



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Machine learning – Large Language Models

Måns Magnusson
Department of Statistics, Uppsala University

Autumn 2023



UPPSALA
UNIVERSITET

- **Introduction**

- "Emergent" abilities
- In-context learning (ICL)
- Hallucinations

- **Architectures**

- **Training**

- **Fine-tuning and Alignment**

- Instruction tuning
- Alignment
- RLHF

- **Use**

- Prompt engineering

Section 1

Introduction



What is a Large Language Model?

- **Introduction**

- "Emergent" abilities
- In-context learning (ICL)
- Hallucinations

- **Architectures**

- **Training**

- **Fine-tuning and Alignment**

- Instruction tuning
- Alignment
- RLHF

- **Use**

- Prompt engineering

- Large Language Models (LLM) are commonly defined as:
 - large natural language models



What is a Large Language Model?

- **Introduction**

- "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations

- **Architectures**

- **Training**

- **Fine-tuning and Alignment**

- Instruction tuning
 - Alignment
 - RLHF

- **Use**

- Prompt engineering

- Large Language Models (LLM) are commonly defined as:
 - large natural language models
 - generative and **autoregressive**: predicting a token at a time, based on previous **context**



- **Introduction**

- "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations

- **Architectures**

- **Training**

- **Fine-tuning and Alignment**

- Instruction tuning
 - Alignment
 - RLHF

- **Use**

- Prompt engineering

What is a Large Language Model?

- Large Language Models (LLM) are commonly defined as:
 - large natural language models
 - generative and **autoregressive**: predicting a token at a time, based on previous **context**
 - having some ability to achieve **general-purpose language "understanding"**
 - show "**emergent**" abilities to solve other more complex tasks, such as few-shot learning and **in-context learning**



- **Introduction**
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

What is a Large Language Model?

- Large Language Models (LLM) are commonly defined as:
 - large natural language models
 - generative and **autoregressive**: predicting a token at a time, based on previous **context**
 - having some ability to achieve **general-purpose language "understanding"**
 - show "**emergent**" abilities to solve other more complex tasks, such as few-shot learning and **in-context learning**
 - pre-trained on very large data
- Large Language Models (LLM) are commonly:
 - decoder-type models



- **Introduction**
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

What is a Large Language Model?

- Large Language Models (LLM) are commonly defined as:
 - large natural language models
 - generative and **autoregressive**: predicting a token at a time, based on previous **context**
 - having some ability to achieve **general-purpose language "understanding"**
 - show "**emergent**" abilities to solve other more complex tasks, such as few-shot learning and **in-context learning**
 - pre-trained on very large data
- Large Language Models (LLM) are commonly:
 - decoder-type models
 - based on transformer architectures



Comparing to pre-trained models (such as BERT)

- **Introduction**

- "Emergent" abilities
- In-context learning (ICL)
- Hallucinations

- **Architectures**

- **Training**

- **Fine-tuning and Alignment**

- Instruction tuning
- Alignment
- RLHF

- **Use**

- Prompt engineering

- Compared to pretrained language models (PLM), LLMs are:
 - larger (billions or trillions, rather than millions of parameters), in practice only possible to train by a few persons



Comparing to pre-trained models (such as BERT)

- **Introduction**

- "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations

- **Architectures**

- **Training**

- **Fine-tuning and Alignment**

- Instruction tuning
 - Alignment
 - RLHF

- **Use**

- Prompt engineering

- Compared to pretrained language models (PLM), LLMs are:
 - larger (billions or trillions, rather than millions of parameters), in practice only possible to train by a few persons
 - possible to use for in-context learning



Comparing to pre-trained models (such as BERT)

- **Introduction**

- "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations

- **Architectures**

- **Training**

- **Fine-tuning and Alignment**

- Instruction tuning
 - Alignment
 - RLHF

- **Use**

- Prompt engineering

- Compared to pretrained language models (PLM), LLMs are:

- larger (billions or trillions, rather than millions of parameters), in practice only possible to train by a few persons
 - possible to use for in-context learning
 - usually interacted with through the prompt



Comparing to pre-trained models (such as BERT)

- **Introduction**

- "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations

- **Architectures**

- **Training**

- **Fine-tuning and Alignment**

- Instruction tuning
 - Alignment
 - RLHF

- **Use**

- Prompt engineering

- Compared to pretrained language models (PLM), LLMs are:

- larger (billions or trillions, rather than millions of parameters), in practice only possible to train by a few persons
 - possible to use for in-context learning
 - usually interacted with through the prompt
 - generate textual responses



UPPSALA UNIVERSITET

● Introduction

- "Emergent" abilities
- In-context learning (ICL)
- Hallucinations

● Architectures

● Training

● Fine-tuning and Alignment

- Instruction tuning
- Alignment
- RLHF

● Use

- Prompt engineering

Historical development

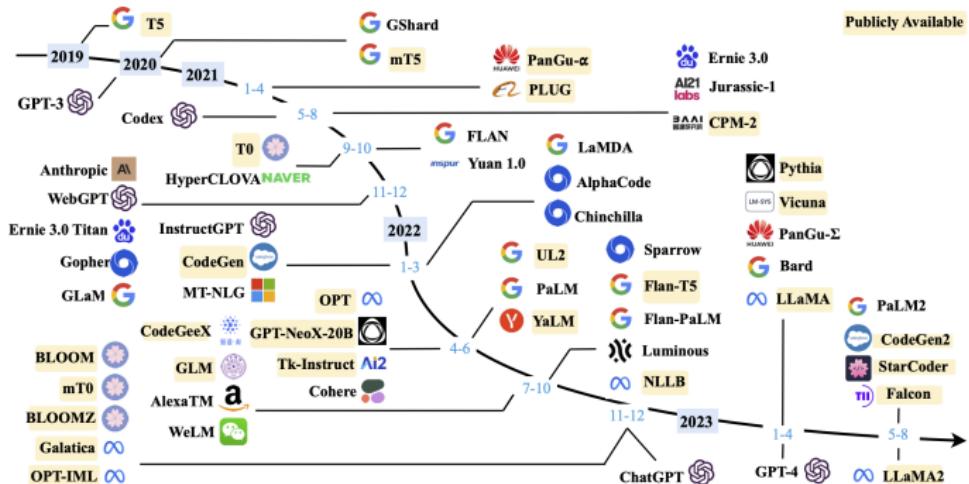


Figure: The development of LLMs (Figure 2, Zhao et al., 2023)



UPPSALA UNIVERSITET

● Introduction

- “Emergent” abilities
- In-context learning (ICL)
- Hallucinations

● Architectures

● Training

● Fine-tuning and Alignment

- Instruction tuning
- Alignment
- RLHF

● Use

- Prompt engineering

A subset of current LLMs

Model	Release Time	Size (B)	Base Model	Adaptation IT	RLHF	Pre-train Data Scale	Latest Data Timestamp	Hardware (GPUs / TPUs)	Training Time	Evaluation ICL	Evaluation CoT
T5 [73]	Oct-2019	11	-	-	-	1T tokens	Apr-2019	1024 TPU v3	-	✓	-
mT5 [74]	Oct-2020	13	-	-	-	1T tokens	-	-	-	✓	-
PanGu- α [75]	Apr-2021	13 [*]	-	-	-	1.1TB	-	2048 Ascend 910	-	✓	-
CPM-2 [76]	Jun-2023	198	-	-	-	2.6TB	-	-	-	-	-
T0 [28]	Oct-2021	11	T5	✓	-	-	-	512 TPU v3	27 h	✓	-
CodeGen [77]	Mar-2022	16	-	-	-	577B tokens	-	-	-	✓	-
GPT-NeoX-20B [78]	Apr-2022	20	-	-	-	825GB	-	96 40G A100	-	✓	-
Tk-Instruct [29]	Apr-2022	11	T5	✓	-	-	-	256 TPU v3	4 h	✓	-
UL2 [80]	May-2022	20	-	-	-	1T tokens	Apr-2019	512 TPU v4	-	✓	✓
OPT [81]	May-2022	175	-	-	-	180B tokens	-	992 80G A100	-	✓	-
NLLB [82]	Jul-2022	54.5	-	-	-	-	-	-	-	✓	-
CodeGenX [83]	Sep-2022	13	-	-	-	850B tokens	-	1536 Ascend 910	60 d	✓	-
GLM [84]	Oct-2022	130	-	-	-	400B tokens	-	768 40G A100	60 d	✓	-
Flan-T5 [64]	Oct-2022	11	T5	✓	-	-	-	-	-	✓	✓
Publicly Available	BLOOM [69]	Nov-2022	176	-	-	366B tokens	-	384 80G A100	105 d	✓	-
mT0 [85]	Nov-2022	13	mT5	✓	-	-	-	-	-	✓	-
Galactica [35]	Nov-2022	120	-	-	-	106B tokens	-	-	-	✓	✓
BLOOMV [85]	Nov-2022	176	BLOOM	✓	-	-	-	-	-	✓	-
OPT-IML [86]	Dec-2022	175	OPT	✓	-	-	-	128 40G A100	-	✓	✓
LLaMA [57]	Feb-2023	65	-	-	-	1.4T tokens	-	2048 80G A100	21 d	✓	-
Pythia [87]	Apr-2023	12	-	-	-	300B tokens	-	256 40G A100	-	✓	-
CodeGen2 [88]	May-2023	16	-	-	-	400B tokens	-	-	-	✓	-
StarCoder [89]	May-2023	15.5	-	-	-	1T tokens	-	512 40G A100	-	✓	✓
LLaMA2 [90]	Jul-2023	70	-	✓	✓	2T tokens	-	2000 80G A100	-	✓	-
Closed Source	GPT-3 [55]	May-2020	175	-	-	300B tokens	-	-	-	✓	-
	GShard [91]	Jun-2020	600	-	-	1T tokens	-	2048 TPU v3	4 d	-	-
	CodeX [92]	Jul-2021	12	GPT-3	-	100B tokens	May-2020	-	-	✓	-
	ERNIE 3.0 [93]	Jul-2021	10	-	-	375B tokens	-	384 V100	-	✓	-
	Jurassic-1 [94]	Aug-2021	178	-	-	300B tokens	-	800 GPU	-	✓	-
	HyperCLOVA [95]	Sep-2021	82	-	-	300B tokens	-	1024 A100	13.4 d	✓	-
	FLAN [62]	Sep-2021	137	LaMDA-PT	✓	-	-	128 TPU v3	60 h	✓	-
	Yuan 1.0 [96]	Oct-2021	245	-	-	180B tokens	-	2128 GPU	-	✓	-
	Anthropic [97]	Dec-2021	52	-	-	400B tokens	-	-	-	✓	-
	WebGPT [72]	Dec-2021	175	GPT-3	✓	-	-	-	-	✓	-
	Gopher [59]	Dec-2021	280	-	-	300B tokens	-	4096 TPU v3	920 h	✓	-
	ERNIE 3.0 Titan [98]	Dec-2021	260	-	-	-	-	-	-	✓	-
	GLaM [99]	Dec-2021	1200	-	-	280B tokens	-	1024 TPU v4	574 h	✓	-
	LaMDA [63]	Jan-2022	137	-	-	768B tokens	-	1024 TPU v3	57.7 d	-	-
	MT-NLG [100]	Jan-2022	530	-	-	270B tokens	-	4480 80G A100	-	✓	-
	AlphaCode [101]	Feb-2022	41	-	-	967B tokens	Jul-2021	-	-	-	-
	InstructGPT [61]	Mar-2022	175	GPT-3	✓	✓	-	-	-	✓	-
	Chinchilla [34]	Mar-2022	70	-	-	1.4T tokens	-	-	-	✓	-
	PaLM [56]	Apr-2022	540	-	-	780B tokens	-	6144 TPU v4	-	✓	✓
	AlexaTM [102]	Aug-2022	20	-	-	1.3T tokens	-	128 A100	120 d	✓	✓
	Sparrow [103]	Sep-2022	70	-	✓	-	-	64 TPU v3	-	✓	-
	WeLM [104]	Sep-2022	10	-	-	300B tokens	-	128 A100 40G	24 d	✓	-
	U-PaLM [105]	Oct-2022	540	PaLM	✓	-	-	512 TPU v4	5 d	✓	✓
	Flan-PaLM [64]	Oct-2022	540	PaLM	✓	-	-	512 TPU v4	37 h	✓	✓
	Flan-U-PaLM [64]	Oct-2022	540	U-PaLM	✓	-	-	-	-	✓	✓
	GPT-4 [46]	Mar-2023	-	✓	✓	-	-	-	-	✓	✓
	PanGu- Σ [106]	Mar-2023	1085	PanGu- α	✓	329B tokens	-	512 Ascend 910	100 d	✓	✓
	PaLM2 [107]	May-2023	16	-	✓	100B tokens	-	-	-	✓	✓

Figure: Statistics of LLMs (Table 1, Zhao et al., 2023)



Examples of LLM prompting

- **Introduction**

- "Emergent" abilities
- In-context learning (ICL)
- Hallucinations

- **Architectures**

- **Training**

- **Fine-tuning and Alignment**

- Instruction tuning
- Alignment
- RLHF

- **Use**

- Prompt engineering

Examples:

1. Can you please add 113329 and 719292? (true is 832621)
2. Who is Olof Palme? Please respond both in English and Swedish.

Llama 2

chatGPT



- **Introduction**
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- **Architectures**
- **Training**
- **Fine-tuning and Alignment**
 - Instruction tuning
 - Alignment
 - RLHF
- **Use**
 - Prompt engineering

Why is this working now?

- **Scaling of models:**
 - More computation thanks to the Transformer architecture



Why is this working now?

