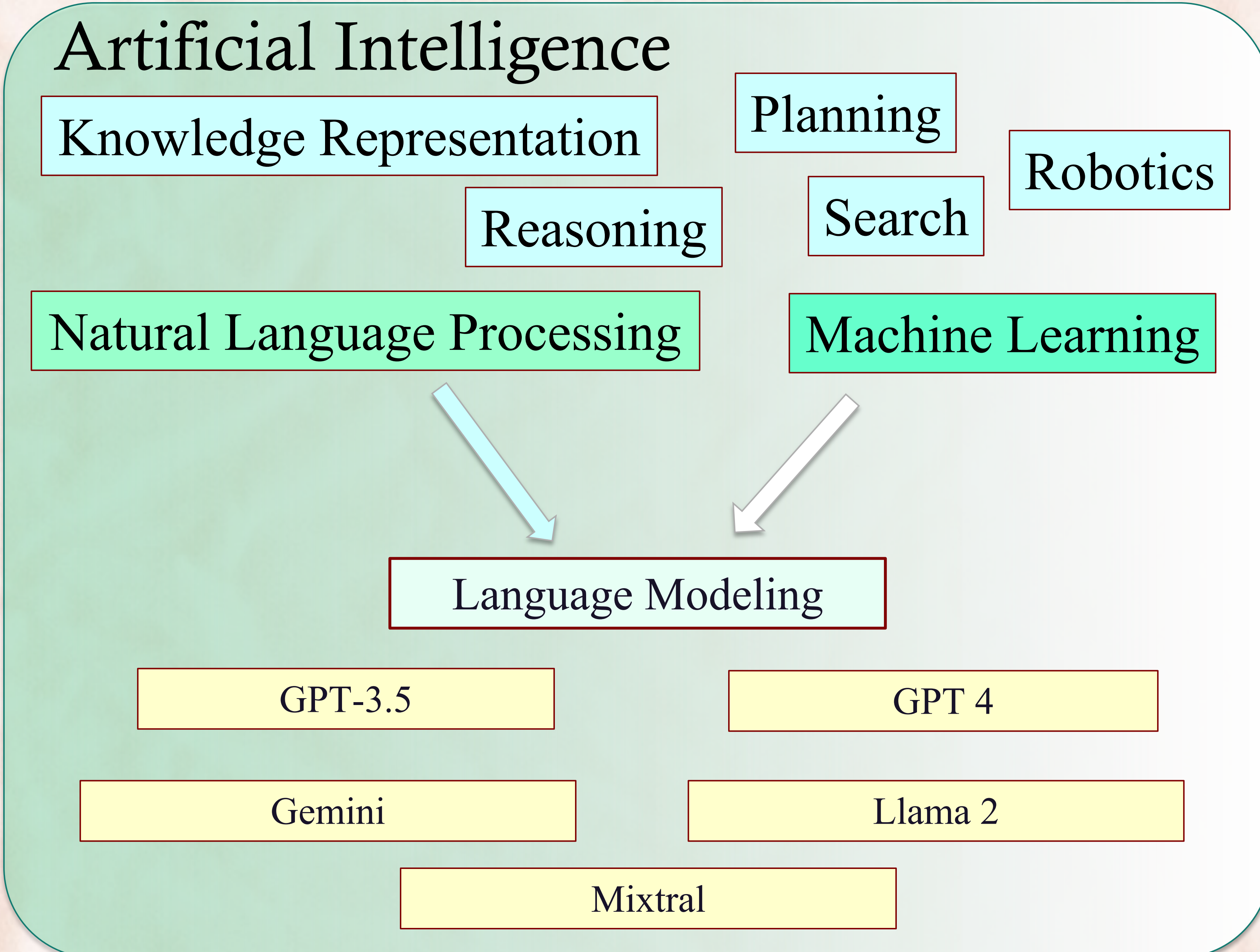
Strenghts and Weaknesses of LLMs for Teaching and Learning

Razvan C. Bunescu

Department of Computer Science

CCI @ UNCC

razvan.bunescu@uncc.edu



Artificial Intelligence
Knowledge Representation
Planning
Robotics
Reasoning
Search
Natural Language Processing
Machine Learning
Language Modeling
GPT-3.5
GPT 4
Gemini
Llama 2
Mixtral

