



UPPSALA
UNIVERSITET

- Introduction
 - 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 2025



UPPSALA
UNIVERSITET

- **Introduction**
 - 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**
 - 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**
 - 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**



What is a Large Language Model?

- **Introduction**
 - 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**
 - having some ability to achieve **general-purpose language "understanding"**



What is a Large Language Model?

- **Introduction**
 - 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**
 - having some ability to achieve **general-purpose language "understanding"**
 - pre-trained on very large data
- Large Language Models (LLM) are commonly:
 - decoder-type models



What is a Large Language Model?

- **Introduction**
 - 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**
 - having some ability to achieve **general-purpose language "understanding"**
 - 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**

- 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)



Comparing to pre-trained models (such as BERT)

- **Introduction**

- 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)
 - possible to use for in-context learning



Comparing to pre-trained models (such as BERT)

- **Introduction**
 - 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)
 - possible to use for in-context learning
 - usually interacted with through prompts



Comparing to pre-trained models (such as BERT)

- **Introduction**
 - 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)
 - possible to use for in-context learning
 - usually interacted with through prompts
 - generate textual responses



- **Introduction**

- In-context learning (ICL)
 - Hallucinations

- **Architectures**

- **Training**

- **Fine-tuning and Alignment**

- Instruction tuning
 - Alignment
 - RLHF

- **Use**

- Prompt engineering

Examples of LLM prompting

Examples:

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

chatGPT



- **Introduction**
 - 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



Why is this working now?

- **Introduction**

- 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)



- **Introduction**
 - 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 does it take to read 1T tokens at 10 tokens/s?



- **Introduction**
 - 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 does it take to read 1T tokens at 10 tokens/s? **(3171 years)**



- **Introduction**
 - 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 does it take to read 1T tokens at 10 tokens/s? **(3171 years)**
- All three are important (Zhou et al., 2023)
Can we connect this back to the standard ML framework?



- **Introduction**
 - 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 does it take to read 1T tokens at 10 tokens/s? **(3171 years)**
 - 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)



- Introduction
 - 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
 - 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 weights - only using existing weights.



- Introduction
 - 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 weights - only using existing weights.
- We demonstrate what to do with a **few examples**



- Introduction
 - 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 weights - only using existing weights.
- We demonstrate what to do with a **few examples**
- The model 'learns' what to do **in context**.



- Introduction
 - 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 weights - only using existing weights.
- We demonstrate what to do with a **few examples**
- The model 'learns' what to do **in context**.
- We "prompt" LLMs for utilization
- Let data be

$$\mathcal{D}_k = \{(x_i, y_i)\}_{i=1}^k,$$

$$\hat{y}_{k+1} \sim p_{\theta}(y \mid I, (x_1, y_1), \dots, (x_k, y_k), x_{k+1}).$$

where I are the general instructions and θ are fixed pre-trained parameters.



- Introduction
 - 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**.



- Introduction
 - 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."
 - Being nice.



- Introduction
 - 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."
 - Being nice.
- The effectiveness of ICL is highly affected by the **design of demonstrations** (Zhou et al., 2023)



Chain of thought (CoT) prompting

- Introduction
 - 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"





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

Chain of thought (CoT) prompting

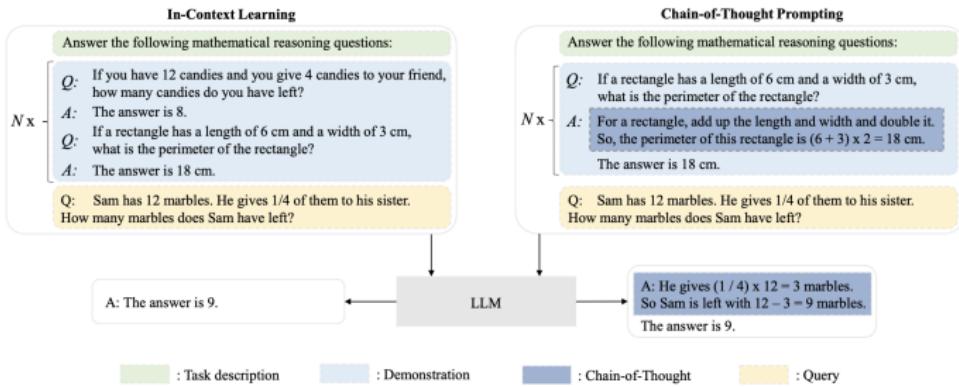


Figure: ICL vs CoT (Figure 12, Zhao et al., 2023)



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

Hallucinations: What is it?

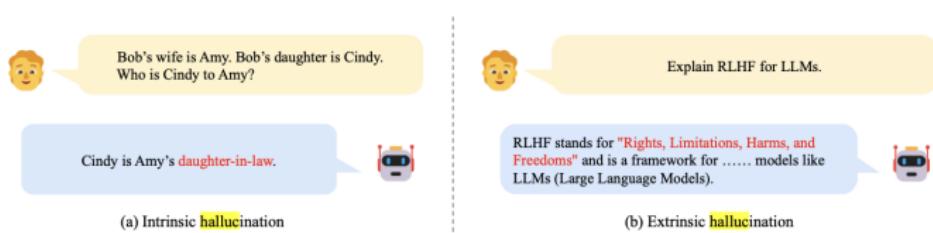


Figure: Hallucinations (Zhao et al., 2023, Figure 14)



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

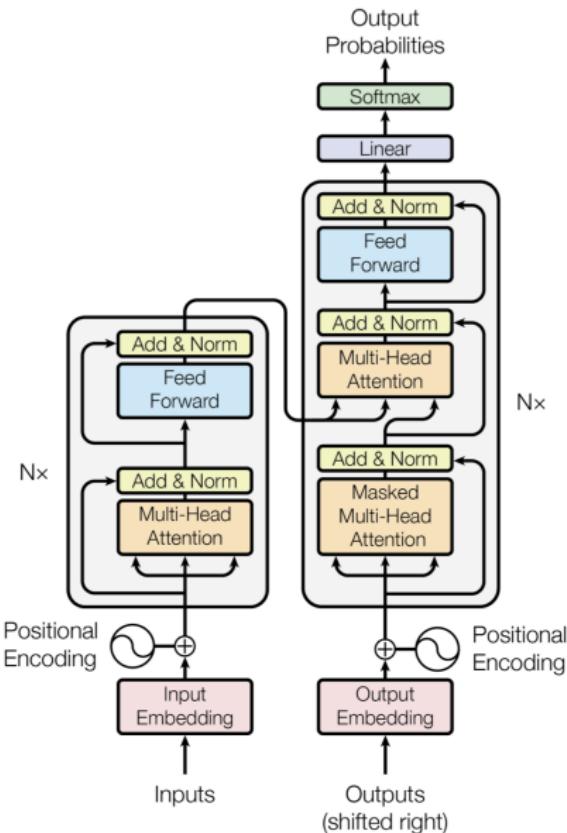
Section 2

Architectures



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

Decoder vs. Encoder models

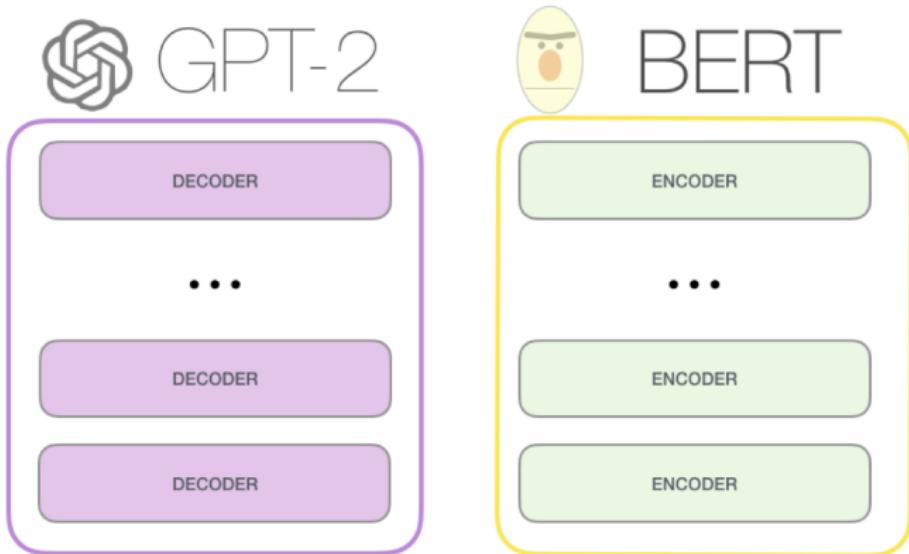


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from [jalalmar.github.io/illustrated-gpt2/](https://jalammar.github.io/illustrated-gpt2/)



