppilan et al., 2022) or limit tool use to task-specific settings only (e.g., Gao et al., 2022; Parisi et al., 2022), hindering a more widespread adoption of tool use in LMs. Therefore, we propose *Toolformer*, a model that learns to use tools in a novel way, which fulfills the following desiderata:

- The use of tools should be learned in a self-supervised way without requiring large amounts of *human annotations*. This is impor-

Recap

- Different models for different purposes:
  - Finetuned models like ChatGPT are better in answering questions and following instructions
  - Base models like davinci-002 are better in generating diverse continuations of a prompt
- The models are stochastic => can’t make conclusions about a prompt working or not working without trying it out multiple times (except if Temperature=0)
- Progress is rapid. If a task is hard today, it might not be in a year.
- No matter the model, two prompting principles apply:
  - Be specific
  - Provide high-quality examples

Extra resources and links

Dr. Strangelove's WarGames adventure in ChatGPT

TTakala · Follow
14 min read · Dec 12, 2022

Recently I was inspired by Frederic Besse's experiments of emulating a Linux terminal with ChatGPT. I realized that it might be fun to roleplay as a hacker, who gets access to a military network responsible for nuclear weapons.

The following roleplaying session is my first dialogue with ChatGPT. The result is quite a thriller, involving nuclear launches, world leaders, and high stakes negotiations. Chat sessions like this could even be used as plot outlines for a movie, novel, or game.

https://medium.com/@tmtakala/dr-strangeloves-wargames-adventure-in-chatgpt-925e0e59b53c

Michael Nielsen
@michael_nielsen

Curious: have you found ChatGPT useful in doing professional work?

If so, what kinds of prompts and answers have been helpful? Detailed examples greatly appreciated! Broader answer also appreciated

Not in theory, but where you've really *done it*, in your work

Thanks!

3:32 AM · Dec 7, 2022

446 393 2.7K 2K

Post your reply Reply

https://twitter.com/michael_nielsen/status/1600302211086458880

Michael Nielsen @michael_nielsen · Dec 7, 2022
Wow, many varied, surprising, and concrete examples.

Please keep them coming!

4 1 59

Michael Nielsen @michael_nielsen · Dec 7, 2022
I have a suspicion future iterations of this will be useful enough as a research assistant that I'll want a voice-to-text version, so I can just ask a question *any time* at all, and get the answer back. An ambient, portable RA.

12 3 165

Roman Pshichenko @romechenko · Dec 7, 2022
I'm trying out projects of various sizes where I let GPT do all the heavy lifting. I'll just copy and paste results.

ANIL ANANTHASWAMY SCIENCE OCT 2, 2022 8:00 AM

Self-Taught AI May Have a Lot in Common With the Human Brain

Neural networks can use self-supervised learning to figure out what matters. This process might be what helps humans do the same.

