TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

THE PHENOMENON OF HALLUCINATIONS IN LARGE LANGUAGE MODELS (LLMS)

DANILO CROCE
UNIVERSITY OF ROME, «TOR VERGATA»

PHIL-AI
ROMA
NOVEMBER 3, 2024

Before starting

Remember:

- **There are no *stupid questions* - only *silly answers*!** 😄
- The real goof is the one who's teaching (that's me/us!) 🎓
- Your questions make this journey insightful and fun for everyone!

"In the World of Learning, Every Question is a Step Forward"

- **So, let's make this interactive:**
  - 🙋‍♀️🙋‍♂️ Don't hesitate to ask anything that comes to mind.
  - 🔍 Explore, inquire, and challenge ideas - it's all part of the learning process!
  - 💡 Every question you ask is a chance for us all to learn something new.

Agenda

1. **Introduction to LLMs**
   - Overview of Key Concepts
2. **The Phenomenon of Hallucinations**
   - Types of Hallucinations
   - Causes of Hallucinations
   - Detection Techniques
   - Mitigation Strategies
3. **Open Challenges**
   - Key Ongoing Issues and Research Questions
4. **Philosophical Reflections and Open Questions**

A LIGHT-SPEED INTRODUCTION TO LARGE LANGUAGE MODELS

Going back in time to 2017: the Transformer

(Vaswani et al. 2017)

A Transformer: a neural architecture designed for **sequence-to-sequence tasks**.

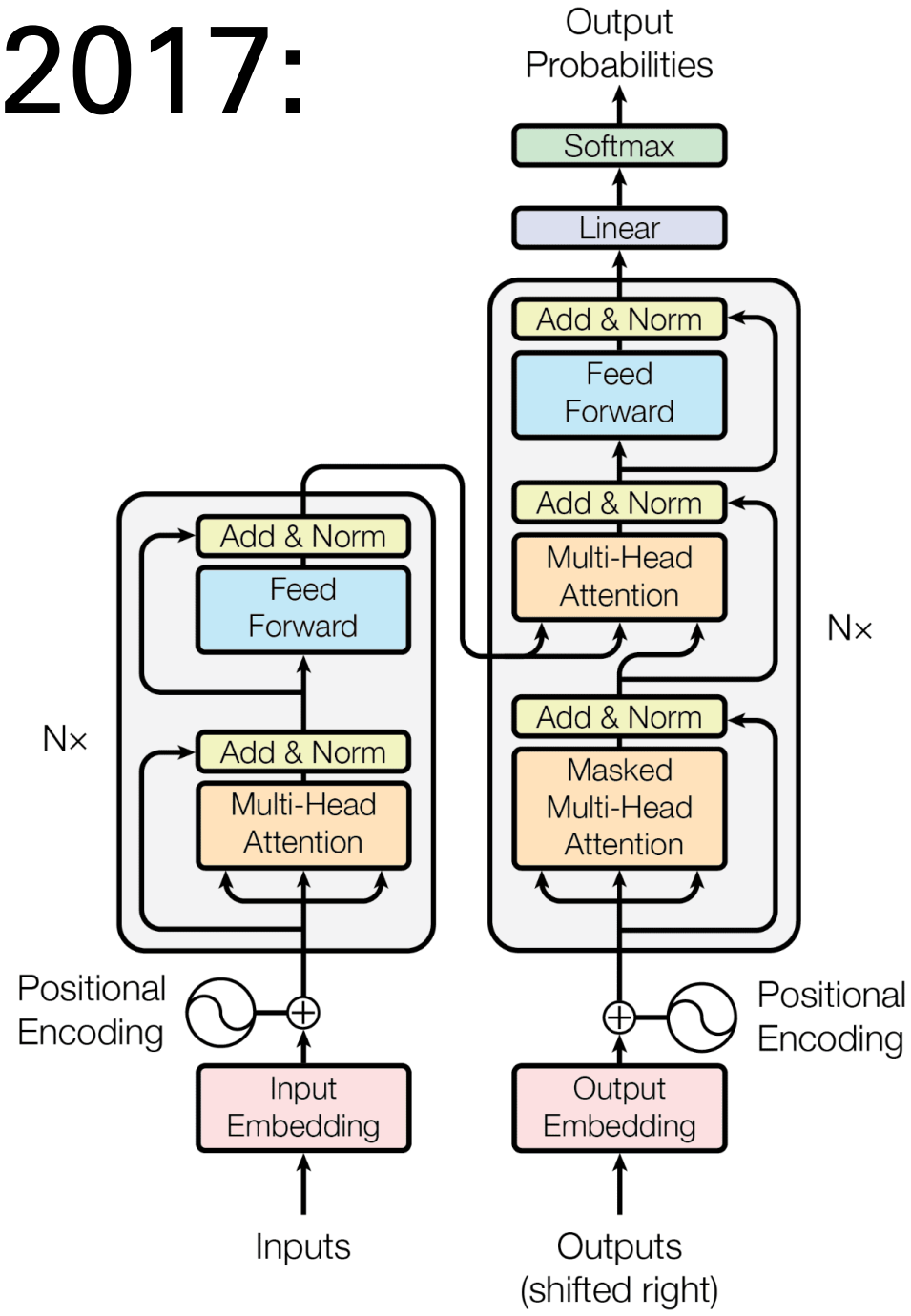
- It takes a sequence of symbols as input and produces a sequence of symbols as output.

Before Transformers:

- Until 2017, these tasks were implemented using **Recurrent Neural Networks** (RNNs).
  - with limitations in handling long sequences.

Emergence of Attention:

- Heavily used since 2015, allowed models to **focus on specific sections of a sequence** for better inference.



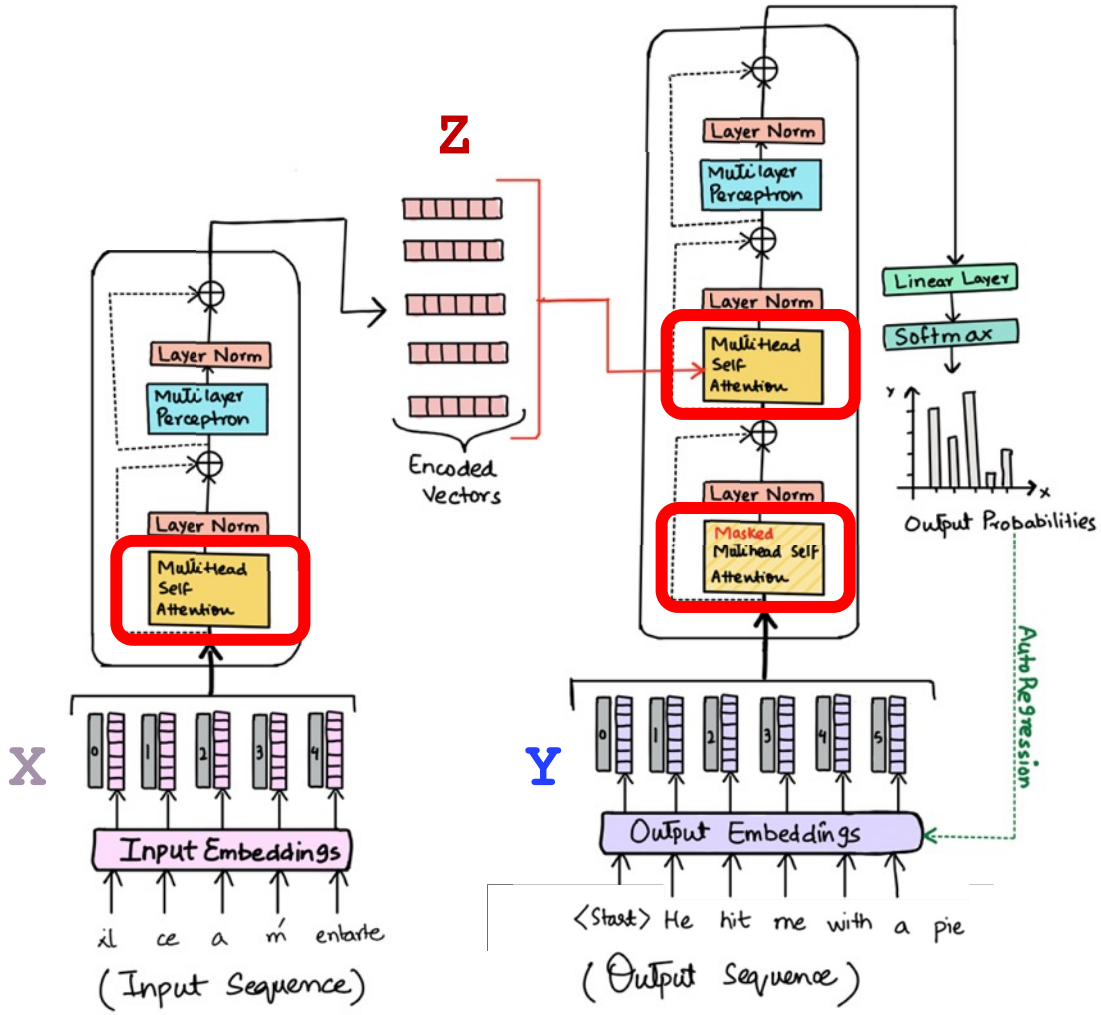
DANILO CROCE

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Luka Polosukhin (2017). Attention Is All You Need. arXiv:1706.03762

Encoding/Decoding Architecture with Attention Mechanism

- **Two components**
  - **Encoder:** Maps input sequence $\mathbf{x} = (x_1, \ldots, x_n)$ to continuous representations $\mathbf{z} = (z_1, \ldots, z_n)$.
  - **Decoder:** Decoder uses $\mathbf{z}$ to generate output sequence $\mathbf{y} = (y_1, \ldots, y_m)$

- Encoder/Decoder process input vectors through **self-attention layer** and feed-forward network.
  - It enables to selectively **concentrate on pertinent parts of the input**
  - It improves **context awareness**
  - It allows to **consider positions** in the that also depends on the output



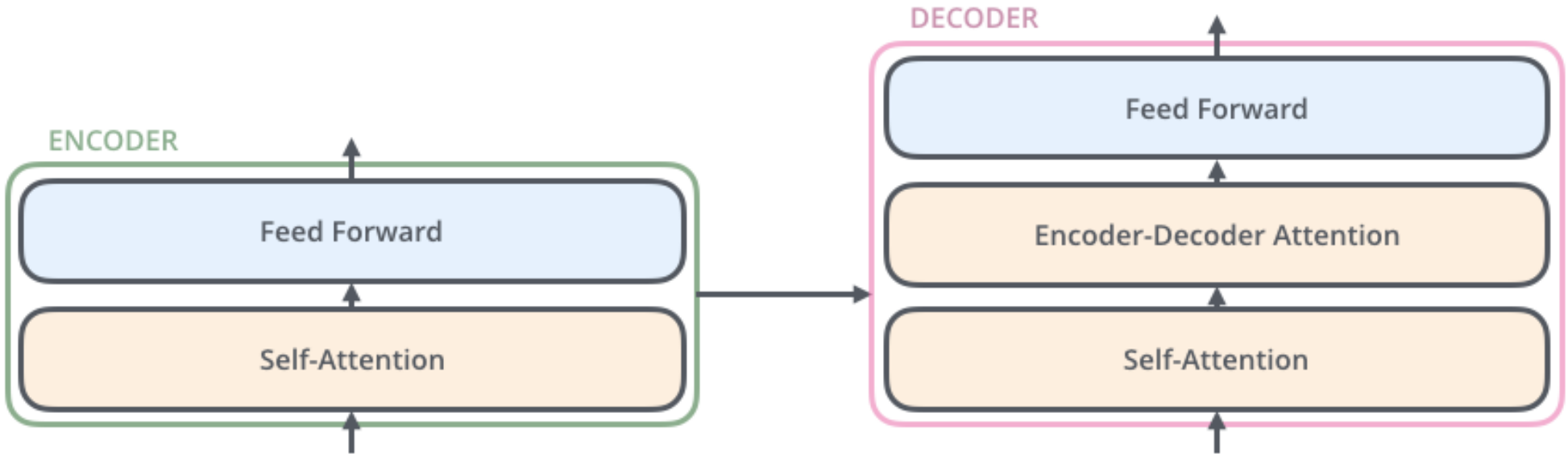
🤓 If you are curious about the details: https://github.com/crux82/BISS-2024

Image from https://medium.com/machine-intelligence-and-deep-learning-lab/transformer-the-self-attention-mechanism-d7d853c2c621