- **Introduction**

- "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations

- **Architectures**

- **Training**

- **Fine-tuning and Alignment**

- Instruction tuning
 - Alignment
 - RLHF

- **Use**

- Prompt engineering

- **Scaling of models:**

- More computation thanks to the Transformer architecture
 - Larger models (GPT3 175B, PaLM 540B)



Why is this working now?

- **Introduction**

- "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations

- **Architectures**

- **Training**

- **Fine-tuning and Alignment**

- Instruction tuning
 - Alignment
 - RLHF

- **Use**

- Prompt engineering

- **Scaling of models:**

- More computation thanks to the Transformer architecture
 - Larger models (GPT3 175B, PaLM 540B)
 - Larger datasets (1-2T tokens)

How long time to read 1T tokens at 10 t/s?



- **Introduction**

- "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations

- **Architectures**

- **Training**

- **Fine-tuning and Alignment**

- Instruction tuning
 - Alignment
 - RLHF

- **Use**

- Prompt engineering

Why is this working now?

- **Scaling of models:**

- More computation thanks to the Transformer architecture
 - Larger models (GPT3 175B, PaLM 540B)
 - Larger datasets (1-2T tokens)
How long time to read 1T tokens at 10 t/s?
 - All three are important (Zhou et al., 2023)
Can we connect this back to the standard ML framework?



Why is this working now?

- **Introduction**

- "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations

- **Architectures**

- **Training**

- **Fine-tuning and Alignment**

- Instruction tuning
 - Alignment
 - RLHF

- **Use**

- Prompt engineering

- **Scaling of models:**

- More computation thanks to the Transformer architecture
 - Larger models (GPT3 175B, PaLM 540B)
 - Larger datasets (1-2T tokens)
How long time to read 1T tokens at 10 t/s?
 - All three are important (Zhou et al., 2023)
Can we connect this back to the standard ML framework?

- **Fine-tuning for to follow instructions (InstructGPT)**



"Emergent" abilities

- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

- "the abilities that are not present in small models but arise in large models" (Zhao et al, 2023)



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

"Emergent" abilities

- "the abilities that are not present in small models but arise in large models" (Zhao et al, 2023)
- One of the main differences between LLMs and PLMs
- Examples:
 - In-context learning (basis for prompting)
 - Instruction following
 - Step-by-step reasoning



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

In-context learning (ICL)

"In context learning is a paradigm that allows language models to learn tasks given only a few examples in the form of demonstrations." (Dong et al., 2023)

- Learning tasks in the actual **context** (previous tokens).



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

In-context learning (ICL)

"In context learning is a paradigm that allows language models to learn tasks given only a few examples in the form of demonstrations." (Dong et al., 2023)

- Learning tasks in the actual **context** (previous tokens).
- No updates of the weight - only using existing weights.



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

In-context learning (ICL)

"In context learning is a paradigm that allows language models to learn tasks given only a few examples in the form of demonstrations." (Dong et al., 2023)

- Learning tasks in the actual **context** (previous tokens).
- No updates of the weight - only using existing weights.
- We demonstrate what to do with a **few examples**



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

In-context learning (ICL)

"In context learning is a paradigm that allows language models to learn tasks given only a few examples in the form of demonstrations." (Dong et al., 2023)

- Learning tasks in the actual **context** (previous tokens).
- No updates of the weight - only using existing weights.
- We demonstrate what to do with a **few examples**
- The model "learn" what to do **in context**.



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

In-context learning (ICL)

"In context learning is a paradigm that allows language models to learn tasks given only a few examples in the form of demonstrations." (Dong et al., 2023)

- Learning tasks in the actual **context** (previous tokens).
- No updates of the weight - only using existing weights.
- We demonstrate what to do with a **few examples**
- The model "learn" what to do **in context**.
- We "prompt" LLMs for utilization
- Let data be

$$D_k = (f(x_1, y_1), \dots, f(x_k, y_k))$$

then

$$\hat{y} = \text{LLM}(I, \underbrace{(f(x_1, y_1), \dots, f(x_k, y_k))}_{\text{demonstrations}}, \underbrace{f(x_{k+1}, \dots)}_{\text{input}})$$

where I are the general instructions.



In-context learning (ICL)

- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

- The problem: How to design the instructions (*I*) in a good way? Prompt engineering.



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

In-context learning (ICL)

- The problem: How to design the instructions (*I*) in a good way? Prompt engineering.
- What happens under the hood? We position the model in embeddings space.
- What is good can be unintuitive?
 - "Take a deep breath and think hard."
 - Beeing nice.



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

In-context learning (ICL)

- The problem: How to design the instructions (*I*) in a good way? **Prompt engineering**.
- What happens under the hood? **We position the model in embeddings space**.
- What is good can be unintuitive?
 - "Take a deep breath and think hard."
 - Beeing nice.
- The effectiveness of ICL is highly affected by the **design of demonstrations** (Zhou et al., 2023)



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
 - Architectures
 - Training
 - Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
 - Use
 - Prompt engineering
- Demonstration selection
 - diversity-based selection strategies: choose the most representative set of examples
 - k-NN
 - LLM-based: e.g. let an LLM itself generate examples



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Demonstration

- Demonstration selection
 - diversity-based selection strategies: choose the most representative set of examples
 - k-NN
 - LLM-based: e.g. let an LLM itself generate examples
- Demonstration format/instruction
 - "Lets think step by step", "Take a deep breath and think."



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Demonstration

- Demonstration selection
 - diversity-based selection strategies: choose the most representative set of examples
 - k-NN
 - LLM-based: e.g. let an LLM itself generate examples
- Demonstration format/instruction
 - "Lets think step by step", "Take a deep breath and think."
- Demonstration order
 - Indications of recency bias



Chain of thought (CoT) prompting

- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

- Prompting strategy to improve performance in "reasoning"



Chain of thought (CoT) prompting

- Introduction

- "Emergent" abilities
- In-context learning (ICL)
- Hallucinations

- Architectures

- Training

- Fine-tuning and Alignment

- Instruction tuning
- Alignment
- RLHF

- Use

- Prompt engineering

- Prompting strategy to improve performance in "**reasoning**"
- Incorporates **intermediate reasoning steps**
- Instead of (input, output), we use (input, chain-of-thought, output)



Chain of thought (CoT) prompting

- Introduction

- "Emergent" abilities
- In-context learning (ICL)
- Hallucinations

- Architectures

- Training

- Fine-tuning and Alignment

- Instruction tuning
- Alignment
- RLHF

- Use

- Prompt engineering

- Prompting strategy to improve performance in "**reasoning**"
- Incorporates **intermediate reasoning steps**
- Instead of (input, output), we use (input, chain-of-thought, output)
- Zero-shot CoT: "Lets think step by step"



- **Introduction**
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- **Architectures**
- **Training**
- **Fine-tuning and Alignment**
 - Instruction tuning
 - Alignment
 - RLHF
- **Use**
 - Prompt engineering

Chain of thought (CoT) prompting

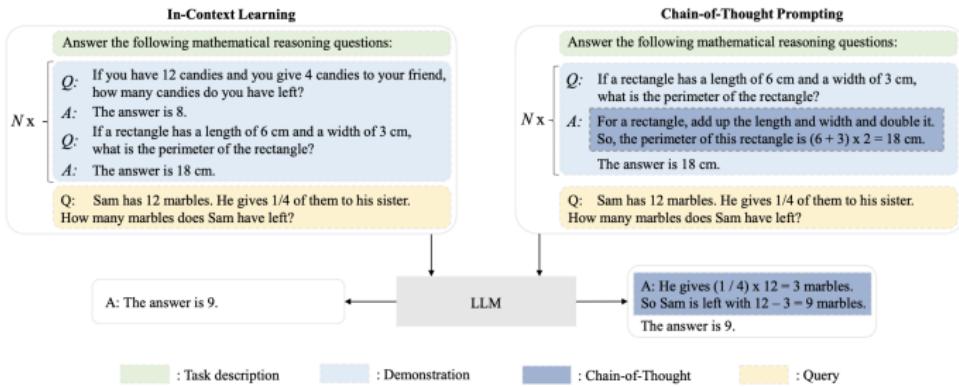


Figure: ICL vs CoT (Figure 12, Zhao et al., 2023)



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Hallucinations: What is it?

"Generated text that is fluent and natural but unfaithful"

- LLMs are prone to generate untruthful information, often called **hallucinations** that
 - logically contradicts the source content (intrinsic hallucination)
 - cannot be verified by the available source (extrinsic hallucination)
- This can be a major problem.
- See Huang et al (2023) for a survey on hallucination





- Introduction

- "Emergent" abilities
- In-context learning (ICL)
- Hallucinations

- Architectures

- Training

- Fine-tuning and Alignment

- Instruction tuning
- Alignment
- RLHF

- Use

- Prompt engineering

Hallucinations: What is it?

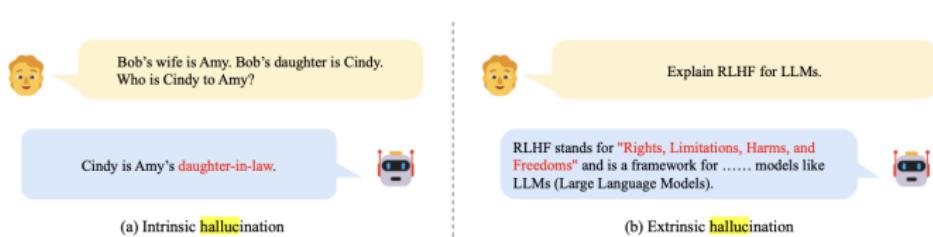


Figure: Hallucinations (Zhao et al., 2023, Figure 14)



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Hallucinations: Example



Who was the sole survivor of Titanic?



There were no survivors of the Titanic disaster. The Titanic sank on April 15, 1912, after colliding with an iceberg during its maiden voyage from Southampton to New York City. Of the more than 2,200 people on board, only around 700 survived the tragedy. The last person believed to have been rescued from the wreckage was a man named Jack Thayer, who was pulled from the water by a lifeboat early in the morning of April 15.

Figure: Hallucinations with Llama 2 (7B, from 2023-11-25)



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Hallucinations: Example



Who was the sole survivor of Titanic?



The sole survivor of the Titanic was a woman named Margaret "Molly" Brown, also known as the "Unsinkable Molly Brown." She was a first-class passenger on the ship and managed to survive the disaster by boarding lifeboat 6, which was launched with only 28 people on board, despite having a capacity of 65. Mrs. Brown helped to row and navigate the lifeboat to safety, and her bravery and determination have become legendary in the history of the Titanic.

Figure: Hallucinations with Llama 2 (70B, from 2023-11-25)



- Introduction

- "Emergent" abilities
- In-context learning (ICL)
- Hallucinations

- Architectures

- Training

- Fine-tuning and Alignment

- Instruction tuning
- Alignment
- RLHF

- Use

- Prompt engineering

Hallucinations: Example



You

Who was the sole survivor of Titanic?



ChatGPT

There was no sole survivor of the Titanic. The sinking of the Titanic in 1912 resulted in the loss of more than 1,500 lives. While some individuals managed to survive, there was no one person who was the sole survivor of the tragedy. The survivors were scattered among lifeboats and were later rescued by the RMS Carpathia, which arrived at the scene to pick up survivors.

Figure: Hallucinations with GPT3.5 (from 2023-11-25)



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Hallucinations: Causes

- LLMs are not trained for factual correctness
- LLMs are statistical models
- Knowledge changes over time - pretraining takes a lot of time.



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Hallucinations: Causes

- LLMs are not trained for factual correctness
- LLMs are statistical models
- Knowledge changes over time - pretraining takes a lot of time.



You

Who is the prime minister of Sweden?



ChatGPT

As of my last knowledge update in January 2022, Magdalena Andersson was the Finance Minister and the head of the Social Democratic Party in Sweden. However, please note that political situations can change, and it's recommended to verify this information for the most current details, as my data might be outdated.

Figure: Hallucinations with GPT3.5, trained up until january 2022 (from 2023-11-26)

- What can be the cause for this hallucination?



Hallucinations: Mitigations

- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

- Decoding strategies
- Retrieval-augmented generation (RAG)
- Instruction-tuning/alignment
- Quality of training data



UPPSALA
UNIVERSITET

- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Section 2

Architectures



- Introduction

- "Emergent" abilities
- In-context learning (ICL)
- Hallucinations

- Architectures

- Training

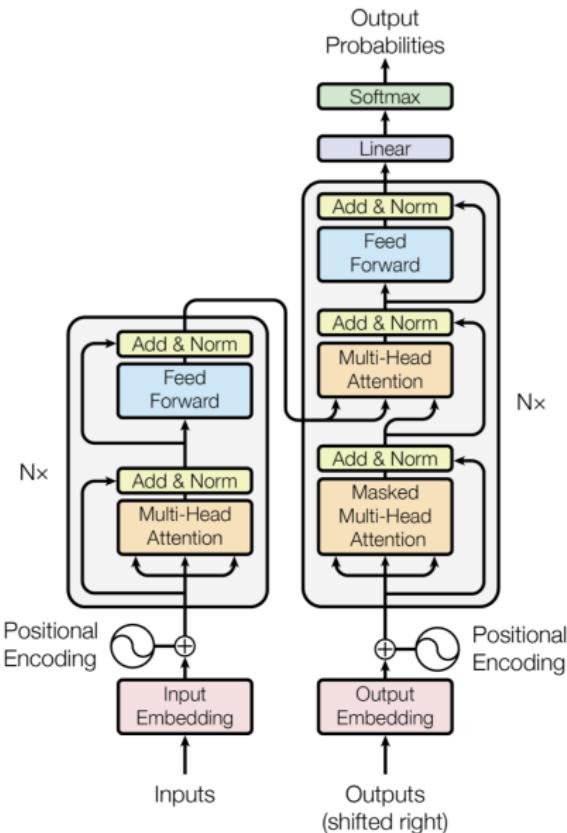
- Fine-tuning and Alignment

- Instruction tuning
- Alignment
- RLHF

- Use

- Prompt engineering

The transformer (Vaswani et al., 2017)





- Introduction

- "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations

- Architectures

- Training

- Fine-tuning and Alignment

- Instruction tuning
 - Alignment
 - RLHF

- Use

- Prompt engineering

- LLMs are usually based on the transformer **decoder**
- "Decoder-only architectures"



- Introduction

- "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations

- Architectures

- Training

- Fine-tuning and Alignment

- Instruction tuning
 - Alignment
 - RLHF

- Use

- Prompt engineering

- LLMs are usually based on the transformer **decoder**
- "Decoder-only architectures"
- Why autoregressive decoders? Why predicting the next word?