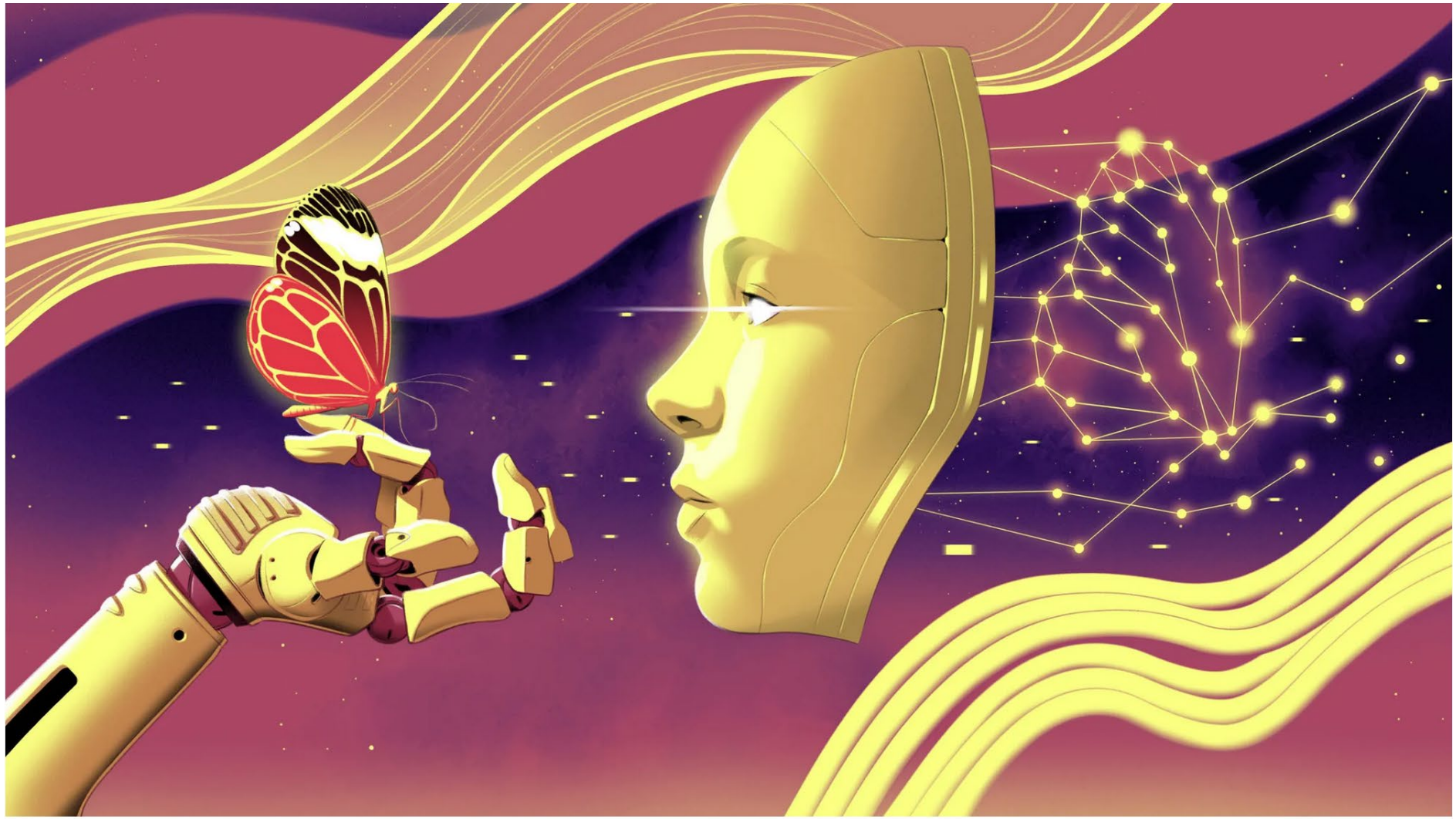


ILLUSTRATION: SEÑOR SALME/QUANTA MAGAZINE

https://www.wired.com/story/self-taught-ai-may-have-a-lot-in-common-with-the-human-brain/

AaltoMediaAI @aaltomediaai · Jun 18, 2023

Interesting tips on how to improve ChatGPT's quality in creative writing

> **Jeremy Nguyen** ✍️ 🚢 @JeremyNguyenPhD · May 19, 2023
>
> I've written hundreds of pages for Disney+, NBCUniversal.
>
> Lazy writers use ChatGPT to crank out trash, but A.I. can help you breathe life into your characters and make us care.
>
> ...
>
> Show more

https://twitter.com/JeremyNguyenPhD/status/1659565899303596045

AaltoMediaAI
@aaltomediaai

An interesting short article on the impact of game writing tools like Ubisoft's AI ghostwriter. For AI to contribute positively, my thinking has been that one should only automate drudgery that people don't enjoy, but it's complicated in practice.

From theverge.com

7:50 AM · May 6, 2023 · **203** Views

AaltoMediaAI @aaltomediaai · May 6, 2023

"Depending on whether you're talking to AI evangelists or entry-level game developers, barks and QA bug hunting are either forms of drudgery or very important inroads and pathways into maintaining steady employment in a tough industry."