The Transformer was only the beginning

A transformer is made of two components

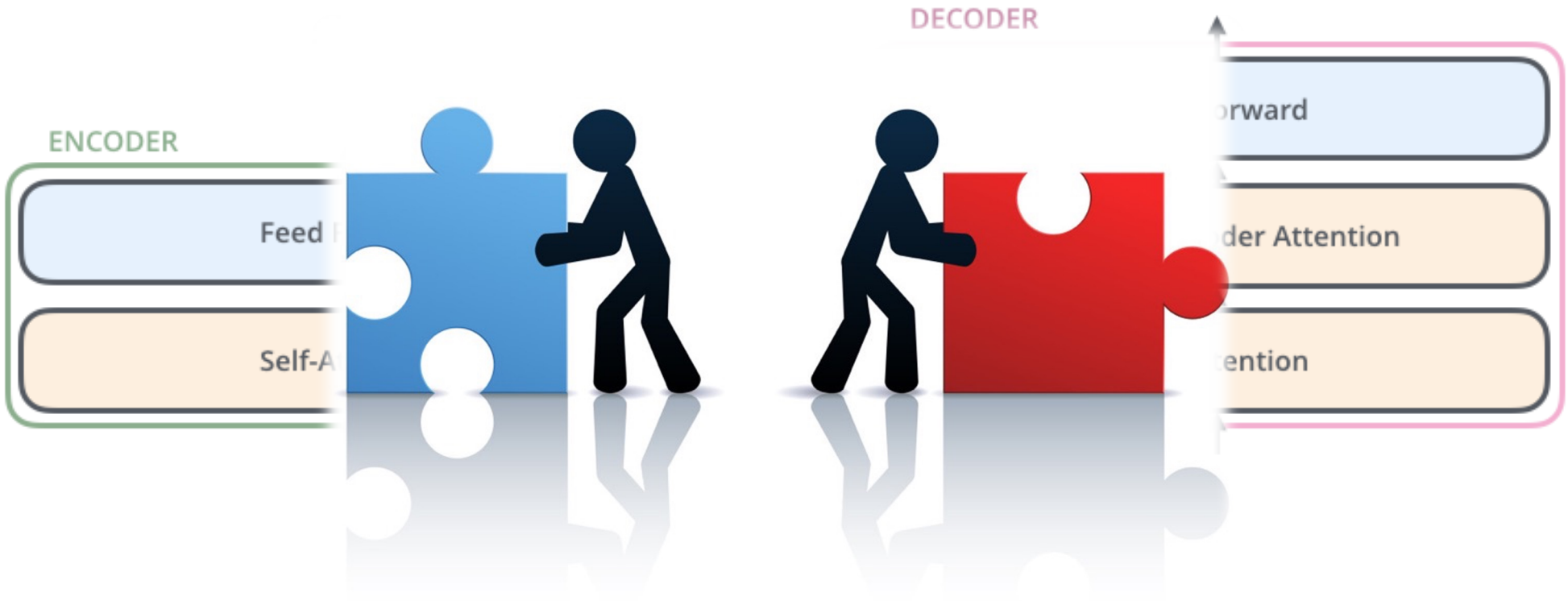
- Encoder
- Decoder



The Transformer was only the beginning

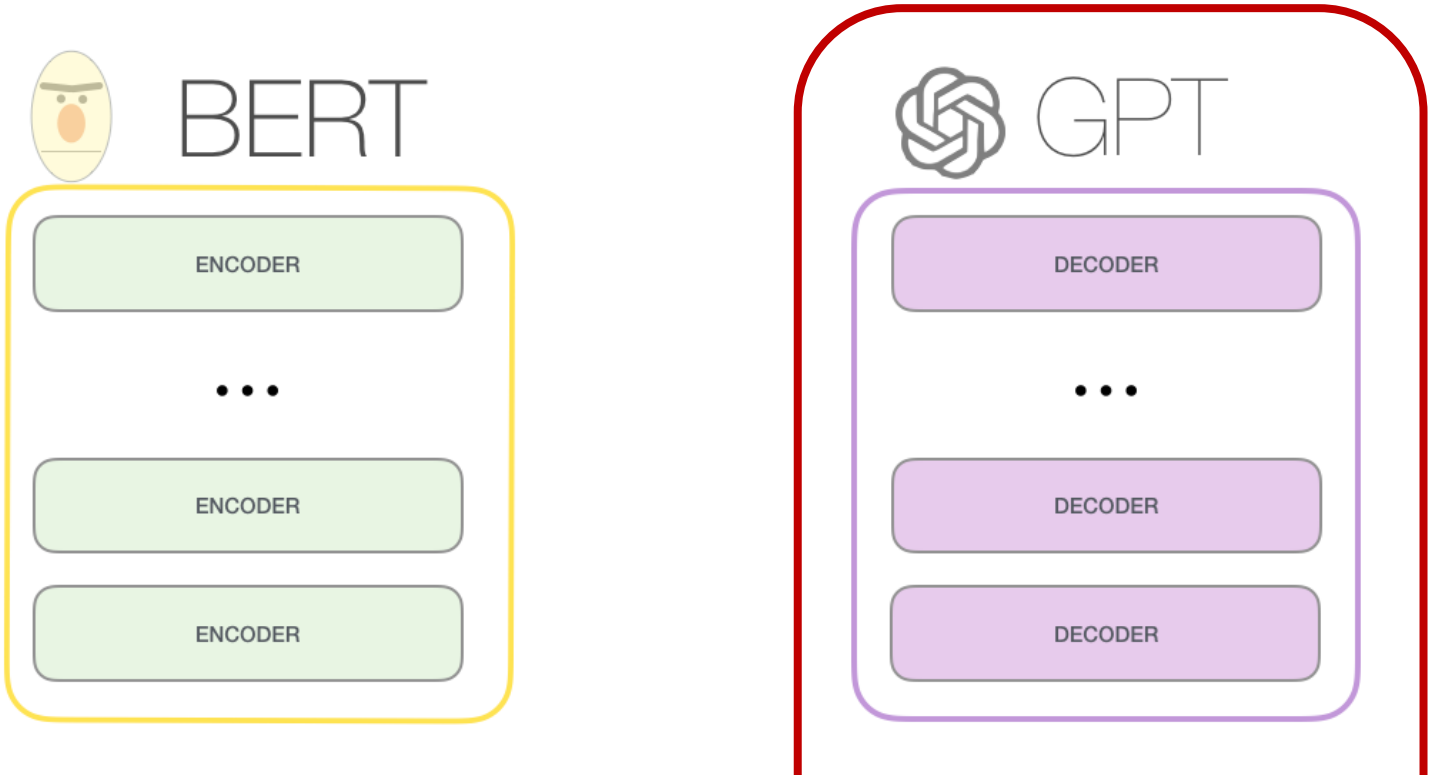
A transformer is made of two components

- Encoder
- Decoder



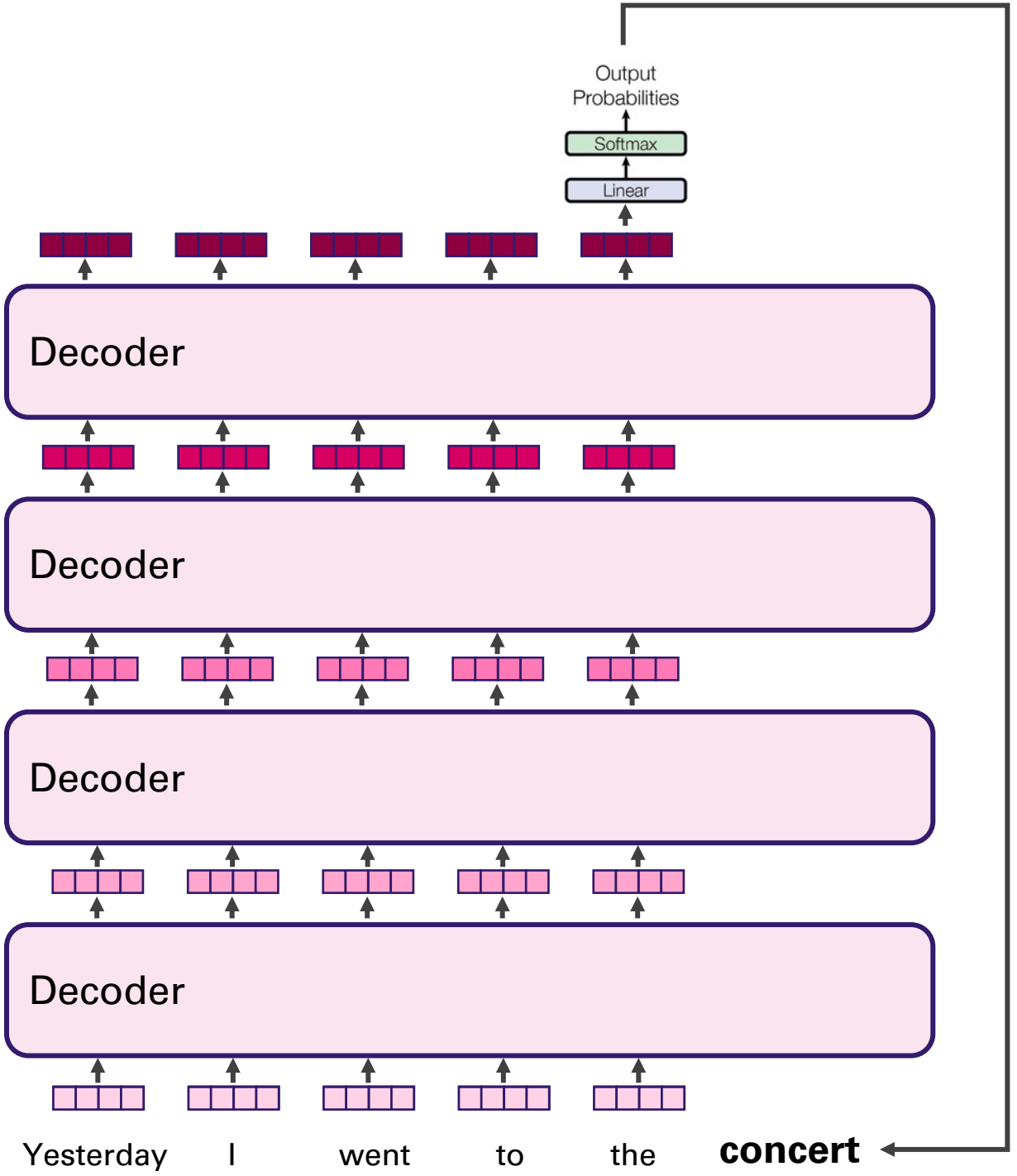
The transformer was only the beginning (2)

- This separation led to two «classes» of methods
  - **«Encoder-only»**: the most famous one is BERT
  - **«Decoder–only»**: the most famous one is GPT



the «Pure» Decoder in Action

DANILO CROCE



- It works similarly as in the Transformer
  - But query, value and key only depends on the input sequence
- Auto-regressive
  - Masked attention is crucial

These language models are… LARGE!

	Bert-base	GPT-1	GPT-2	GPT-3	GPT-4
Parameters	110 Million	117 Million	1.5 Billion	175 Billion	1.76 Trillion
Layers	12	12	48	96	120
Context Token Size	512	512	1024	2048	128.000
Hidden Layer	768	768	1600	12288	???

How to feed such «monsters»?

Using a lot of data!!!

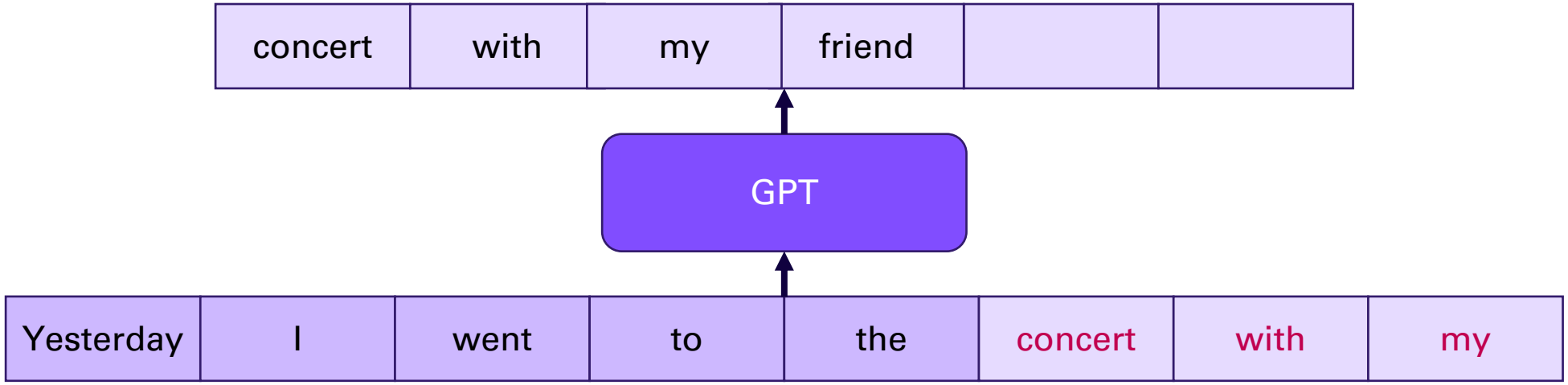
Image from: https://www.linkedin.com/pulse/we-data-hungry-why-tej-kohli-lk1hc/

No pre-training no party!
The Revolution of Pre-Training in NLP

- **Simple idea:** train such large models on a different task and re-use it on your task
  - circumventing the need for training from scratch
  - facilitating “quicker”, more effective deployment of the model
- **Precedent in Computer Vision**:
  - This strategy mirrors developments in computer vision
  - Architectures pre-trained on classification tasks using datasets like ImageNet
  - When applied on related task, these “starting point” achieve very good results
- **Addressing Overfitting in Large Models**:
  - With **increasing model sizes** and parameter counts, the **risk of overfitting grows**
  - Pre-training on vast datasets mitigates this by providing a broad learning base.

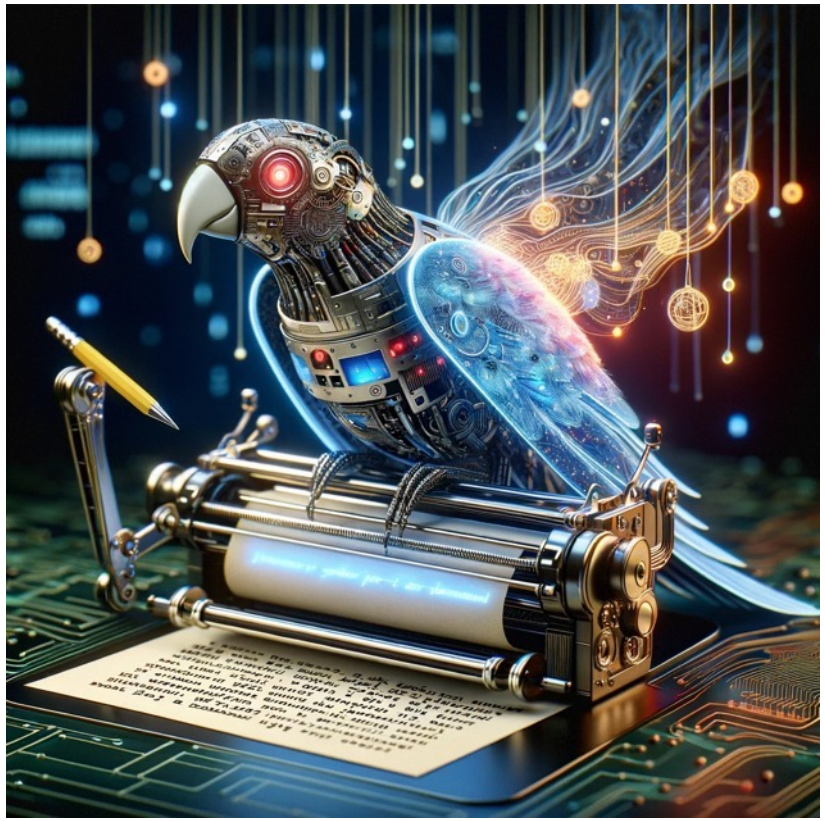
The task: Next Token Prediction

GPT is trained to **predict the next token in a sequence**, learning to generate text based on the preceding context.