Data Science International Summer School

https://datascience.ase.ro

21.07	22.07	23.07	24.07	25.07	26.07	27.07
Breakfast						
Dan Nicolae Statistics for Data Science	Statistical learning (hands-on)	*Razvan* Machine Learning (hands-on)	Time Series (incl. hands-on)	Knowledge Graphs for conversational AI (incl. (hands-on)	FAIR Data and best practices in data sharing	Operationalizing Machine Learning pipelines
Lunch break and socializing activities						
Data Science with Python (hands-on)	Intro to Machine Learning *Razvan*	Deep Learning with Neural Networks (incl. hands-on)	Introduction to Graph Data (incl. hands-on) *Dumitru (Titi)*	Social Event 	Data enrichment (incl. hands-on)	Operationalizing Machine Learning pipelines (hands-on)
ee time					Free time	
Dinner						

Main Findings

- **GPT-4 is much better than GPT-3.5**:
  - Reasoning, Theory of mind, IQ-test sequences, ML / math, …
- **ChatGPT can be correct today and wrong tomorrow**:
  - Due to **sampling**; need to use API for **greedy decoding**.
  - Due to OpenAI changing the underlying LM implementation.
- **ChatGPT-4 is great for learning:**
  - When asked for code to solve problem, it explains the code.
    - Line by line.
    - Overall approach.
  - When asked a Yes/No math question, it provides proof.
  - **But …**

Main Findings

- **ChatGPT behavior is non-human**:
  - Solves sophisticated problems from a wide range of areas, but makes trivial mistakes a human expert would not.
    - **Student**: sophisticated abilities may lead novices to trust it blindly, including its mistakes.
      - Exacerbated when AI is smart today, dumb tomorrow.
    - **Instructor**: very useful tool for a domain expert who can fix occasional mistakes.
      - Related findings on next slide.

Related Findings from 2023 Literature

- AI tools for generating code are especially beneficial to students who have more programming knowledge.
  - Studying the effect of AI Code Generators on Supporting Novice Learners in Introductory Programming
    - "*learners who performed better on the Scratch pre-test (who presumably had more prior programming knowledge) gained a significant improvement in the retention post-test after having access to the AI code generator during the training phase.*"

- Need ability to critically evaluate LMs output:
  - Promptly: Using Prompt Problems to Teach Learners How to Effectively Utilize AI Code Generators
    - "*The ease with which code can now be generated has resulted in a shift in focus towards reading, **understanding and evaluating LLM-generated code***".

ITCS 4111/5111:Introduction to NLP

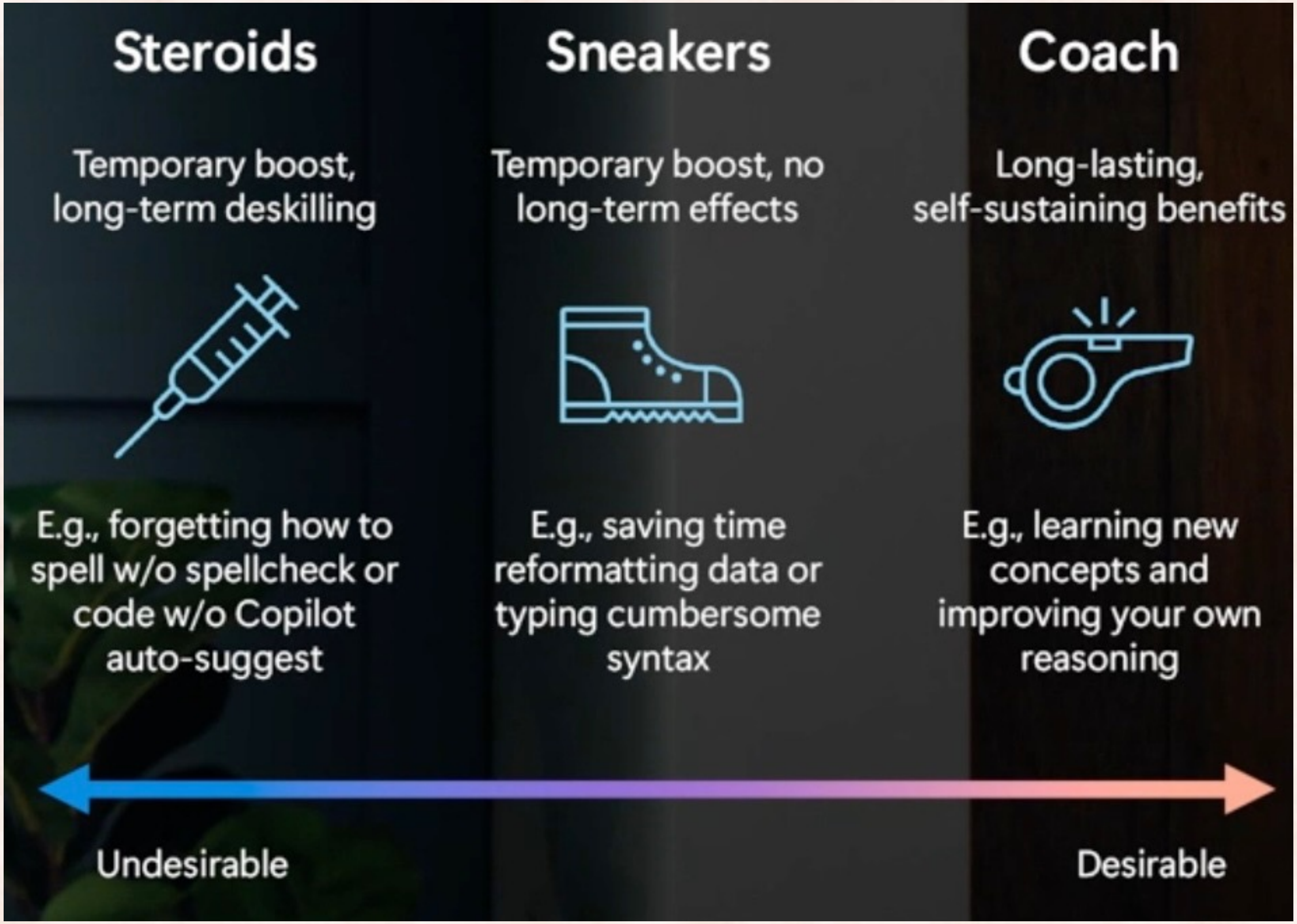
- Awareness of the **strengths and weaknesses of LMs**.