1 36

https://www.theverge.com/2023/5/4/23700619/ai-game-development-jobs-gdc-2023

AaltoMediaAI @aaltomediaai · May 2, 2023

A good example of how LLMs are better viewed as mere building blocks instead of complete solutions to intelligence. Adding a bit of application-specific error prevention/correction/regeneration logic on top of a pretrained LLM can make a big difference.

rahul @rahulgs · May 2, 2023

Problem: Getting LLMs to output valid JSON in the format you want is hard

Solution: ONLY generate values, feed model with keys and JSON structure. Constrain outputs with custom sampling...

Show more

Jsonformer: A Bulletproof Way to Generate Structured JSON from Language Models

keys, quotes, braces, and brackets **from schema** fed into model during generation

```
{
  temperature: 24,
  humidity: 48.0,
  wind_speed: {
    value: 12,
    unit: "mph"
  }
}
```

model generates **only content tokens**

https://github.com/1rgs/jsonformer

Generating booleans - compares logits for **true** and **false**
Generating numbers - squash logits for non-digit tokens before sampling
Generating strings - stops generation on second "
Terminating arrays - compare logits for "," and "]"

AaltoMediaAI @aaltomediaai · Apr 20, 2023

AI-generated ads coming to facebook. Meta certainly has the data to train an engagement-maximizing AI copywriter.

https://asia.nikkei.com/Business/Technology/Meta-to-debut-ad-creating-generative-AI-this-year-CTO-says

AaltoMediaAI @aaltomediaai · Apr 15, 2023

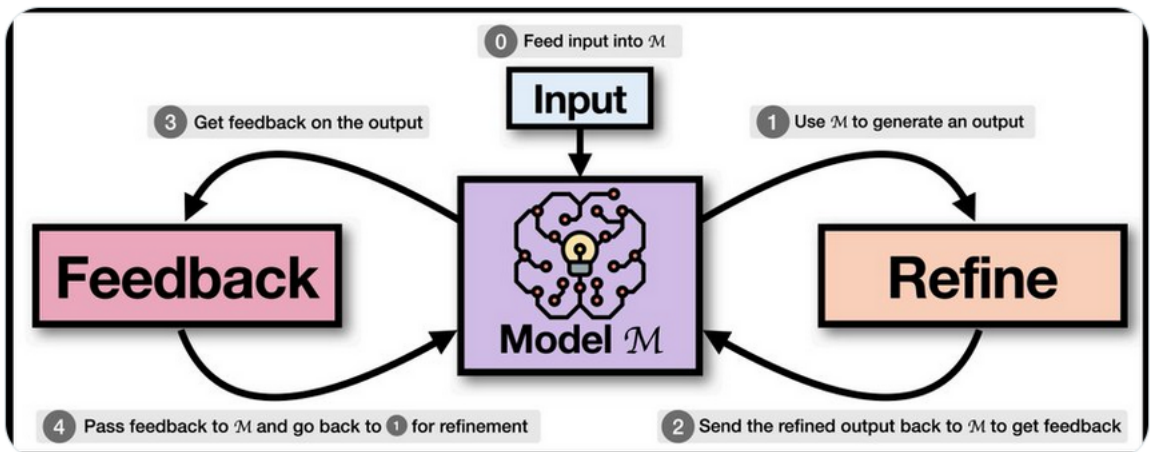
”The tool has even helped him win a grant from the U.K. government ... and complete coursework for a master’s degree necessary to receive a university teaching qualification he was too busy to handle by himself while working three jobs.”

https://www.vice.com/en/article/v7begx/overemployed-hustlers-exploit-chatgpt-to-take-on-even-more-full-time-jobs

Aman Madaan @aman_madaan · Apr 3, 2023

Can LLMs enhance their own output without human guidance? In some cases, yes! With Self-Refine, LLMs generate feedback on their work, use it to improve the output, and repeat this process. Self-Refine improves GPT-3.5/4 outputs for a wide range of tasks.

selfrefine.info



https://selfrefine.info/

finbarr @finbarrtimbers · Apr 1, 2023

I wrote about the development of GPT models since 2017, starting from GPT-1 and ending with GPT-4.

finbarrtimbers.substack.com/p/five-years-o...