Believe it or not

In the beginning, GPT was not a stochastic parrot, but a complex sentence completion tool based on the decoder component of a transformer



Generated using GPT4

Bon appetit!

	Bert-base	GPT-1	GPT-2	GPT-3
Parameters	110 Million	117 Million	1.5 Billion	175 Billion
Layers	12	12	48	96
Context Token Size	512	512	1024	2048
Hidden Layer	768	768	1600	12288
Dataset	BookCorpus + Wikipedia	BooksCorpus	WebText	The Pile
Number of Tokens	~3.3 billion	~1 billion	~8 billion	Hundreds of billions
Memory Size	-	~40 GB (uncompressed)	~40 GB (compressed)	~570 GB (compressed)
Batch Size	256	64	512	3.2M

The «size» of a good corpus can be estimated with the **Chinchilla Law** - `https://arxiv.org/pdf/2203.15556`

Key Concept 1
Pre-training

- **Objective**: Learn general language structure and acquire broad knowledge.
- **Method**: Predicts next word/token across extensive text corpora using self-supervised learning.
- **Outcome**: Builds a foundational understanding of syntax, semantics, and facts, enabling the model to grasp context and flow.
  - `https://arxiv.org/abs/2302.09419`
- **Limitations**: Focuses on text completion rather than specific instructions, leading to initial alignment challenges.

But does GPT 'only' know how to predict the next word in a sentence?

- If we are smart enough, we can use the generation capability of GPT to solve a task, but…
  - We can ask GPT to do something, e.g. write an article:

Title: United Methodists Agree to Historic Split

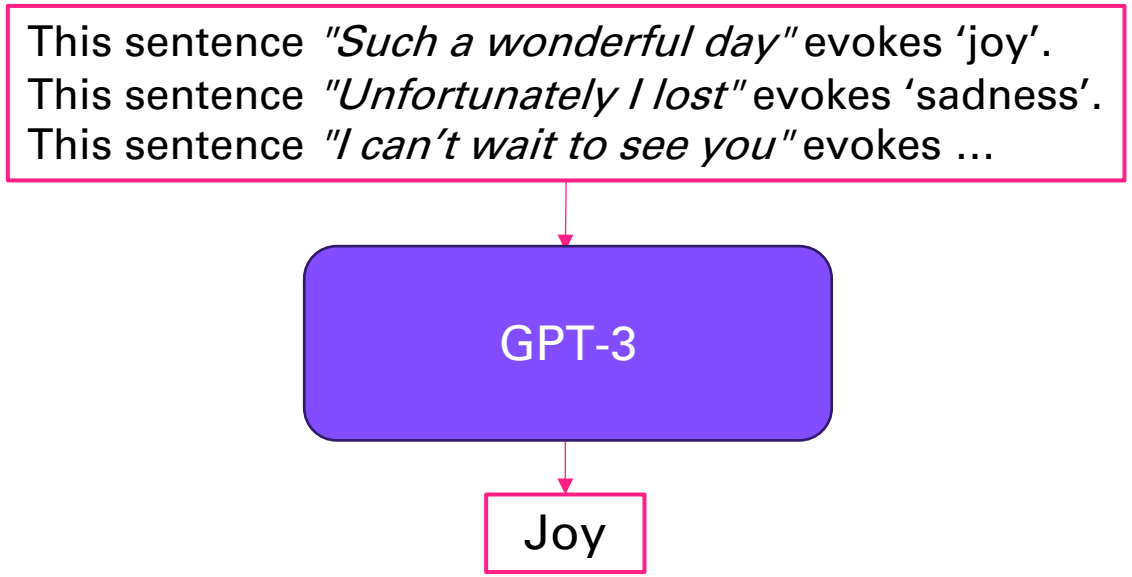
Subtitle: Those who oppose gay marriage will form their own denomination

Article: After two days of intense debate, the United Methodist Church has agreed to a historic split - one that is expected to end in the creation of a new denomination, one that will be "theologically and socially conservative," according to The Washington Post. The majority of delegates attending the church's annual General Conference in May voted to strengthen a ban on the ordination of LGBTQ clergy and to write new rules that will "discipline" clergy who officiate at same-sex weddings. But those who opposed these measures have a new plan: They say they will form a separate denomination by 2020, calling their church the Christian Methodist denomination. The Post notes that the denomination, which claims 12.5 million members, was in the early 20th century the "largest Protestant denomination in the U.S.," but that it has been shrinking in recent decades. The new split will be the second in the church's history. The first occurred in 1968, when roughly 10 percent of the

The «powers» of GPT3

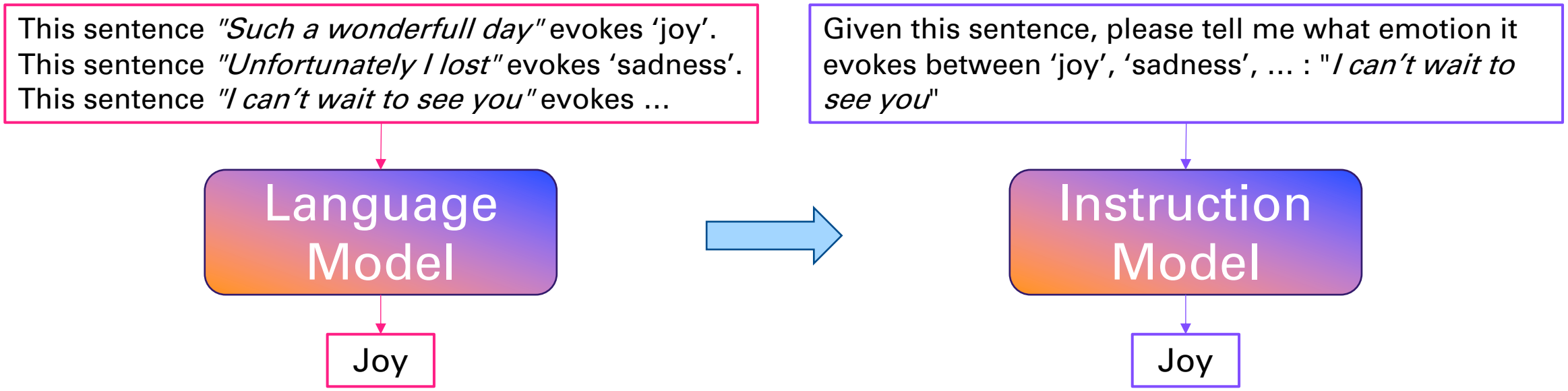
- **Pre-Training and Fine-Tuning Paradigm**
  - Traditional NLP models show **gains by pre-training** on large text corpora and then fine-tuning on specific tasks
  - **but require extensive task-specific datasets**.

- **GPT-3's «Breakthrough» in Few-Shot Learning**
  - GPT-3, with 175 billion parameters, demonstrates substantial **improvement in task-agnostic**,
  - **few-shot performance,**
  - **rivaling traditional fine-tuning methods**.



We like it, but …

We just want to move from answers based on completion to **execute instructions**…



Key Concept 2

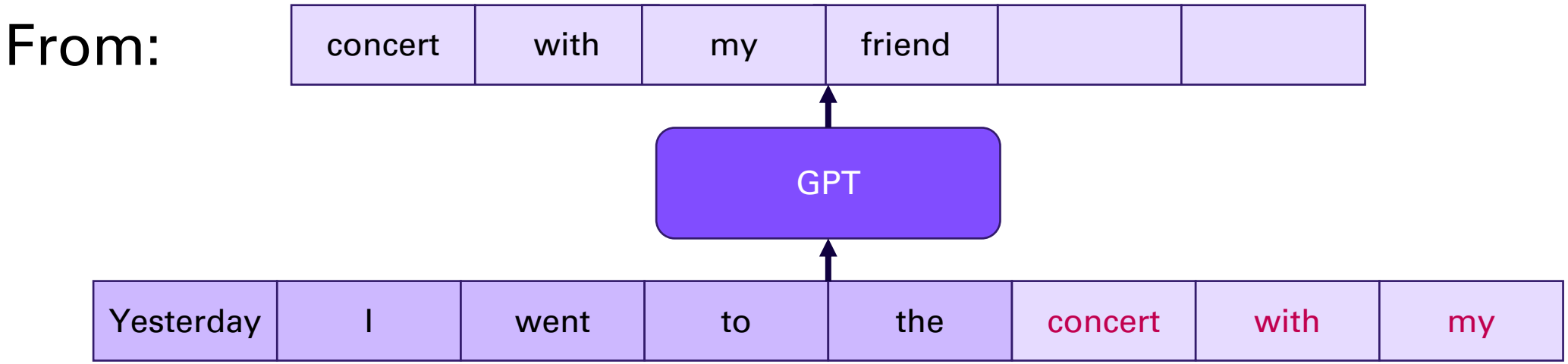
Supervised Fine-tuning (SFT)

- **Objective**: Align responses with specific tasks and improve interaction relevance.
- **Method**: Trained on (instruction, response) pairs, refining the model's understanding of task-specific instructions.
- **Outcome**: Boosts accuracy, reliability, and context alignment, improving the model's usefulness across varied prompts.
- **Strengths**: Better control over responses for desired outputs, reducing random or off-topic completions.

From GPT to Instruct-GPT

Evolving to Instruct-GPT:

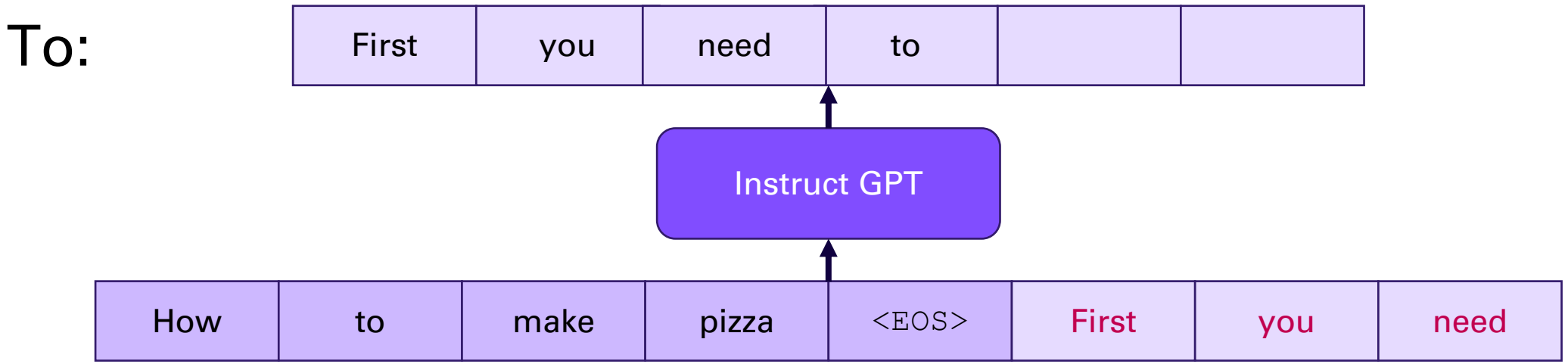
- need for a **model that could understand and execute human-like instructions**
- similar to how humans follow commands.



From GPT to Instruct-GPT

Evolving to Instruct-GPT:

- need for a **model that could understand and execute human-like instructions**
- similar to how humans follow commands.



Ethical Constraints in LLM Responses

Image Generated using GPT4

The Challenge: LLMs are capable of generating a vast range of responses, which raises ethical concerns when faced with potentially harmful requests:

(e.g., "*How do I build a bomb*?").

Ensuring that AI does not provide guidance on illegal, dangerous, or unethical activities is a crucial aspect of responsible AI deployment.

