Large Language Models: A Brief Introduction

Aalto University's Game Analysis course 2024

Prof. Perttu Hämäläinen

Vocabulary

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

GPT: Generative Pretrained Transformer

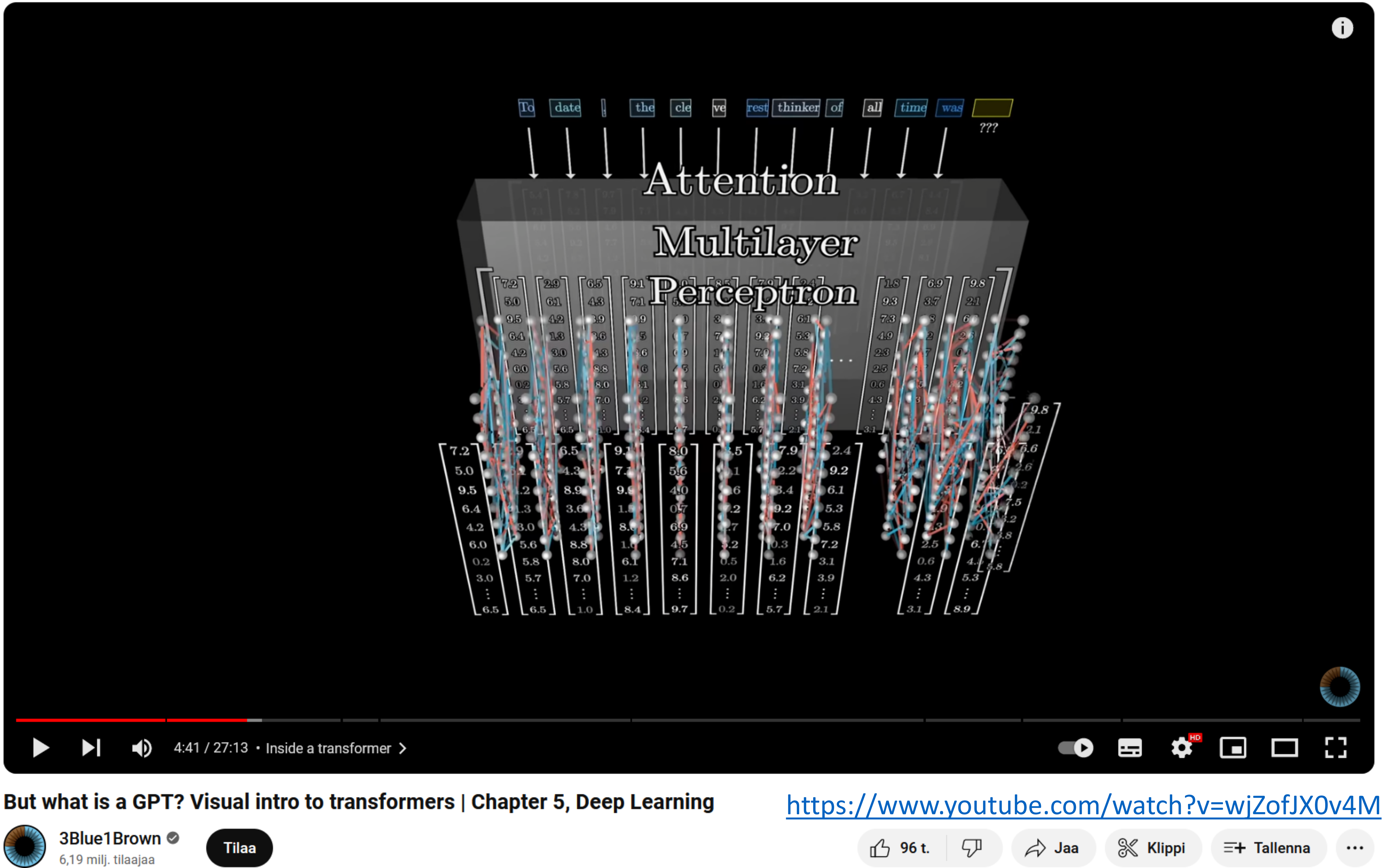
Generative: The model generates samples (text snippets, images…)

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.

Context size: The maximum amount of text that a LLM can "hold in its mind" at once. If you provide more text, rest will be simply ignored.

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



But what is a GPT? Visual intro to transformers | Chapter 5, Deep Learning

3Blue1Brown
6,19 milj. tilaajaa

https://www.youtube.com/watch?v=wjZofJX0v4M

Some notes on the videos:

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

GPT-3 (2020)

- First real LLM that started the current craze
- ChatGPT 3.5 = GPT-3 fine-tuned with human feedback
- GPT-3 established an emergent property of LLMs: **few-shot learning** (a.k.a. **in-context learning**)

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.