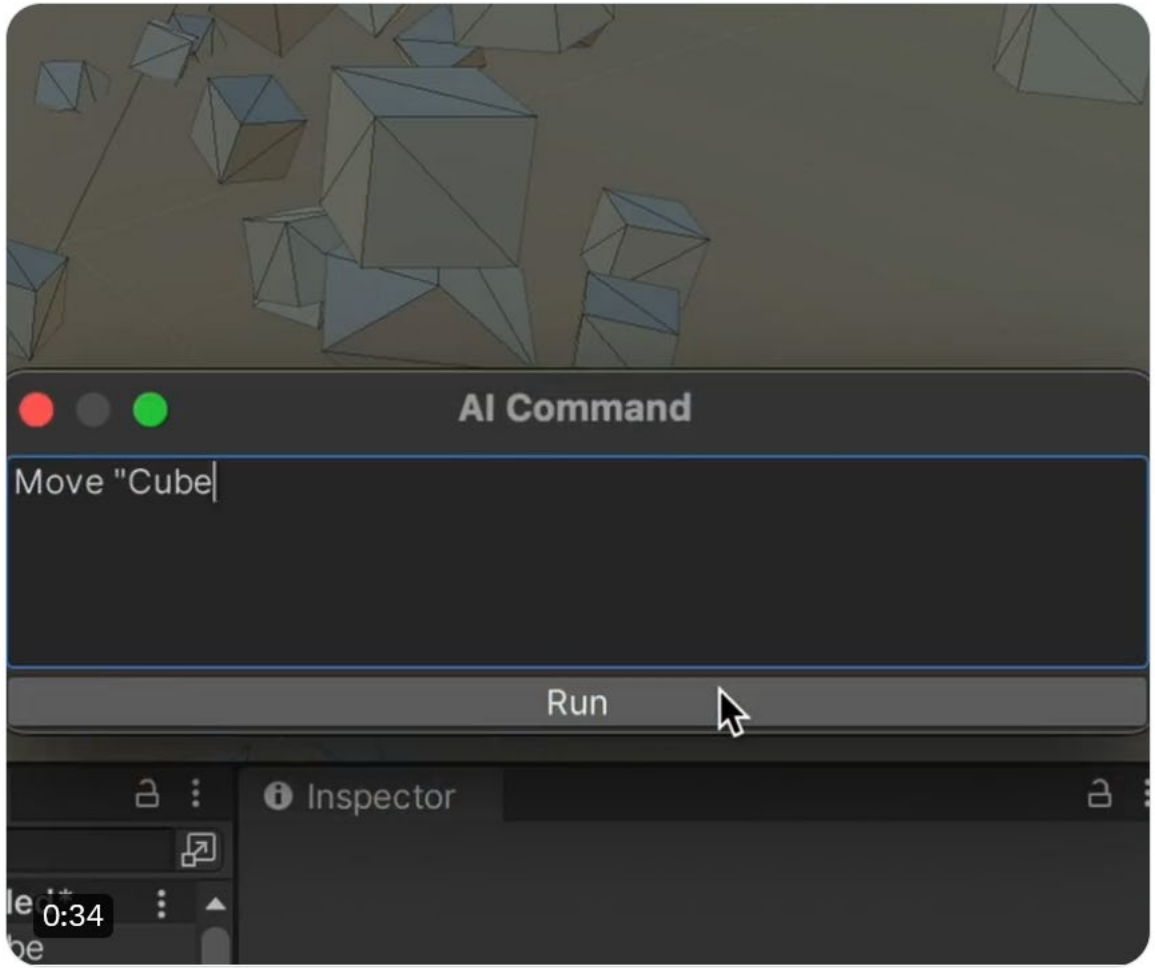
In the article, I go into depth about the technical details, and particularly, the differences between each successive SOTA model