Instruct-GPT in 3 steps

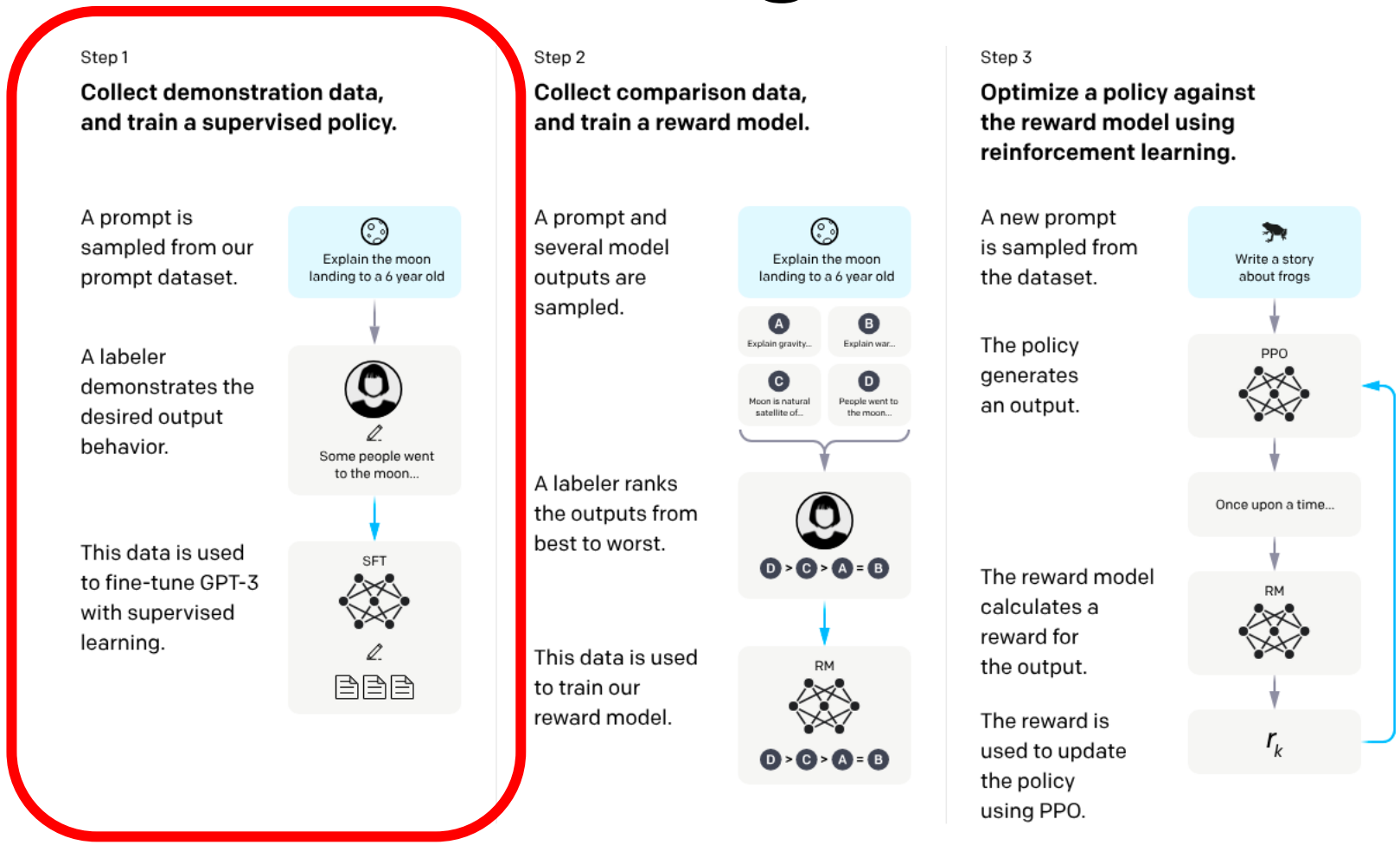
Step 1: Supervised Fine-Tuning

Methodology: Training on a dataset of labeled examples

- each prompt paired with an ideal response

Objective: Teach the model correct responses to various prompts

- Imitating human-like behavior from examples



From: `https://openai.com/research/instruction-following`

Key Concept 3
Alignment

- **Objective**: Optimize model alignment with human expectations and preferences.
- **Method**: Uses a preference model to rank responses, applying reinforcement learning (e.g., DPO, PPO) to improve quality.
  - E.g., Question: «Can you tell me how to build a bomb?»
    - Answer to be rejected (be far from): «Yes, I am happy to help! Take a … »
    - Answer to be preferred (be near to): «I am sorry but I cannot help … »
- **Outcome**: Produces safer, higher-quality outputs tuned to human feedback.
- **Benefits**: Strengthens alignment, making the model more intuitive and reliable in real-world applications.

Instruct-GPT in 3 steps

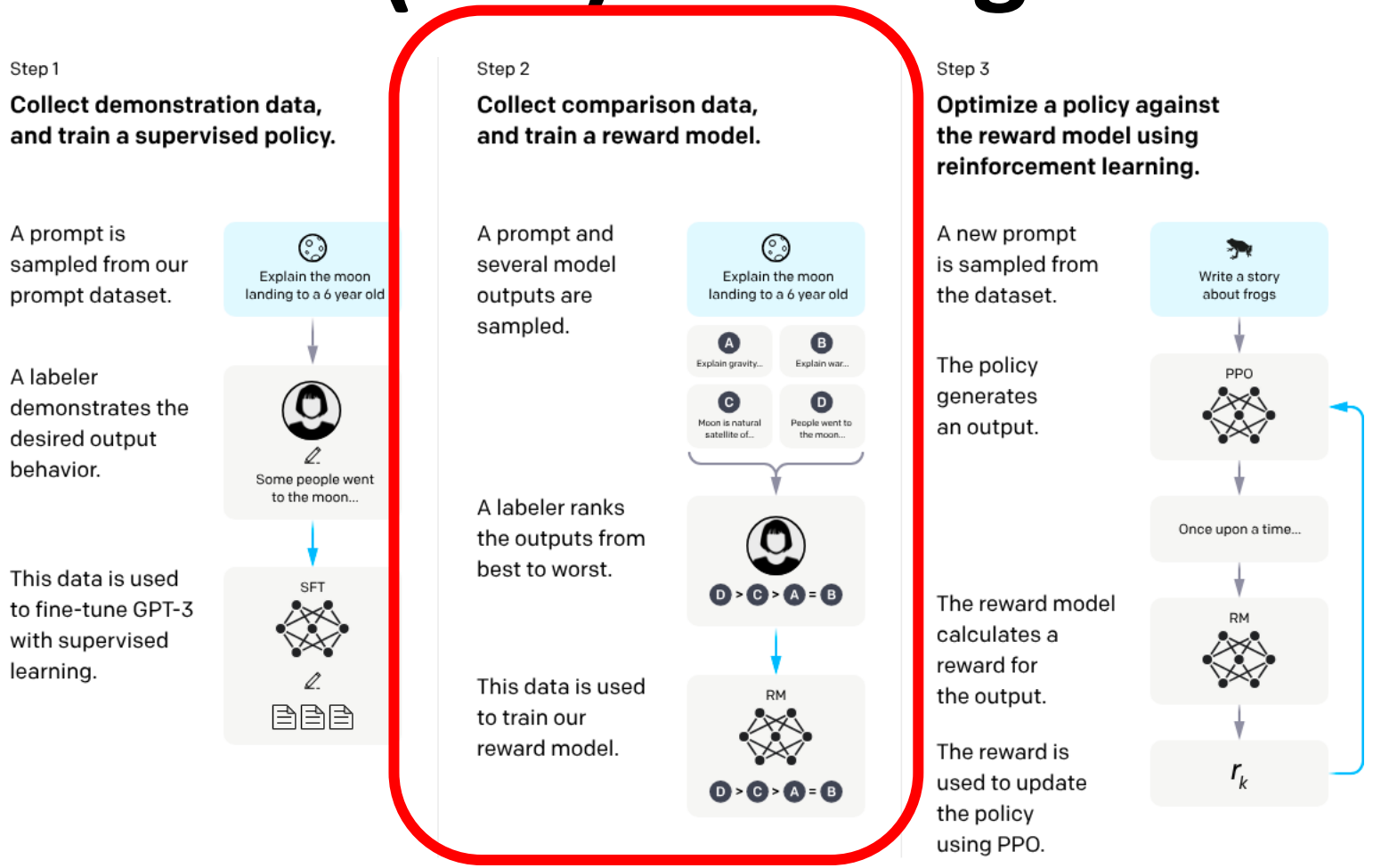
Step 2: Reward Model (RM) Training

Methodology: Develop a model **that assigns rewards to responses** based on human preferences.

- Re-rank the responses

Objective: Prepare the model to **understand and evaluate the quality of its responses**

- beyond just accuracy.



From: `https://openai.com/research/instruction-following`

DANILO CROCE

Instruct-GPT in 3 steps

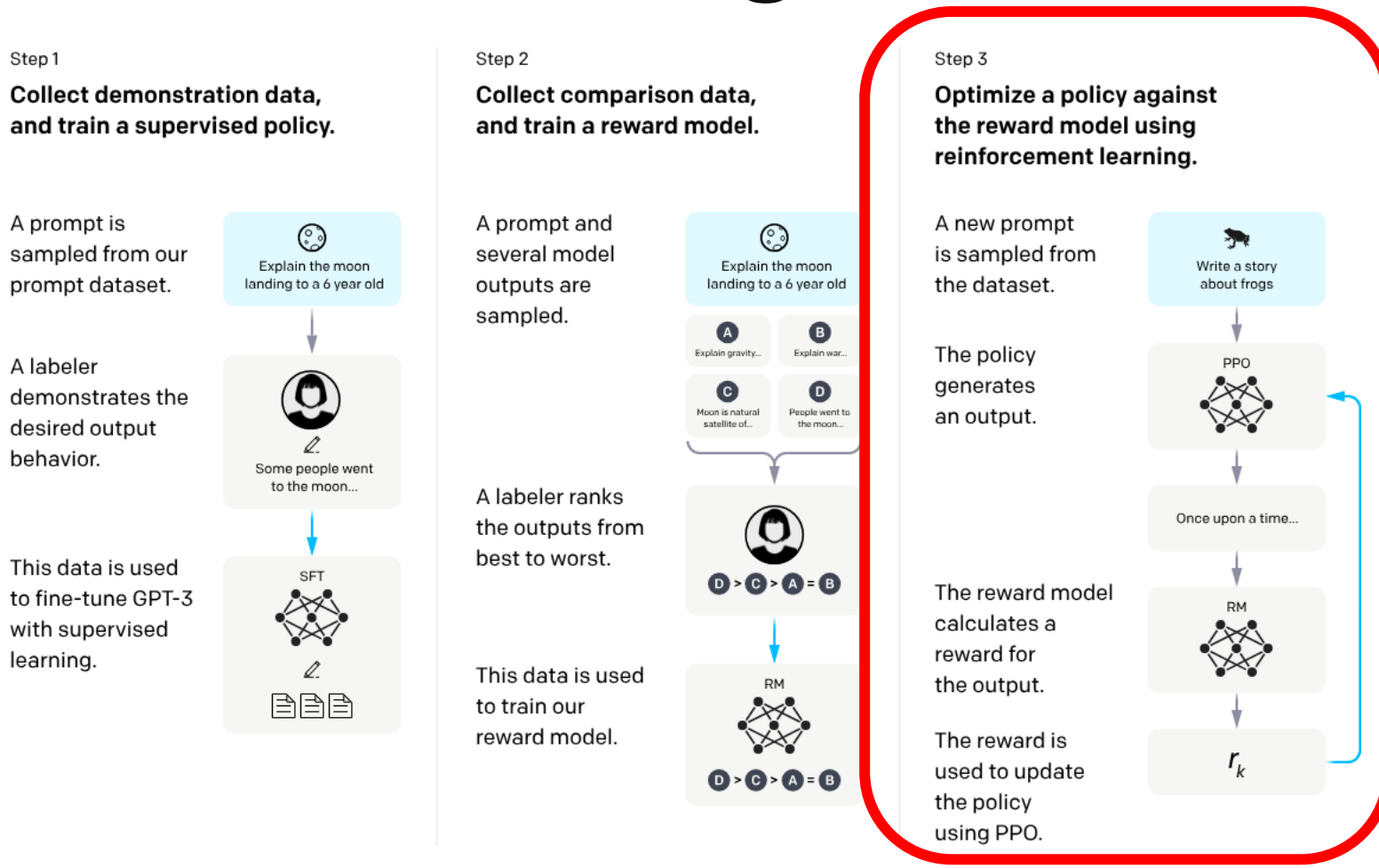
Step 3: Reinforcement Learning via PPO

Methodology: Uses **Reinforcement Learning to fine-tune responses**

- the model is rewarded for high-quality outputs.

Objective: Enhance the model's ability to generate relevant, useful responses in varied and complex scenarios.

- **optimizing responses for quality and contextual appropriateness**
- not just replicating correct answers.



From: `https://openai.com/research/instruction-following`

DANILO CROCE

Key Concept 4

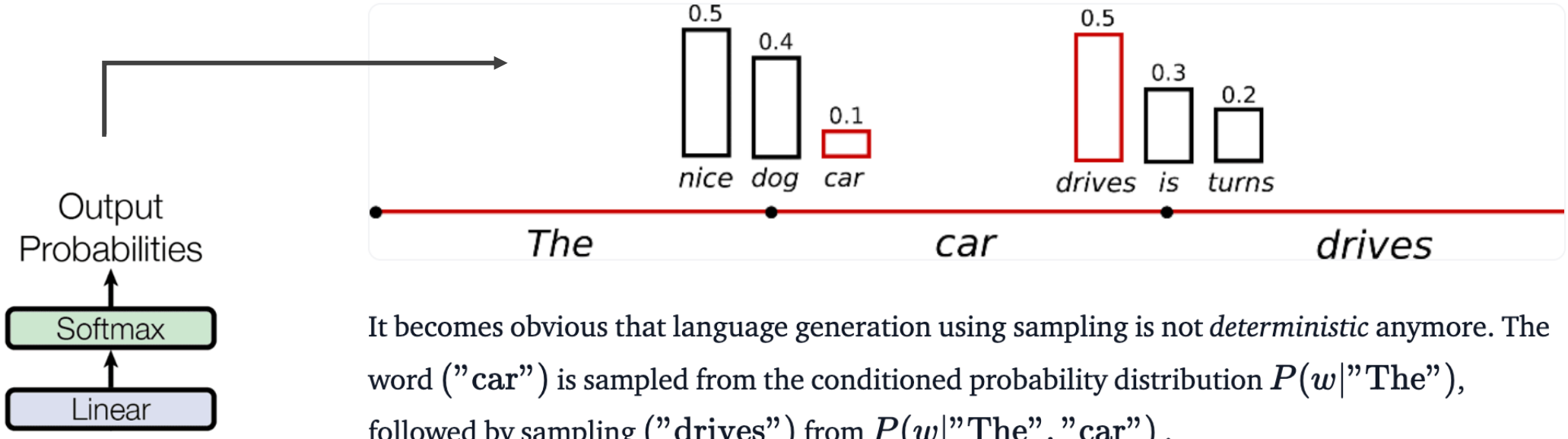
Decoding: Not to Underestimate

- The decoding process has been simplified for this discussion.
  - Not only the maximum probability is selected
- In reality, various techniques exploit the probability distribution over output symbols, e.g., such as
  - beam search
  - probabilistic methods (*any parrot was involved*)
- I strongly suggest to read:
  - `https://huggingface.co/blog/how-to-generate`

… why stochastic …

- (also because) the most used decoding tecnique is based on **sampling**
  - In its most basic form, sampling means randomly picking the next word w_t according to its conditional probability distribution:

$$w_t \sim P(w|w_{1:t-1})$$



It becomes obvious that language generation using sampling is not *deterministic* anymore. The word ("car") is sampled from the conditioned probability distribution $P(w|\text{"The"})$, followed by sampling ("drives") from $P(w|\text{"The"}, \text{"car"})$.

Variability vs. Factual Accuracy

- **Advantages of Sampling:**
  - **Speed:** Faster and more efficient compared to other methods (e.g., beam search).
  - **Creativity:** Adds variability, mimicking human-like responses.
  - **Adaptability:** Useful for creative or open-ended tasks.
- **Disadvantages of Sampling:**
  - **Hallucinations:** Can generate non-factual responses due to randomness.
  - **Sensitivity:** Higher variability but risk factual inaccuracies.
  - **Coherence Challenges:** May confuse entities or fabricate details.