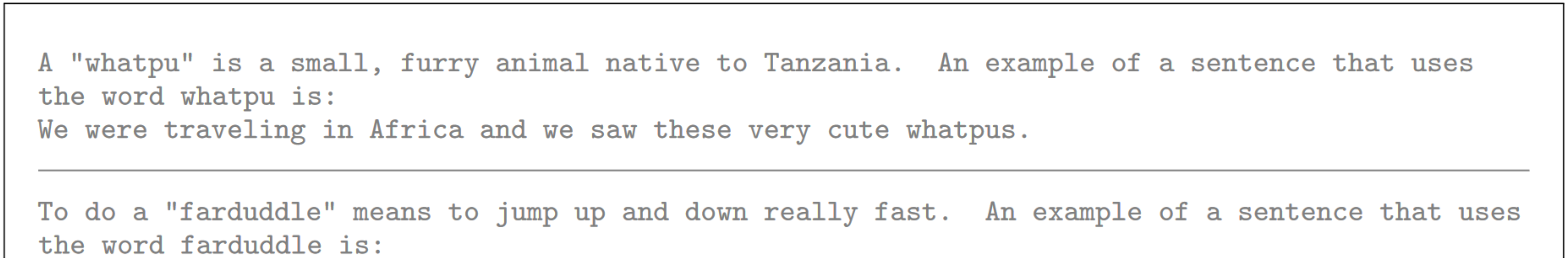
1 Introduction

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

Few-shot learning example

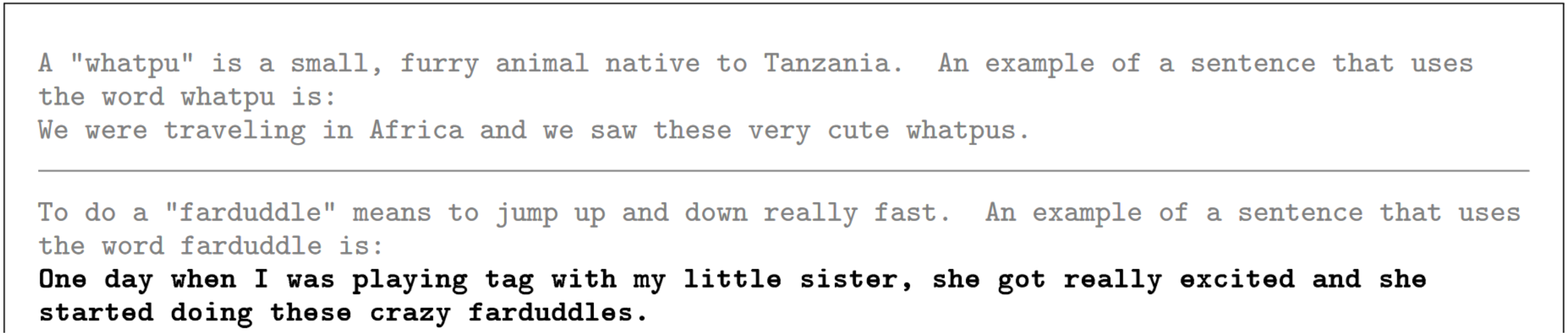


A "whatpu" is a small, furry animal native to Tanzania. An example of a sentence that uses the word whatpu is:
We were traveling in Africa and we saw these very cute whatpus.

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

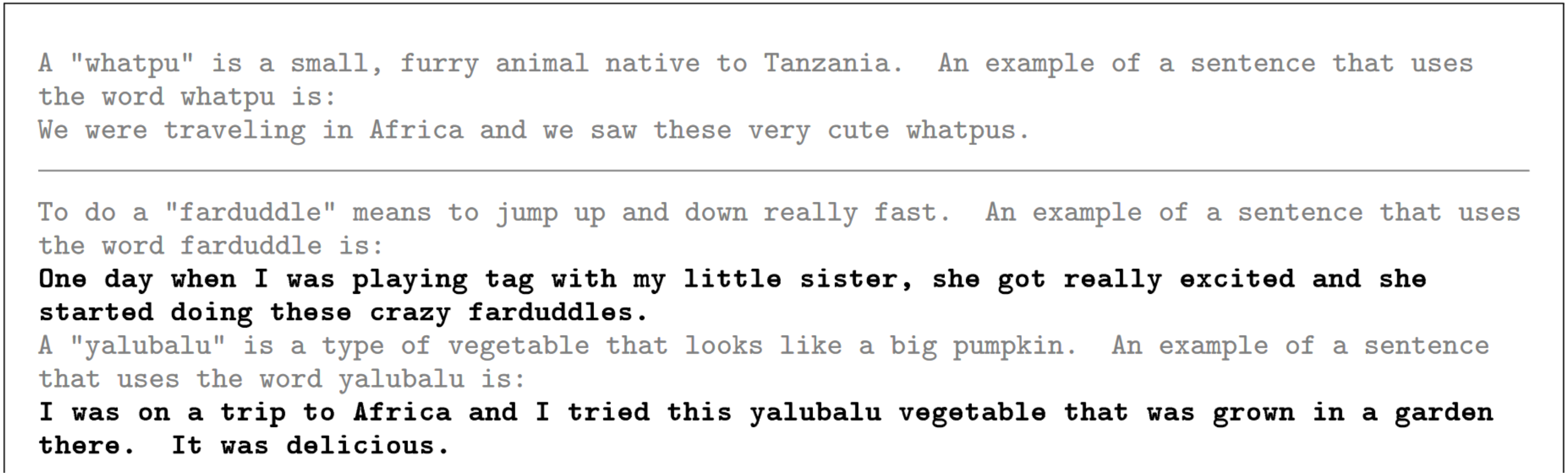


A "whatpu" is a small, furry animal native to Tanzania. An example of a sentence that uses the word whatpu is:
We were traveling in Africa and we saw these very cute whatpus.