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

The GPT Architecture

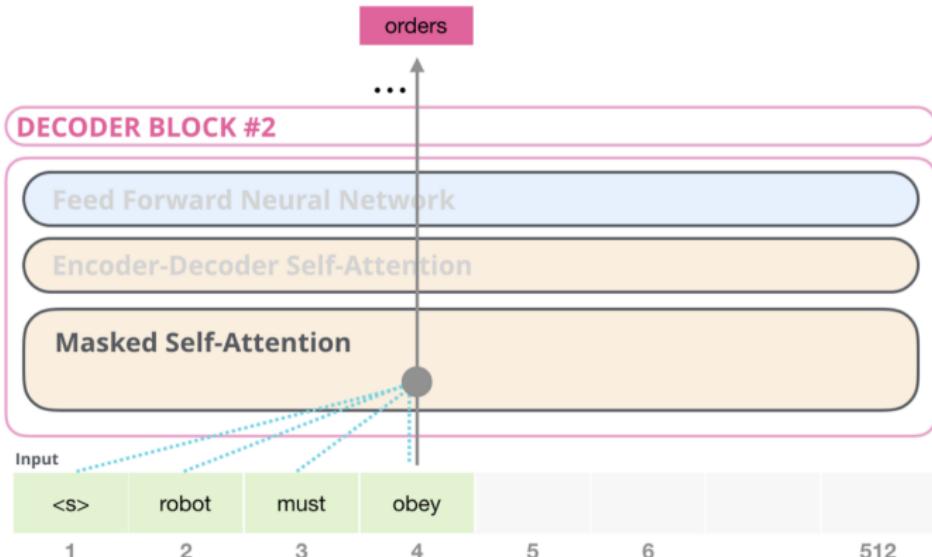


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



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

The GPT Architecture

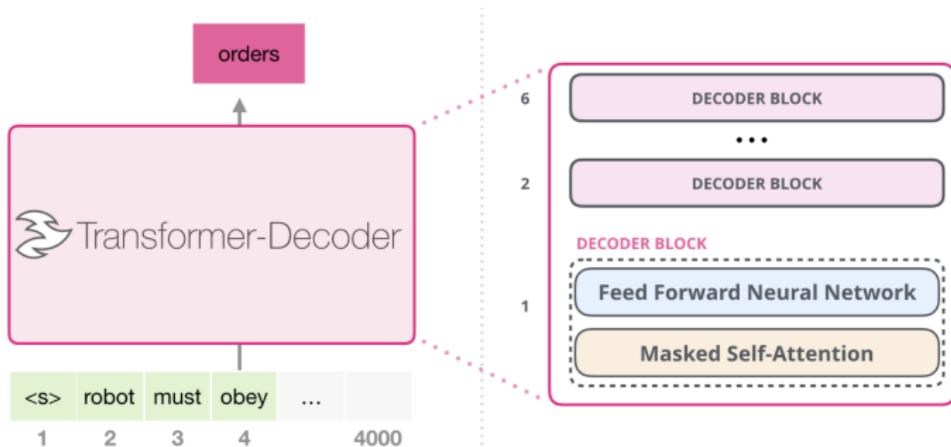


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post].
Retrieved from jalammar.github.io/illustrated-gpt2/



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

The GPT Architecture

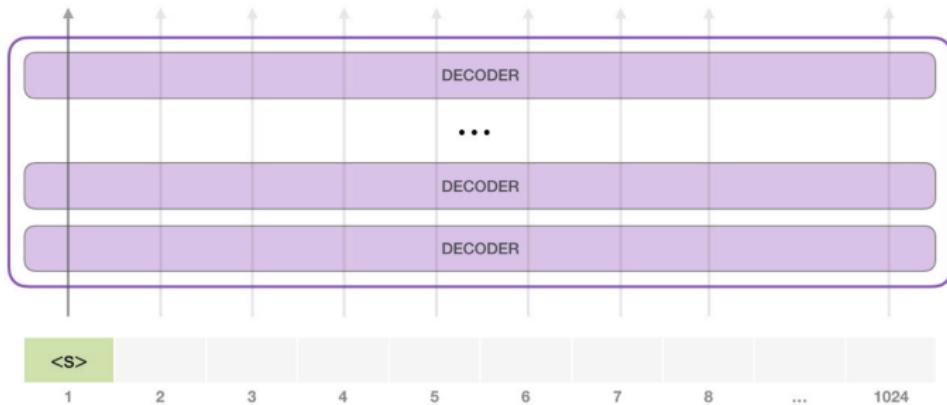


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post].
Retrieved from jalammar.github.io/illustrated-gpt2/



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

The GPT Architecture

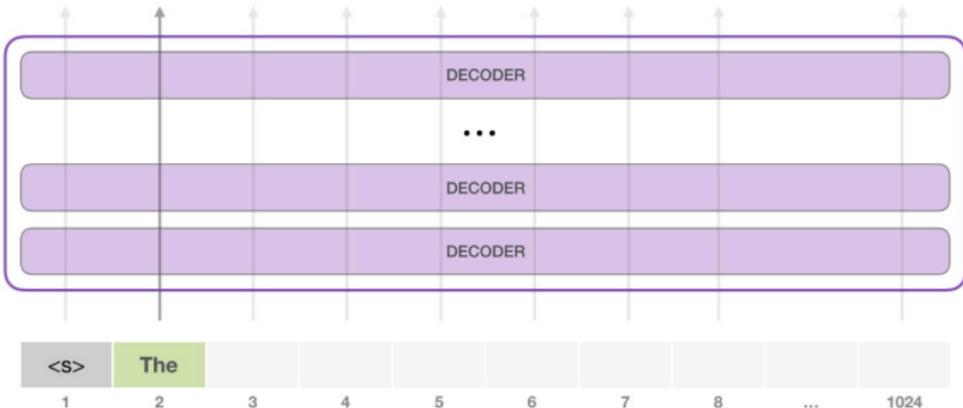


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post].
Retrieved from jalammar.github.io/illustrated-gpt2/