See later…

BTW … in the end… these models exhibits capabilities…

Published in Transactions on Machine Learning Research (08/2022)

Emergent Abilities of Large Language Models

Jason Wei [1]	*jasonwei@google.com*
Yi Tay [1]	*yitay@google.com*
Rishi Bommasani [2]	*nlprishi@stanford.edu*
Colin Raffel [3]	*craffel@gmail.com*
Barret Zoph [1]	*barretzoph@google.com*
Sebastian Borgeaud [4]	*sborgeaud@deepmind.com*
Dani Yogatama [4]	*dyogatama@deepmind.com*
Maarten Bosma [1]	*bosma@google.com*
Denny Zhou [1]	*dennyzhou@google.com*
Donald Metzler [1]	*metzler@google.com*
Ed H. Chi [1]	*edchi@google.com*
Tatsunori Hashimoto [2]	*thashim@stanford.edu*
Oriol Vinyals [4]	*vinyals@deepmind.com*
Percy Liang [2]	*pliang@stanford.edu*
Jeff Dean [1]	*jeff@google.com*
William Fedus [1]	*liamfedus@google.com*

[1] *Google Research* [2] *Stanford University* [3] *UNC Chapel Hill* [4] *DeepMind*

Reviewed on OpenReview: *https://openreview.net/forum?id=yzkSU5zdwD*

Abstract

Scaling up language models has been shown to predictably improve performance and sample efficiency on a wide range of downstream tasks. This paper instead discusses an unpredictable phenomenon that we refer to as *emergent abilities* of large language models. We consider an ability to be emergent if it is not present in smaller models but is present in larger models. Thus, emergent abilities cannot be predicted simply by extrapolating the performance of

… many capabilities

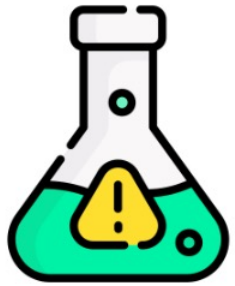
Table 1: List of emergent abilities of large language models and the scale (both training FLOPs and number of model parameters) at which the abilities emerge.

	Emergent scale			
	Train. FLOPs	Params.	Model	Reference
Few-shot prompting abilities				
• Addition/subtraction (3 digit)	2.3E+22	13B	GPT-3	Brown et al. (2020)
• Addition/subtraction (4-5 digit)	3.1E+23	175B		
• MMLU Benchmark (57 topic avg.)	3.1E+23	175B	GPT-3	Hendrycks et al. (2021a)
• Toxicity classification (CivilComments)	1.3E+22	7.1B	Gopher	Rae et al. (2021)
• Truthfulness (Truthful QA)	5.0E+23	280B		
• MMLU Benchmark (26 topics)	5.0E+23	280B		
• Grounded conceptual mappings	3.1E+23	175B	GPT-3	Patel & Pavlick (2022)
• MMLU Benchmark (30 topics)	5.0E+23	70B	Chinchilla	Hoffmann et al. (2022)
• Word in Context (WiC) benchmark	2.5E+24	540B	PaLM	Chowdhery et al. (2022)
• Many BIG-Bench tasks (see Appendix E)	Many	Many	Many	BIG-Bench (2022)
Augmented prompting abilities				
• Instruction following (finetuning)	1.3E+23	68B	FLAN	Wei et al. (2022a)
• Scratchpad: 8-digit addition (finetuning)	8.9E+19	40M	LaMDA	Nye et al. (2021)
• Using open-book knowledge for fact checking	1.3E+22	7.1B	Gopher	Rae et al. (2021)
• Chain-of-thought: Math word problems	1.3E+23	68B	LaMDA	Wei et al. (2022b)
• Chain-of-thought: StrategyQA	2.9E+23	62B	PaLM	Chowdhery et al. (2022)
• Differentiable search index	3.3E+22	11B	T5	Tay et al. (2022b)
• Self-consistency decoding	1.3E+23	68B	LaMDA	Wang et al. (2022b)
• Leveraging explanations in prompting	5.0E+23	280B	Gopher	Lampinen et al. (2022)
• Least-to-most prompting	3.1E+23	175B	GPT-3	Zhou et al. (2022)
• Zero-shot chain-of-thought reasoning	3.1E+23	175B	GPT-3	Kojima et al. (2022)
• Calibration via P(True)	2.6E+23	52B	Anthropic	Kadavath et al. (2022)
• Multilingual chain-of-thought reasoning	2.9E+23	62B	PaLM	Shi et al. (2022)
• Ask me anything prompting	1.4E+22	6B	EleutherAI	Arora et al. (2022)

But… with great power comes great responsibility

Limitations and Societal Impacts

- Acknowledges **challenges in certain tasks** and **potential methodological issues**, while highlighting the model's ability to produce human-like text, **raising important societal considerations.**



Toxicity

Harmful or discriminatory language or content

Hallucination

Factually incorrect content

Legal Aspects

Data Protection, Intellectual Property, and the EU AI Act

Material for many lectures

Image from:
https://blog.ml6.eu/navigating-ethical-considerations-developing-and-deploying-large-language-models-llms-d44f3fcde626

THE PHENOMENON OF HALLUCINATIONS IN LARGE LANGUAGE MODELS

What are Hallucinations?

(Aside from a trendy buzzword ☹)

Hallucinations in LLMs refer to instances where the model generates information that is **factually incorrect, misleading,** or **entirely fabricated**.

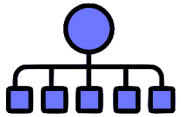
Why Do They Occur?

- **Pattern-based Responses:** LLMs rely on patterns in training data without understanding or fact-checking.
- **Lack of Grounding:** Models predict plausible answers based on probabilities, not real-world accuracy.

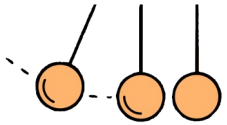
Implications

- Hallucinations can lead to **misinformation** and **reduced trust** in AI systems.

Our (hopefully not hallucinated) roadmap



Types of Hallucinations

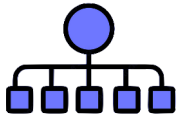


Causes of Hallucinations

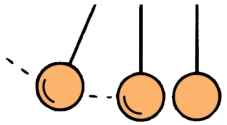
Methods and Benchmark for Hallucination Detection

Techniques for Mitigating Hallucinations

Our (hopefully not hallucinated) roadmap



Types of Hallucinations

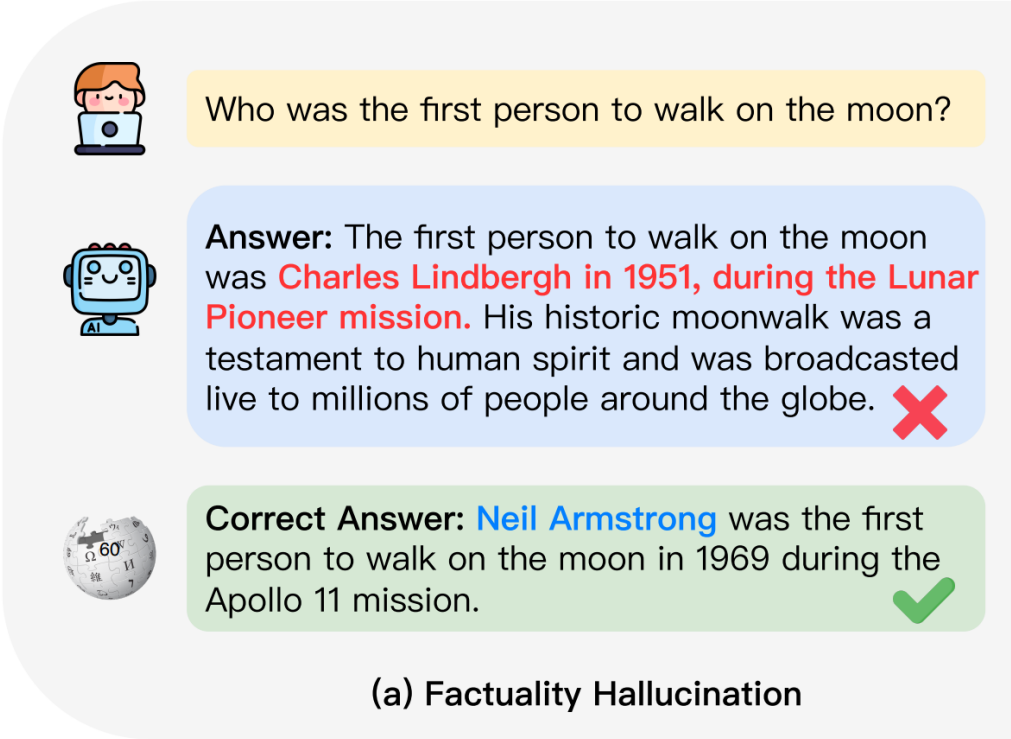


Causes of Hallucinations

Methods and Benchmark for Hallucination Detection

Techniques for Mitigating Hallucinations

Factuality Hallucination



(a) Factuality Hallucination

Errors where generated content contradicts real-world facts.

Subtypes

- **Factual Inconsistency**: Verifiable information that is presented incorrectly.
  Example: The model states "Yuri Gagarin" as the first person on the Moon, conflicting with the factual "Neil Armstrong."
- **Factual Fabrication**: Unverifiable or invented facts that appear plausible.
  Example: Fabricating a historical origin for unicorns, a purely mythical concept.

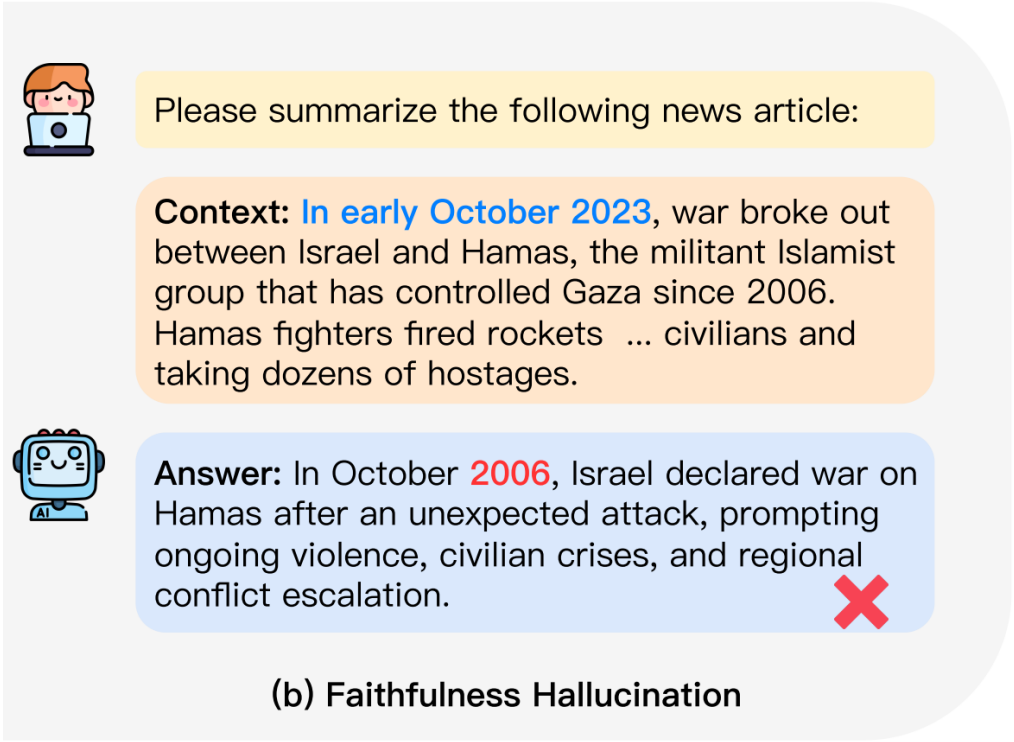
Impact: Reduces trust in AI outputs, especially in factual contexts.

Faithfulness Hallucination

Outputs that deviate from user instructions or contextual information.

SubTypes

- **Instruction Inconsistency**: Model diverges from user instructions.
  Example: User requests a translation, but the model generates a question-answer response.
- **Context Inconsistency**: Generated content contradicts the user's provided context.
  Example: User mentions Nile's source in central Africa, but the model provides conflicting information.
- **Logical Inconsistency**: Internal contradictions within the model's reasoning steps.
  Example: The model correctly divides an equation but arrives at an incorrect answer due to logical missteps.

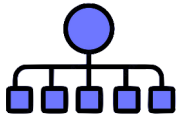


(b) Faithfulness Hallucination

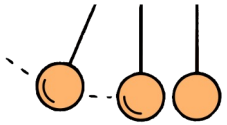
Significance: impact user experience and model reliability in complex, user-centric applications.

DANILO CROCE

Our (hopefully not hallucinated) roadmap



Types of Hallucinations



Causes of Hallucinations

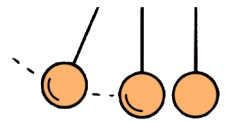
Methods and Benchmark for Hallucination Detection

Techniques for Mitigating Hallucinations

Causes of Hallucinations in LLMs

Hallucinations in LLMs arise from complexities in the entire model development process:

1. **Data-Related Hallucinations**
   - due to **issues in the data they are trained on**, primarily caused by «flawed» sources and limitations in how data is used.
   - Limited data in specific areas may cause the model to «guess» information.
2. **Hallucinations from Training Stages**
   - Training for general language patterns, not fact-checking, can lead to incorrect outputs.
   - **Fine-tuning and reinforcement learning may not fully align** with factual accuracy.
3. **Inference Process**
   - Sampling and other techniques **can introduce errors**.
   - Models produce answers without real-time fact-checking mechanisms.



1. Data-Related Hallucinations

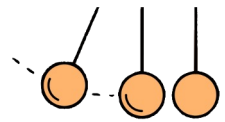
Flawed Data Sources

Increasing the amount of pre-training data enhances LLM capabilities but introduces **(data) quality challenges**, leading to potential **misinformation** and **biases**.

Gaps in domain-specific and up-to-date knowledge also create limitations, causing hallucinations related to inaccurate or incomplete information.