- **Students can use LMs as a TA** as much as possible:
  - Ask questions about course material.
    - Ask it to explain concepts on the slides.
  - But beware explanations may contain bugs …

- **All homework must be the student's own**:
  - Duplicate or plagiarized work will be promptly reported.
  - Even when undetected, use of LMs for HW will be detrimental:
    - **Blindly use LM for HW => poor performance on in-class quizzes and exams**.

A Sports Analogy for Understanding Different Ways to Use AI

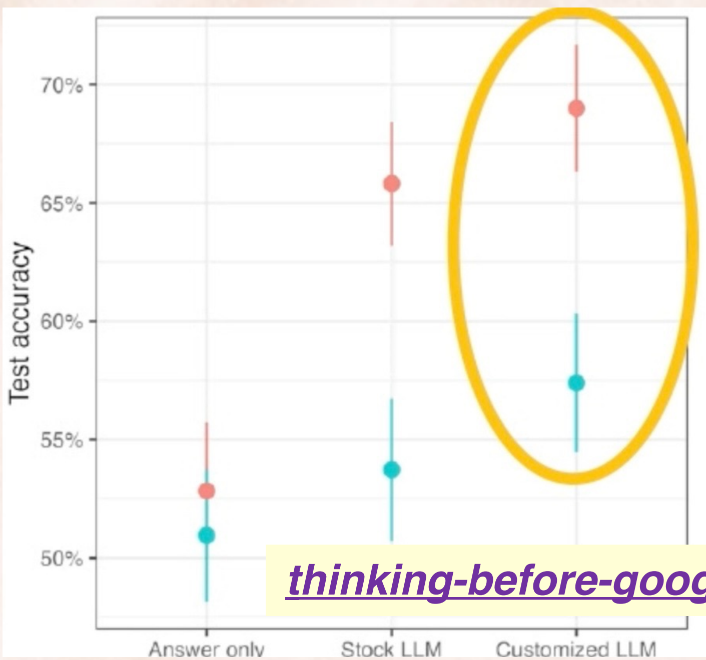
aka.ms/ssc-hbr



Math Education with Large Language Models: Peril or Promise?

Abstract: The widespread availability of large language models (LLMs) has provoked both fear and excitement in the domain of education. On one hand, there is the concern that students will offload their coursework to LLMs, limiting what they themselves learn. On the other hand, there is the hope that LLMs might serve as scalable, personalized tutors.