> We're starting to see a consistent pattern emerge:
>
> - Papers introduce a bunch of changes, their own dataset, and have a new SOTA
> - but they don't do a proper ablation, so it's tough to understand what was important and what *drove* the improvements
>
> GPT-2, GPT-3, Jurassic-1, etc. all did this.

https://www.artfintel.com/p/five-years-of-progress-in-gpts

Keijiro Takahashi @_kzr · Mar 19, 2023

I created a proof-of-concept integration of ChatGPT into Unity Editor. You can edit your Unity projects using natural language prompts. github.com /keijiro/AIComm...



https://github.com/keijiro/AICommand

Environment: You are in the middle of a room. Looking quickly around you, you see a bed 1, a desk 2, a desk 1, a drawer 6, a drawer 5, a drawer 4, a drawer 3, a drawer 2, a drawer 1, a garbagecan 1, a laundryhamper 1, a safe 1, a shelf 6, a shelf 5, a shelf 4, a shelf 3, a shelf 2, and a shelf 1.
Task: examine the mug with the desklamp.
Trial #1
> think: To solve the task, I need to find and take a mug, then find and use a desklamp.
OK.
> think: First I need to find a mug. A mug is more likely to appear in drawer (1-6), desk (1-2), shelf (1-6), garbagecan (1), laundryhamper (1). I can check one by one, starting with drawer 1.
OK.
> go to drawer 1
The drawer 1 is closed.
...
> open drawer 6
You open the drawer 6. The drawer 6 is open. In it, you see a keychain 2.
> go to desk 1
On the desk 1, you see a creditcard 3, a desklamp 1, a laptop 2, a mug 1, a pen 1, and a pencil 1.
> think: Now I find a mug (1). Next, I need to take it.
OK.
> take mug 1 from desk 1
You pick up the mug 1 from the desk 1.
> think: Now I take a mug (1). Next, I need to find a desklamp. A desklamp is more likely to appear in desk (1-2), sidetable (1-2), shelf (1-6), bed (1), drawer (1-6). I can check one by one, starting with desk 1.
OK.
> go to desk 1
Nothing happens.
> go to desk 2
On the desk 2, you see a alarmclock 1, a bowl 1, a mug 3, a pencil 3, and a pencil 2.
> think: Now I find a desklamp (1). Next, I need to use it.
OK.
> use desklamp 1
Nothing happens.
> use desklamp 1
Nothing happens.
Status: Fail

Reflection: In this environment, my plan was to find a mug then find and use a desklamp. However, the task says to examine the mug with the desklamp. I should have looked for the desklamp first, then looked for the mug. I noticed that the desklamp was found on desk 1. In the next trial, I will go to desk 1, find the lamp, then look for the mug and examine it with the desklamp.

Trial #2
> go to desk 1
On the desk 1, you see a creditcard 3, a desklamp 1, a laptop 2, a mug 1, a pen 1, and a pencil 1.
> think: To solve the task, I need to find and take a mug, then find and use a desklamp.
OK.
> take mug 1 from desk 1
You pick up the mug 1 from the desk 1.
> think: To solve the task, I need to find and take a mug, then find and use a desklamp.
OK.
> use desklamp 1
You turn on the desklamp 1.
Status: Success

Figure 4: [Top] An AlfWorld trajectory in which the agent failed due to inefficient planning. In the reflection, the agent recognizes that it should have looked for the desklamp then the mug, not the mug then the desklamp. [Bottom] The agent is able to correct its reasoning trace and execute a sequence of actions in a concise manner.