Architecture

- Introduction

- "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations

- Architectures

- Training

- Fine-tuning and Alignment

- Instruction tuning
 - Alignment
 - RLHF

- Use

- Prompt engineering

- LLMs are usually based on the transformer **decoder**
- "Decoder-only architectures"
- Why autoregressive decoders? Why predicting the next word?
- Here, **generative pre-trained transformer** (GPT) 2 is a running example.



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Decoder vs. Encoder models

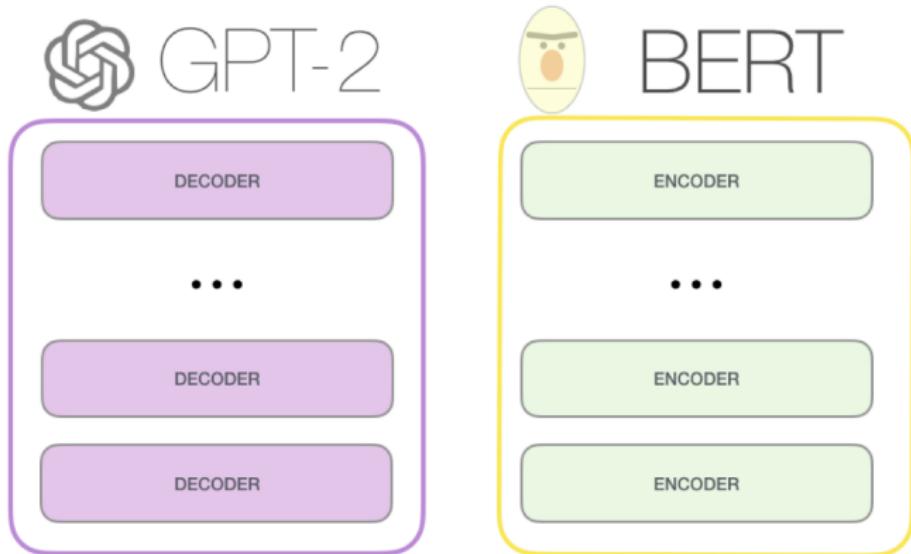


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from [jalalmar.github.io/illustrated-gpt2/](https://jalammar.github.io/illustrated-gpt2/)



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The GPT Architecture



THE TRANSFORMER

DECODER BLOCK

Feed Forward Neural Network

Encoder-Decoder Self-Attention

Masked Self-Attention

Input

<s>	robot	must	obey						512
1	2	3	4	5	6				

Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The GPT Architecture

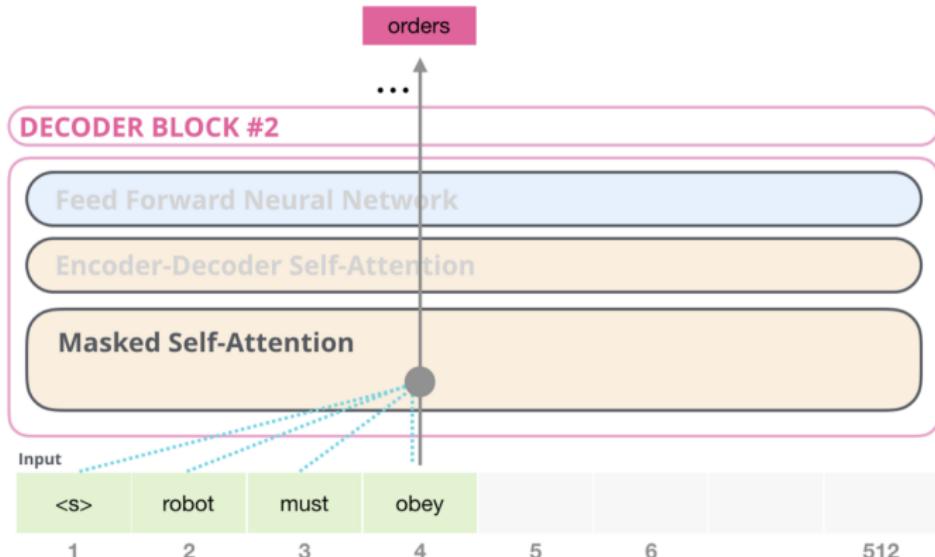


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The GPT Architecture

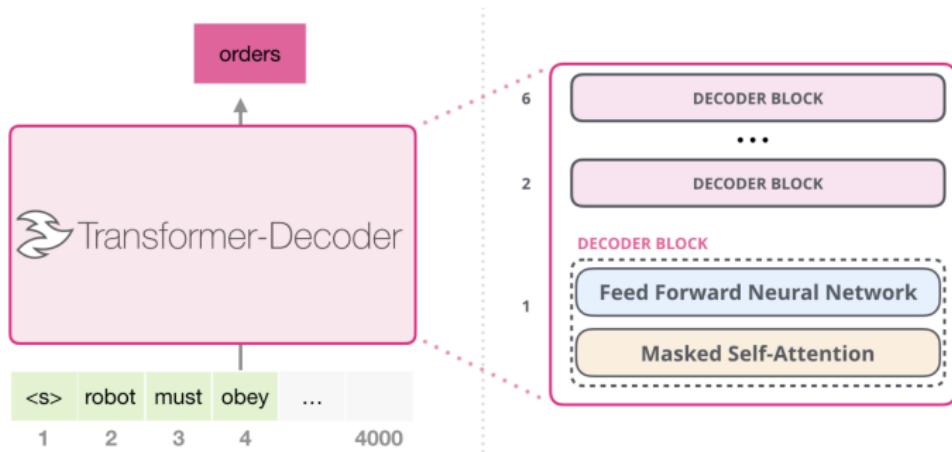


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post].
Retrieved from jalammar.github.io/illustrated-gpt2/



- **Introduction**
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- **Architectures**
- **Training**
- **Fine-tuning and Alignment**
 - Instruction tuning
 - Alignment
 - RLHF
- **Use**
 - Prompt engineering

The GPT Architecture

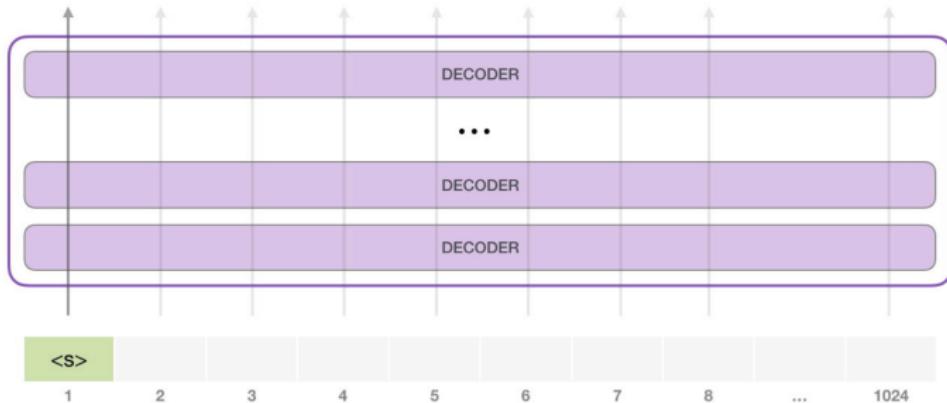


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post].
Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The GPT Architecture

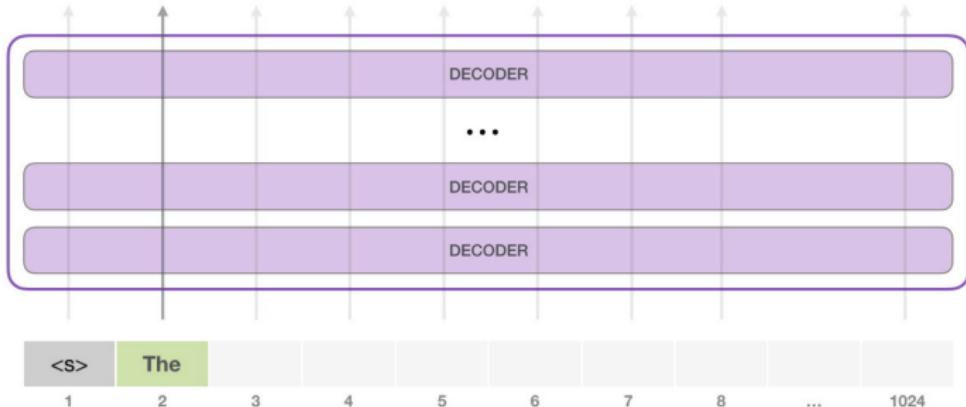


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post].
Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The GPT input and output

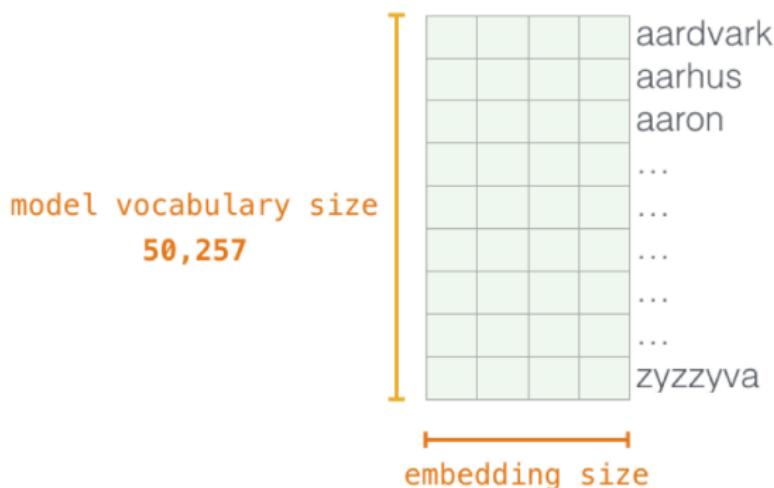


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The GPT input and output



Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The GPT input and output

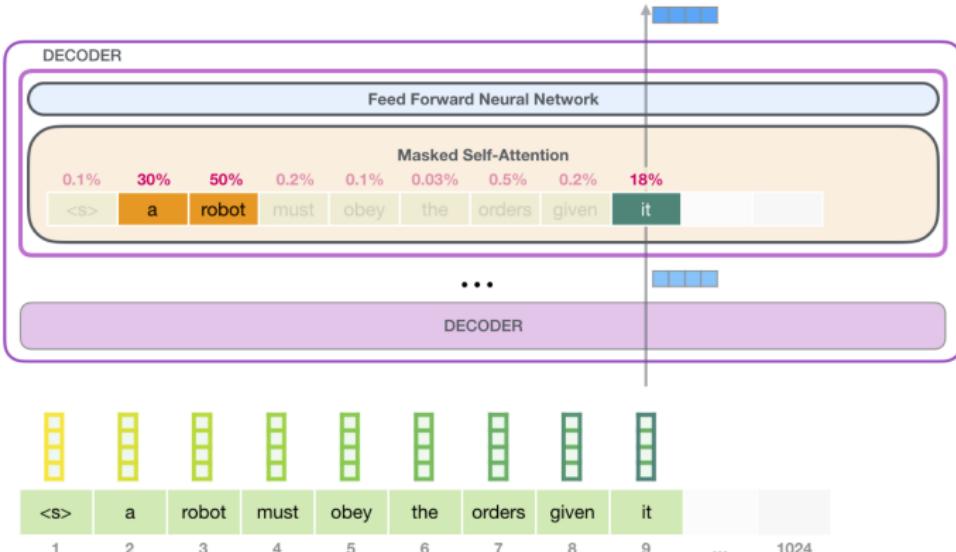


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The GPT input and output

Word	Value vector	Score	Value X Score
<S>		0.001	
a		0.3	
robot		0.5	
must		0.002	
obey		0.001	
the		0.0003	
orders		0.005	
given		0.002	
it		0.19	
Sum:			

Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction

- "Emergent" abilities
- In-context learning (ICL)
- Hallucinations

- Architectures

- Training

- Fine-tuning and Alignment

- Instruction tuning
- Alignment
- RLHF

- Use

- Prompt engineering

The GPT input and output

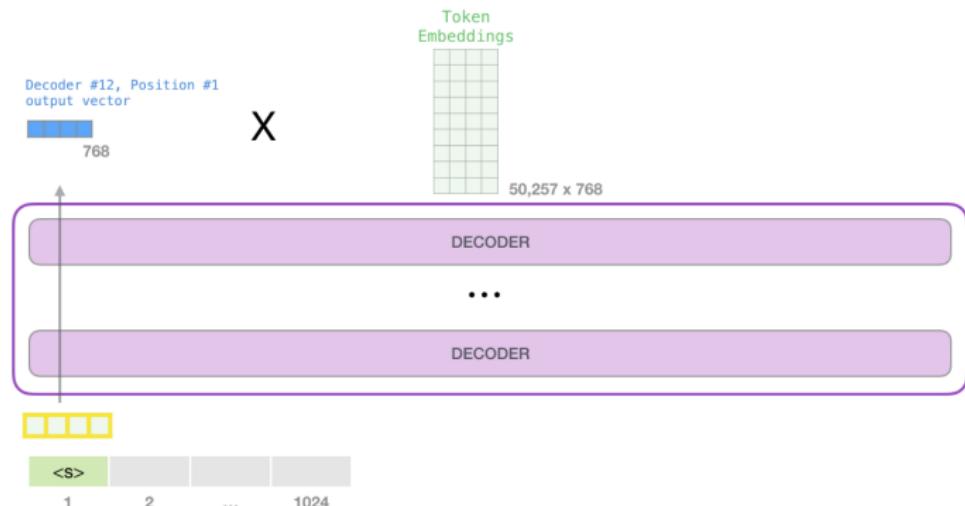


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The (masked) self attention

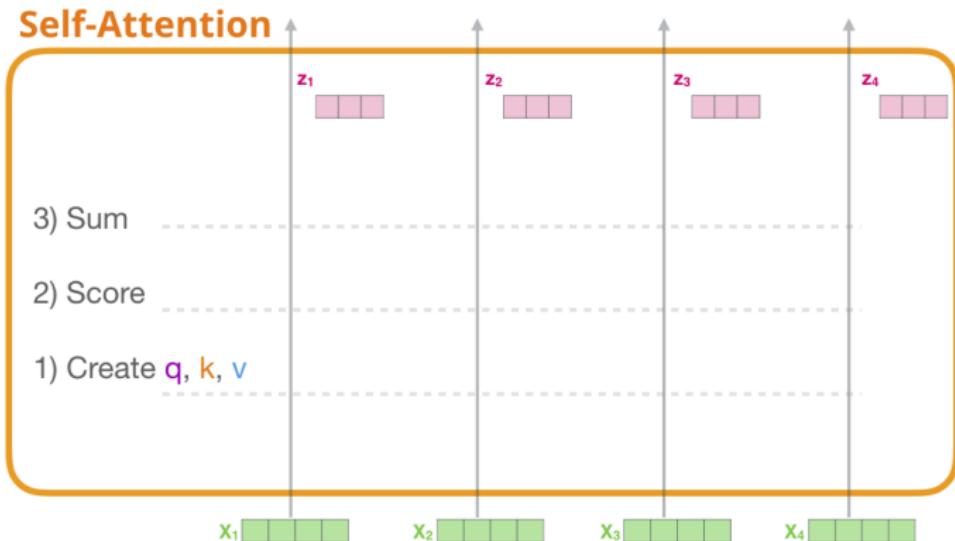


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The (masked) self attention

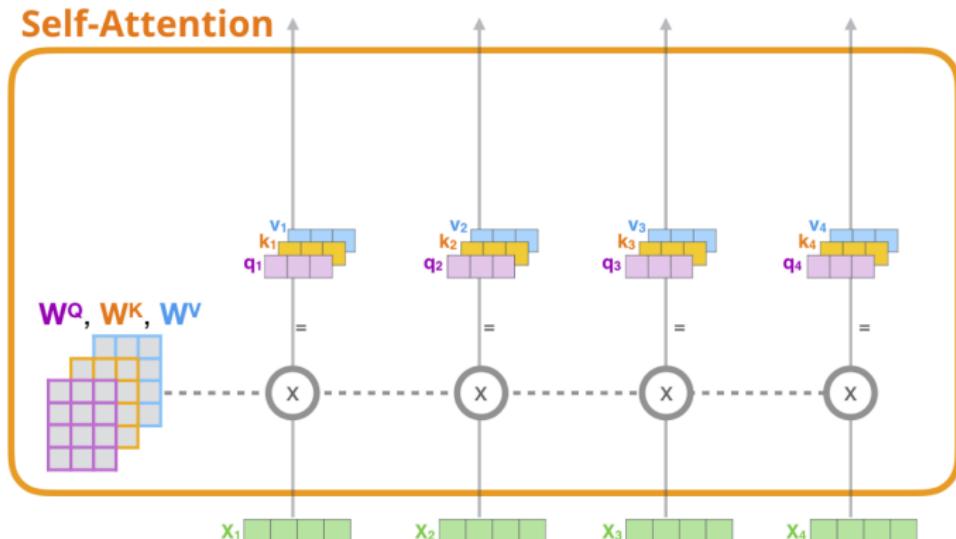


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The (masked) self attention

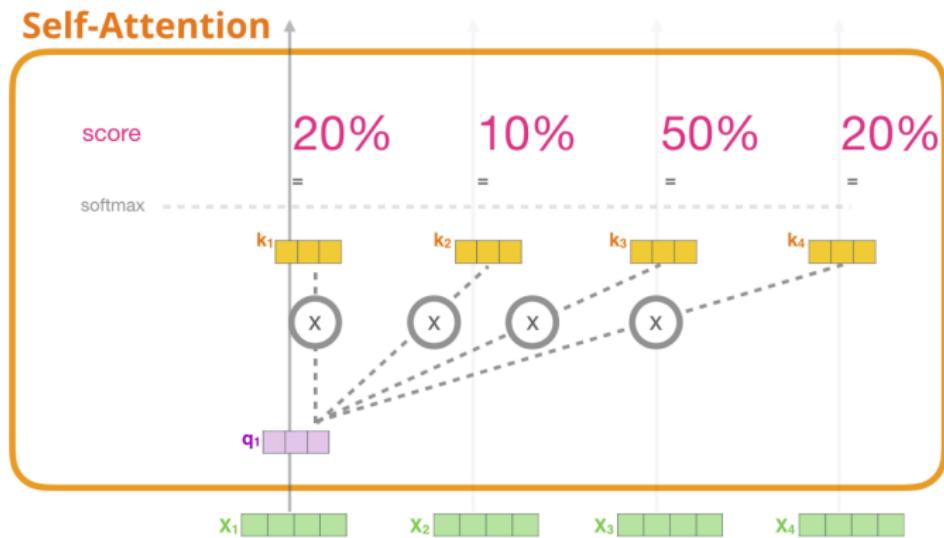


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The (masked) self attention

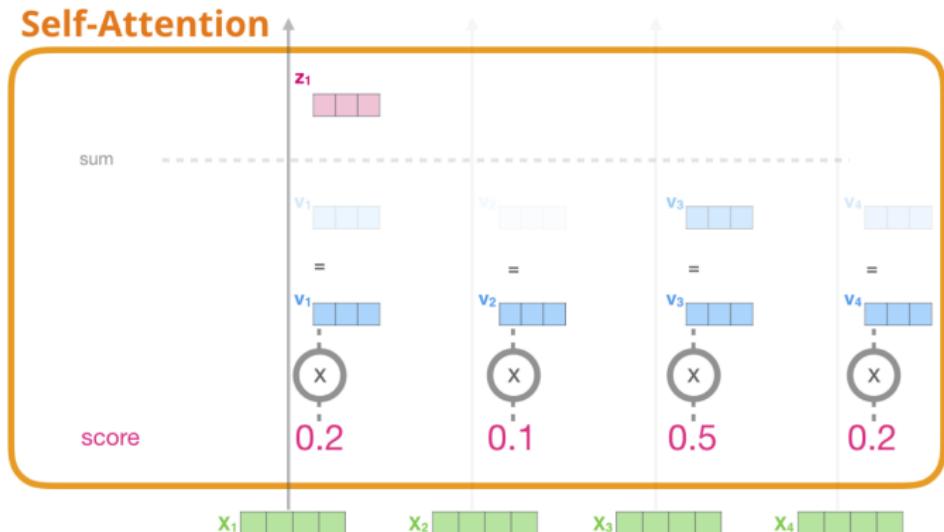


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post].
Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The (masked) self attention

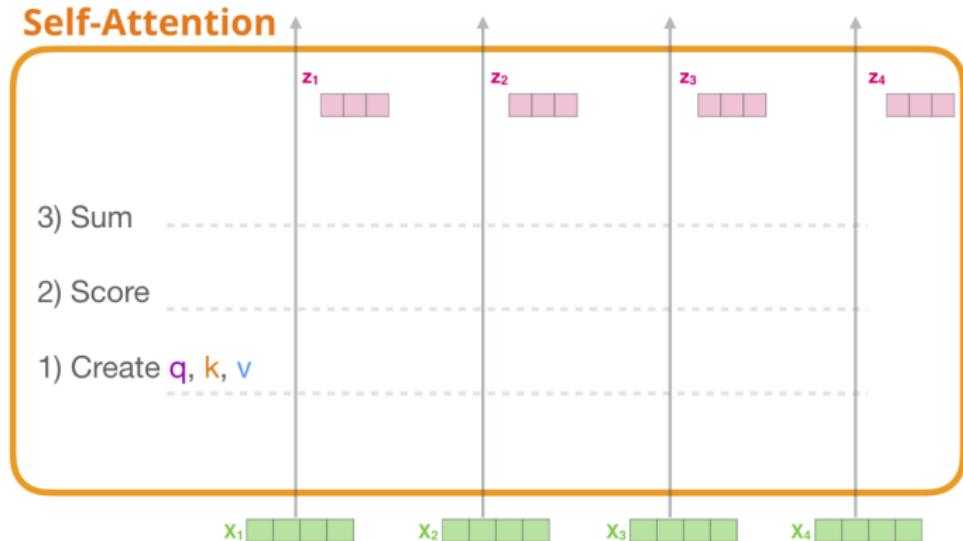


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from [jalalmar.github.io/illustrated-gpt2/](https://jalammar.github.io/illustrated-gpt2/)



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The (masked) self attention

Masked Self-Attention

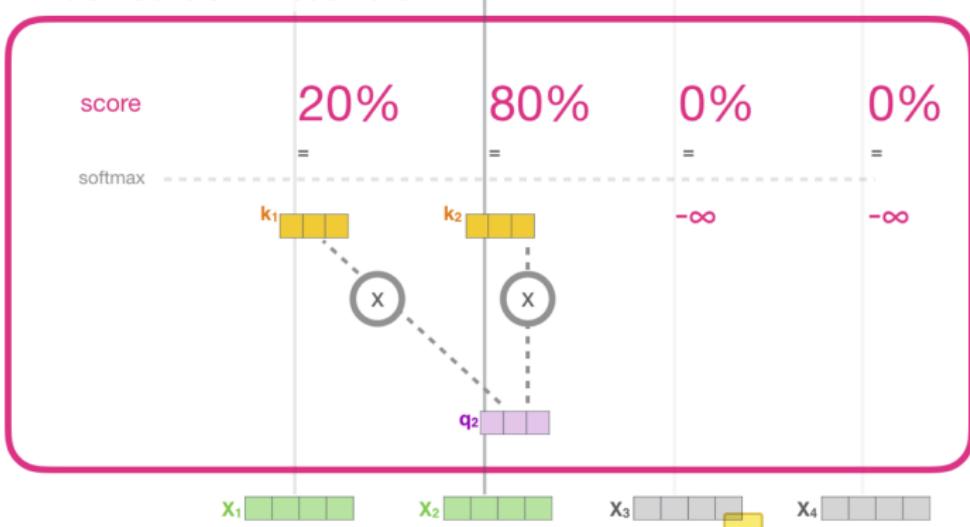


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- **Introduction**
 - “Emergent” abilities
 - In-context learning (ICL)
 - Hallucinations
 - **Architectures**
 - **Training**
 - **Fine-tuning and Alignment**
 - Instruction tuning
 - Alignment
 - RLHF
 - **Use**
 - Prompt engineering

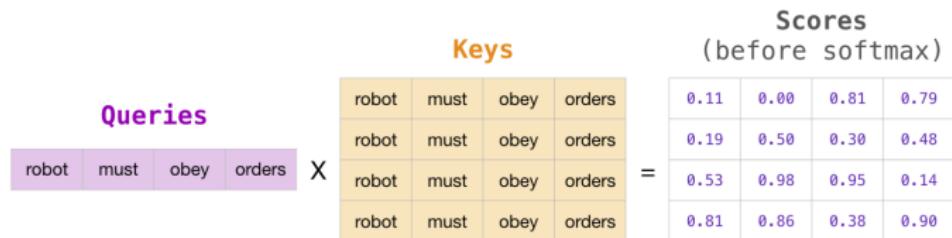


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The (masked) self attention



Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post].
Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The (masked) self attention

Masked Scores (before softmax)				Scores
				Softmax (along rows) →
0.11	-inf	-inf	-inf	1 0 0 0
0.19	0.50	-inf	-inf	0.48 0.52 0 0
0.53	0.98	0.95	-inf	0.31 0.35 0.34 0
0.81	0.86	0.38	0.90	0.25 0.26 0.23 0.26

Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- **Introduction**
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- **Architectures**
- **Training**
- **Fine-tuning and Alignment**
 - Instruction tuning
 - Alignment
 - RLHF
- **Use**
 - Prompt engineering