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

The GPT input and output

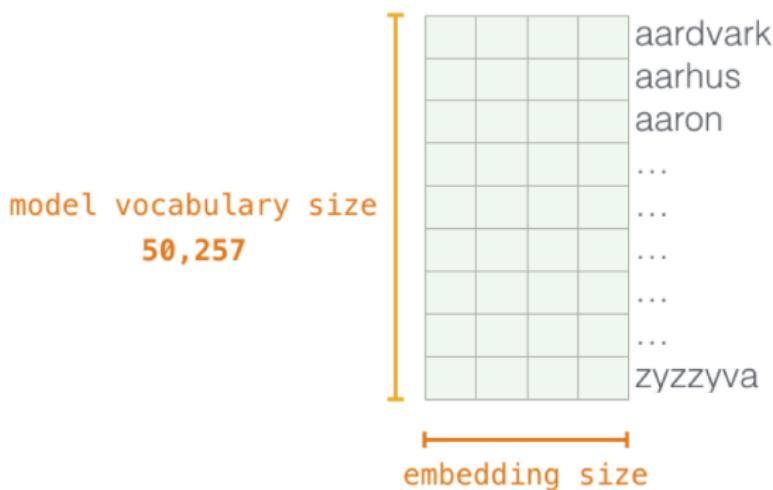


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



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

The GPT input and output

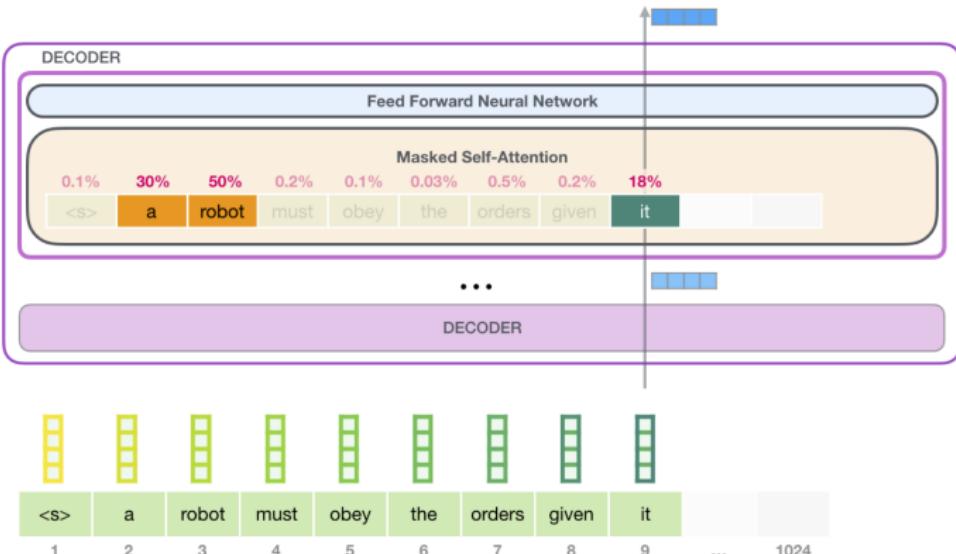


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



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

The GPT input and output

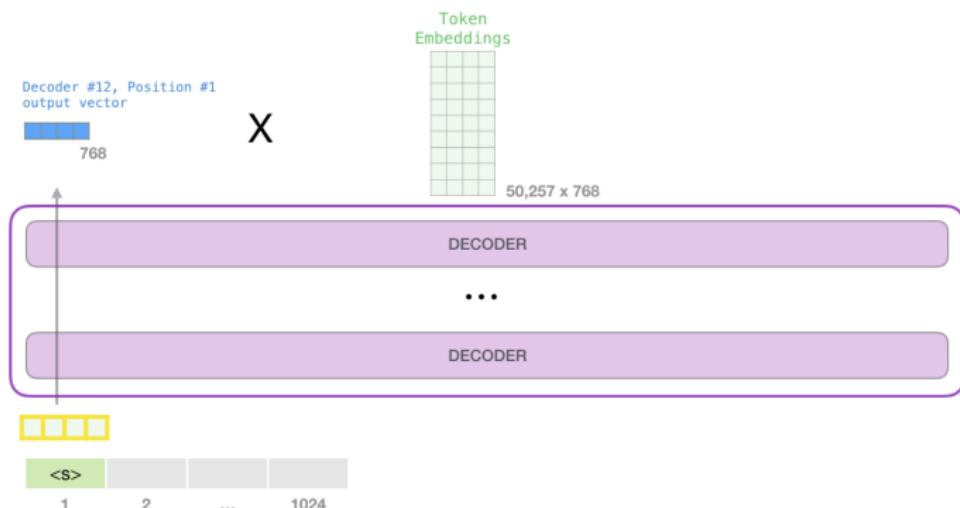


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



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

The (masked) self attention

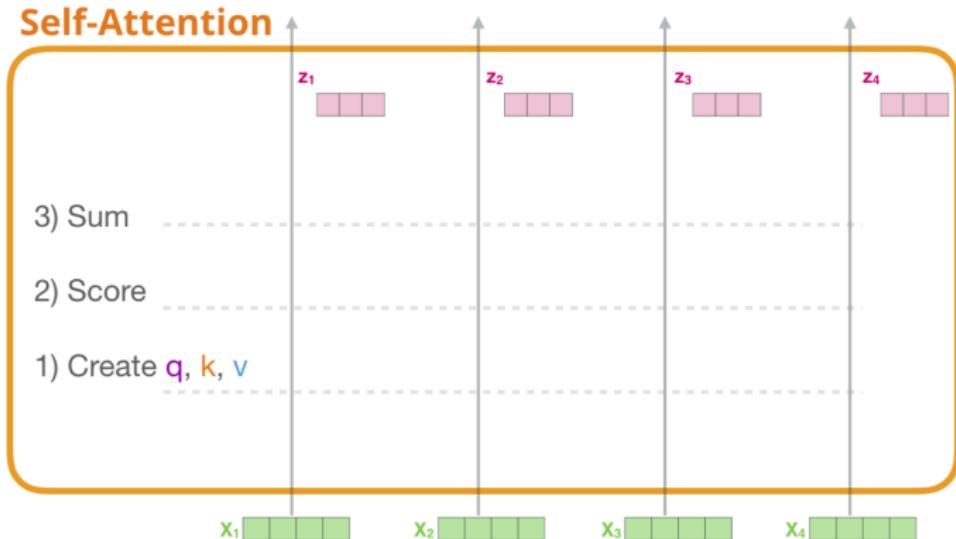


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



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

The (masked) self attention

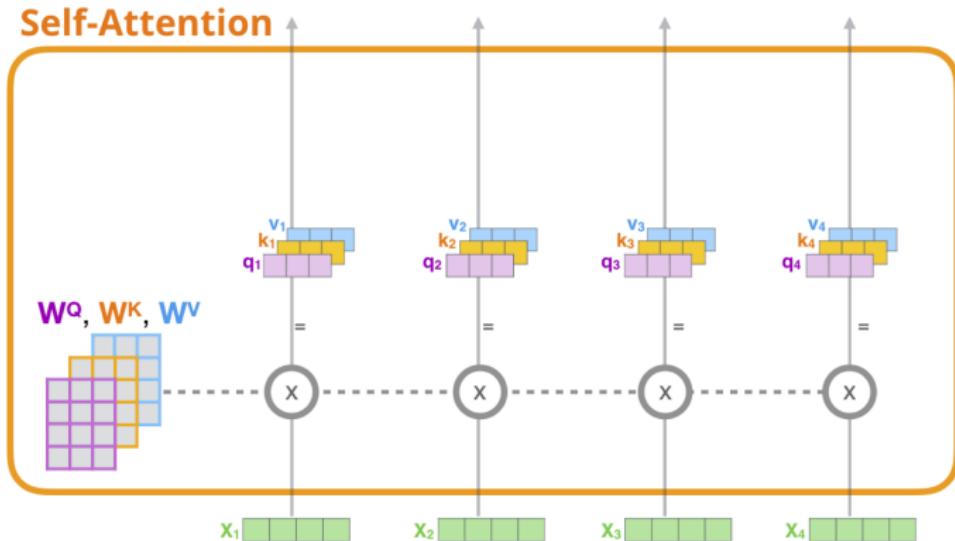


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



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

