Large Language Models

Sukanya Nath
Data Science Lab (DSL)
University of Bern

What are Large Language Models?

What are Large Language Models?

LLM models

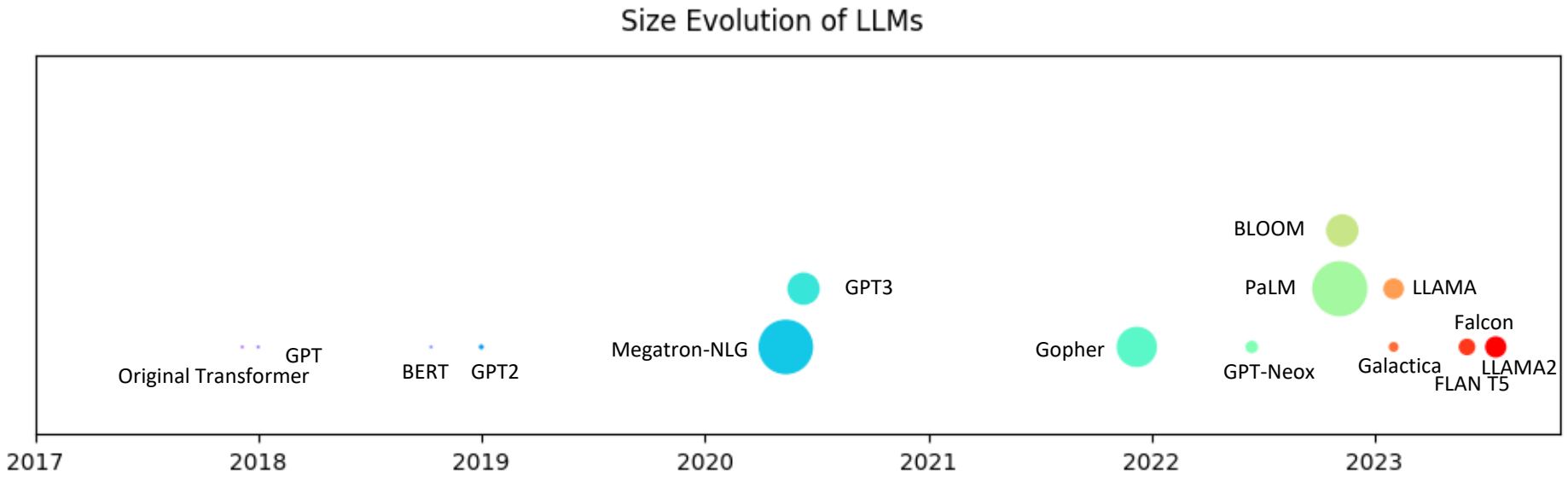
- Contain hundreds of billions or more parameters.
- Are trained on large amount of data.
- Show an understanding of language and are suited to solve complex tasks by following instructions and in context information.
- Examples are GPT-3, PaLM, LLaMA.

What are the applications of Large Language Models?