The GPT output

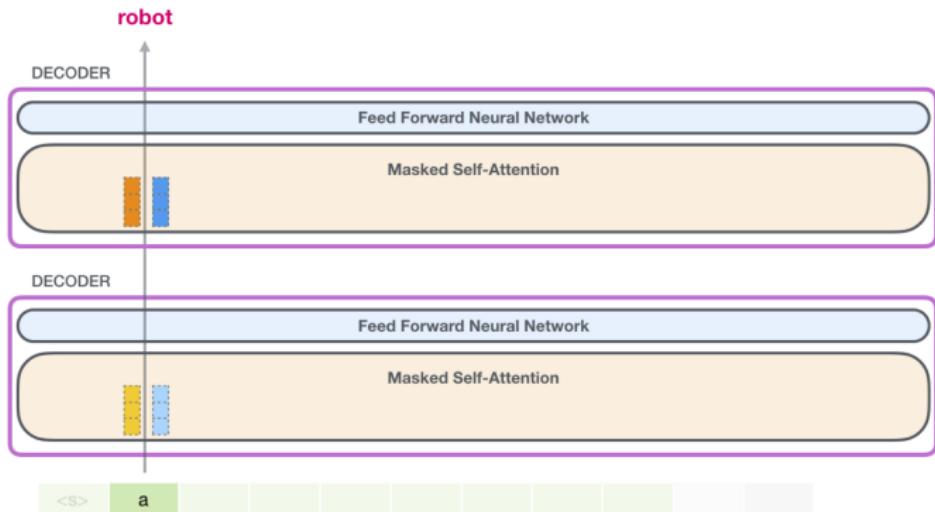


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The GPT output

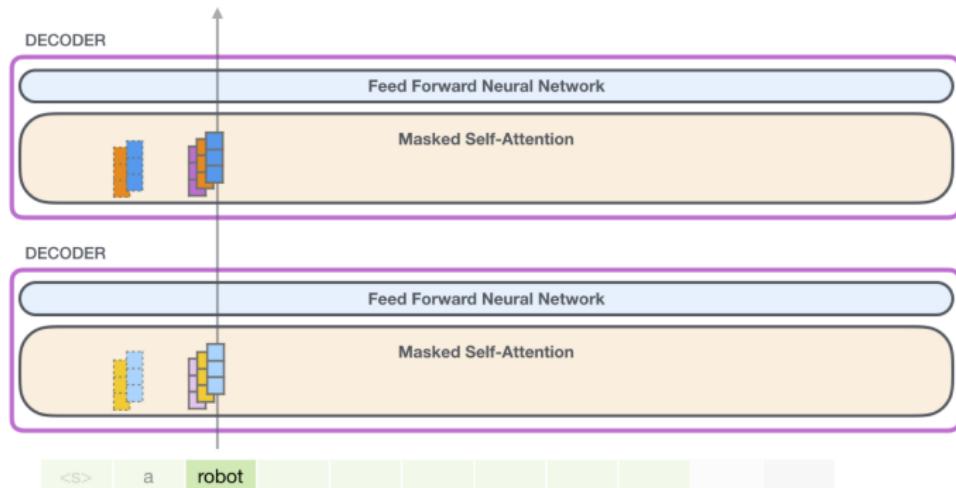


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The GPT output

GPT2 Self-Attention

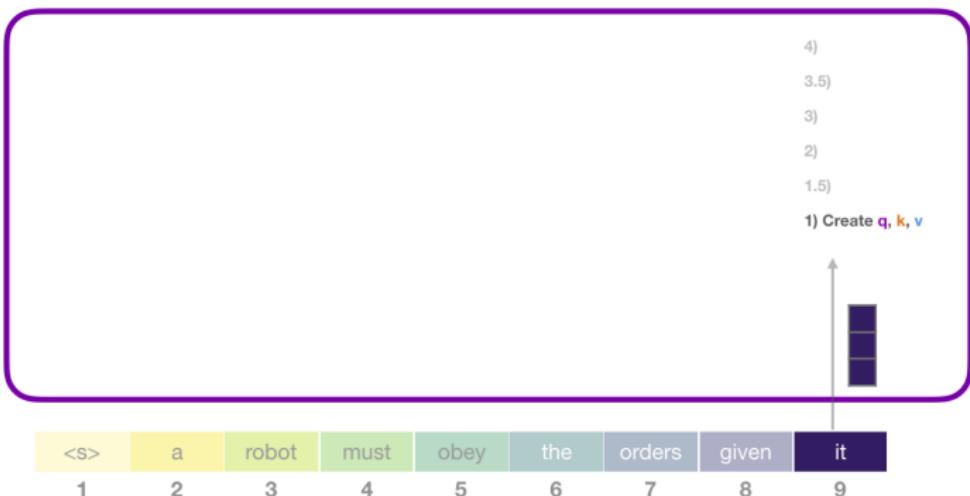


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The GPT output

GPT2 Self-Attention

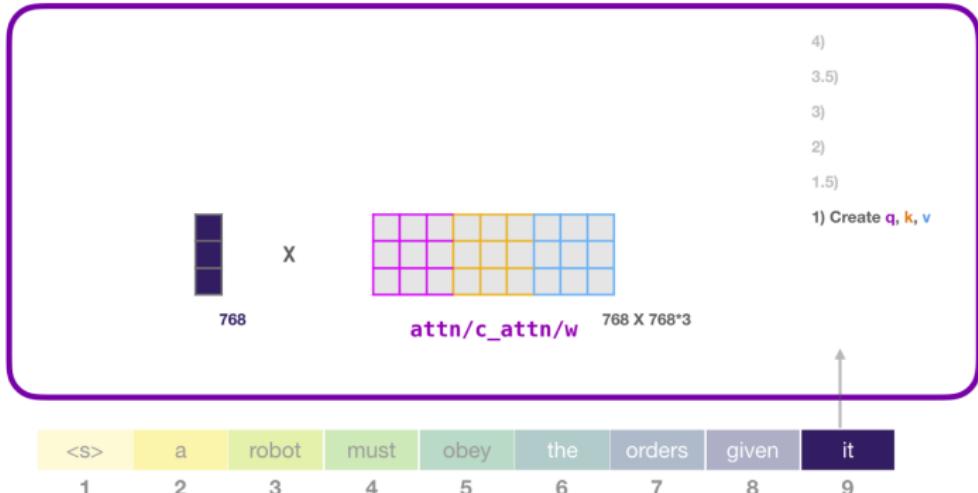


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The GPT output

GPT2 Self-Attention

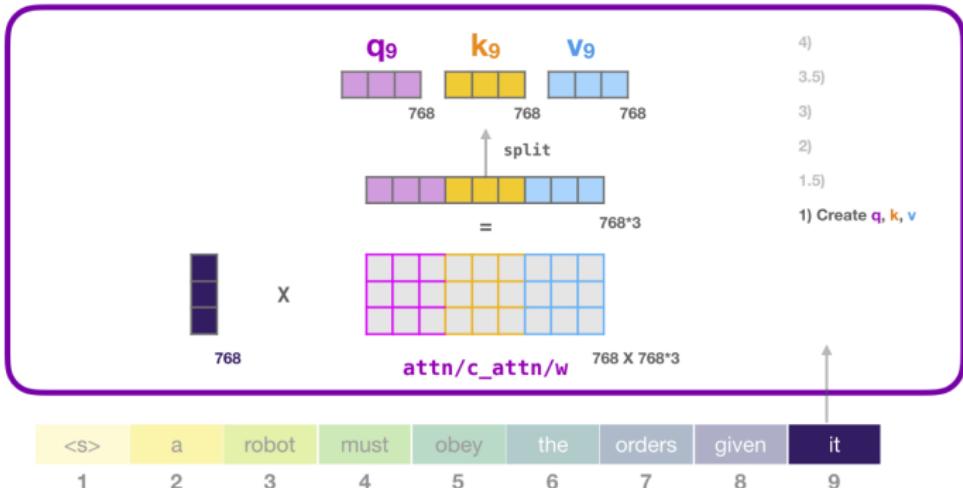


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The GPT output

GPT2 Self-Attention

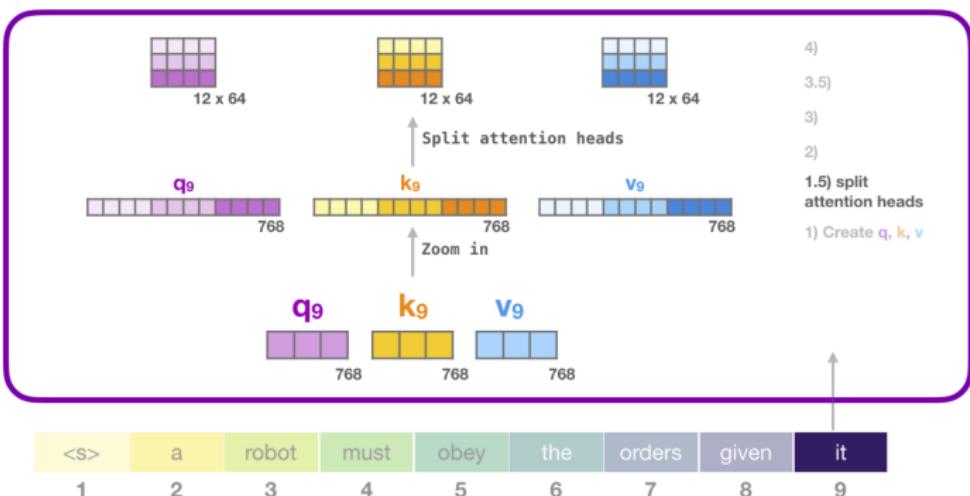


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The GPT output

GPT2 Self-Attention

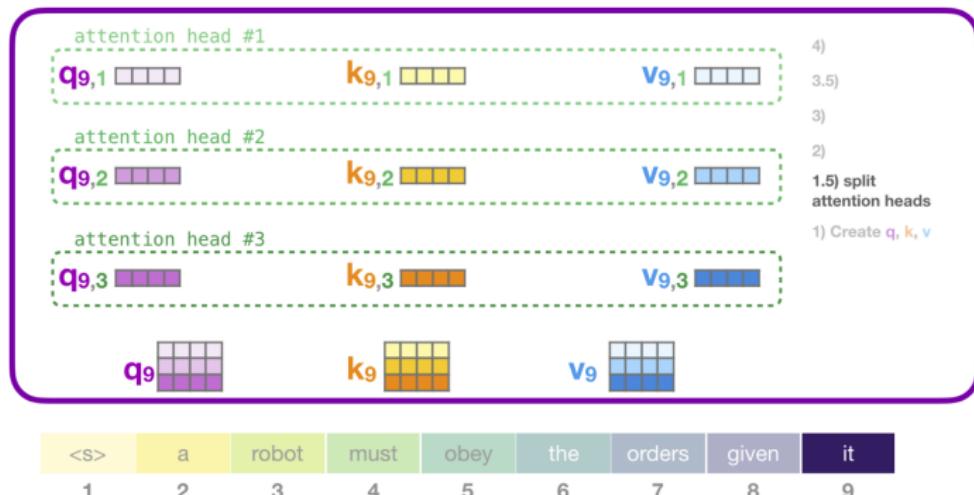


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The GPT output

GPT2 Self-Attention

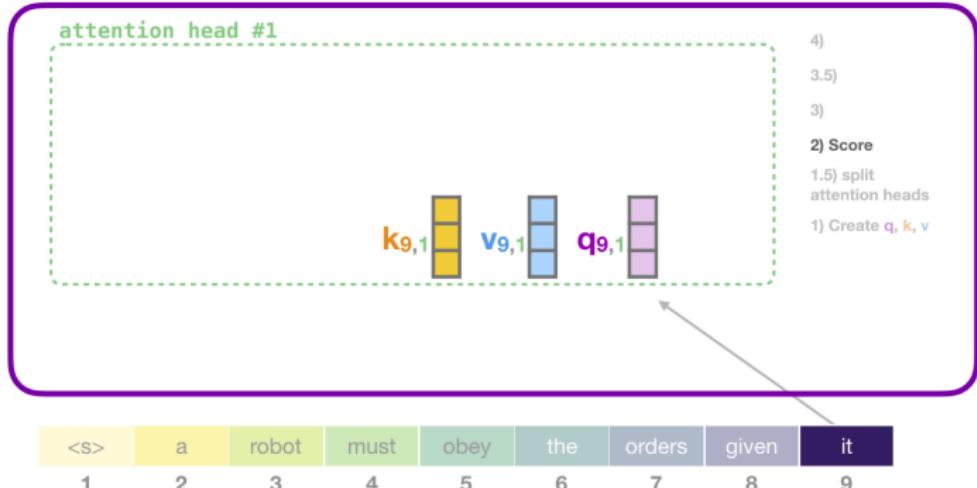


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post].
Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The GPT output

GPT2 Self-Attention

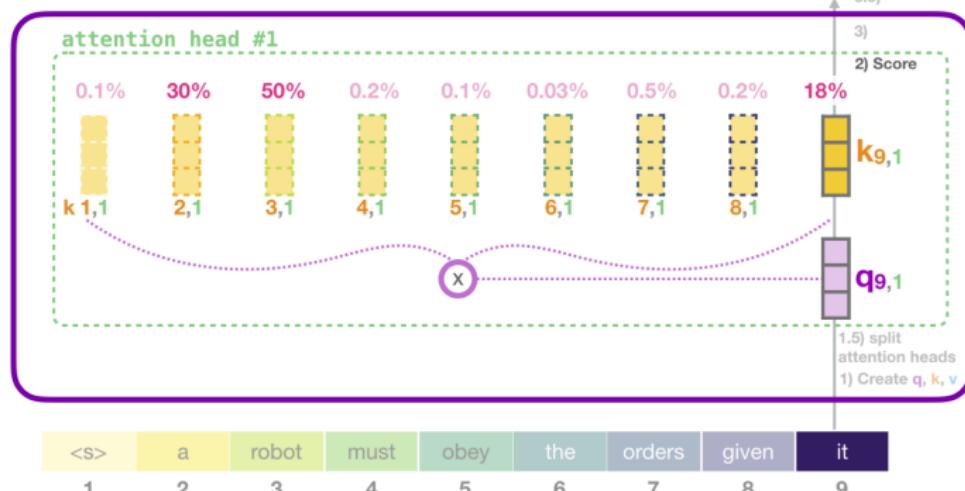


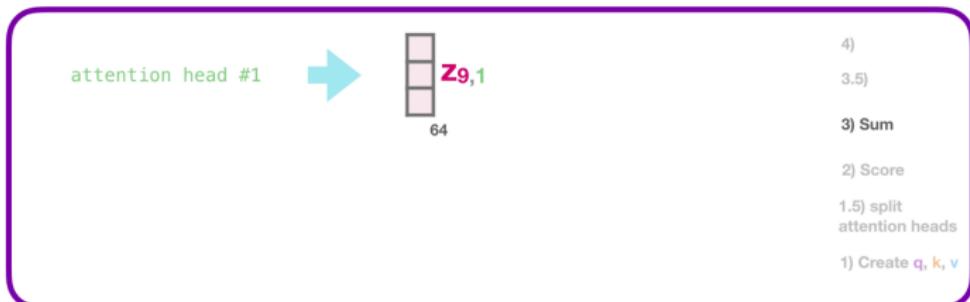
Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- **Introduction**
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- **Architectures**
- **Training**
- **Fine-tuning and Alignment**
 - Instruction tuning
 - Alignment
 - RLHF
- **Use**
 - Prompt engineering