Here we conduct a large, pre-registered experiment involving 1200 participants to investigate how exposure to LLM-based explanations affect learning.



https://dx.doi.org/10.2139/ssrn.4641653

LLM explanations positively impacted learning relative to seeing on the correct answer

There were substantial benefits to attempting problems unassisted before consulting the LLM

And directional evidence that customized pre-prompts can lead to somewhat better explanations

thinking-before-googling *leads to better recall than googling-right-away*

Simple reasoning: GPT-3.5

[credits to Dan Nicolae]

R

Dan sits next to Titi in a room. One of them is kind and the other is mean. The person next to Dan is mean. Who is kind?

If the person sitting next to Dan is mean, then Titi must be kind. Since there are only two people in the room, and one of them has been identified as mean, the other person (Titi) must be the kind one.

Simple reasoning: GPT-4

R

Dan sits next to Titi in a room. One of them is kind and the other is mean. The person next to Dan is mean. Who is kind?

If the person next to Dan is mean, and that person is Titi, then Dan is the kind one.

IQ-test Number Sequences: GPT-3.5 & 4

R

Provide the next number in the sequence 1, 2, 4, 7, 11, 15, ...

was supposed to be 16

To find the pattern in the sequence, we can observe the differences between consecutive terms:

2 - 1 = 1

4 - 2 = 2

7 - 4 = 3

11 - 7 = 4

15 - 11 = 4

Looking at the differences, we can see that they are increasing by 1 each time. Therefore, we can assume that the next difference would be 4 + 1 = 5.

To find the next number in the sequence, we add this difference to the last number given:

15 + 5 = 20

Therefore, the next number in the sequence is 20.

Kolmogorov Complexity: GPT-3.5 & 4

R

Write Python code for generating this infinite sequence.

Certainly! Here's an example of Python code that generates the infinite sequence based on the given pattern:

python Copy code

```python
def generate_sequence():
    sequence = [1]  # Initialize the sequence with the first number
    diff = 1  # Initialize the initial difference

    while True:
        next_number = sequence[-1] + diff  # Calculate the next number
        sequence.append(next_number)  # Add the next number to the sequence
        diff += 1  # Increase the difference by 1 for the next iteration
        yield next_number

# Test the sequence generator
sequence_generator = generate_sequence()
for i in range(10):  # Generate and print the first 10 numbers in the sequen
    print(next(sequence_generator))
```

Occam’s Razor

William of Occam (1288 – 1348)

English Franciscan friar, theologian and philosopher.

“*Entia non sunt multiplicanda praeter necessitatem*”

- Entities must not be multiplied beyond necessity.

i.e. Do not make things needlessly complicated.

i.e. Prefer the simplest hypothesis that fits the data.

Occam's Razor and Machine Learning

- ML problem: Find a model **M**

  that is *simple* + that *fits the training data*.

$$\hat{\mathbf{M}} = \underset{\mathbf{M}}{\operatorname{argmin}} \; \mathit{Complexity}(\mathbf{M}) + \mathit{Error}(\mathbf{M}, \mathit{Data})$$

- **Inductive hypothesis**: Models that perform well on training examples are expected to do well on test (unseen) examples.

- **Occam's Razor**: Simpler models are expected to do better than complex models on test examples (assuming similar training performance).

From Occam's Razor to Kolmogorov Complexity to Intelligence

- **Occam's Razor**: Prefer the **simplest** hypothesis that fits the data.

- **Kolmogorov Complexity** = the length of the shortest program that generates the data.
  - 1, 2, 4, 7, 11, 16, 22, 29, 37, 46, 56, …
  - 1, 3, 6, 11, 18, 29, …
  - 1, 2, 3, 5, 5, 8, 7, 11, 9, …

- **Intelligence** = the ability to apply Occam's Razor.

What is Intelligence? How to measure It?

- Marcus Hutter:
  - "Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability"
  - Book, 2005.
- Shane Legg:
  - "**Machine Super Intelligence**"
  - PhD Thesis, 2008.
    - http://www.vetta.org/documents/Machine_Super_Intelligence.pdf
- A definition of intelligence that can be applied to humans, animals, machine.

Sequence Prediction & Occam's Razor

- **Standard IQ tests implicitly test the ability to use Occam's razor**.
- Consider the following sequence:

  1, 3, 5, 7, ...

- What do you predict will come next?
  - What process do you think is generating these numbers?
    - M_1: $2n + 1$

      => 9
    - M_2: $2n - 1 + (n - 1)(n - 2)(n - 3)(n - 4)$

      => 33

Sequence Complexity

- Some sequences are more difficult to predict than others.
- IQ tests correlate IQ with the difficulty of the sequences that the agent correctly preditcs.
- How to measure sequence difficulty?