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

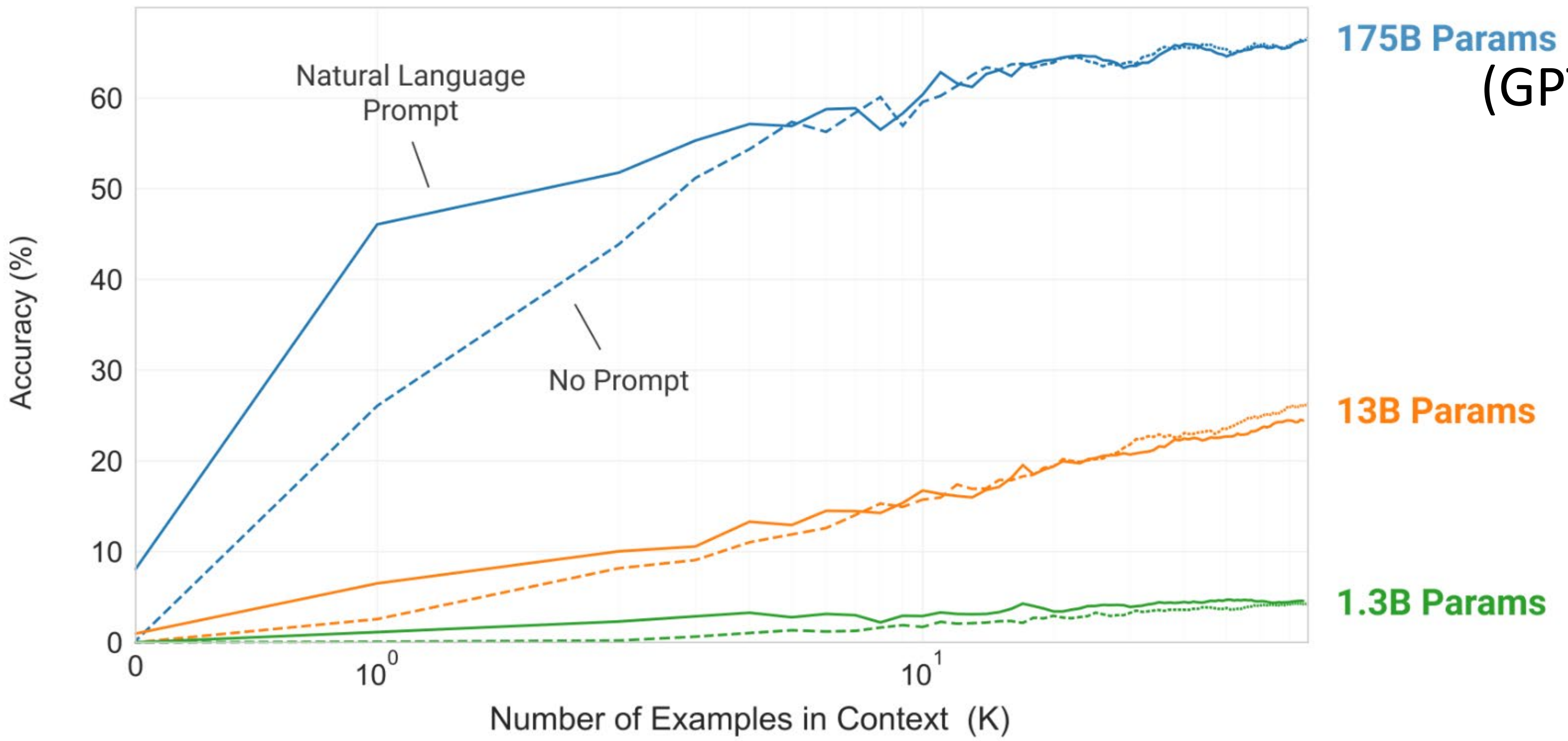


A "whatpu" is a small, furry animal native to Tanzania. An example of a sentence that uses the word whatpu is:
We were traveling in Africa and we saw these very cute whatpus.