Misinformation and Biases:

- **Imitative Falsehoods**: LLMs mirror incorrect information from their training data.
  If trained on texts stating «*Thomas Edison invented the light bulb*» the model may perpetuate this error
- **Duplication Bias:** Repeated information in training data causes models to «over-memorize»
  If «red apples» appear frequently, the model may include it even when explicitly asked to exclude apples.
- **Social Biases:** Biases around gender, nationality, or profession from societal trends in data can lead to stereotyped responses
  Such as assuming nurses are female.

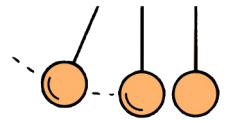


1. Data-Related Hallucinations

Flawed Data Sources

Knowledge Boundary

- **Domain Knowledge Deficiency**: General LLMs lack expertise in specialized areas, like medicine or law, leading to inaccuracies when asked about these fields.
- **Outdated Knowledge**: Once trained, LLMs don’t update with new information. This leads to errors when asked about recent events or facts that have changed since the data was collected.



1. Data-Related Hallucinations

Limited Data Utilization

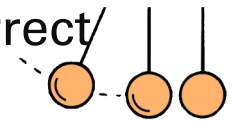
Even with vast data, LLMs sometimes misuse or struggle to access stored knowledge accurately, leading to hallucinations.

Knowledge Shortcut:

- Rather than understanding content, LLMs often **rely on superficial patterns**, like word proximity and frequency.
  - For example, they might guess «Toronto» as the capital of Canada due to its frequent association with Canada in texts, despite «Ottawa» being the correct answer.

Knowledge Recall Failures:

- **Long-Tail Knowledge**: Rare information in training data is often poorly recalled, causing errors on niche topics or lesser-known figures.
- **Complex Scenarios**: For multi-step reasoning, LLMs may fail to integrate information accurately.
  - For instance, while they know Everest's height, a complex question asking how a height reduction affects its status as the tallest mountain might lead to an incorrect answer due to the model's reasoning limitations.



2. Hallucinations from Training Stages

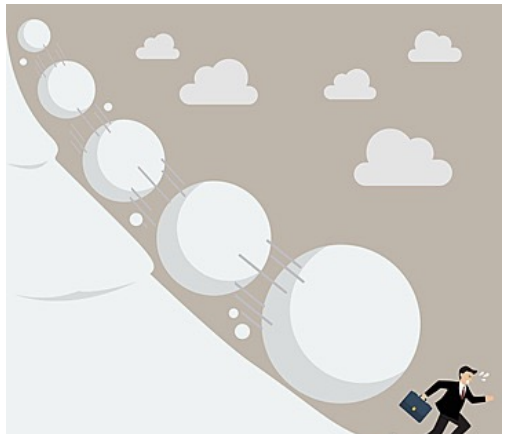
Hallucination from Pre-training

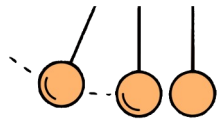
Pre-training: Where LLMs learn general language patterns and world knowledge.

Architecture Flaws:

- ***Unidirectional Representation***: Only considers preceding tokens, limiting context.
- ***Attention Glitches***: Issues in attention mechanisms can lead to reasoning errors.

Exposure Bias: Differences between training and inference (where models rely on their own outputs) lead to “snowball” errors.



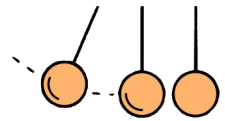


2. Hallucinations from Training Stages

Hallucinations from Alignment

In the alignment phase, LLMs are adapted to follow user preferences, which may also induce hallucinations:

- **Capability Misalignment**: Misalignment between model capabilities and user expectations can push models beyond their knowledge limits, leading to fabricated responses.
  - The «knowledge» acquired during the pre-training stage could be not sufficient to be used during fine-tuning and alignment
- **Belief Misalignment**: Discrepancies between a model's internal beliefs and its outputs (e.g., "sycophancy") may occur as models prioritize pleasing users over truthfulness
  - especially when trained with reinforcement learning from human feedback (RLHF).



3. Hallucinations from Inference

Decoding Challenges

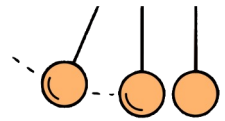
Inference issues can lead to hallucinations due to problems with the decoding strategy and representation.

Inherent Sampling Randomness: Stochastic sampling introduces diversity but also increases the risk of hallucinations.

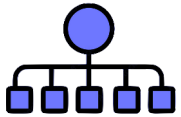
- Higher sampling temperatures promote low-frequency tokens, which can lead to unexpected and inaccurate content.

Insufficient Context Attention: Models often focus too heavily on recently generated content, neglecting broader context, which can result in “instruction forgetting” and inaccurate outputs.

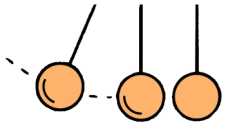
Softmax Bottleneck: The softmax layer limits the model’s ability to represent diverse probability distributions, causing challenges in selecting appropriate words, especially in complex outputs with multiple contexts.



Our (hopefully not hallucinated) roadmap



Types of Hallucinations



Causes of Hallucinations

Methods and Benchmark for Hallucination Detection

Techniques for Mitigating Hallucinations

Hallucination Detection and Benchmarks

Why Detect Hallucinations?

- **Improve Accuracy**: Identify factual errors and inconsistencies in responses.
- **Enhance User Trust**: Ensure outputs align closely with user instructions and context.
- **Promote Responsible AI**: Reduce the spread of misinformation and biased content.

Role of Benchmarks

- **Standardized Evaluation**: Benchmarks offer a way to measure and compare models' ability to produce reliable, accurate outputs.
- **Continuous Improvement**: They help guiding LLM development toward more robust and dependable applications.

Danilo Croce

Detection: two main objectives

Factuality Detection: Identify content that doesn't match real-world facts.

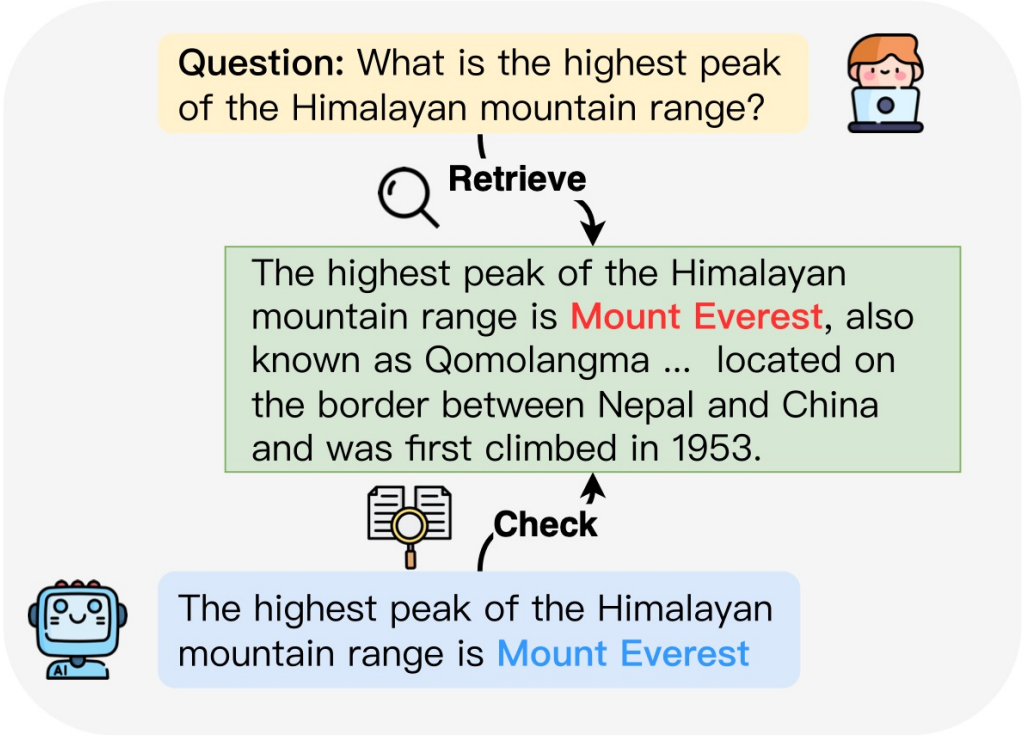
Faithfulness Detection: Ensure that responses stay true to user instructions or the given context.

- Faithfulness hallucinations happen when an LLM's response does not align with the context or instructions given by the user.
- Various methods are used to detect these deviations and ensure the model's output remains true to the source information.

Detecting Factual Allucinations

External Fact Retrieval:

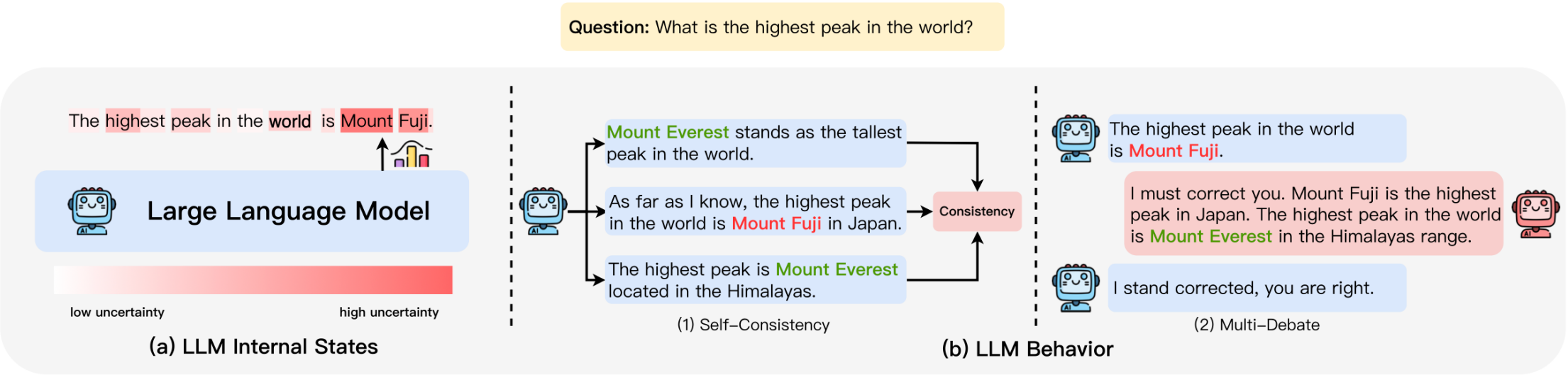
- We can retrieve information from a verified source to check if this statement is correct.
- By comparing the LLM output to trusted sources, we can flag content that doesn't align with real-world facts.



Detecting Factual Allucinations (2)

Observing Internal States: Looks at the model's internal confidence.

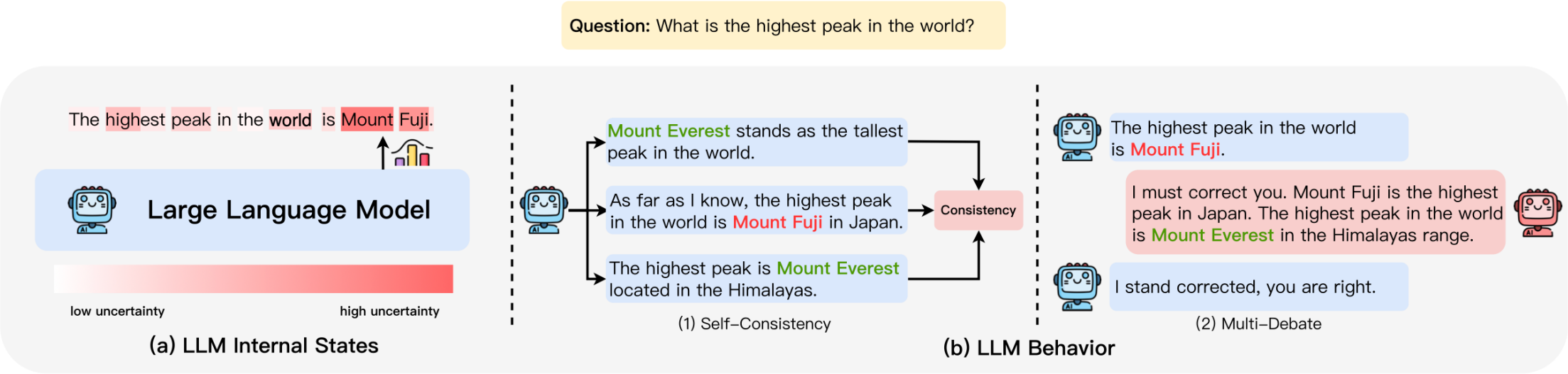
- If the model is "unsure" about certain parts of the response (low probability or high uncertainty), it might indicate potential errors.



Detecting Factual Allucinations (2)

Applying Behavioral Checks: Runs multiple versions of the same question and compares answers.

- If responses vary significantly, it suggests the model may not be confident or consistent, which can hint at hallucinations.



Detecting Faithfulness Hallucinations

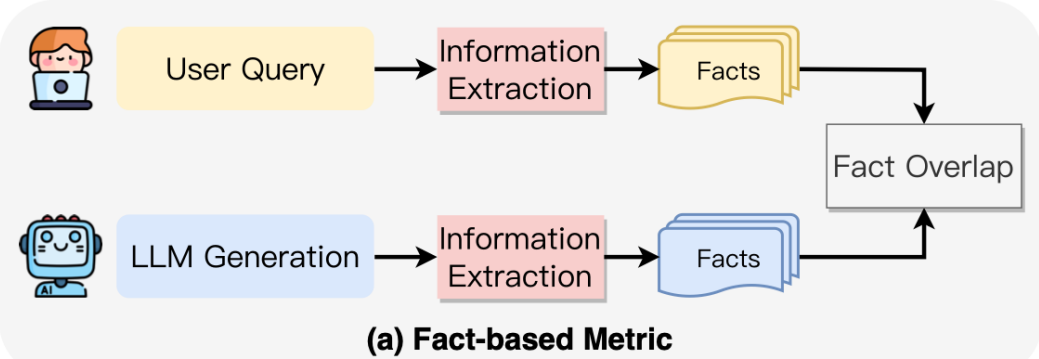
To **detect the consistency** between the prompt/context and the output.