Reflexion: Language Agents with Verbal Reinforcement Learning

Noah Shinn
Northeastern University
noahshinn024@gmail.com

Federico Cassano
Northeastern University
cassano.f@northeastern.edu

Edward Berman
Northeastern University
berman.ed@northeastern.edu

Ashwin Gopinath
Massachusetts Institute of Technology
agopi@mit.edu

Karthik Narasimhan
Princeton University
karthikn@princeton.edu

Shunyu Yao
Princeton University
shunyuy@princeton.edu

Abstract

Large language models (LLMs) have been increasingly used to interact with external environments (e.g., games, compilers, APIs) as goal-driven agents. However, it remains challenging for these language agents to quickly and efficiently learn from trial-and-error as traditional reinforcement learning methods require extensive training samples and expensive model fine-tuning. We propose *Reflexion*, a novel framework to reinforce language agents not by updating weights, but instead through linguistic feedback. Concretely, Reflexion agents verbally reflect on task feedback signals, then maintain their own reflective text in an episodic memory buffer to induce better decision-making in subsequent trials. Reflexion is flexible enough to incorporate various types (scalar values or free-form language) and sources (external or internally simulated) of feedback signals, and obtains significant improvements over a baseline agent across diverse tasks (sequential decision-making, coding, language reasoning). For example, Reflexion achieves a 91% pass@1 accuracy on the HumanEval coding benchmark, surpassing the previous state-of-the-art GPT-4 that achieves 80%. We also conduct ablation and analysis studies using different feedback signals, feedback incorporation methods, and agent types, and provide insights into how they affect performance. We release all code, demos, and datasets at https://github.com/noahshinn024/reflexion.

https://arxiv.org/abs/2303.11366

1 Introduction

Recent works such as ReAct [30], SayCan [1], Toolformer [22], HuggingGPT [23], generative agents [19], and WebGPT [17] have demonstrated the feasibility of autonomous decision-making agents that are built on top of a large language model (LLM) core. These methods use LLMs to generate text and 'actions' that can be used in API calls and executed in an environment. Since they rely on massive models with an enormous number of parameters, such approaches have been so far limited to using in-context examples as a way of teaching the agents, since more traditional optimization schemes like reinforcement learning with gradient descent require substantial amounts of compute and time.

Short on details of the model and data, but many interesting points on the development process.

See slide notes.

GPT-4 Technical Report

OpenAI*

Abstract

We report the development of GPT-4, a large-scale, multimodal model which can accept image and text inputs and produce text outputs. While less capable than humans in many real-world scenarios, GPT-4 exhibits human-level performance on various professional and academic benchmarks, including passing a simulated bar exam with a score around the top 10% of test takers. GPT-4 is a Transformer-based model pre-trained to predict the next token in a document. The post-training alignment process results in improved performance on measures of factuality and adherence to desired behavior. A core component of this project was developing infrastructure and optimization methods that behave predictably across a wide range of scales. This allowed us to accurately predict some aspects of GPT-4's performance based on models trained with no more than 1/1,000th the compute of GPT-4.

https://arxiv.org/abs/2303.08774

1 Introduction

This technical report presents GPT-4, a large multimodal model capable of processing image and text inputs and producing text outputs. Such models are an important area of study as they have the potential to be used in a wide range of applications, such as dialogue systems, text summarization, and machine translation. As such, they have been the subject of substantial interest and progress in recent years [1–34].

One of the main goals of developing such models is to improve their ability to understand and generate natural language text, particularly in more complex and nuanced scenarios. To test its capabilities in such scenarios, GPT-4 was evaluated on a variety of exams originally designed for humans. In these evaluations it performs quite well and often outscores the vast majority of human test takers. For example, on a simulated bar exam, GPT-4 achieves a score that falls in the top 10% of test takers. This contrasts with GPT-3.5, which scores in the bottom 10%.

On a suite of traditional NLP benchmarks, GPT-4 outperforms both previous large language models and most state-of-the-art systems (which often have benchmark-specific training or hand-engineering). On the MMLU benchmark [35, 36], an English-language suite of multiple-choice questions covering 57 subjects, GPT-4 not only outperforms existing models by a considerable margin in English, but also demonstrates strong performance in other languages. On translated variants of MMLU, GPT-4 surpasses the English-language state-of-the-art in 24 of 26 languages considered. We discuss these model capability results, as well as model safety improvements and results, in more detail in later sections.