The (masked) self attention

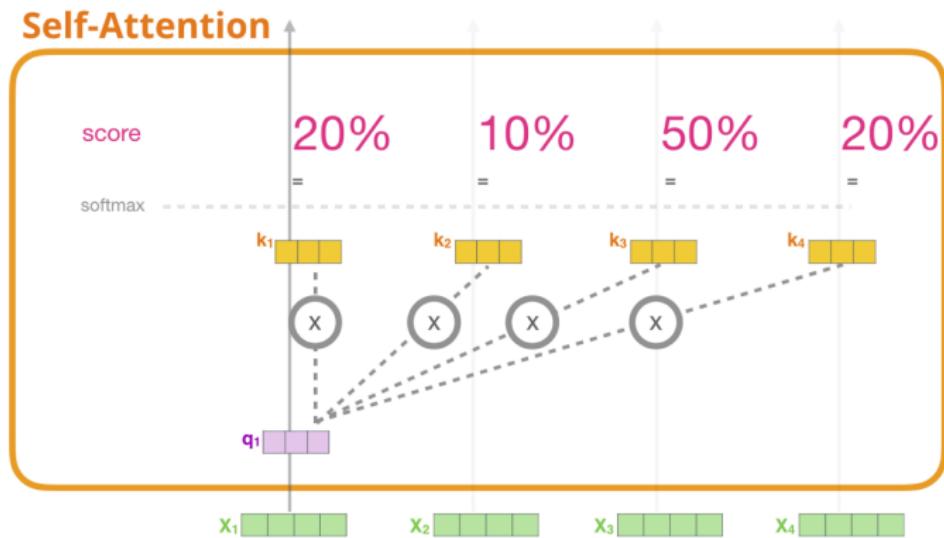


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



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

The (masked) self attention

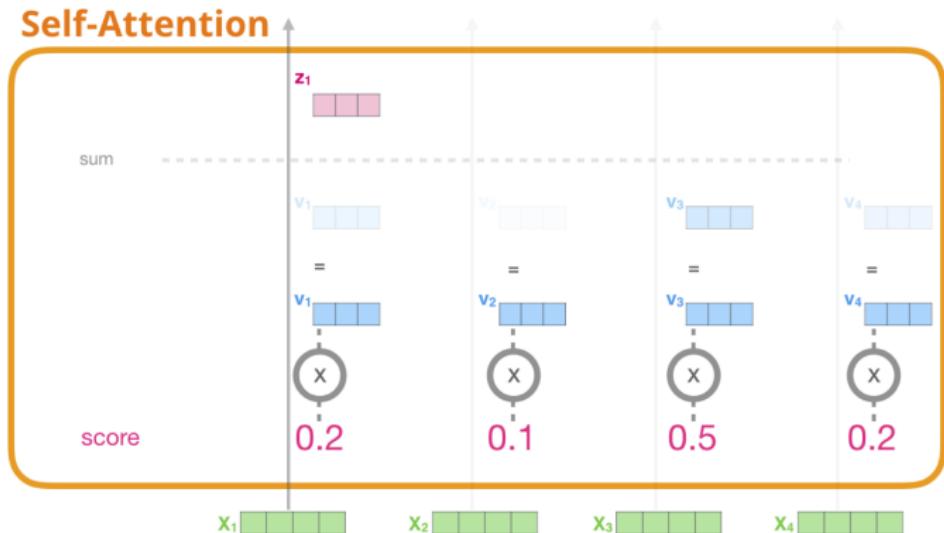


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



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

The (masked) self attention

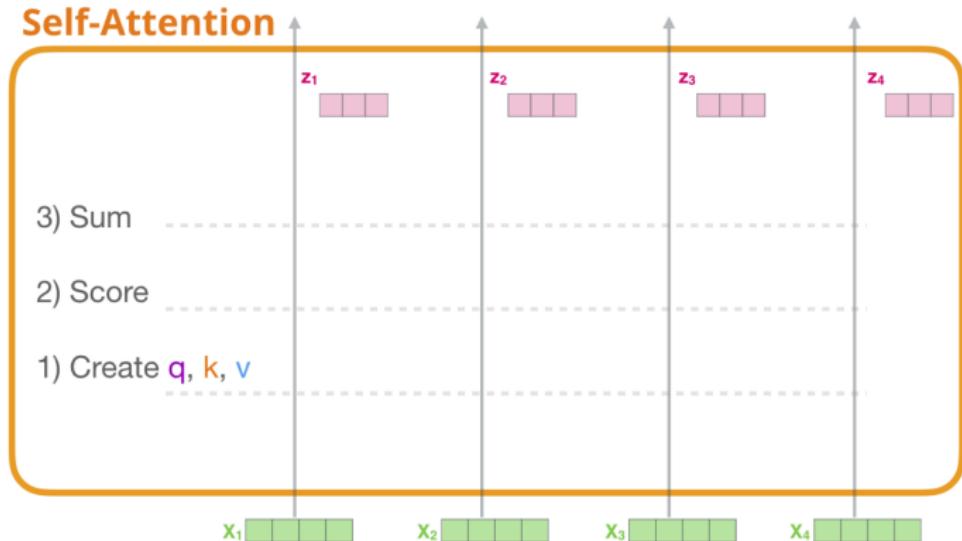


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



- Introduction
 - 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

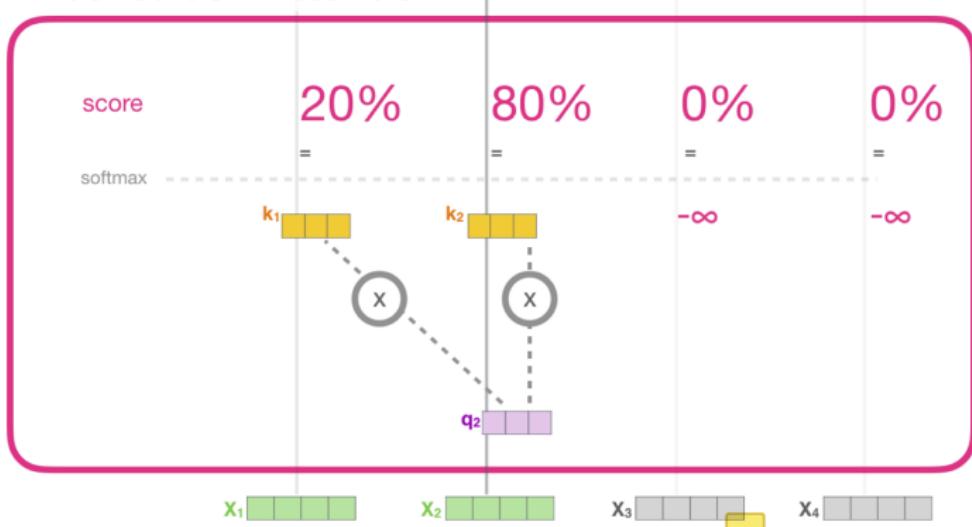


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



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

The (masked) self attention

Queries				X	Keys	Scores (before softmax)			
robot	must	obey	orders		robot	0.11	0.00	0.81	0.79
					robot	0.19	0.50	0.30	0.48
					robot	0.53	0.98	0.95	0.14
					robot	0.81	0.86	0.38	0.90