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

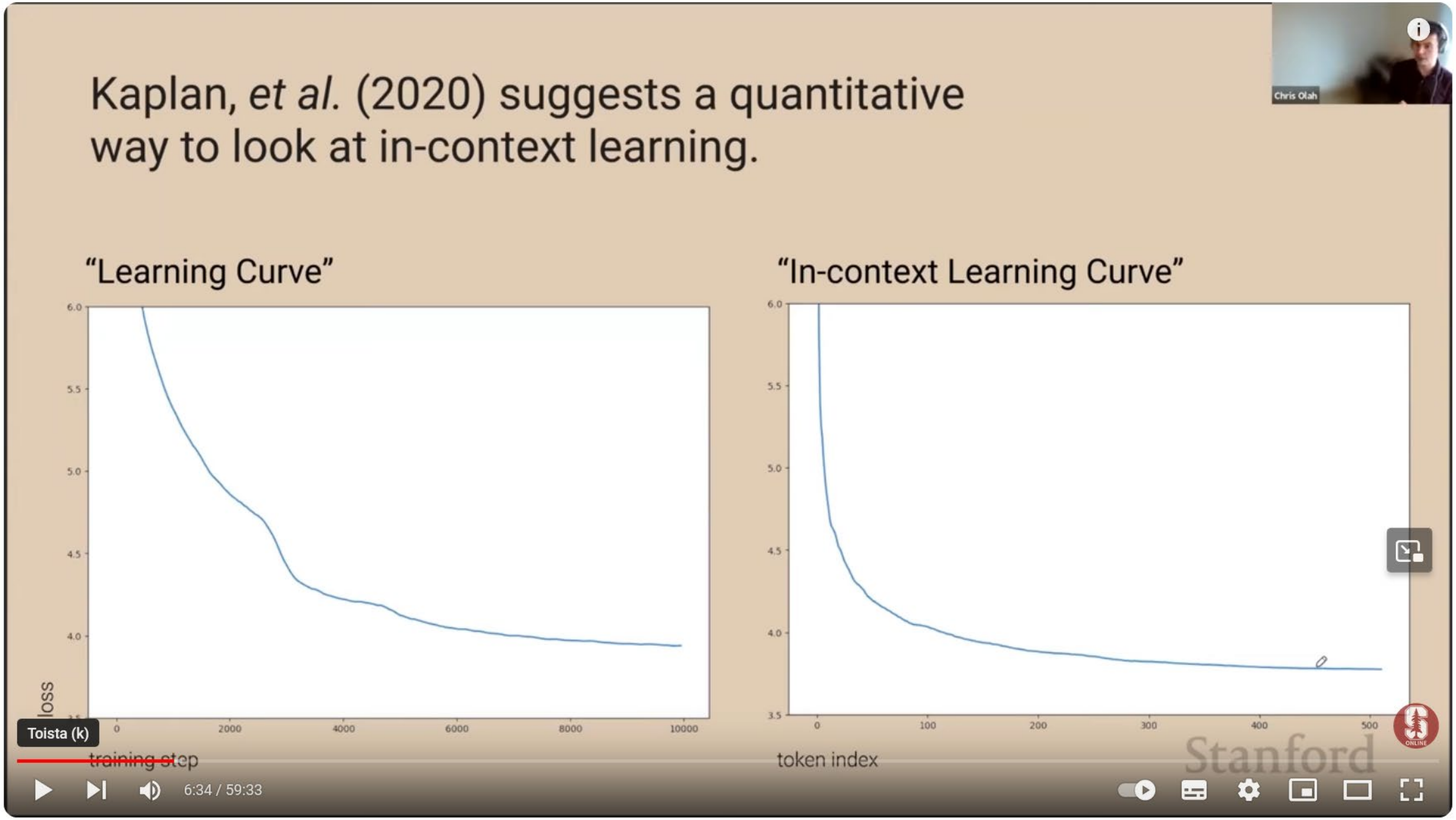
Two key lessons from the GPT-3 paper



(GPT-3 davinci)

(≈GPT-2)

1. Adding few-shot examples makes a huge difference
2. Few-shot learning only emerges in sufficiently large models



Kaplan, *et al.* (2020) suggests a quantitative way to look at in-context learning.

“Learning Curve”

“In-context Learning Curve”

Stanford CS25: V1 I Transformer Circuits, Induction Heads, In-Context Learning

https://www.youtube.com/watch?v=pC4zRb_5noQ

Stanford Online

598 t. tilaajaa

Tilaa

311

Jaa

Tallenna

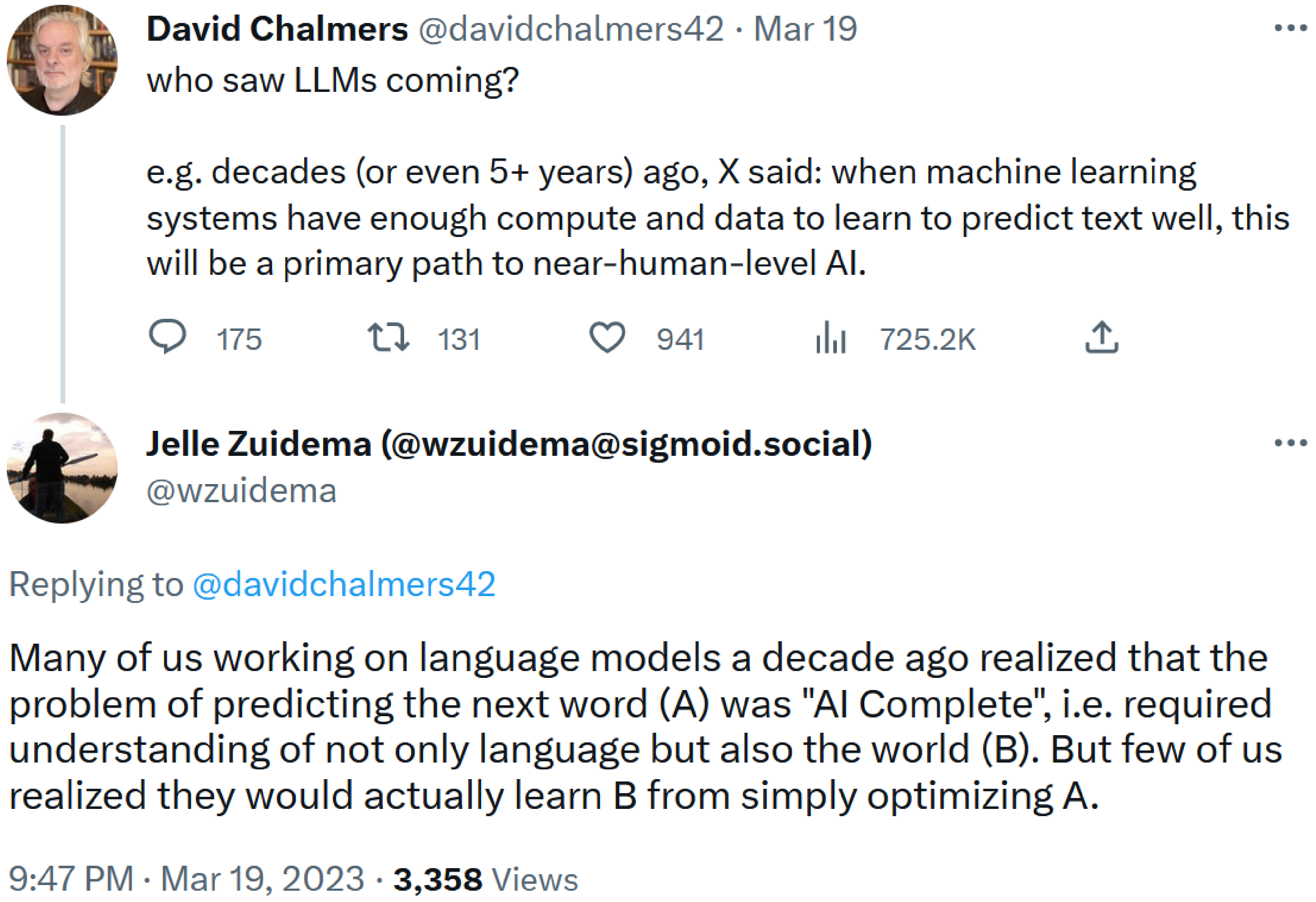
ChatGPT (2022)