The GPT output

GPT2 Self-Attention



<S>	a	robot	must	obey	the	orders	given	it
1	2	3	4	5	6	7	8	9

Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The GPT output

GPT2 Self-Attention

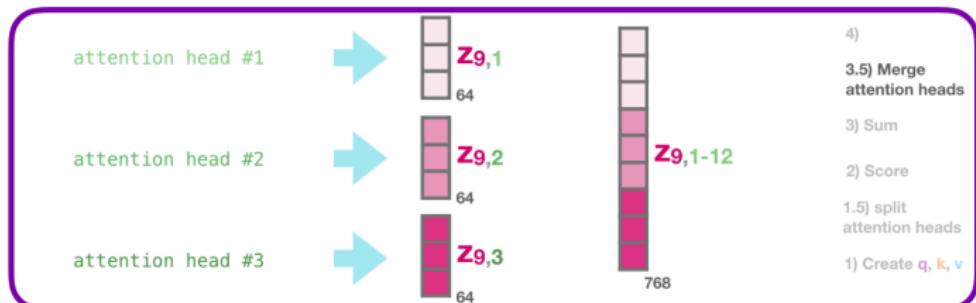


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The GPT output

GPT2 Self-Attention

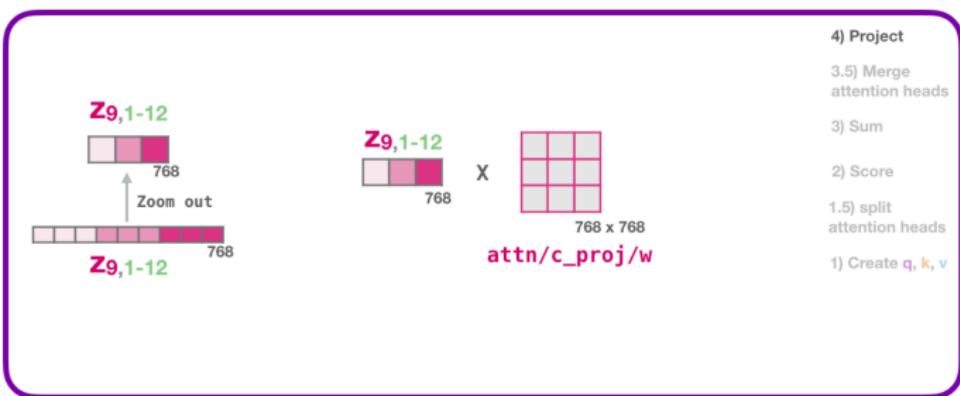


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The GPT output

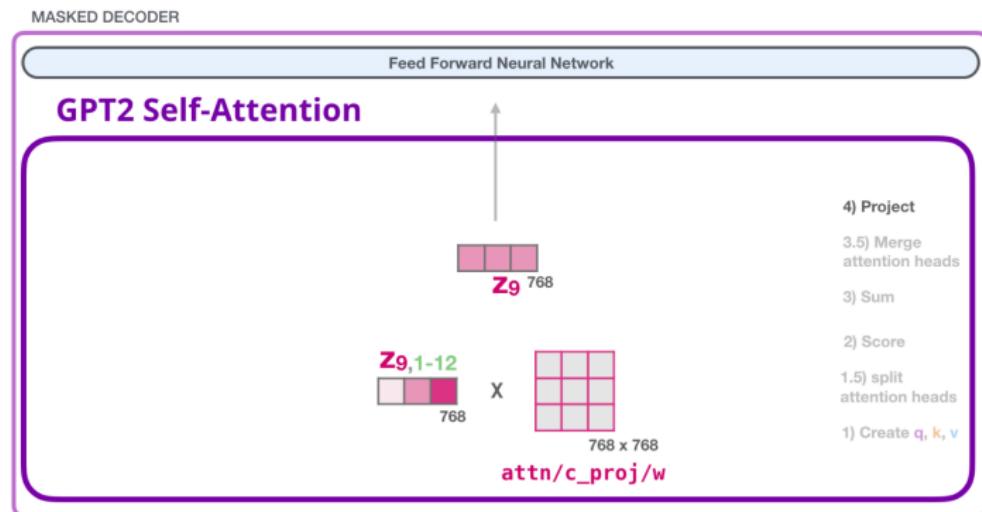


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

The GPT output

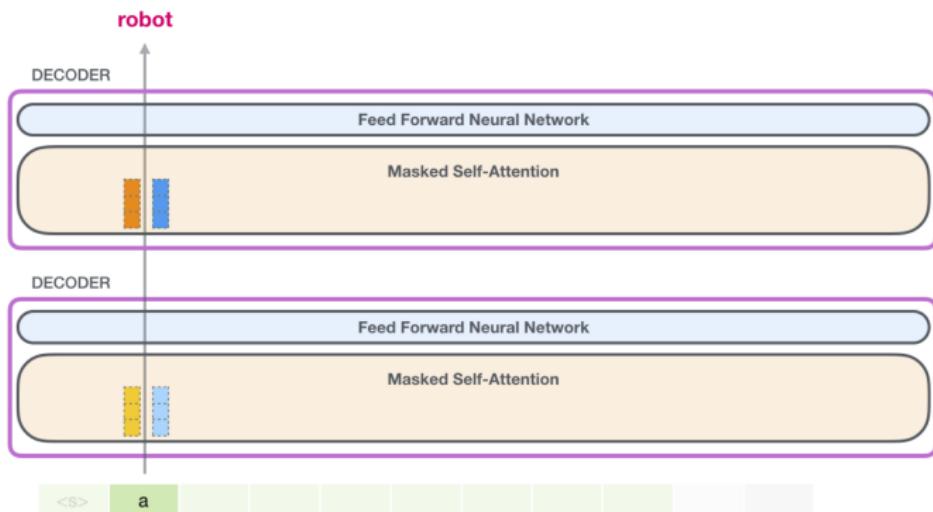


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post].
Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction

- "Emergent" abilities
- In-context learning (ICL)
- Hallucinations

- Architectures

- Training

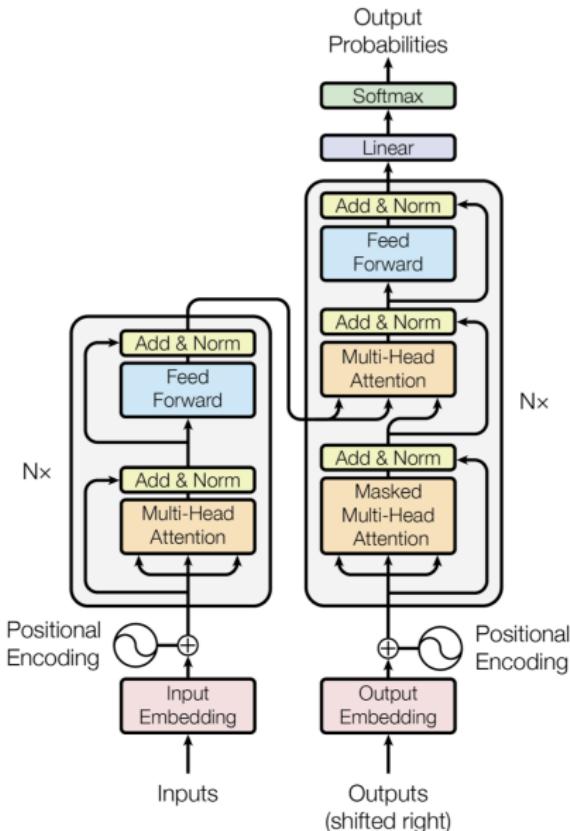
- Fine-tuning and Alignment

- Instruction tuning
- Alignment
- RLHF

- Use

- Prompt engineering

The transformer





- **Introduction**
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- **Architectures**
- **Training**
- **Fine-tuning and Alignment**
 - Instruction tuning
 - Alignment
 - RLHF
- **Use**
 - Prompt engineering

Architecture configurations

Configuration	Method	Equation
Normalization position	Post Norm [22] Pre Norm [26] Sandwich Norm [201]	$\text{Norm}(\mathbf{x} + \text{Sublayer}(\mathbf{x}))$ $\mathbf{x} + \text{Sublayer}(\text{Norm}(\mathbf{x}))$ $\mathbf{x} + \text{Norm}(\text{Sublayer}(\text{Norm}(\mathbf{x})))$
Normalization method	LayerNorm [202] RMSNorm [203] DeepNorm [204]	$\frac{\mathbf{x} - \mu}{\sqrt{\sigma}} \cdot \gamma + \beta, \quad \mu = \frac{1}{d} \sum_{i=1}^d x_i, \quad \sigma = \sqrt{\frac{1}{d} \sum_{i=1}^d (x_i - \mu)^2}$ $\text{RMS}(\mathbf{x}) \cdot \gamma, \quad \text{RMS}(\mathbf{x}) = \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}$ $\text{LayerNorm}(\alpha \cdot \mathbf{x} + \text{Sublayer}(\mathbf{x}))$
Activation function	ReLU [205] GeLU [206] Swish [207] SwiGLU [208] GeGLU [208]	$\text{ReLU}(\mathbf{x}) = \max(\mathbf{x}, \mathbf{0})$ $\text{GeLU}(\mathbf{x}) = 0.5x \otimes [1 + \text{erf}(\mathbf{x}/\sqrt{2})], \quad \text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$ $\text{Swish}(\mathbf{x}) = \mathbf{x} \otimes \text{sigmoid}(\mathbf{x})$ $\text{SwiGLU}(\mathbf{x}_1, \mathbf{x}_2) = \text{Swish}(\mathbf{x}_1) \otimes \mathbf{x}_2$ $\text{GeGLU}(\mathbf{x}_1, \mathbf{x}_2) = \text{GeLU}(\mathbf{x}_1) \otimes \mathbf{x}_2$
Position embedding	Absolute [22] Relative [73] RoPE [209] Alibi [210]	$\mathbf{x}_i = \mathbf{x}_i + \mathbf{p}_i$ $A_{ij} = \mathbf{W}_q \mathbf{x}_i \mathbf{x}_j^T \mathbf{W}_k^T + r_{i-j}$ $A_{ij} = \mathbf{W}_q \mathbf{x}_i \mathbf{R}_{\theta, i-j} \mathbf{x}_j^T \mathbf{W}_k^T$ $A_{ij} = \mathbf{W}_q \mathbf{x}_i \mathbf{R}_{\theta, i-j} \mathbf{x}_j^T \mathbf{W}_k^T \quad A_{ij} = \mathbf{W}_q \mathbf{x}_i \mathbf{x}_j^T \mathbf{W}_k^T - m(i-j)$

Figure: Common configurations (Zhou et al., 2023)

where "sublayer" indicates the previous layer in the network.



● Introduction

- "Emergent" abilities
- In-context learning (ICL)
- Hallucinations

● Architectures

● Training

● Fine-tuning and Alignment

- Instruction tuning
- Alignment
- RLHF

● Use

- Prompt engineering

Common models and their configuration

Model	Category	Size	Normalization	PE	Activation	Bias	#L	#H	d_{model}	MCL
GPT3 [55]	Causal decoder	175B	Pre LayerNorm	Learned	GeLU	✓	96	96	12288	2048
PanGU- α [75]	Causal decoder	207B	Pre LayerNorm	Learned	GeLU	✓	64	128	16384	1024
OPT [81]	Causal decoder	175B	Pre LayerNorm	Learned	ReLU	✓	96	96	12288	2048
PaLM [56]	Causal decoder	540B	Pre LayerNorm	RoPE	SwiGLU	✗	118	48	18432	2048
BLOOM [69]	Causal decoder	176B	Pre LayerNorm	ALiBi	GeLU	✓	70	112	14336	2048
MT-NLG [100]	Causal decoder	530B	-	-	-	-	105	128	20480	2048
Gopher [59]	Causal decoder	280B	Pre RMSNorm	Relative	-	-	80	128	16384	2048
Chinchilla [34]	Causal decoder	70B	Pre RMSNorm	Relative	-	-	80	64	8192	-
Galactica [35]	Causal decoder	120B	Pre LayerNorm	Learned	GeLU	✗	96	80	10240	2048
LaMDA [63]	Causal decoder	137B	-	Relative	GeGLU	-	64	128	8192	-
Jurassic-1 [94]	Causal decoder	178B	Pre LayerNorm	Learned	GeLU	✓	76	96	13824	2048
LLaMA [57]	Causal decoder	65B	Pre RMSNorm	RoPE	SwiGLU	✗	80	64	8192	2048
LLaMA 2 [90]	Causal decoder	70B	Pre RMSNorm	RePE	SwiGLU	✗	80	64	8192	4096
Falcon [127]	Causal decoder	40B	Pre LayerNorm	RoPE	GeLU	✗	60	64	8192	2048
GLM-130B [84]	Prefix decoder	130B	Post DeepNorm	RoPE	GeGLU	✓	70	96	12288	2048
T5 [73]	Encoder-decoder	11B	Pre RMSNorm	Relative	ReLU	✗	24	128	1024	512