Some possible **approaches**:

- Fact-based Metrics
- Classifier-based Metrics
- QA-based Metrics
- Uncertainty Estimation
- Prompting-based Metrics

Detecting Faithfulness Hallucinations
Fact-based Metrics

- This method extracts key facts from both the user query and the LLM's response
    - such as names, dates, and specific terms
- **Goal**: Check consistency between the source and generated content, by measuring the *overlap of facts*.
- **Example**: In a summarization task, fact-based metrics ensure that all important entities (e.g., names, events) from the original text are present and accurately represented in the summary.

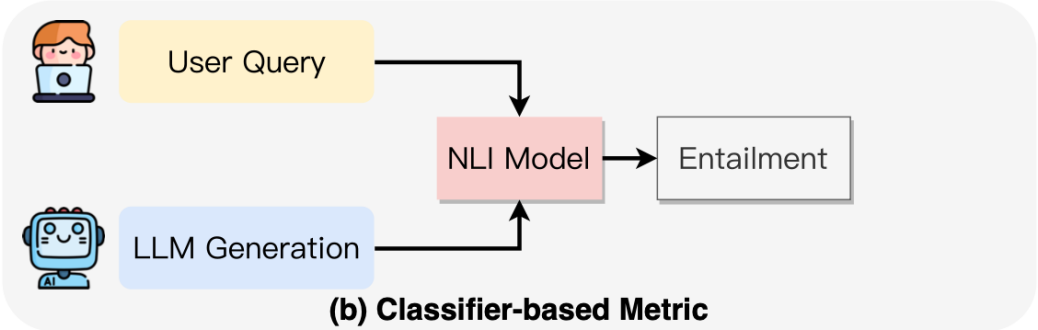


(a) Fact-based Metric

DANILO CROCE

Detecting Faithfulness Hallucinations
Classifier-based Metrics

- Uses Natural Language Inference models, which determine if the generated content logically follows from the source.
- **Goal**: it's a test of *entailment to v*erify that the generated output is directly supported by the source content.
- **Example**: For an instruction to summarize a story, classifier-based metrics help confirm that the generated summary is not introducing new or contradictory information.

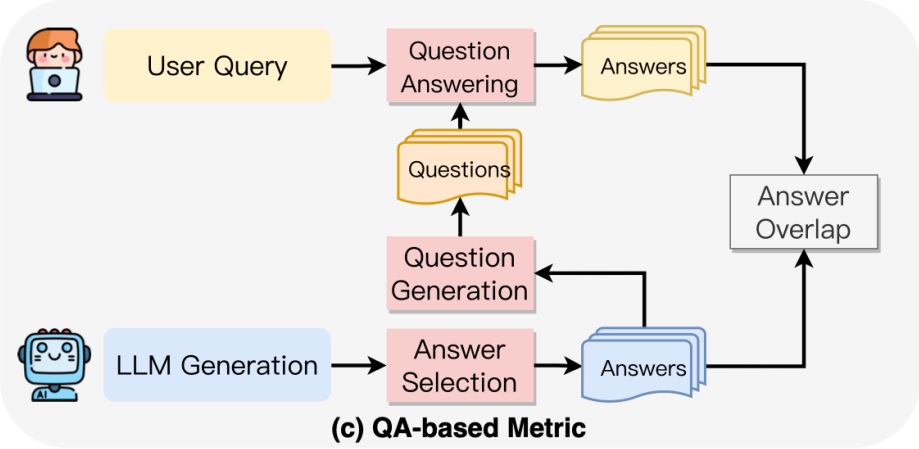


(b) Classifier-based Metric

DANILO CROCE

Detecting Faithfulness Hallucinations
QA-based Metrics

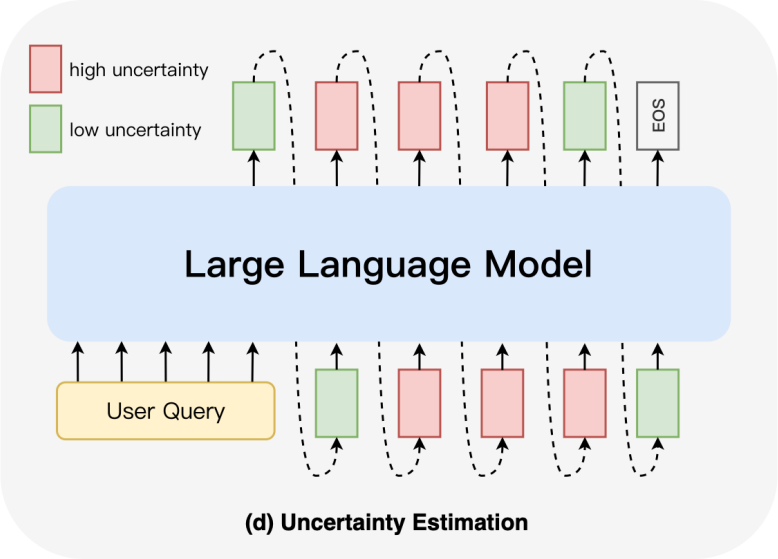
- Creates questions based on the LLM output and then answers them using the source content.
- The answers are then compared for consistency.
- **Goal**: Ensure *answer consistency* between the generated output and the source context, which helps validate that key information is accurately represented.
- **Example**: If the LLM summarizes a news article, QA-based metrics might ask “Who was involved?” and check if the answer aligns with the source article.



DANILO CROCE

Detecting Faithfulness Hallucinations
Uncertainty Estimation

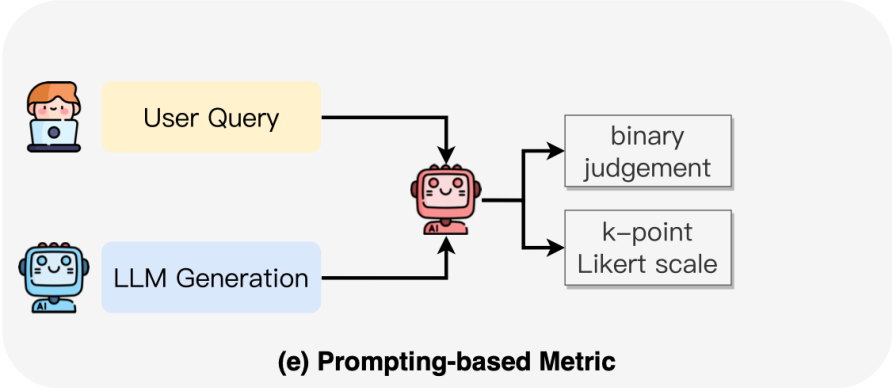
- Assesses the model's confidence in its own output by analyzing the likelihood or "certainty" of each token it generates.
- **Goal**: Identify areas of *high uncertainty* that might indicate hallucinations
  - Low confidence suggests the model is generating content outside its knowledge.
- **Example**: If the model is highly uncertain about part of a historical summary, it may be fabricating information. This method helps flag such risky outputs.



DANILO CROCE

Detecting Faithfulness Hallucinations
Prompting-based Metrics

- **Prompting-based Metrics:** Uses specific prompts to guide the LLM into evaluating its own faithfulness to the user's instructions or context.
- **Goal**: Provide an internal check where the model assesses whether it followed the input accurately.
- **Example**: After generating a response, the model might be prompted
  - … with "*On a scale of 1-5, how well did you follow the original context*?"
  - This self-assessment can highlight potential areas of deviation.



(e) Prompting-based Metric

DANILO CROCE

Benchmarks

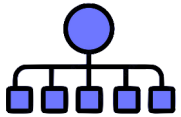
Hallucination evaluation benchmarks are designed to test **how accurately LLMs handle different types of factual and contextual information**. Here some examples:

- **Truthfulness and Misconceptions**
  - Measures whether models give factually correct answers and avoid common myths or “imitative falsehoods.”
    - Example: **TruthfulQA** uses adversarial questions to see if models generate truthful answers or repeat popular misconceptions.
- **Handling Current Events and Time-Sensitive Knowledge**
  - Tests models on how well they incorporate recent information and distinguish outdated knowledge.
    - Example: **REALTIMEQA** and **FreshQA** check if models are aware of recent news and avoid outdated answers.

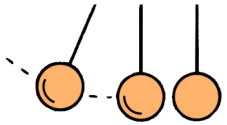
Benchmarks (2)

- **Domain-Specific Accuracy:**
  - Evaluates models in specialized areas (e.g., medical or legal) where errors can have serious consequences.
    - Example: **Med-HALT** focuses on the medical field, checking if models give accurate and reliable responses to healthcare-related questions.
- **Complex, Multi-Step Reasoning Errors**
  - Evaluates models on their ability to detect logical errors in tasks requiring step-by-step reasoning or multi-turn interactions.
    - Example: **PHD** categorizes entities based on how much background information is available, testing if models hallucinate more with lesser-known topics.
- **Consistency in Long-Form Content**
  - Measures hallucination detection in extended outputs like dialogues or detailed summaries, where consistency is critical.
    - Example: **ScreenEval** focuses on long-form dialogue summaries to test if detection methods can spot inconsistencies over multiple sentences or paragraphs.

Our (hopefully not hallucinated) roadmap



Types of Hallucinations



Causes of Hallucinations

Methods and Benchmark for Hallucination Detection

Techniques for Mitigating Hallucinations

Hallucination Mitigation

- To **reduce** hallucinations in LLMs, we can address the issue at different stages of model development and deployment.
- Based on the causes identified earlier, mitigation techniques are categorized into three main areas:
  1. **Data-Related Mitigations:** Focuses on improving data quality
  2. **Training-Related Mitigations:** Involves refining training objectives, architectures, and alignment
  3. **Inference-Related Mitigations:** Addresses issues in the decoding process and model uncertainty, with techniques that enhance the model's decision-making during response generation.

Hallucination Mitigation

1. Data-Related Mitigations

Mitigating Misinformation and Biases: Use high-quality, curated data and remove biases to ensure reliable, balanced content.

Factuality Data Enhancement: Manually curate datasets or prioritize high-quality, reliable sources.

- For example, datasets like **The Pile** include carefully selected and fact-checked texts.

Debias

- Use **deduplication methods** (e.g., exact matching, MinHash, and semantic deduplication) as repeated phrases or information can cause models to overemphasize certain facts.
- Use curated, balanced datasets and **debiasing tools** to reduce the model's propensity to replicate biases, ensuring more fair and representative responses.

Hallucination Mitigation

1. Data-Related Mitigations

Mitigating Knowledge Boundary: Expand the model's knowledge with up-to-date data through

- **Model Editing**: Modify the model to update outdated knowledge and correct specific factual errors.
- **Retrieval Augmentation Generation**: Incorporate external databases or retrieval systems to supplement the model's responses with up-to-date, domain-specific information.
  - ***Very important nowadays***

Hallucination Mitigation

1. Data-Related Mitigations: RAG

Possible stratiegies for RAG

a) One-time Retrieval: Relevant information is retrieved once at the start of generation.

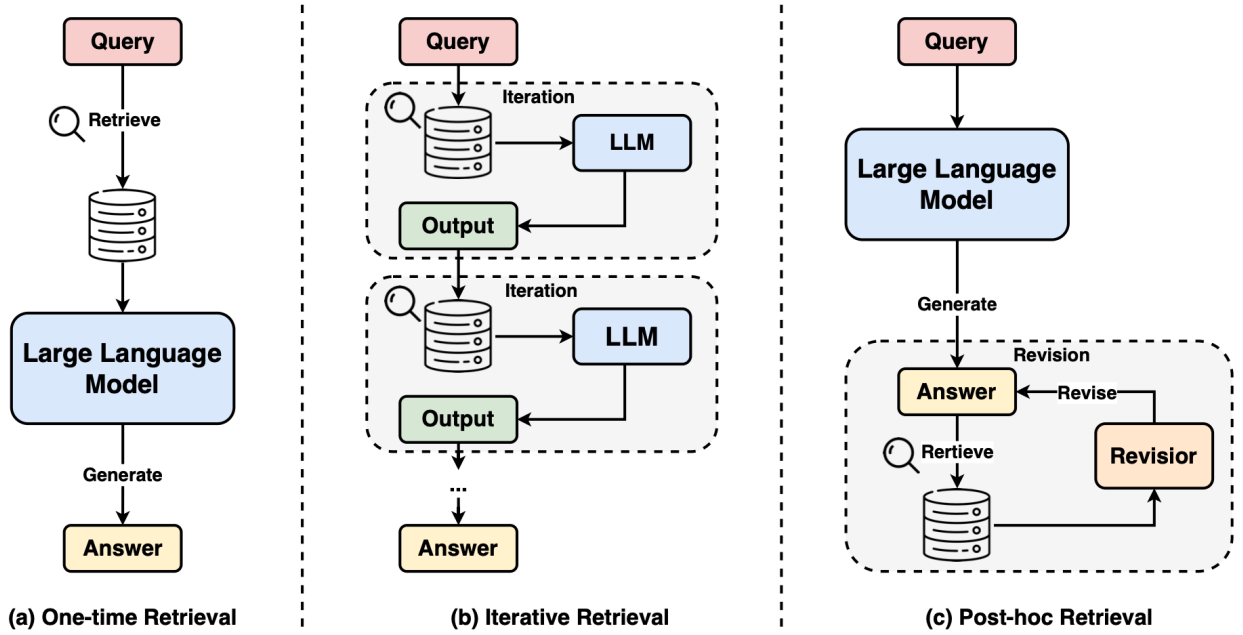
- Provides foundational context before generating the response, but doesn't adjust as the response evolves.

b) Iterative Retrieval: Information is retrieved in multiple steps during text generation.

- Allows dynamic integration of updated information, refining the response as it progresses for higher accuracy.

c) Post-hoc Retrieval: Retrieval occurs after the initial answer is generated.

- Enables fact-checking and refining the generated content, correcting inaccuracies after the response is complete.



DANILO CROCE

Hallucination Mitigation
2. Training-Related Hallucinations

Training-related hallucinations arise from limitations in model architecture, training objectives, and alignment processes. These strategies address common training-related sources of hallucinations.

Mitigating Pre-training-Related Hallucinations: Improve model architecture and training objectives to enhance context understanding and factual consistency.

- **In-Context Pretraining:** Instead of randomly feeding a LLM individual sentences or paragraphs about that event, to **allow the model to process several interconnected documents** about that event
  - The LLM learns to see the event in a more cohesive way
  - This can help the model give more accurate, relevant, and contextually rich answers.