- The initial version (ChatGPT-3.5) was just a finetuned variant of GPT-3
- The real innovation was the chat interface that allowed people easy and natural way to start interacting and exploring what the model can do

What is happening?

- To do the next token prediction really well, an LLM needs to “understand” 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.

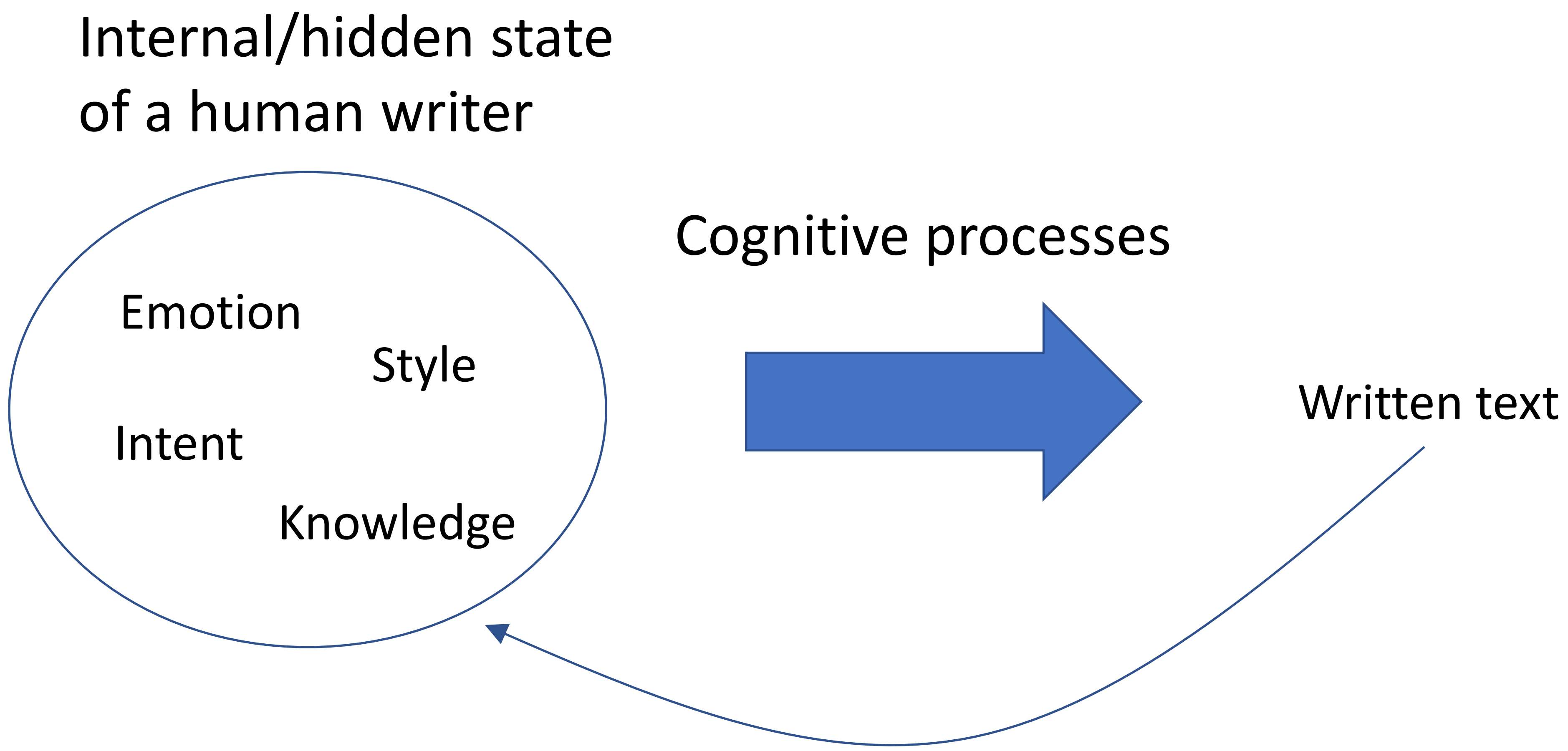
9:47 PM · Mar 19, 2023 · **3,358** Views

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.