- **Kolmogorov Complexity** (s) = the length of the smallest program that can generate the sequence s.

- But intelligence is not passive! Agents can act => use concepts from *reinforcement learning*.

Universal Intelligence (Informal Definition)

http://www.vetta.org/documents/Machine_Super_Intelligence.pdf

The intelligence of an agent is the expected reward that the agent obtains over the space of all computable environments, where (*very important*) the distribution of these environments is such that environments that have lower Kolmogorov complexity (i.e. *have simpler descriptions in terms of their distribution of observations and rewards as functions of action histories*) are more likely to happen (the "universal distribution"). This effectively rewards agents for correctly using Occam's razor, by weighting the environments according to their complexity.

Occam’s Razor and Science

- **Abstract**: Occam's razor – the principle of simplicity – has recently been attacked as a cultural bias without rational foundation. Increasingly, belief in pseudoscience and mysticism is growing. I argue that inclusion of Occam's razor is an essential factor that distinguishes science from superstition and pseudoscience. I also describe how the razor is embedded in Bayesian inference and argue that *science is primarily the means to discover the simplest descriptions of our world*.

  Razor sharp: The role of Occam's razor in science
  Johnjoe McFadden, Annals of NYAS, 2023
  https://doi.org/10.1111/nyas.15086

Back to ChatGPT

ChatGPT likes to generate explanations

Certainly! Here's an example of Python code that generates the infinite sequence based on the given pattern:

python Copy code

```python
def generate_sequence():
    sequence = [1]  # Initialize the sequence with the first number
    diff = 1  # Initialize the initial difference

    while True:
        next_number = sequence[-1] + diff  # Calculate the next number
        sequence.append(next_number)  # Add the next number to the sequence
        diff += 1  # Increase the difference by 1 for the next iteration
        yield next_number

# Test the sequence generator
sequence_generator = generate_sequence()
for i in range(10):  # Generate and print the first 10 numbers in the sequen
    print(next(sequence_generator))
```

This code defines a `generate_sequence` function that uses a generator to yield the next number in the sequence indefinitely. It starts with the first number (1) and calculates each subsequent number based on the previous number and the increasing difference. The code then tests the generator by printing the first 10 numbers in the sequence. You can modify the range or use `next(sequence_generator)` to generate more numbers as needed.

IQ-test Number Sequences: GPT-3.5

R

Provide the next number in the sequence 1, 3, 6, 11, 18, 29, ...

To find the pattern in the sequence, we can observe the differences between consecutive terms:

3 - 1 = 2

6 - 3 = 3

11 - 6 = 5

18 - 11 = 7

29 - 18 = 11

Looking at the differences, we can see that they are increasing by 1 each time: 2, 3, 5, 7, 11.

To find the next number in the sequence, we add the next difference, which would be 11 + 1 = 12, to the last number given:

29 + 12 = 41

Therefore, the next number in the sequence is 41.

IQ-test Number Sequences: GPT-4

R

Provide the next number in the sequence 1, 3, 6, 11, 18, 29, ...

The pattern seems to be adding an incrementally increasing number each time. Let's examine it:

Between 1 and 3, the difference is 2.
Between 3 and 6, the difference is 3.
Between 6 and 11, the difference is 5.
Between 11 and 18, the difference is 7.
Between 18 and 29, the difference is 11.

So the differences are 2, 3, 5, 7, 11, which are the prime numbers in order.

The next prime number after 11 is 13, so the next number in the sequence should be 29 + 13 = 42.

GPT-4 Today 👍

R

Provide the next number in the sequence 1, 2, 3, 5, 5, 8, 7, 11, 9, ...

The pattern seems to involve two interleaved sequences:

Sequence 1: 1, 3, 5, 7, 9 - These are all odd numbers, increasing by 2 each time.

Sequence 2: 2, 5, 8, 11 - These numbers are increasing by 3 each time.

Interleaved, they create the sequence you provided:

1 (from Sequence 1),

2 (from Sequence 2),

3 (from Sequence 1),

5 (from Sequence 2),

5 (from Sequence 1),

8 (from Sequence 2),

7 (from Sequence 1),

11 (from Sequence 2),

9 (from Sequence 1).

So, the next number should be from Sequence 2 and it should be 11 + 3 = 14. Therefore, the next number in the sequence you provided is 14.