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



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

The GPT output

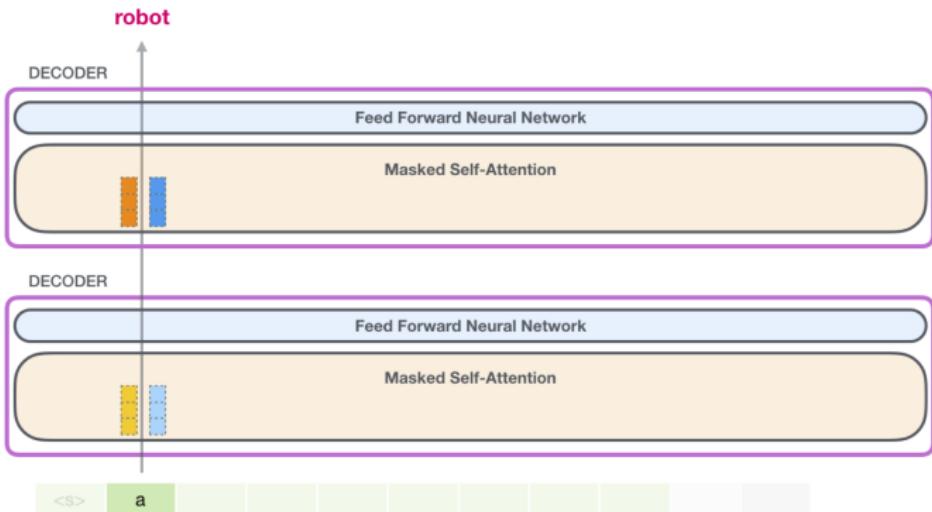


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



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

The GPT output

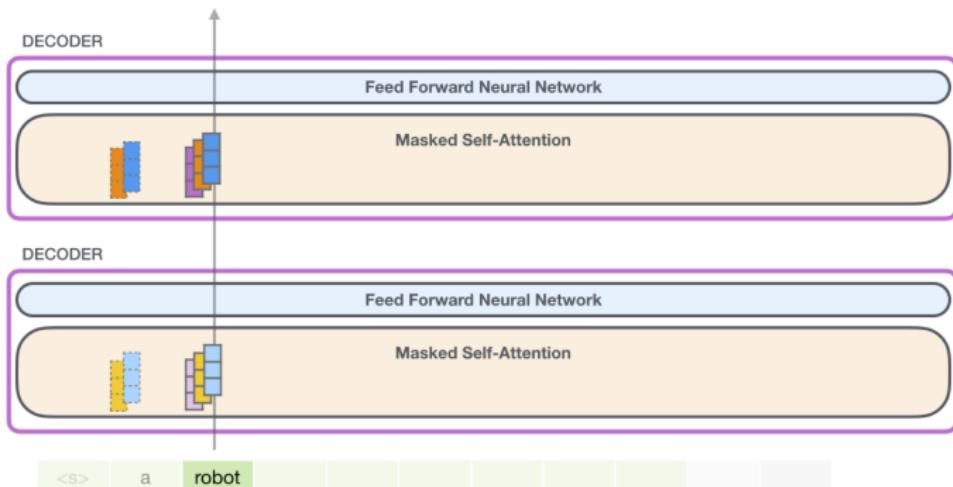


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



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

The GPT output

GPT2 Self-Attention

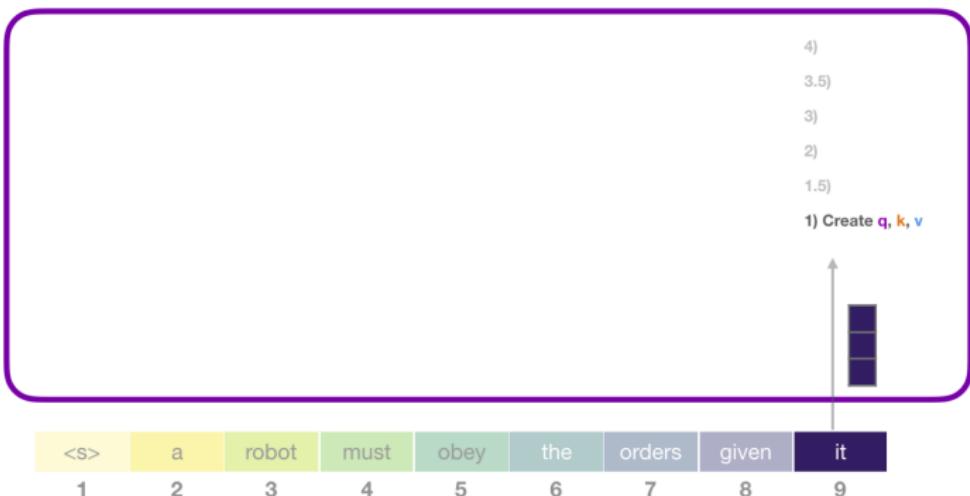


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



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

The GPT output

GPT2 Self-Attention

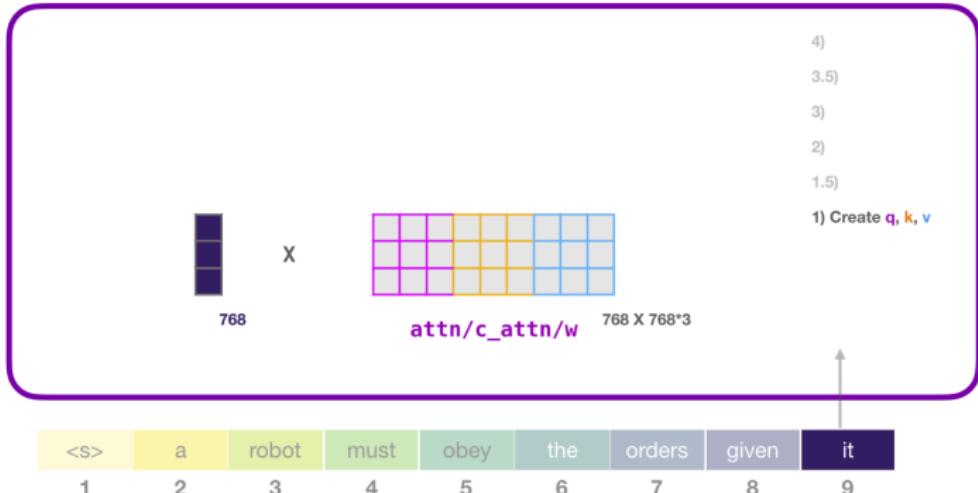


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



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

The GPT output

GPT2 Self-Attention

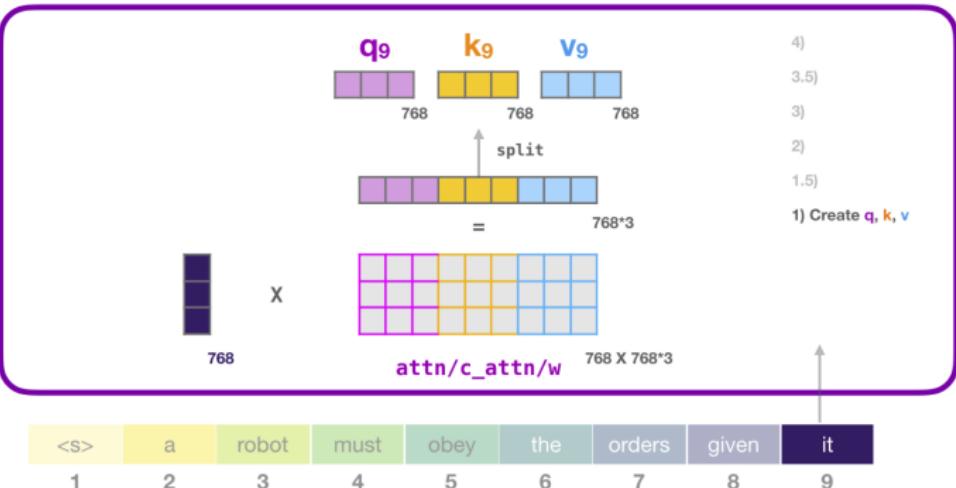