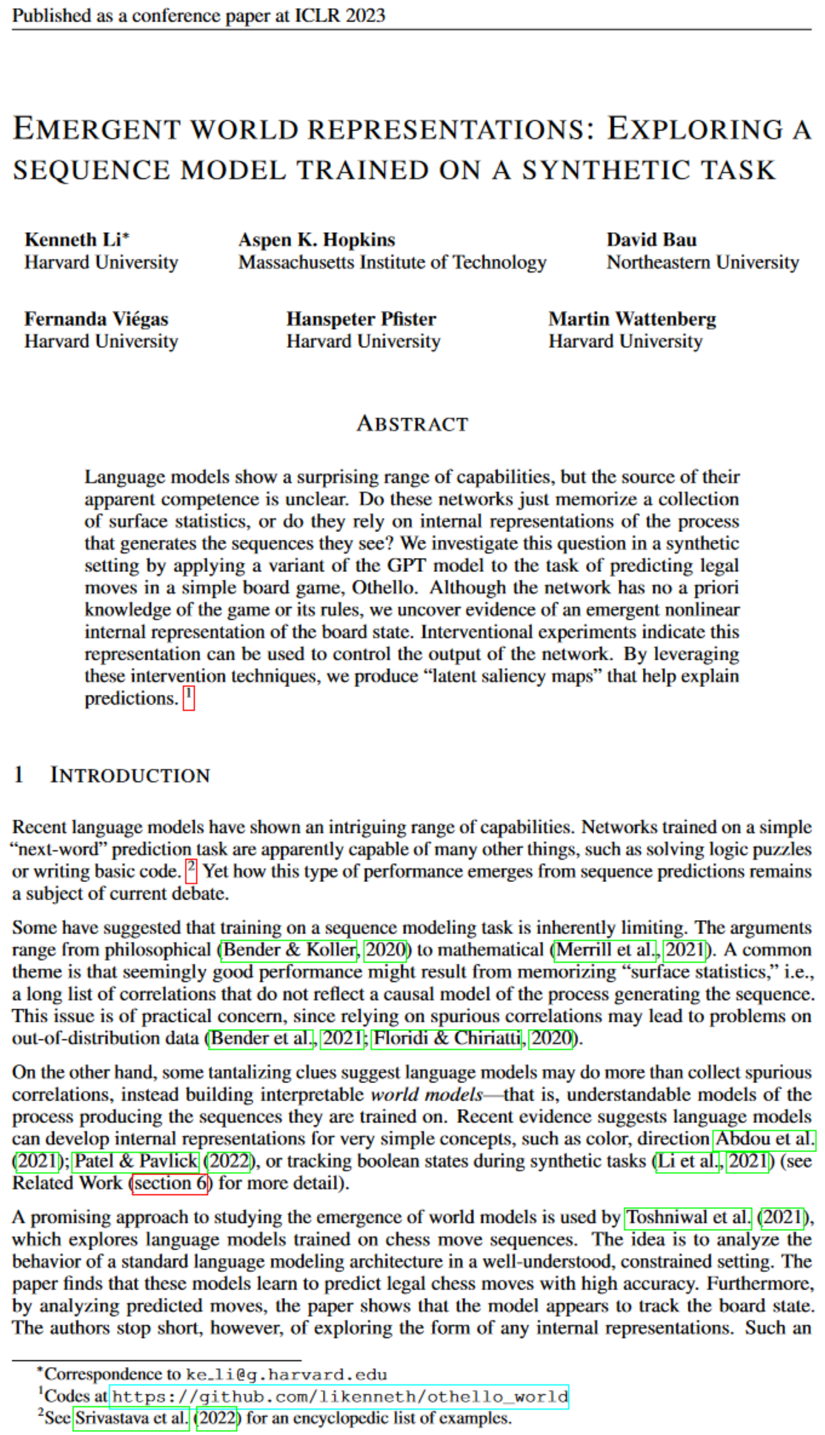
The supervised training of high-capacity models on large labeled datasets is critical to the recent success of deep learning techniques for a wide range of applications such as image classification (Krizhevsky et al., 2012), speech recognition (Hinton et al., 2012), and machine translation (Wu et al., 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