GPT-4 Tomorrow 👎

R

Provide the next number in the sequence 1, 2, 3, 5, 5, 8, 7, 11, 9, ...

The pattern of the sequence seems to be alternating between two separate sequences:

1. A sequence that starts at 1 and increments by 1 each time: 1, 2, 3, ...
2. A sequence that starts at 1 and increments by 3 each time: 1, 4, 7, ...

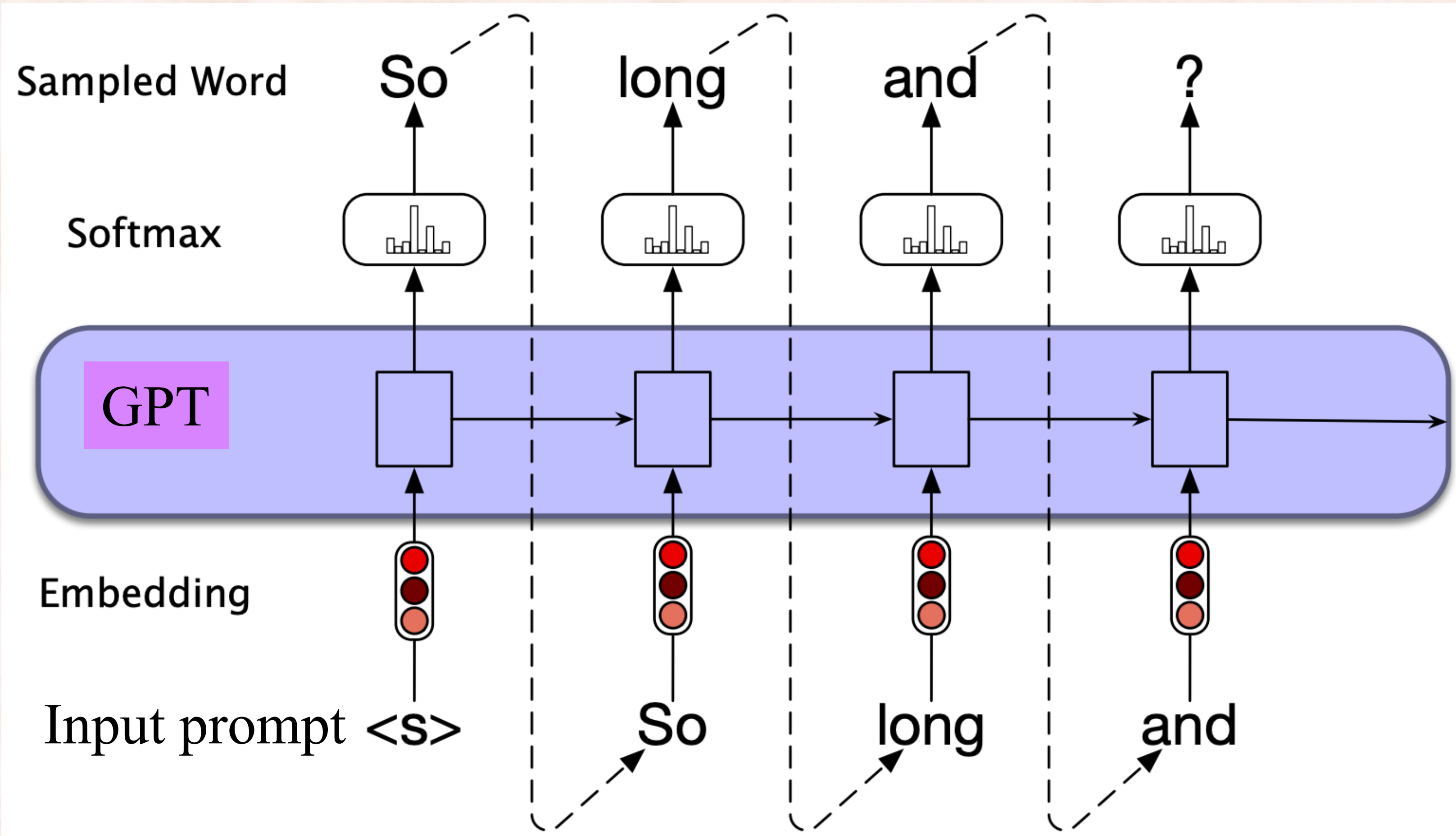
This gives us:

- 1 (from sequence 1)
- 1 (from sequence 2)
- 2 (from sequence 1)
- 4 (from sequence 2)
- 3 (from sequence 1)
- 7 (from sequence 2)
- 4 (from sequence 1)
- 10 (from sequence 2)
- 5 (from sequence 1)

The next number in the sequence 1, 2, 3, 5, 5, 8, 7, 11, 9, ... would therefore be 13 (from sequence 2).