Figure: Common configurations (Zhou et al., 2023, Table 3)



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Decoding (word generation)

- LLMs are trained to predict the next word (called decoding), i.e the discrete probability model (pmf)

$$p_{x,i} = P(x_i | \mathbf{x}_{j < i})$$



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Decoding (word generation)

- LLMs are trained to predict the next word (called decoding), i.e the discrete probability model (pmf)

$$p_{x,i} = P(x_i | \mathbf{x}_{j < i})$$

- Two approaches to generate/decode:

greedy $x_i = \arg_x \max P(x | \mathbf{x}_{j < i})$

sampling $x_i \sim P(x | \mathbf{x}_{j < i})$



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Decoding (word generation)

- Temperature sampling:

$$\tilde{P}(x_i | \mathbf{x}_{j < i}) = \frac{\exp(\log(p_{x,i})/t)}{\sum_{i'} \exp(\log(p_{x,i'})/t)}$$

$$x_i \sim \tilde{P}(x_i | \mathbf{x}_{j < i})$$

As $t \rightarrow 0$: greedy, and $t = 1$: ordinary sampling



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
 - Architectures
 - Training
 - Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
 - Use
 - Prompt engineering
- Temperature sampling:
- $$\tilde{P}(x_i | \mathbf{x}_{j < i}) = \frac{\exp(\log(p_{x,i})/t)}{\sum_{i'} \exp(\log(p_{x,i'})/t)}$$
- $$x_i \sim \tilde{P}(x_i | \mathbf{x}_{j < i})$$
- As $t \rightarrow 0$: greedy, and $t = 1$: ordinary sampling
- Top- k sampling: sample proportionally from top k most probable words



UPPSALA UNIVERSITET

- Introduction

- "Emergent" abilities
- In-context learning (ICL)
- Hallucinations

- Architectures

- Training

- Fine-tuning and Alignment

- Instruction tuning
- Alignment
- RLHF

- Use

- Prompt engineering

Section 3

Training



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

(Pre-training) Data

Corpora	Size	Source	Latest Update Time
BookCorpus [138]	5GB	Books	Dec-2015
Gutenberg [139]	-	Books	Dec-2021
C4 [73]	800GB	CommonCrawl	Apr-2019
CC-Stories-R [140]	31GB	CommonCrawl	Sep-2019
CC-NEWS [27]	78GB	CommonCrawl	Feb-2019
REALNEWS [141]	120GB	CommonCrawl	Apr-2019
OpenWebText [142]	38GB	Reddit links	Mar-2023
Pushift.io [143]	2TB	Reddit links	Mar-2023
Wikipedia [144]	21GB	Wikipedia	Mar-2023
BigQuery [145]	-	Codes	Mar-2023
the Pile [146]	800GB	Other	Dec-2020
ROOTS [147]	1.6TB	Other	Jun-2022

Figure: Common datasets (Zhou et al., 2023, Table 2)



UPPSALA UNIVERSITET

● Introduction

- "Emergent" abilities
- In-context learning (ICL)
- Hallucinations

● Architectures

● Training

● Fine-tuning and Alignment

- Instruction tuning
- Alignment
- RLHF

● Use

- Prompt engineering

(Pre-training) Data

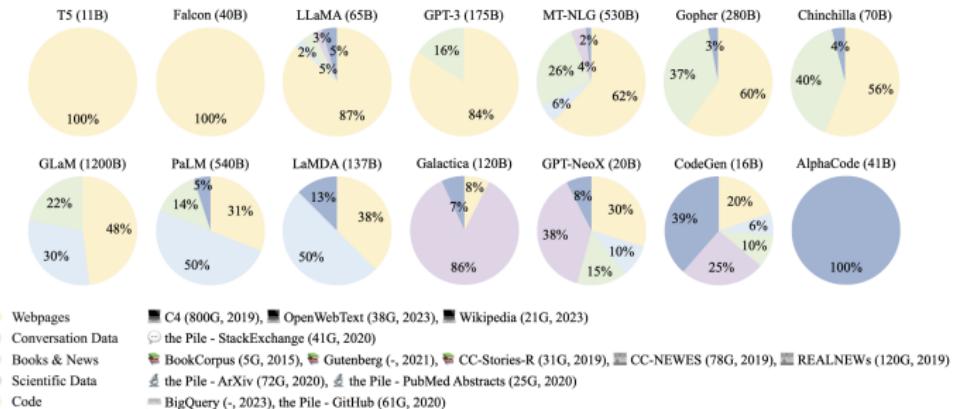


Figure: Common datasets (Zhou et al., 2023, Figure 5)



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Pre-training Data Processing Pipeline

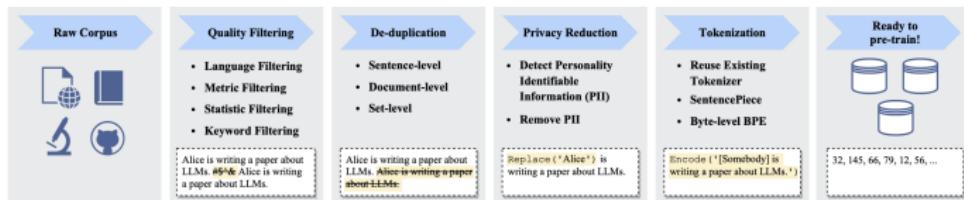


Figure: Pre-training data processing (Zhou et al., 2023, Figure 6)



Tokenization

- Introduction

- "Emergent" abilities
- In-context learning (ICL)
- Hallucinations

- Architectures

- Training

- Fine-tuning and Alignment

- Instruction tuning
- Alignment
- RLHF

- Use

- Prompt engineering

- Simple tokenization: Word based



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Tokenization

- Simple tokenization: Word based
 - Difficult in some languages (e.g. Chinese)



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Tokenization

- Simple tokenization: Word based
 - Difficult in some languages (e.g. Chinese)
 - Generate large vocabularies with many different words



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Tokenization

- Simple tokenization: Word based
 - Difficult in some languages (e.g. Chinese)
 - Generate large vocabularies with many different words
 - Out-of-vocabulary problem



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Tokenization

- Simple tokenization: Word based
 - Difficult in some languages (e.g. Chinese)
 - Generate large vocabularies with many different words
 - Out-of-vocabulary problem
- **Byte-pair encoding (BPE):**
Combine most common characters until predefined vocabulary is reached



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Tokenization

- Simple tokenization: Word based
 - Difficult in some languages (e.g. Chinese)
 - Generate large vocabularies with many different words
 - Out-of-vocabulary problem
- **Byte-pair encoding (BPE):**
Combine most common characters until predefined vocabulary is reached
- **Wordpiece encoding (WE):**
Similar to BPE, but uses a language model to score which to merge



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Tokenization

- Simple tokenization: Word based
 - Difficult in some languages (e.g. Chinese)
 - Generate large vocabularies with many different words
 - Out-of-vocabulary problem
- **Byte-pair encoding (BPE):**
Combine most common characters until predefined vocabulary is reached
- **Wordpiece encoding (WE):**
Similar to BPE, but uses a language model to score which to merge
- **Unigram tokenizer:**
Like WE, but start with a large vocabulary and trim it down.



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Pretraining task/Loss

- The most common task is language modeling(LM), i.e. maximizing

$$L_{LM}(\mathbf{x}) = \sum_i^n \log P(x_i | \mathbf{x}_{<i})$$

- Why is this good/result in good models?



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Pretraining task/Loss

- The most common task is language modeling(LM), i.e. maximizing

$$L_{LM}(\mathbf{x}) = \sum_i^n \log P(x_i | \mathbf{x}_{<i})$$

- Why is this good/result in good models?
 - To better and better predict text we need to increase the **understanding** of the model.
e.g. "Biden and Xi had a meeting on the role of ..." (truth: artificial intelligence)



UPPSALA UNIVERSITET

● Introduction

- "Emergent" abilities
- In-context learning (ICL)
- Hallucinations

● Architectures

● Training

● Fine-tuning and Alignment

- Instruction tuning
- Alignment
- RLHF

● Use

- Prompt engineering

Optimization/Training

Model	Batch Size (#tokens)	Learning Rate	Warmup	Decay Method	Optimizer	Precision Type	Weight Decay	Grad Clip	Dropout
GPT3 (175B)	32K→3.2M	6×10^{-5}	yes	cosine decay to 10%	Adam	FP16	0.1	1.0	-
PanGu- α (200B)	-	2×10^{-5}	-	-	Adam	-	0.1	-	-
OPT (175B)	2M	1.2×10^{-4}	yes	manual decay	AdamW	FP16	0.1	-	0.1
PaLM (540B)	1M→4M	1×10^{-2}	no	inverse square root	Adafactor	BF16	lr^2	1.0	0.1
BLOOM (176B)	4M	6×10^{-5}	yes	cosine decay to 10%	Adam	BF16	0.1	1.0	0.0
MT-NLG (530B)	64 K→3.75M	5×10^{-5}	yes	cosine decay to 10%	Adam	BF16	0.1	1.0	-
Gopher (280B)	3M→6M	4×10^{-5}	yes	cosine decay to 10%	Adam	BF16	-	1.0	-
Chinchilla (70B)	1.5M→3M	1×10^{-4}	yes	cosine decay to 10%	AdamW	BF16	-	-	-
Galactica (120B)	2M	7×10^{-6}	yes	linear decay to 10%	AdamW	-	0.1	1.0	0.1
LaMDA (137B)	256K	-	-	-	-	BF16	-	-	-
Jurassic-1 (178B)	32 K→3.2M	6×10^{-5}	yes	-	-	-	-	-	-
LLaMA (65B)	4M	1.5×10^{-4}	yes	cosine decay to 10%	AdamW	-	0.1	1.0	-
LLaMA 2 (70B)	4M	1.5×10^{-4}	yes	cosine decay to 10%	AdamW	-	0.1	1.0	-
Falcon (40B)	2M	1.85×10^{-4}	yes	cosine decay to 10%	AdamW	BF16	0.1	-	-
GLM (130B)	0.4M→8.25M	8×10^{-5}	yes	cosine decay to 10%	AdamW	FP16	0.1	1.0	0.1
T5 (11B)	64K	1×10^{-2}	no	inverse square root	AdaFactor	-	-	-	0.1
ERNIE 3.0 Titan (260B)	-	1×10^{-4}	-	-	Adam	FP16	0.1	1.0	-
PanGu- Σ (1.085T)	0.5M	2×10^{-5}	yes	-	Adam	FP16	-	-	-

Figure: Optimization during pre-training (Zhao et al, 2023, Table 5)



UPPSALA
UNIVERSITET

- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Section 4

Fine-tuning and Alignment



Instruction tuning/following

- Introduction

- "Emergent" abilities
- In-context learning (ICL)
- Hallucinations

- Architectures

- Training

- Fine-tuning and Alignment

- Instruction tuning
- Alignment
- RLHF

- Use

- Prompt engineering

- We have a very large, pre-trained model
- We want it **perform tasks** (e.g. ICL): **instruction tuning**
- We want it **to be safe**: **alignment**



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - **Instruction tuning**
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Instruction tuning/following

- **Instruction tuning:** fine-tuning pre-trained LLMs on a collection of formatted instances



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - **Instruction tuning**
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Instruction tuning/following

- **Instruction tuning:** fine-tuning pre-trained LLMs on a collection of formatted instances
- Effective approach to adapting existing general LLMs to be domain-specific experts
 - e.g. fine-tune Flan-PaLM using medical datasets to create Med-PaLM



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - **Instruction tuning**
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Instruction tuning/following

- **Instruction tuning:** fine-tuning pre-trained LLMs on a collection of formatted instances
- Effective approach to adapting existing general LLMs to be domain-specific experts
 - e.g. fine-tune Flan-PaLM using medical datasets to create Med-PaLM
- **Effect of instruction tuning:** Performance improvement, a general approach to enhancing the abilities of existing language models



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Instruction tuning/following

- An instruction-formatted instance consists of:
 - a task description (called an instruction),
 - an optional input,
 - the corresponding output,
 - and a small number of demonstrations (optional).



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - [Instruction tuning](#)
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Instruction tuning/following

- An instruction-formatted instance consists of:
 - a task description (called an instruction),
 - an optional input,
 - the corresponding output,
 - and a small number of demonstrations (optional).
- Two kinds of important instruction data,
 - task-formatted instructions
 - chat instructions.



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - [Instruction tuning](#)
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Instruction tuning/following

- An instruction-formatted instance consists of:
 - a task description (called an instruction),
 - an optional input,
 - the corresponding output,
 - and a small number of demonstrations (optional).
- Two kinds of important instruction data,
 - task-formatted instructions
 - chat instructions.
- Key aspects for instruction data:
 - Scaling, more data is better (to a point)
 - Diversity and quality of instructions more important than the number of instances
 - Including things to (1) avoid, (2) reasons, and (3) suggestions may have a negligible or adverse effect



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Instruction tuning training

- Unlike pre-training, instruction tuning is often **more efficient**.
- Usually, **only a moderate number of instances** are used for training.