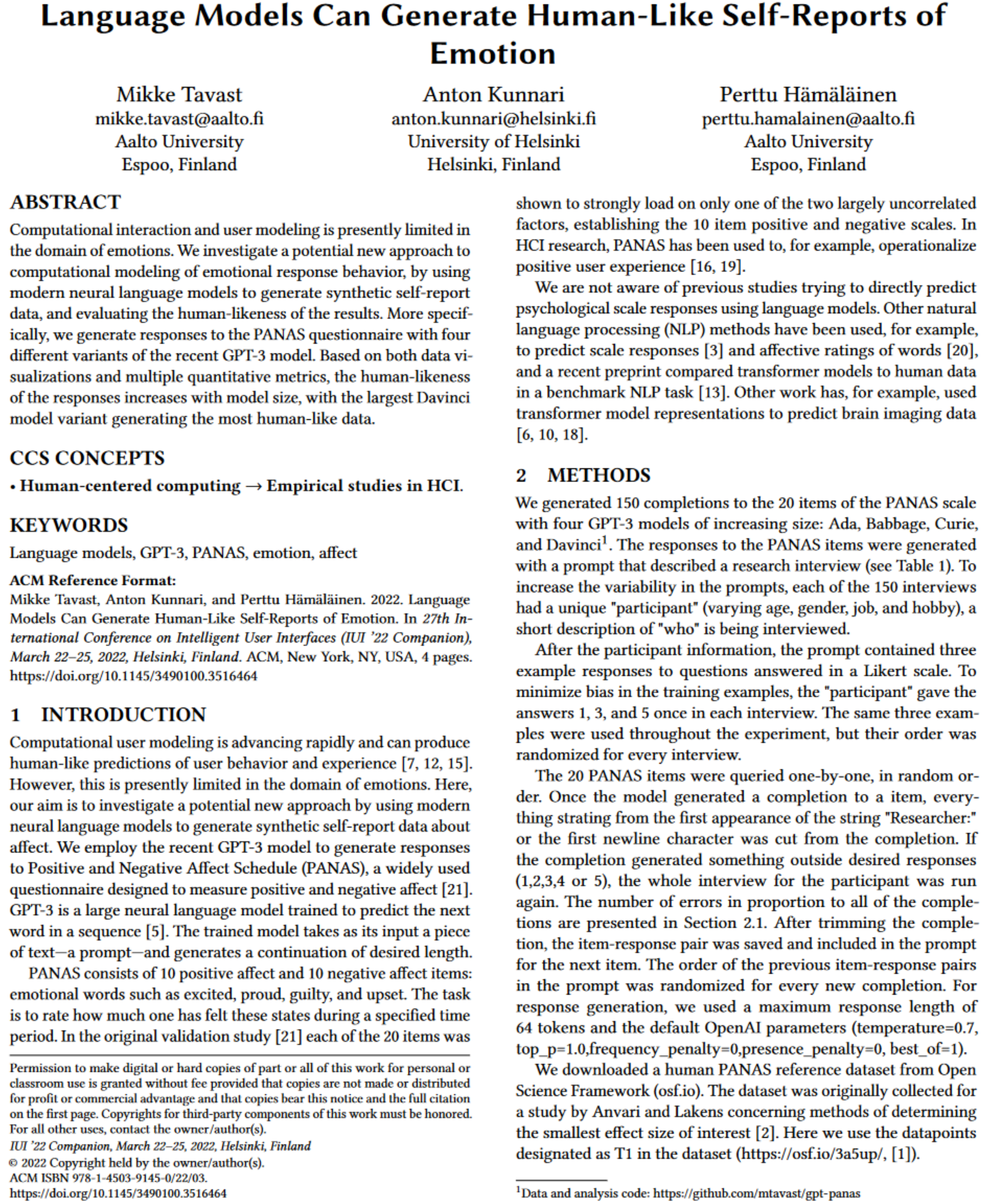
Recent language models have shown an intriguing range of capabilities. Networks trained on a simple "next-word" prediction task are apparently capable of many other things, such as solving logic puzzles or writing basic code. [2] Yet how this type of performance emerges from sequence predictions remains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

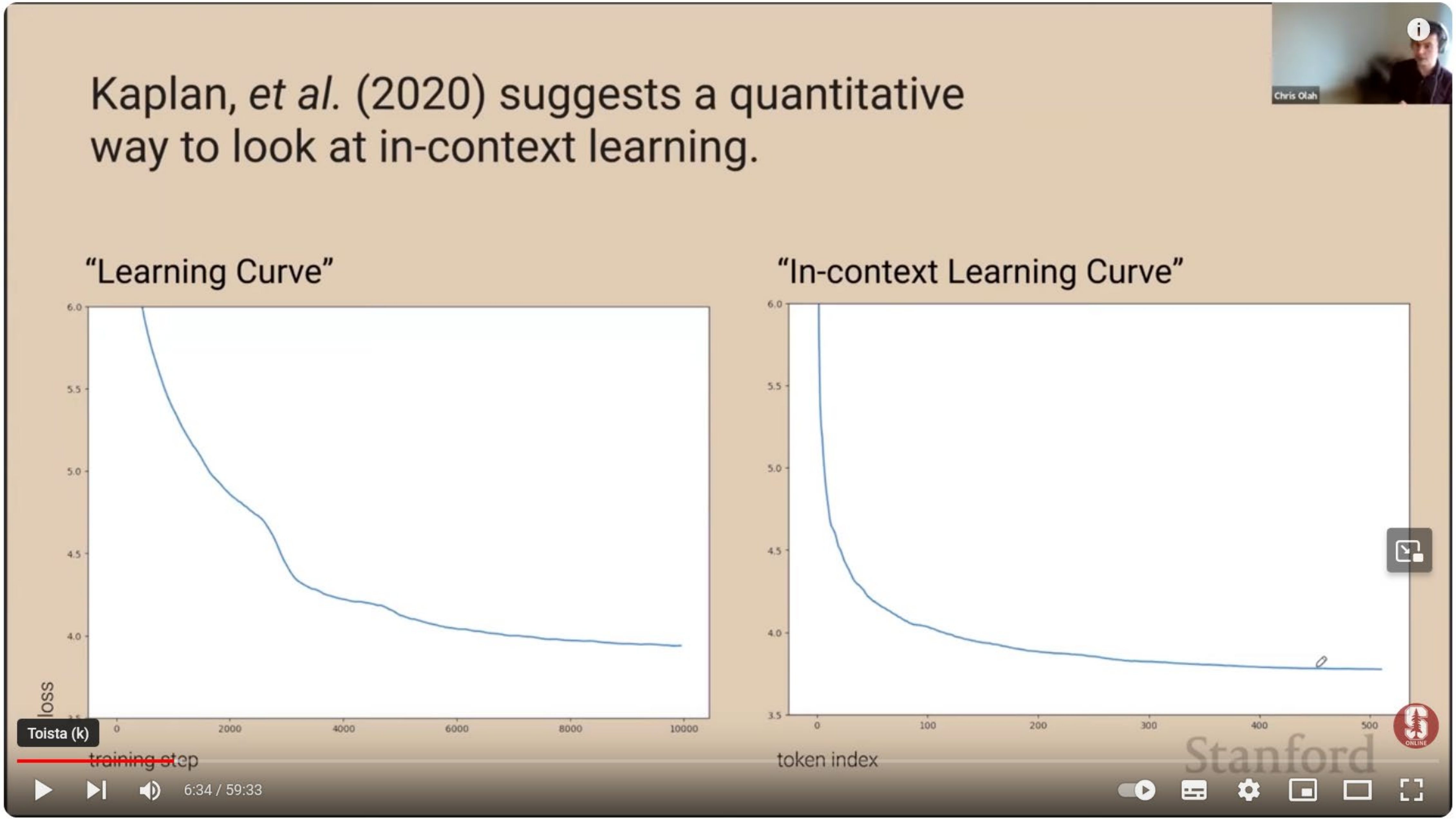
We generated 150 completions to the 20 items of the PANAS scale with four GPT-3 models of increasing size: Ada, Babbage, Curie, and Davinci[1]. The responses to the PANAS items were generated with a prompt that described a research interview (see Table 1). To increase the variability in the prompts, 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