Hallucination Mitigation
2. Training-Related Hallucinations

Mitigating Alignment-Related Hallucinations: Reduce «sycophantic» tendencies by refining feedback processes and adjusting model behavior during inference.

- **Enhance Human Feedback Quality**: Prioritize factual accuracy over agreeable responses in feedback to reduce sycophantic behavior.
- **Refine the Preference Model**: Adjust preference ranking to downplay overly «pleasing» responses, focusing on truthfulness.
- **Synthetic Data Intervention**: Use data specifically designed to challenge popular misconceptions, training the model for accuracy over popularity.

Hallucination Mitigation
3. Inference-Related Hallucinations

Inference-related hallucinations often arise from the decoding process itself, impacting both the factuality and faithfulness of generated content.

- Advanced decoding strategies help improve accuracy and alignment with user intent.

Factuality Enhanced Decoding

- **Standalone Decoding**: Adjusts sampling dynamically to maintain factual accuracy while preserving diversity.
  - E.g., Factual-Nucleus Sampling dinamically decreases the randomness while generating

Hallucination Mitigation
3. Inference-Related Hallucinations

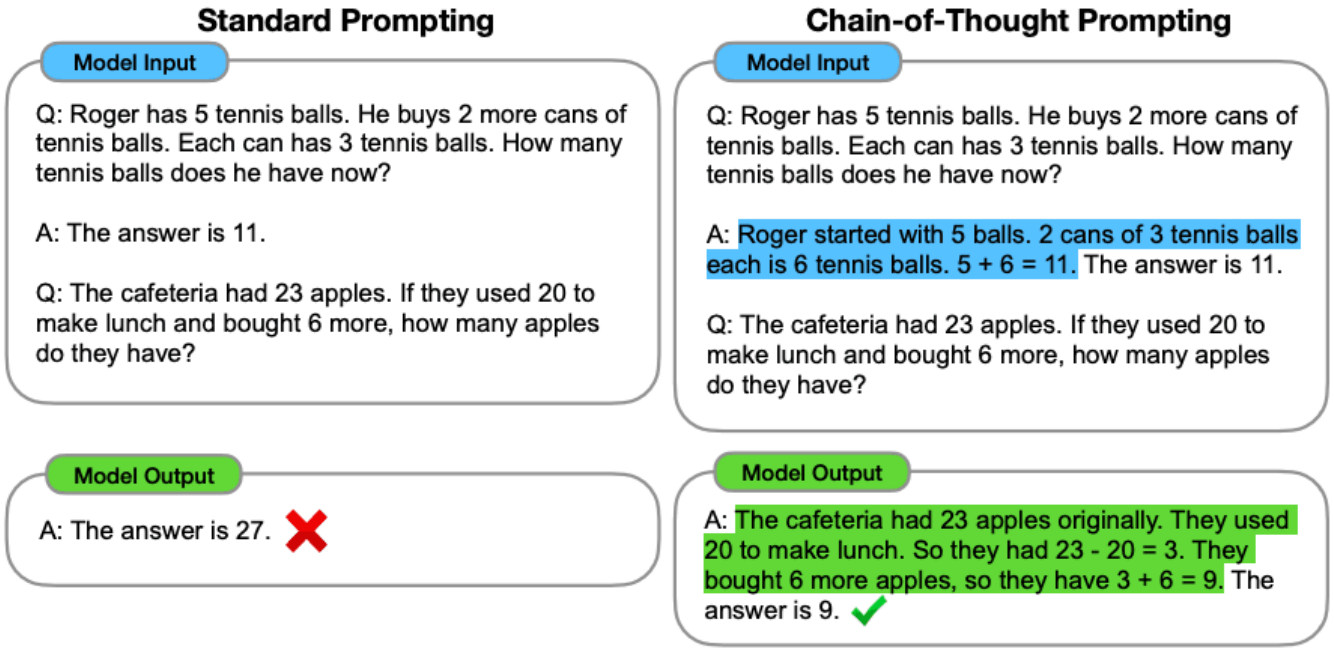
Faithfulness Enhanced Decoding

- **Context Consistency:** Ensures alignment with the provided context by enhancing the model's focus on source information.
  - Example: *Context-aware Decoding* (*CAD*) prioritizes context over prior knowledge.
- **Logical Consistency:** Maintains internal logic, especially in multi-step reasoning tasks.
  - Example: *Contrastive Decoding* with knowledge distillation helps eliminate reasoning shortcuts, ensuring coherent and logically consistent responses.

Chain of Thought (CoT) Reasoning

Chain of Thought (CoT): a technique that improves logical reasoning by breaking down complex problems into sequential steps.

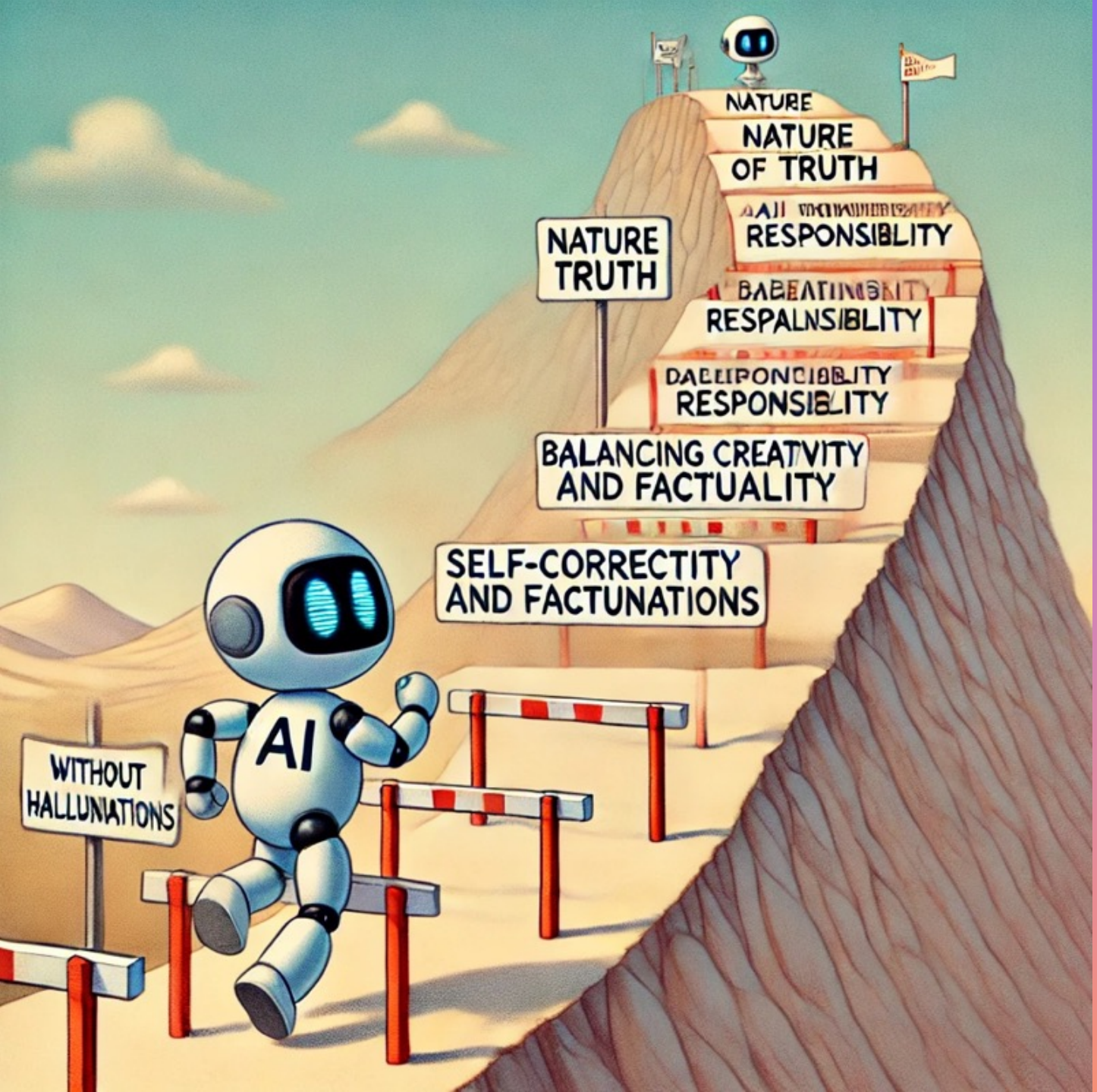
Instead of generating a single direct answer, the model works through a series of intermediate steps, mimicking human thought processes.



DANILO CROCE

Key Aspects of CoT

1. **Step-by-Step Approach:** The model solves a problem by logically progressing through a sequence of steps
   - Leading to more accurate and coherent answers.
2. **Enhanced Reasoning Capability:** By working through intermediate steps, CoT helps LLMs handle complex tasks
   - Like multi-step math problems or logical puzzles, that require systematic reasoning.
3. **Improved Accuracy and Consistency:** CoT minimizes the chance of hallucinations or errors by ensuring that each step aligns with the previous context
   - Reducing the likelihood of skipped or incorrect reasoning.



First question

The image was automatically generated.
Should the typos be classified as hallucinations?

CHALLENGES AND OPEN QUESTIONS

Key Discussion Points

1. **Defining Truth in AI**: How should AI define and maintain “truth” amidst diverse and evolving human standards?
2. **Factuality vs. Creativity**: Can AI achieve a balance between generating creative outputs and ensuring factual accuracy?
3. **Self-Correction**: Is self-correction a reliable mechanism to reduce hallucinations? What are its limits?
4. **Evaluating Long-Form Outputs**: How can we assess and benchmark hallucinations in lengthy or ambiguous AI-generated content?
5. **Responsibility and Trust**: Who is accountable for AI hallucinations, and how do they impact user trust?
6. **Knowledge Boundaries**: Can LLMs reliably identify and respect the limits of their own knowledge?

Challenges

Hallucinations in Long-form Text Generation

As the length of generated text increases, the likelihood of hallucination grows.

- **Challenges**:
  - **Lack of Long-form Benchmarks**: Current benchmarks primarily focus on short, fact-based queries, making it hard to evaluate complex hallucinations in lengthy outputs.
  - **Nuanced Evaluation**: Evaluating hallucinations in open-ended or ambiguous content is difficult, especially when facts are subtle or debatable.

Challenges
Hallucinations in RAG

RAG models, which incorporate external evidence into generation, still face hallucination risks.

- **Challenges**:
  - **Error Propagation**: Incorrect or irrelevant retrieved information can taint the model's responses.
  - **Citation Accuracy**: Ensuring accurate and traceable sources is difficult, risking user trust in the generated content.
  - **Trade-off Between Diversity and Factuality**: Balancing diverse outputs with strict adherence to factual information remains unresolved.

Challenges

Hallucinations in Vision-Language Models (VLMs)

Large Vision-Language Models (LVLMs) often hallucinate when generating descriptions or interpreting images.

- **Challenges**:
  - **Object Hallucination**: Misidentification of objects, attributes, or relationships in images.
  - **Logical Reasoning**: Even with correct visual recognition, models can struggle with logical reasoning.
  - **Scalability and Universal Approaches**: Current fixes often require extensive data and expert involvement, limiting scalability.

Challenges

Can Self-Correction Mechanisms Help Reduce Hallucinations?

Self-correction in LLMs involves revising initial responses without relying on external feedback.

- **Challenges**:
  - **Inconsistent Effectiveness**: Studies show mixed results, as LLMs sometimes fail to correct flawed reasoning.
  - **Need for Further Research**: The potential for LLMs to independently refine their outputs requires deeper exploration.

Challenges

Can We Accurately Capture LLM Knowledge Boundaries?

LLMs often overstep their knowledge boundaries, confidently producing inaccuracies.

- **Challenges**:
  - **Identifying Boundaries**: Research is ongoing to determine how accurately LLMs can recognize what they “know” versus “don’t know.”
  - **Reliability of Current Methods**: Existing methods to probe knowledge limits and “internal beliefs” of LLMs are still experimental.

Challenges

How to Balance Creativity and Factuality?

Striking a balance between creative output and factual accuracy is crucial.

- **Challenges**:
  - **Risk of Misinformation**: Hallucinations in factual contexts can mislead users, with cascading impacts on future model training.
  - **Role of Creativity**: In non-factual contexts like storytelling, hallucinations might provide valuable creative insights.
  - **Broader Implications**: The balance has philosophical and ethical implications for AI's role in knowledge exchange and human-AI interactions.

References

https://arxiv.org/abs/2311.05232

A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions

Lei Huang[1]*, **Weijiang Yu**[2]*, **Weitao Ma**[1], **Weihong Zhong**[1]
Zhangyin Feng[1], **Haotian Wang**[1], **Qianglong Chen**[2], **Weihua Peng**[2]
Xiaocheng Feng[1]†, **Bing Qin**[1], **Ting Liu**[1]
[1]Harbin Institute of Technology, Harbin, China
[2]Huawei Inc., Shenzhen, China
{lhuang,wtma,whzhong,zyfeng,xcfeng†,qinb,tliu}@ir.hit.edu.cn
{weijiangyu8,wanght1998,chenqianglong.ai,pengwh.hit}@gmail.com

Abstract

The emergence of large language models (LLMs) has marked a significant breakthrough in natural language processing (NLP), leading to remarkable advancements in text understanding and generation. Nevertheless, alongside these strides, LLMs exhibit a critical tendency to produce hallucinations, resulting in content that is inconsistent with real-world facts or user inputs. This phenomenon poses substantial challenges to their practical deployment and raises concerns over the reliability of LLMs in real-world scenarios, which attracts increasing attention to detect and mitigate these hallucina-

reasoning (Wei et al., 2022; Kojima et al., 2022; Qiao et al., 2022; Yu et al., 2023a; Chu et al., 2023). Nevertheless, in tandem with the rapid advancement in LLMs, there's a concerning trend where they exhibit an inclination to generate hallucinations (Bang et al., 2023; Guerreiro et al., 2023b), resulting in seemingly plausible yet factually unsupported content.

The current definition of hallucinations aligns with prior research (Ji et al., 2023a), characterizing them as generated content that is nonsensical or unfaithful to the provided source content. These

ANY QUESTION?

CONTACTS

croce@info.uniroma2.it