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



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

The GPT output

GPT2 Self-Attention

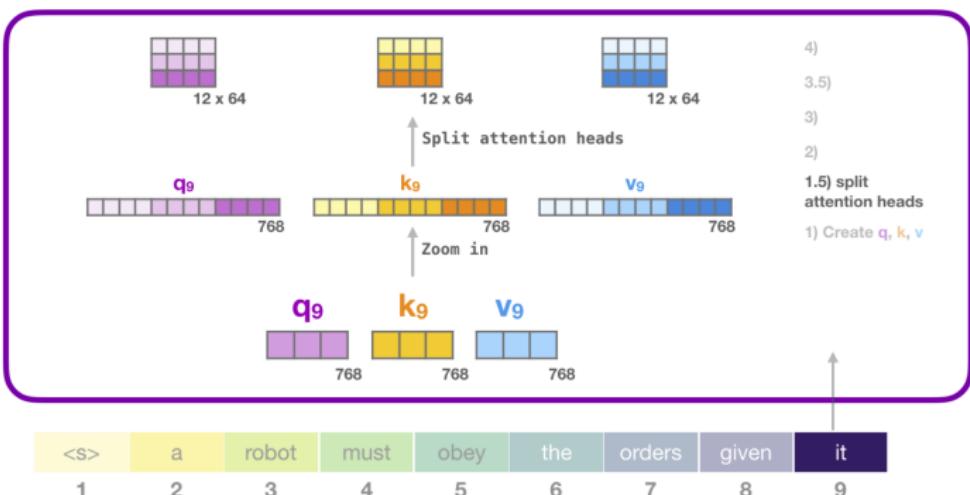


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



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

The GPT output

GPT2 Self-Attention

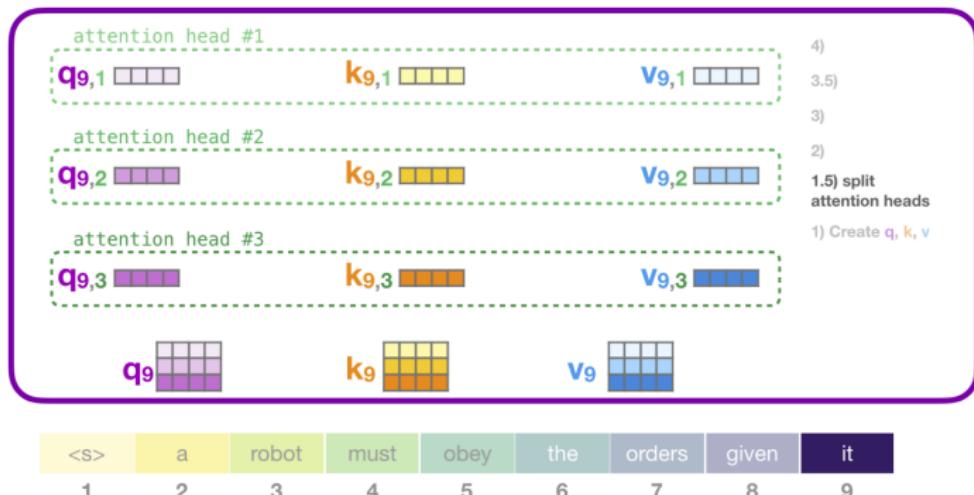


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



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

The GPT output

GPT2 Self-Attention

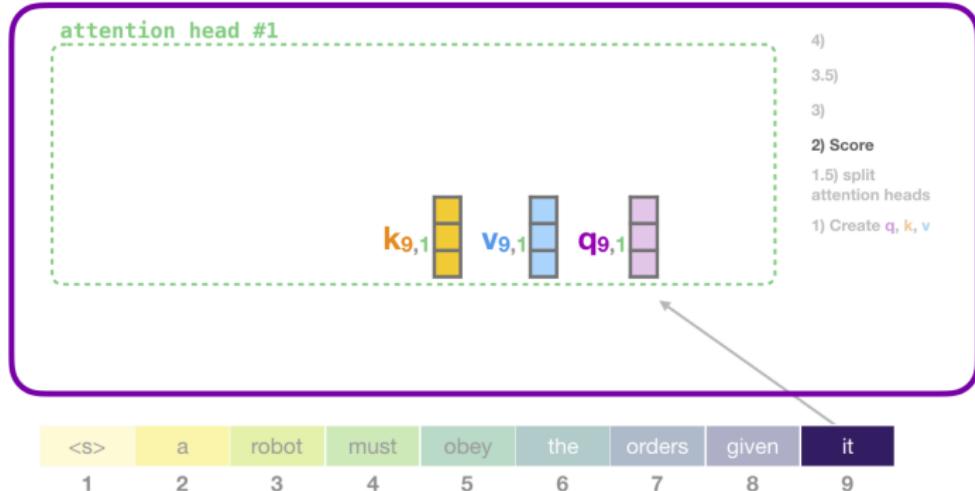


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post].
Retrieved from jalammar.github.io/illustrated-gpt2/



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

The GPT output

GPT2 Self-Attention

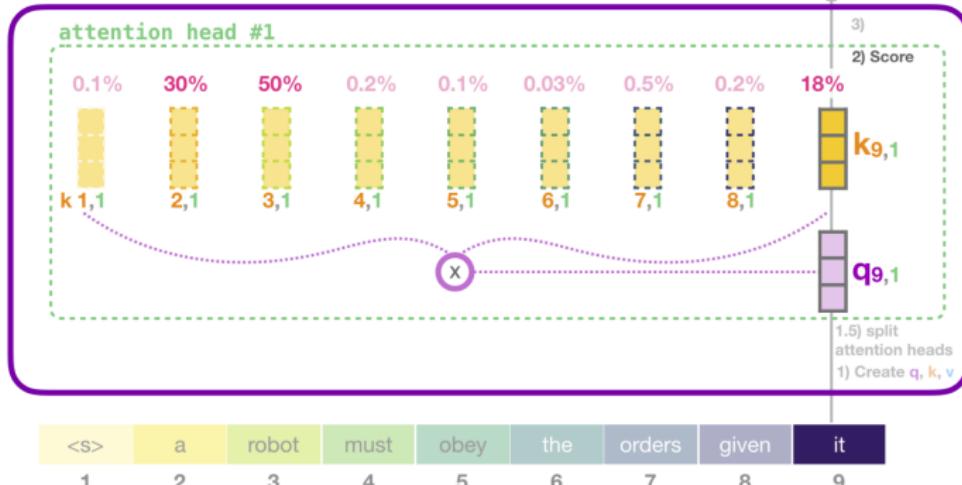


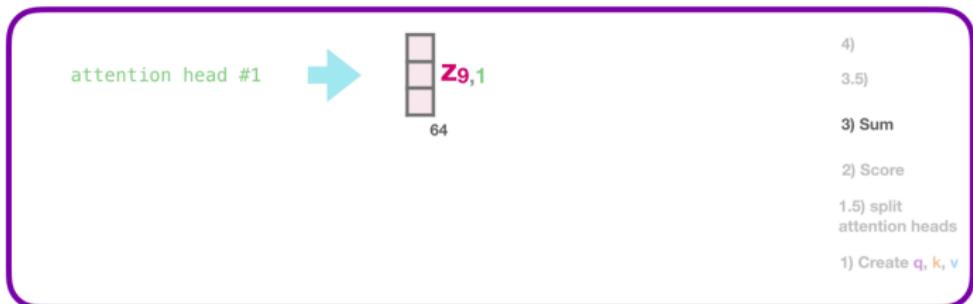
Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