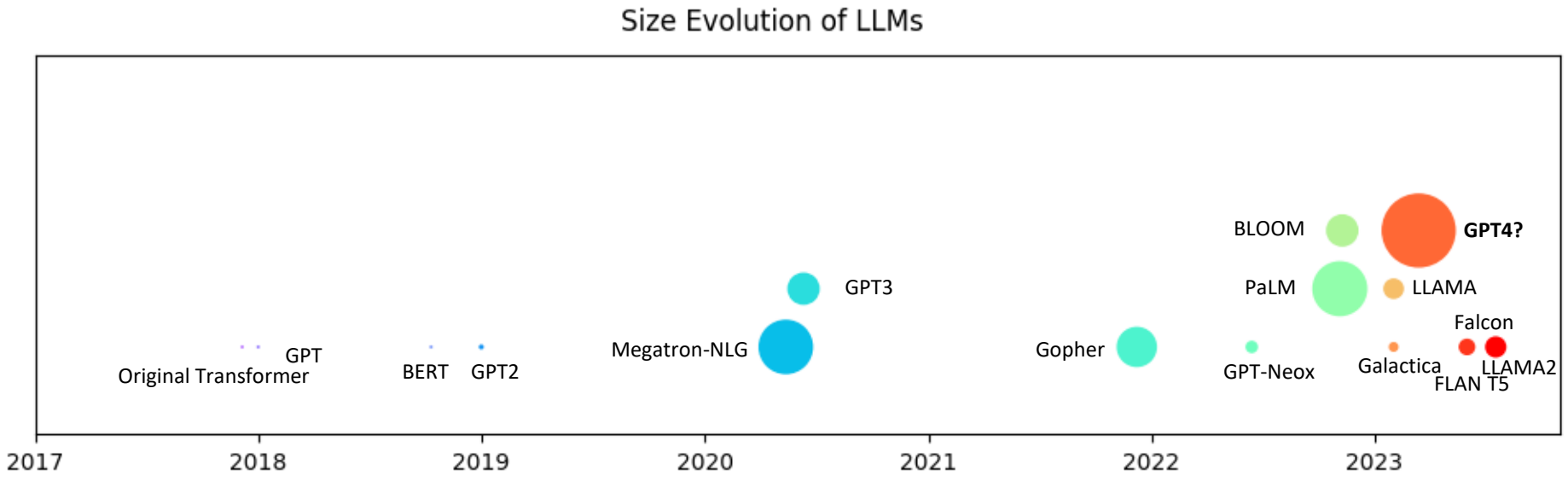
Applications of Large Language Models

1. Chatbots
2. Information Retrieval
3. Writing Assistance
4. Learning Assistance
5. Research
6. Creative Arts

LLM Timeline



LLM Timeline



LLM Timeline Table

Model Name	Release Year	Number of parameters
Original Transformer	2017	65 M
GPT	2018	100 M
BERT-Large	2018	110 M
GPT2	2019	1.5 B
Megatron-NLG	2020	530 B
GPT3	2020	1.75 B
Gopher	2021	280 B
GPT-Neox	2022	20 B
PaLM	2022	540 B
BLOOM	2022	176 B
LLAMA	2023	65 B
FlanT5	2023	11 B
GPT4	2023	1 T ??
Falcon	2023	40B
LLAMA2	2023	70B
GPT4 Turbo	2023	??

?? indicates missing information or speculation.

Why models continue to get larger?

Emergent abilities

1. Emergent abilities are those abilities in LLMs which appear in large models but are absent in smaller models.
2. Examples of emergent abilities
   - In context learning: Model can carry out the instructions in the prompt without additional training.
     - Zero shot, One shot and Few shot learning.
   - Instruction following: LLMs can perform well on unseen tasks.
   - Step-by-step reasoning: Chain-of-thought prompting strategy LLMs can show intermediate reasoning steps.

Pretraining Strategies Summary

	Encoder Only	Decoder Only	Encoder Decoder
Pretraining Task Examples	• Masked Language Modelling (bidirectional context)	• Causal Language Modelling (unidirectional context)	• Combination of Encoder and Decoder pretraining tasks • Span Corruption
Pretraining Objective	Reconstruct text (“denoising”)	Predict next token	Reconstruct span
Common use cases	• Text Classification • Token Classification (NER)	• Text Generation	• Translation • Question Answering • Summarization
Example Models	BERT, RoBERTa	GPT, BLOOM	T5, BART

Benefits of hosting your own LLMs

1. Customization and more flexibility.
2. Privacy and security.
3. Compliance.
4. Cost Control.

Challenges of hosting LLMs

1. High cost of infrastructure.
2. Skilled experts are required for creating, deploying and maintaining the LLM.
3. Requirement of updating and improving the model from time to time.