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - **Instruction tuning**
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Instruction tuning training

- Unlike pre-training, instruction tuning is often **more efficient**.
- Usually, **only a moderate number of instances** are used for training.
- The training objective (i.e., usually sequence-to-sequence loss with teacher forcing)



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
 - Architectures
 - Training
 - Fine-tuning and Alignment
 - **Instruction tuning**
 - Alignment
 - RLHF
 - Use
 - Prompt engineering
-
- Unlike pre-training, instruction tuning is often **more efficient**.
 - Usually, **only a moderate number of instances** are used for training.
 - The training objective (i.e., usually sequence-to-sequence loss with teacher forcing)
 - Smaller batch size and learning rate



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - **Instruction tuning**
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Instruction tuning training

- Unlike pre-training, instruction tuning is often **more efficient**.
- Usually, **only a moderate number of instances** are used for training.
- The training objective (i.e., usually sequence-to-sequence loss with teacher forcing)
- Smaller batch size and learning rate
- Balance the proportion of different tasks during fine-tuning



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Instruction tuning training

- Unlike pre-training, instruction tuning is often **more efficient**.
- Usually, **only a moderate number of instances** are used for training.
- The training objective (i.e., usually sequence-to-sequence loss with teacher forcing)
- Smaller batch size and learning rate
- Balance the proportion of different tasks during fine-tuning
- Pre-training data during instruction tuning as regularization



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - *Instruction tuning*
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Fine-tuning Llama

Models	A800 Full Training			A800 LoRA Training		
	#GPU	BS	Time	#GPU	BS	Time
LLaMA (7B)	2	8	3.0h	1	80	3.5h
LLaMA (13B)	4	8	3.1h	1	48	5.1h
LLaMA (30B)	8	4	6.1h	1	24	14.3h
LLaMA (65B)	16	2	11.2h	1	4	60.6h

Figure: Training time for fine-tuning Llama



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Alignment tuning

- Problems with LLMs:
 - fabricating false information (hallucinations)
 - producing harmful, misleading, and biased output



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Alignment tuning

- Problems with LLMs:
 - fabricating false information (hallucinations)
 - producing harmful, misleading, and biased output
- Alignment of LLMs: Fine-tuning (like instruction tuning) but different criteria:
 - helpfulness
 - honesty
 - harmlessness



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Alignment tuning

- Problems with LLMs:
 - fabricating false information (hallucinations)
 - producing harmful, misleading, and biased output
- Alignment of LLMs: Fine-tuning (like instruction tuning) but different criteria:
 - helpfulness
 - honesty
 - harmlessness
- Alignment might harm the general performance of LLMs : alignment tax
- Red teaming:
 1. probe the model in an adversarial way to generate harmful outputs
 2. updates LLMs to prevent such outputs



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Alignment tuning: Criterias

● Helpfulness

- Demonstrate a clear attempt to assist users in solving their tasks
- Answering questions in a concise and efficient manner



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Alignment tuning: Criterias

● Helpfulness

- Demonstrate a clear attempt to assist users in solving their tasks
- Answering questions in a concise and efficient manner

● Honesty

- Present accurate content to users instead of fabricating information
- Convey appropriate degrees of uncertainty



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Alignment tuning: Criterias

- Helpfulness
 - Demonstrate a clear attempt to assist users in solving their tasks
 - Answering questions in a concise and efficient manner
- Honesty
 - Present accurate content to users instead of fabricating information
 - Convey appropriate degrees of uncertainty
- Harmlessness
 - No offensive or discriminatory language
 - Detect covert endeavors aimed for malicious purposes.



Alignment tuning: Human feedback

- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

- High-quality human feedback is crucial for aligning LLMs



Alignment tuning: Human feedback

- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - **Alignment**
 - RLHF
- Use
 - Prompt engineering

- High-quality human feedback is crucial for aligning LLMs
- **Human Labeler Selection:** Usually native speakers.



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Alignment tuning: Human feedback

- High-quality human feedback is crucial for aligning LLMs
- **Human Labeler Selection:** Usually native speakers.
- Three ways to collect feedback (from humans):
 1. Ranking based (choosing the best out of many suggestions)



Alignment tuning: Human feedback

- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

- High-quality human feedback is crucial for aligning LLMs
- **Human Labeler Selection:** Usually native speakers.
- Three ways to collect feedback (from humans):
 1. Ranking based (choosing the best out of many suggestions)
 2. Answering question about the model output (multiple choice)



Alignment tuning: Human feedback

- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

- High-quality human feedback is crucial for aligning LLMs
- **Human Labeler Selection:** Usually native speakers.
- Three ways to collect feedback (from humans):
 1. Ranking based (choosing the best out of many suggestions)
 2. Answering question about the model output (multiple choice)
 3. Collect data about if the LLM is "breaking the rules"



Reinforcement learning with human feedback

- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
 - Architectures
 - Training
 - Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
 - Use
 - Prompt engineering
- Method to align a model (e.g. in chatGPT)
 - Three components:
 1. Pre-trained LLM
 2. Reward Model (RM) learned from human feedback
 3. Reinforcement learning algorithm aligning the LLM





- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
 - Architectures
 - Training
 - Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
 - Use
 - Prompt engineering
-
- Method to align a model (e.g. in chatGPT)
 - Three components:
 1. Pre-trained LLM
 2. Reward Model (RM) learned from human feedback
 3. Reinforcement learning algorithm aligning the LLM
 - Idea: The reward model (RM) learn human preferences
 - RM: A fine-tuned smaller LM or a LM trained from scratch using human preference data.



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
 - Architectures
 - Training
 - Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
 - Use
 - Prompt engineering
-
- Method to align a model (e.g. in chatGPT)
 - Three components:
 1. Pre-trained LLM
 2. Reward Model (RM) learned from human feedback
 3. Reinforcement learning algorithm aligning the LLM
 - Idea: The reward model (RM) learn human preferences
 - RM: A fine-tuned smaller LM or a LM trained from scratch using human preference data.
 - The reinforcement learning algorithm:
 - the (LLM) agent will perform an action by generate text
 - the (RM) will give the agent a reward signal



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

RLHF: Three steps

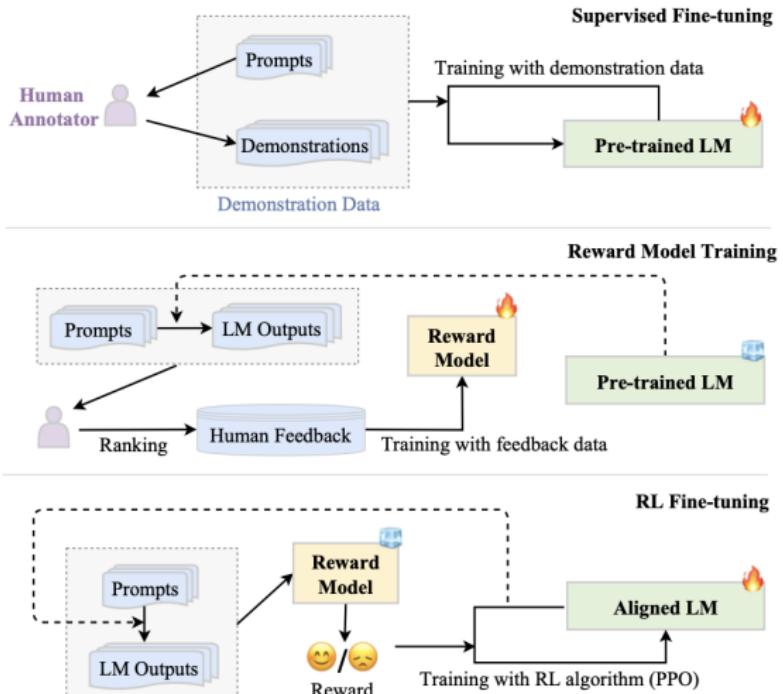


Figure: RLHF (Zhao, 2023, Figure 10)



Values of LLMs

- Introduction

- "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations

- Architectures

- Training

- Fine-tuning and Alignment

- Instruction tuning
 - Alignment
 - RLHF

- Use

- Prompt engineering

- Aligning will implicitly encode values into LLMs



Values of LLMs

- Introduction

- "Emergent" abilities
- In-context learning (ICL)
- Hallucinations

- Architectures

- Training

- Fine-tuning and Alignment

- Instruction tuning
- Alignment
- RLHF

- Use

- Prompt engineering

- Aligning will **implicitly encode values** into LLMs
- What are the values of an LLM?



Values of LLMs

- Introduction

- "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations

- Architectures

- Training

- Fine-tuning and Alignment

- Instruction tuning
 - Alignment
 - RLHF

- Use

- Prompt engineering

- Aligning will **implicitly encode values** into LLMs
- What are the values of an LLM?
- Atari et al (2023) asked ChatGPT World Value Survey questions:



- **Introduction**
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- **Architectures**
- **Training**
- **Fine-tuning and Alignment**
 - Instruction tuning
 - Alignment
 - RLHF
- **Use**
 - Prompt engineering

Values of LLMs

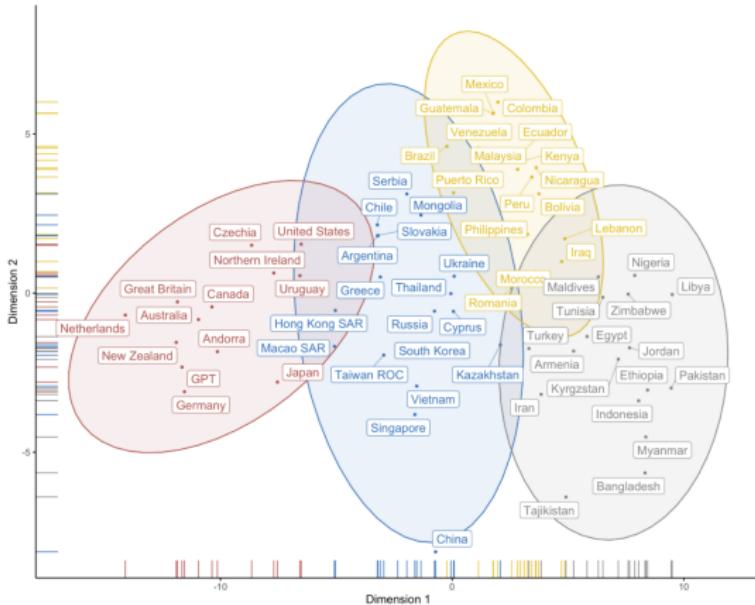


Figure: Value dimensions (Atari et al., 2023, Figure 2)



UPPSALA UNIVERSITET

- Introduction

- "Emergent" abilities
- In-context learning (ICL)
- Hallucinations

- Architectures

- Training

- Fine-tuning and Alignment

- Instruction tuning
- Alignment
- RLHF

- Use

- Prompt engineering

Section 5

Use



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Usage of LLM

- Many areas of applications
- Probably just in the starting phase



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Usage of LLM

- Many areas of applications
- Probably just in the starting phase
- Example applications:
 - ChatBots/Q&A in domains (Medical, Law etc)
 - Virtual Assistants
 - Text summarization
 - Code Generation
 - Information retrieval



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
 - Architectures
 - Training
 - Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
 - Use
 - Prompt engineering
-
- Many areas of applications
 - Probably just in the starting phase
 - Example applications:
 - ChatBots/Q&A in domains (Medical, Law etc)
 - Virtual Assistants
 - Text summarization
 - Code Generation
 - Information retrieval
 - Still BERT is competitive in many tasks (see Ollion et al, 2023)



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Prompt engineering

- **Prompt engineering:**
Designing prompts to produce good results with
in-context learning
- **In practice:** Moving into the **right position in embedding space.**



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Prompt engineering

- **Prompt engineering:**
Designing prompts to produce good results with
in-context learning
- **In practice:** Moving into the **right position in embedding space.**
- **Four different areas affecting the performance (Zhou et al (2023)):**



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Prompt engineering

- **Prompt engineering:**
Designing prompts to produce good results with
in-context learning
- **In practice:** Moving into the **right position in embedding space.**
- Four different areas affecting the performance (Zhou et al (2023)):
 1. **Task description:** Describe the task clearly.



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Prompt engineering

- **Prompt engineering:**
Designing prompts to produce good results with
in-context learning
- **In practice:** Moving into the **right position in embedding space.**
- Four different areas affecting the performance (Zhou et al (2023)):
 1. **Task description:** Describe the task clearly.
 2. **Demonstration data:** Describe the example data for the LLM so it can be used.



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Prompt engineering

- **Prompt engineering:**
Designing prompts to produce good results with
in-context learning
- **In practice:** Moving into the **right position in embedding space.**
- Four different areas affecting the performance (Zhou et al (2023)):
 1. **Task description:** Describe the task clearly.
 2. **Demonstration data:** Describe the example data for the LLM so it can be used.
 3. **Contextual information:** Background information on the task



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

Prompt engineering

- **Prompt engineering:**
Designing prompts to produce good results with **in-context learning**
- **In practice:** Moving into the **right position in embedding space**.
- Four different areas affecting the performance (Zhou et al (2023)):
 1. **Task description:** Describe the task clearly.
 2. **Demonstration data:** Describe the example data for the LLM so it can be used.
 3. **Contextual information:** Background information on the task
 4. **Prompt style:**
 - OpenAI suggest to separate examples with hashtags.
 - Most pre-training data is in English, so using English improves performance
 - "Think through this step by step", and unintuitive "Take a deep breath and think it through".



- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
- Architectures
- Training
- Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
- Use
 - Prompt engineering

How to prompt: Design Principles

- Express your goal clearly, e.g. make the prompt self-contained (avoid propositions like they, it etc)



How to prompt: Design Principles

- Introduction
 - "Emergent" abilities
 - In-context learning (ICL)
 - Hallucinations
 - Architectures
 - Training
 - Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
 - Use
 - Prompt engineering
- **Express your goal clearly**, e.g. make the prompt self-contained (avoid propositions like they, it etc)
 - **Decompose into detailed, easy-to-solve sub-tasks**, e.g. "Check if the above solution is correct"
 - **Provide few-shot examples/demonstrations**
 - **Use LLM friendly formats**, e.g. use quotation marks and hashtags to separate examples