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

The GPT output

GPT2 Self-Attention

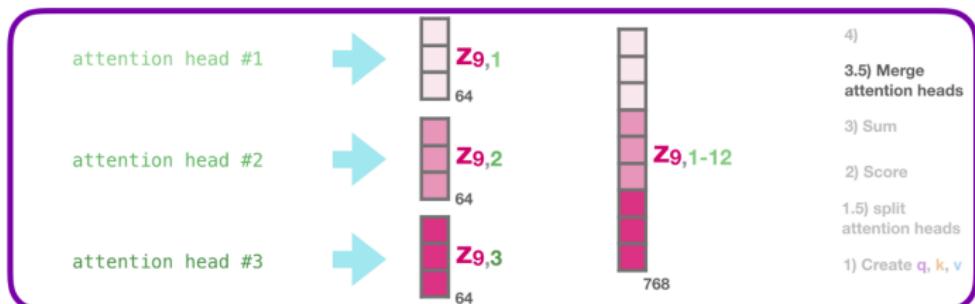


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



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

The GPT output

GPT2 Self-Attention

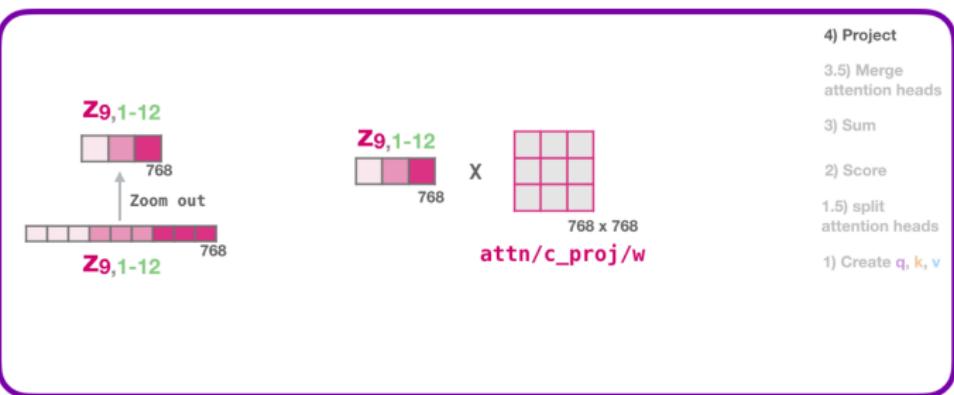


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



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

The GPT output

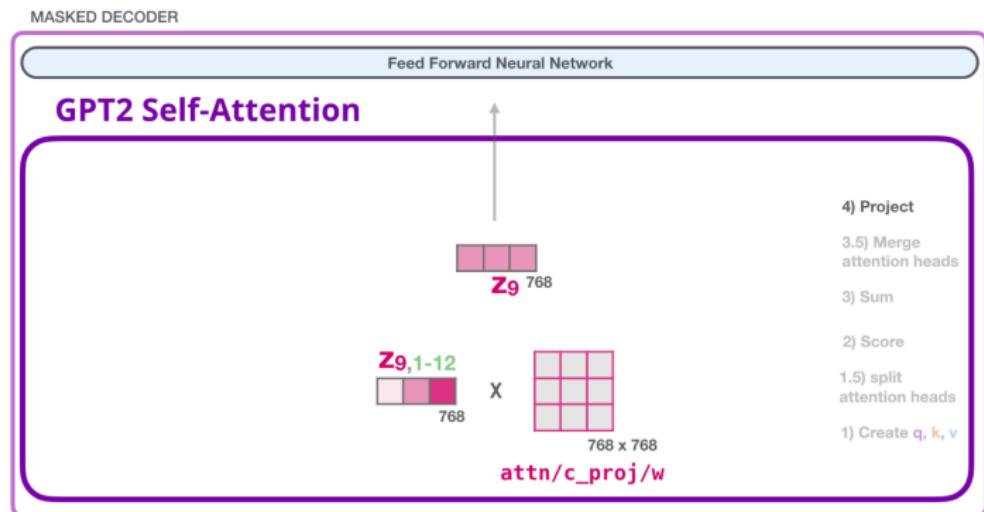


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post]. Retrieved from jalammar.github.io/illustrated-gpt2/



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

The GPT output

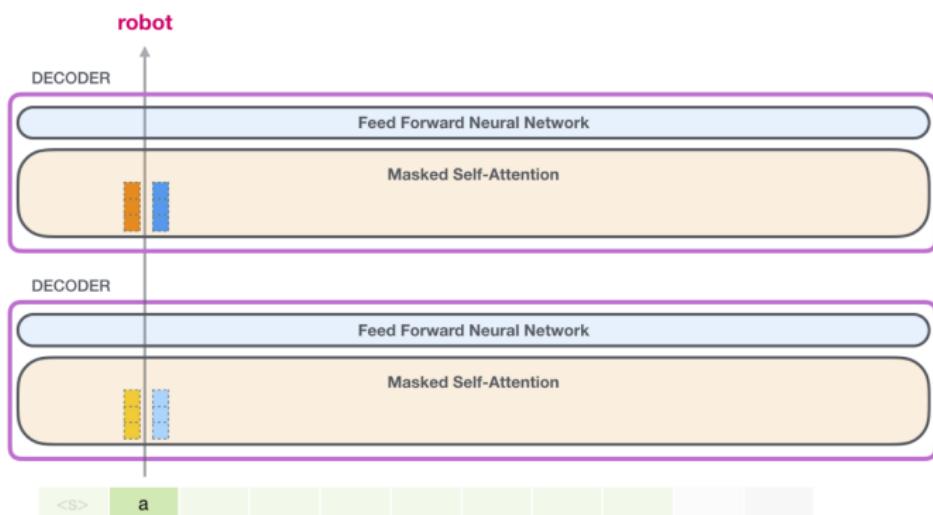
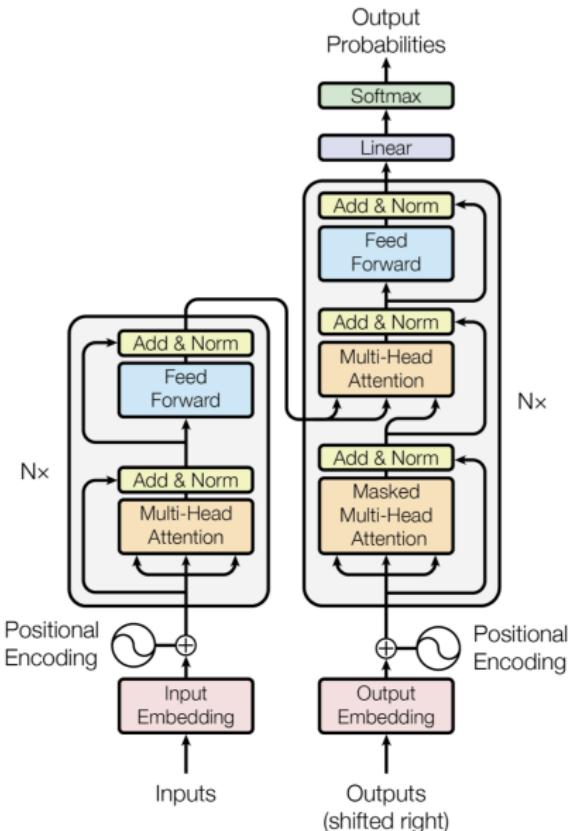


Figure: Alammar, J (2019). The Illustrated GPT2 [Blog post].
Retrieved from jalammar.github.io/illustrated-gpt2/



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

The transformer





- Introduction
 - 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)

$$p_{\theta}(x_i \mid x_{<i}) = P_{\theta}(X_i = x_i \mid X_{<i} = x_{<i}).$$



- Introduction
 - 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)

$$p_{\theta}(x_i \mid x_{<i}) = P_{\theta}(X_i = x_i \mid X_{<i} = x_{<i}).$$

- Two approaches to generate/decode:

greedy $x_i = \arg_x \max p_{\theta}(x \mid \mathbf{x}_{j < i})$

sampling $x_i \sim p_{\theta}(x \mid \mathbf{x}_{j < i})$



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

Decoding (word generation)

- Temperature sampling:

$$\tilde{p}_\theta(x_i \mid x_{<i}) = \frac{p_\theta(x_i \mid x_{<i})^{\frac{1}{T}}}{\sum_{x'} p_\theta(x' \mid x_{<i})^{\frac{1}{T}}}.$$

$$x_i \sim \tilde{p}_\theta(x_i \mid \mathbf{x}_{j < i})$$

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



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

Decoding (word generation)

- Temperature sampling:

$$\tilde{p}_\theta(x_i \mid x_{<i}) = \frac{p_\theta(x_i \mid x_{<i})^{\frac{1}{T}}}{\sum_{x'} p_\theta(x' \mid x_{<i})^{\frac{1}{T}}}.$$

$$x_i \sim \tilde{p}_\theta(x_i \mid \mathbf{x}_{j < i})$$

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

- Top- k sampling: sample proportionally from top k most probable words



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

Decoding (word generation): Top- p

- Top- p (nucleus) sampling samples from the smallest set of tokens whose cumulative probability exceeds a threshold $p \in (0, 1)$.
- Let \mathcal{V}_p be the smallest set such that

$$\sum_{x \in \mathcal{V}_p} p_\theta(x \mid x_{<i}) \geq p.$$

- Sample

$$x_i \sim p_\theta(x \mid x_{<i}), \quad x \in \mathcal{V}_p.$$

- Adapts the number of candidate tokens to model uncertainty.
- Common default in modern LLMs (e.g. $p = 0.9$).