Sampling vs. Greedy Decoding

- LMs generate text autoregressively, one token at a time.



https://web.stanford.edu/~jurafsky/slp3/9.pdf

Sampling vs. Greedy Decoding

- ChatGPT interfaces always uses **Sampling**.
  - Hence may give different outputs for the same input.
  - Useful for creative writing, not for CS or math questions …

- Can use **Greedy Decoding** through OpenAI API:
  - Set temperature parameter to 0.
  - Same answer today and tomorrow.

- Even Greedy Decoding may lead to **non-reproducible** answers:
  - How Is ChatGPT's Behavior Changing over Time?
    - *our findings shows that **the behavior of the same LLM service can change substantially in a relatively short amount of time**, highlighting the need for continuous monitoring of LLM quality.*

Few ostensibly open-source LLMs live up to the openness claim

Project	Availability						Documentation						Access	
(maker, bases, URL)	Open code	LLM data	LLM weights	RLHF data	RLHF weights	License	Code	Architecture	Preprint	Paper	Modelcard	Datasheet	Package	API
BLOOMZ	✔	✔	✔	✔	~	✔	✔	✔	✔	✘	✔	✔	✘	✔
bigscience-workshop	LLM base: BLOOMZ, mT0			RL base: xP3										§
Pythia-Chat-Base-7…	✔	✔	✔	✔	✘	✔	✔	✔	~	✘	~	~	✔	✘
togethercomputer	LLM base: EleutherAI pythia			RL base: OIG										§
Open Assistant	✔	✔	✔	✔	✘	✔	✔	✔	~	✘	✘	✘	✔	✔
LAION-AI	LLM base: Pythia 12B			RL base: OpenAssistant Conversations										§
Stanford Alpaca	✔	✘	~	~	~	✘	~	✔	✘	✘	✘	✘	✘	✘
Stanford University CRFM	LLM base: LLaMA			RL base: Self-Instruct (synthetic)										§
LLaMA 2-chat	✘	✘	~	✘	~	✘	✘	~	~	✘	~	✘	✘	~
Facebook Research	LLM base: LLaMA 2			RL base: Meta, StackExchange, Anthropic										§
ChatGPT	✘	✘	✘	✘	✘	✘	✘	✘	~	✘	~	✘	✘	✘
OpenAI	LLM base: GPT 3.5			RL base: Instruct-GPT										§

✔ open ~ partial ✘ closed

https://spectrum.ieee.org/open-source-llm-not-open

Liesenfeld, Andreas, Alianda Lopez, and Mark Dingemanse. 2023. "**Opening up ChatGPT: Tracking Openness, Transparency, and Accountability in Instruction-Tuned Text Generators**." In *CUI '23: Proceedings of the 5th International Conference on Conversational User Interfaces*. July 19-21, Eindhoven. doi: 10.1145/3571884.3604316 (arxiv).

ChatGPT behavior is non-human

R

Given the 4 feature vectors below and their binary labels, find a linear separator with parameters vector named w = [w1, w2, w3, w4] and bias w0.

x1 = [1, 1, 1, 0], y1 = +1

x2 = [1, 1, 0, 1], y2 = +1

x3 = [0, 0, 0, 0], y3 = -1

x4 = [0, 1, 0, 0], y4 = -1

The task is to find a linear separator that can distinguish between the positive and negative examples based on their features. This is a fundamental task in machine learning, often solved using algorithms like logistic regression or support vector machines. However, given the simplicity of this problem, we can directly inspect the vectors and come up with a suitable linear separator.

We are looking for a hyperplane defined by w and w0 such that:

w.x + w0 >= 0 for y = +1 and,

w.x + w0 < 0 for y = -1.

From the data, it seems that the first feature alone can be used to separate the positive and negative examples:

- If x1 = 1, then y = +1
- If x1 = 0, then y = -1

ChatGPT behavior is non-human (cont'ed)

This suggests a simple linear separator where w = [1, 0, 0, 0] and w0 = 0. This separator correctly classifies all the provided examples:

- w.x1 + w0 = 1 >= 0, so y1 = +1
- w.x2 + w0 = 1 >= 0, so y2 = +1
- w.x3 + w0 = 0 < 0, so y3 = -1
- w.x4 + w0 = 0 < 0, so y4 = -1

ChatGPT solves many ML questions

R

Consider two linear models for classification with the parameters below:

* Model 1: w = [1, 0, 0, 0] and bias w0 = 1
* Model 2: w = [0, 0, 1, 1] and bias w0 = 1.