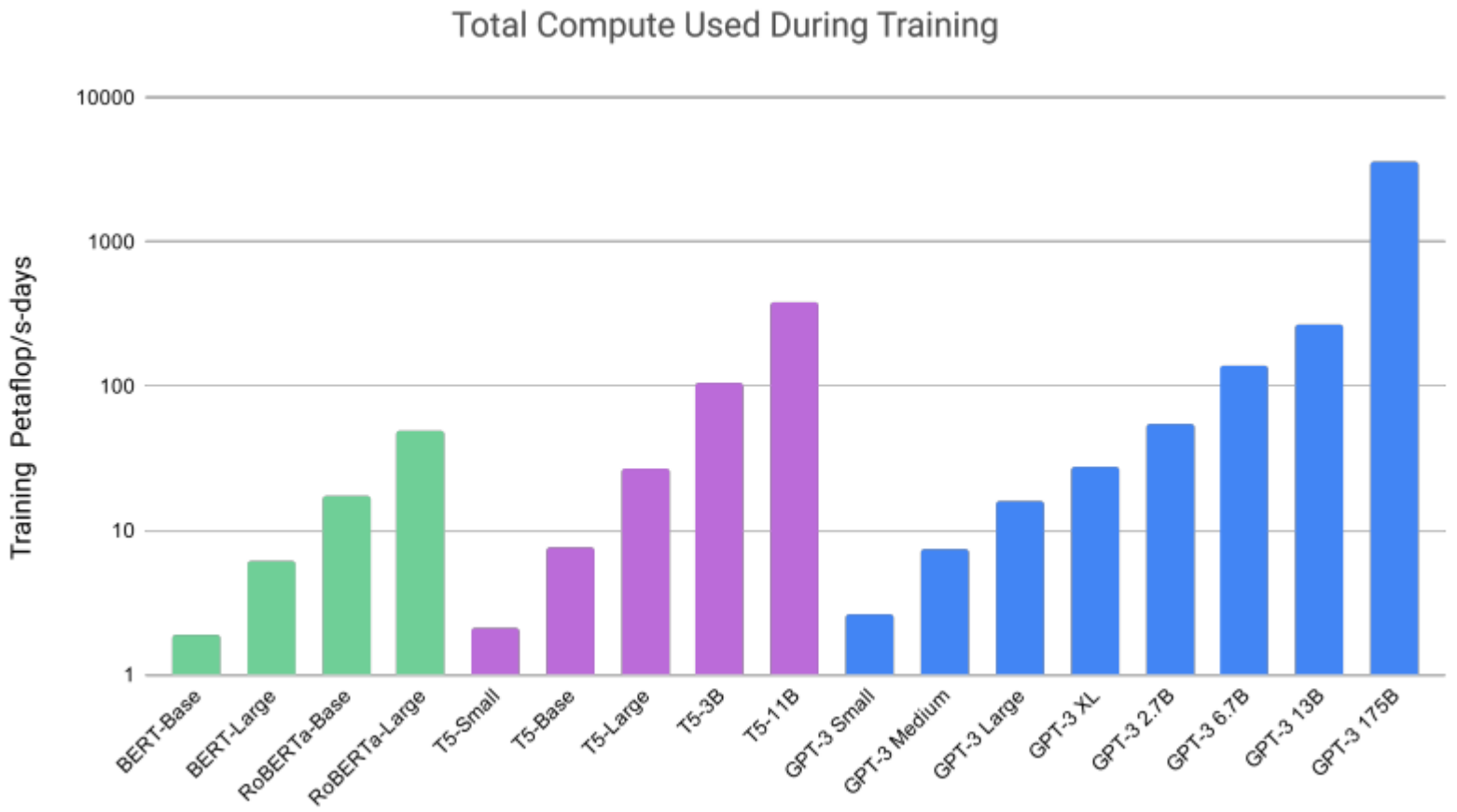
Memory requirements for training LLMs

1. How can we estimate the memory requirements for storing LLMs?
   - 1 parameter occupies 4 bytes (32-bit float) space.
   - Thus, to store a model with 1B parameters = $4\text{x}10^9$bytes = 4 GB space is required.