UPPSALA
UNIVERSITET

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

Section 3

Training



Tokenization

- **Introduction**

- In-context learning (ICL)
 - Hallucinations

- **Architectures**

- **Training**

- **Fine-tuning and Alignment**

- Instruction tuning
 - Alignment
 - RLHF

- **Use**

- Prompt engineering

- Simple tokenization: Word based



- Introduction
 - 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
 - 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
 - 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
 - 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
 - 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
 - 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
 - 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

$$\mathcal{L}(\theta) = - \sum_{i=1}^n \log p_\theta(x_i | x_{<i}).$$

- Why is this good/result in good models?



- Introduction
 - 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

$$\mathcal{L}(\theta) = - \sum_{i=1}^n \log p_\theta(x_i | x_{<i}).$$

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

Section 4

Fine-tuning and Alignment



- Introduction
 - 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
 - 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
 - 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
- **Effect of instruction tuning:** Performance improvement, a general approach to enhancing the abilities of existing language models



- Introduction
 - 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
 - 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
 - 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



- Introduction
 - 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.



- Introduction
 - 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)



- Introduction
 - 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)
 - Smaller batch size and learning rate



- Introduction
 - 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)
 - Smaller batch size and learning rate
 - Balance the proportion of different tasks during fine-tuning



- Introduction
 - 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)
- Smaller batch size and learning rate
- Balance the proportion of different tasks during fine-tuning
- Pre-training data during instruction tuning as regularization



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





- Introduction
 - 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.



- Introduction
 - 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)



- Introduction
 - 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)
 2. Answering question about the model output (multiple choice)



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

RLHF: Three steps

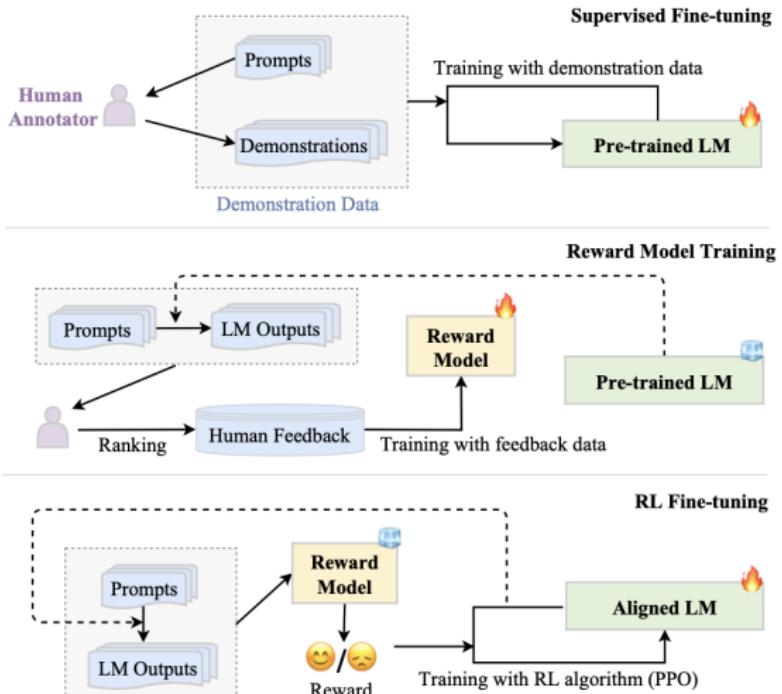


Figure: RLHF (Zhao, 2023, Figure 10)



Values of LLMs

- Introduction
 - 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



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

Values of LLMs

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



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

Values of LLMs

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

Values of LLMs

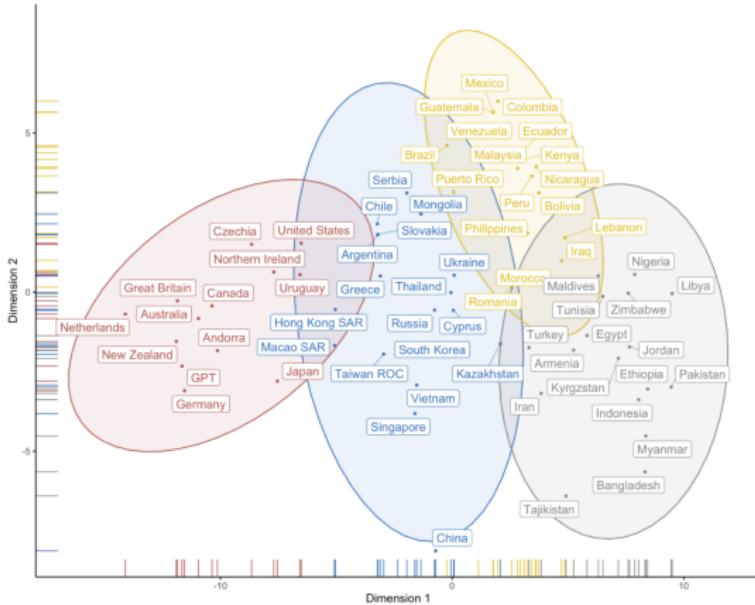


Figure: Value dimensions (Atari et al., 2023, Figure 2)



UPPSALA
UNIVERSITET

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

Section 5

Use



- Introduction
 - 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
 - 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



- Introduction
 - 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 ML tasks (e.g. see Sargeant et al 2025)



UPPSALA UNIVERSITET

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

Retrieval-Augmented Generation (RAG)

- Retrieval-Augmented Generation (RAG) combines:
 - a *retrieval model* (search over external data)
 - a *generation model* (LLM)



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

Retrieval-Augmented Generation (RAG)

- Retrieval-Augmented Generation (RAG) combines:
 - a *retrieval model* (search over external data)
 - a *generation model* (LLM)
- **Goal:** improve factuality and reduce hallucinations.
- Instead of relying only on parametric memory (weights), the model conditions on **retrieved documents**.

User query → Retriever → Relevant documents → LLM →
Output

- External data can be:
 - databases, document collections, PDFs
 - knowledge bases, APIs, logs, Wikipedia
- Knowledge can be updated **without retraining** the LLM.



Retrieval-Augmented Generation (RAG): Formalization

- Let q denote the user query.

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





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

Retrieval-Augmented Generation (RAG): Formalization

- Let q denote the user query.
- A retriever selects a set of documents:

$$D_q = \{d_1, \dots, d_k\} \sim p_\phi(d \mid q)$$





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

Retrieval-Augmented Generation (RAG): Formalization

- Let q denote the user query.
- A retriever selects a set of documents:

$$D_q = \{d_1, \dots, d_k\} \sim p_\phi(d \mid q)$$

- The LLM generates an answer conditioned on both query and documents:

$$y \sim p_\theta(y \mid q, D_q)$$

- Overall, RAG approximates:

$$p(y \mid q) = \sum_D p_\theta(y \mid q, D) p_\phi(D \mid q)$$

- Retriever often implemented via vector search (embeddings).
- Generator is a standard autoregressive LLM.
- Can be interpreted as latent-variable modeling.



- Introduction
 - 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
 - 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
 - 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
 - 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
 - In-context learning (ICL)
 - Hallucinations
 - Architectures
 - Training
 - Fine-tuning and Alignment
 - Instruction tuning
 - Alignment
 - RLHF
 - Use
 - 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
 - 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
 - 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)



- Introduction
 - 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