Kaplan, *et al.* (2020) suggests a quantitative way to look at in-context learning.

Stanford CS25: V1 I Transformer Circuits, Induction Heads, In-Context Learning

https://www.youtube.com/watch?v=pC4zRb_5noQ

Stanford Online
598 t. tilaajaa
Tilaa
311
Jaa
Tallenna

All models are wrong, but some are useful

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

Why do LLMs make factual errors?

- LLMs are infamous for generating text that sounds plausible but may not be factually correct
- The training goal is to minimize the average next token prediction error over all the training data
- The datasets are so large that LLMs don't have the capacity to memorize everything
- Getting a particular fact wrong only affects the error for a small portion of the data
- Not understanding writing tone, emotion, and style will affect the error for almost all the data

$\Rightarrow$It makes sense to prioritize tone and style over facts. LLMs are only doing what they are engineered to do.

GPT-4, Gemini and others

- 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

Non-OpenAI models

- Google Gemini
- Anthropic Claude
- Meta Llama 3: The best fully open source model family. The biggest variants very near the best commercial ones, but also decent small models one can run on a personal computer
- Microsoft Phi 3: Another high-quality and small open source model, runs on a potato.
- For running open source models on your computer, check out https://lmstudio.ai/ and https://ollama.com/

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?

How to use AI for data coding?

- Prompt LLMs (ChatGPT etc.) with examples and data to code
- Provide the examples by manually coding at least some data
- More examples => the LLM will follow your coding style more accurately

Example prompt

Below, I will give you a game experience description from a research experiment about experiencing video games as art. Your task is to assist in analyzing the experience description.

The research question is: What feelings, emotions and sensations do players feel when experiencing video games as art?

Please carry out the following task:

- Identify and highlight statements relevant to the research question.
- Respond by repeating the original text, but surrounding the statements with double asterisks (**), as if they were bolded text in a Markdown document. Add the codes for each statement as superscript, between superscript tags <sup>, </sup>. If there's multiple codes per statement, separate them with semicolons.

Below, I first give you an example of the output you should produce given the input.

After that, I give you the actual input to process.

<add examples here, followed by the text to process>

Example LLM input and output:

Input:

I was overlooking a beautiful valley surrounded by snow-capped mountains. There was a massive lake in the middle. Everything seemed so peaceful and serene it was like being inside a painting.

Output:

I was overlooking a beautiful valley surrounded by snow-capped mountains. There was a massive lake in the middle. **Everything seemed so peaceful and serene it was like being inside a painting.** Peacefulness; serenity; being inside a painting

Example prompt

We will use LLMCode, an open source toolkit that implements such prompting under the hood.

https://github.com/PerttuHamalainen/LLMCode

Below, I will give you a game experience description from a research experiment about experiencing video games as art. Your task is to assist in analyzing the experience description.

The research question is: What feelings, emotions and sensations do players feel when experiencing video games as art?

Please carry out the following task:

- Identify and highlight statements relevant to the research question.
- Respond by repeating the original text, but surrounding the statements with double asterisks (**), as if they were bolded text in a Markdown document. Add the codes for each statement as superscript, between superscript tags <sup>, </sup>. If there's multiple codes per statement, separate them with semicolons.

Below, I first give you an example of the output you should produce given the input.

After that, I give you the actual input to process.

<add examples here, followed by the text to process>

LLMCode other key features

- Verifying that the output contains no hallucinated text, e.g., imaginary quotes
- Supporting multiple workflows with varying degrees of automation – you choose what parts of the process you want to do yourself and what parts to automate