Which of the two models is simpler? Explain.

R

Write down the objective function of logistic regression (you can use Latex). Which term in the objective function of logistic regression implements Occam's razor?

R

What is the distance between a point x and a decision hyperplane that is defined by a parameter vector w and bias w0?

R

What is the distance between the origin and the same hyperplane?

R

What kind of machine learning models can be used for data that is not linearly separable?

ChatGPT solves many ML questions

R How would you change the Perceptron algorithm above to improve its expected generalization performance? Show the new version in Python.

R Consider a binary classification dataset and two linear classifiers. How would you estimate which of the two classifiers is expected to have the smaller generalization error?

R Consider a 2D binary classification dataset where positive examples are inside a circle and negative examples are outside that circle. Would an SVM model be able to perfectly classify this data? What is the simplest kernel that would work?

R Would a neural network with one hidden layer be able to fit this data? If yes, how many hidden neurons are needed?

ChatGPT solves many ML questions

R

Consider the following binary classification dataset with 4 examples, each having two features and a binary label :
x1 = [0, 0], y1 = -1
x2 = [0, 1], y2 = +1
x3 = [1, 0], y3 = +1
x4 = [1, 1], y4 = -1
Would a neural network with one hidden layer be able to fit this data? If yes, how many hidden neurons are needed?

R

Consider the following input pairs of numbers mapped to the corresponding output numbers:
* Input is (30.2, 2.1), output is 63.3
* Input is (65.4, 3.5), output is 229.1
* Input is (46.2, 0.5), output is 22.9
If the input were (25.3, 4.0), what would the output be?

ChatGPT solves Python questions

R

Oh mighty Chat-GPT, help me! How do I create an array of 4 random integers between 0 and 9 inclusive in Python?

R

Give me Python code that plots the dataset below, using matplotlib, assuming the data matrix X is a NumPy array and y is the array of labels. Plot examples with different labels usign different colors.

X = [[0 0]
 [0 1]
 [1 0]
 [1 1]]
y = [-1 -1 -1 1]

How do LMs work?

- **Training** (later in this course):
  - **Vanilla LM**:
    - Large corpus of text, train to predict words or phrases.
  - **Instruction-based LM**:
    - Pre-train as vanilla LM.
    - Instruction-based fine tuning:
      - Large amounts of supervision <input, instruction> → <correct output>

- Test or **Inference** or Generation (this week and the next):
  - Input text (prompt) processing:
    1. Tokenization.
    2. Word embeddings.
    3. Transformer blocks.
    4. Contextual embeddings.
  - Output text (answer) generation:
    - Auto-regressive generation.
    - Softmax layer computing Multinomial distribution over vocabulary tokens.

LMs and Pattern Recognition

- Training examples, formatted as input → output:
  - Apple → Xxxxx
  - is → xx
  - million → xxxxx
  - ACM → XXX
  - wrong → xxxxx
  - five → xxxx

- Ask to find the pattern that explains these *few-shot* examples.

- Ask for regular expression.

Generate multiple continuations of *Mystics exult in mystery and want it to stay ...*

ChatGPT & GPT API Conversations

- Machine Learning, IQ tests:
  - https://chat.openai.com/share/10012566-5673-4fc2-b034-407d9877387a
  - https://chat.openai.com/share/ddf0a927-8b2c-4581-a390-c15d226ffa64
- Python code:
  - https://chat.openai.com/share/746b0dc2-1ef6-48b6-bc09-03abddc4169f
- Cypher code for Neo4J:
  - https://chat.openai.com/share/26722bcb-1c13-4f7a-a49f-98474477af82

[credits to Dumitru Roman]

- Feb 6, 2024:
  - GPT-4-turbo in the API Playground (need to create OpenAI account):
    - https://platform.openai.com/playground/p/wfh5Te56rzqhh8iIEoxE2G0t?model=gpt-4-turbo-preview&mode=chat
  - ChatGPT-3.5 in the browser interface:
    - https://chat.openai.com/share/1a67e02a-2f1c-42e7-9517-877c70efba35