This report also discusses a key challenge of the project, developing deep learning infrastructure and optimization methods that behave predictably across a wide range of scales. This allowed us to make predictions about the expected performance of GPT-4 (based on small runs trained in similar ways) that were tested against the final run to increase confidence in our training.

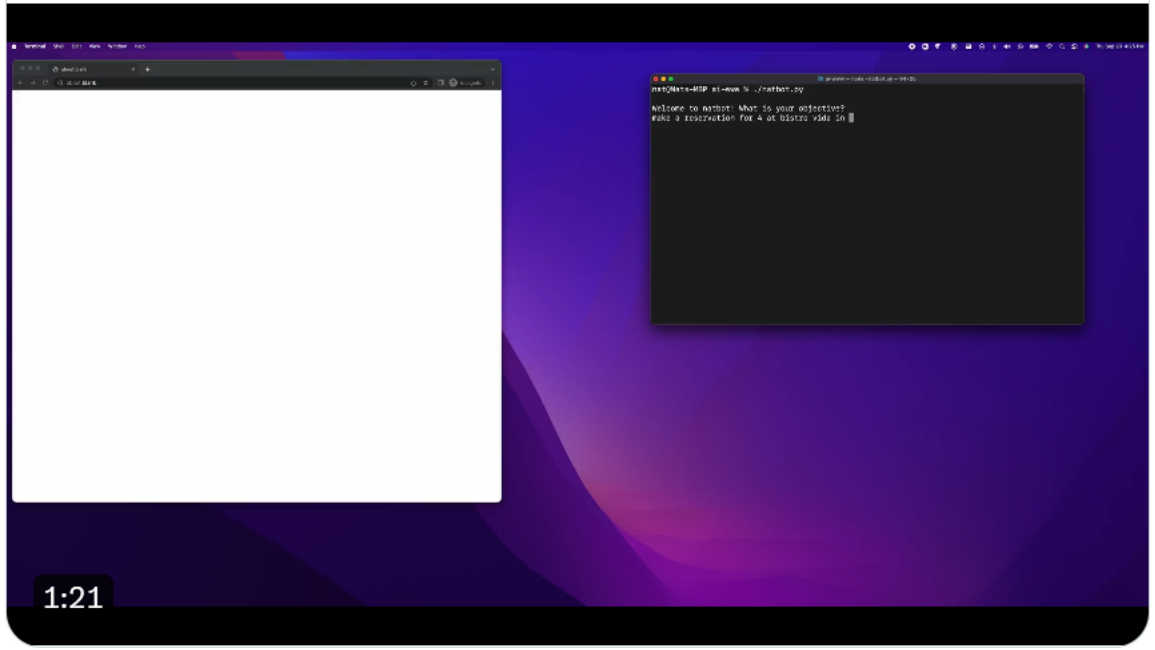
Despite its capabilities, GPT-4 has similar limitations to earlier GPT models [1, 37, 38]: it is not fully reliable (e.g. can suffer from "hallucinations"), has a limited context window, and does not learn

*Please cite this work as "OpenAI (2023)". Full authorship contribution statements appear at the end of the document. Correspondence regarding this technical report can be sent to gpt4-report@openai.com

AaltoMediaAI @aaltomediaai · Feb 10, 2023

If you'd like to see how a language model can use web pages based on user instructions, see github.com/nat/natbot/blo.... The whole thing is less than 1k line of Python. Old but relevant again, with the Bing-ChatGPT integration.

> **Nat Friedman** @natfriedman · Sep 30, 2022
>
> I've been playing with using GPT-3 to control a browser the last couple days. Here's a quick demo. As you can see it's pretty neat! But also quite flakey.
>
> Will publish the source code shortly for others to try and improve.
>
> Show this thread



https://github.com/nat/natbot/blob/main/natbot.py