Memory requirements for training LLMs

1. In addition to Model Parameter weights, other components required for training:
   - Adam Optimizer
   - Gradients
   - Activations

Computational requirements for training LLMs



Peta Flop/s days is the number of floating-point operations at a rate of 1 petaflop per second for one day

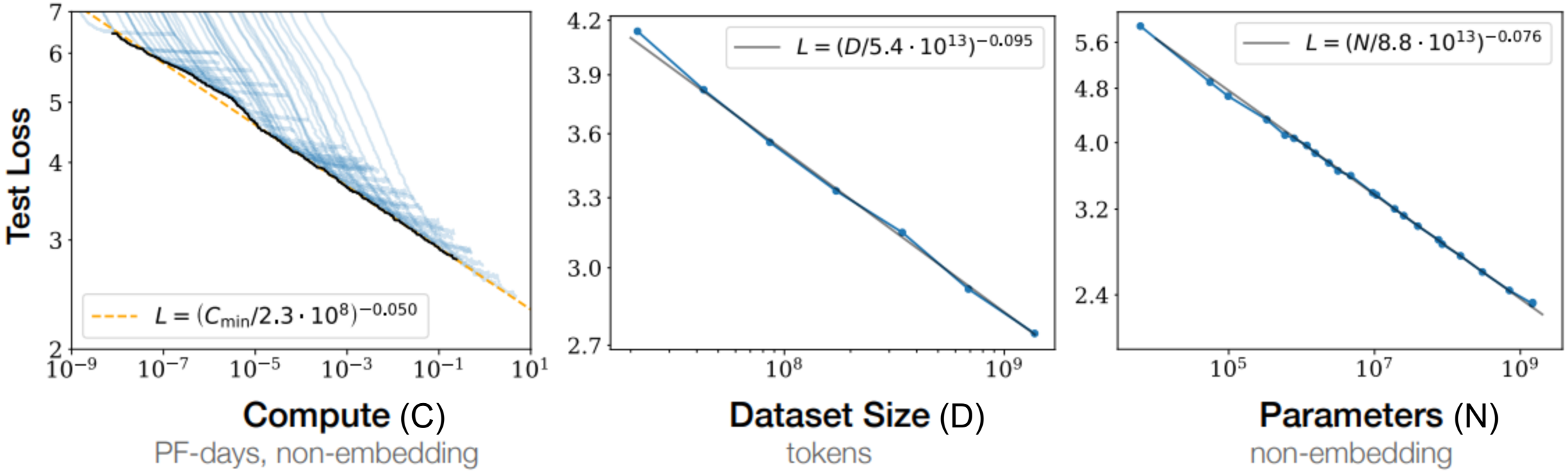
Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877-1901.

Scaling Laws

1. Scaling laws help to predict a model's ability to perform generative modelling tasks when scaled up along three dimensions:
   - Amount of data they are trained on/ number of training tokens.
   - Model size measured in parameters.
   - Computational budget (i.e the amount of computation required for training) measured in FLOPs (floating point operations).

Bowman, S. R. (2023). Eight things to know about large language models. *arXiv preprint arXiv:2304.00612*.

Scaling Laws (Kaplan et al. 2020)



Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., ... & Amodei, D. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.
https://youtu.be/Suhp3OLASSo

Scaling Laws (Kaplan et al. 2020)

Kaplan et al. 2020 showed that

1. Model performance depends most strongly on three factors: the number of model parameters, the size of the dataset D, and the amount of compute C used for training.
2. When given an 10x increase in computational budget, the model size should increase 5.5x while the number of training tokens should increase to 1.8x.
3. Large models should not be trained to their lowest possible loss to be compute optimal.
4. Large models are more sample-efficient than small models, reaching the same level of performance with fewer optimization steps
5. Larger models will continue to perform better and big models may be more important than big data.

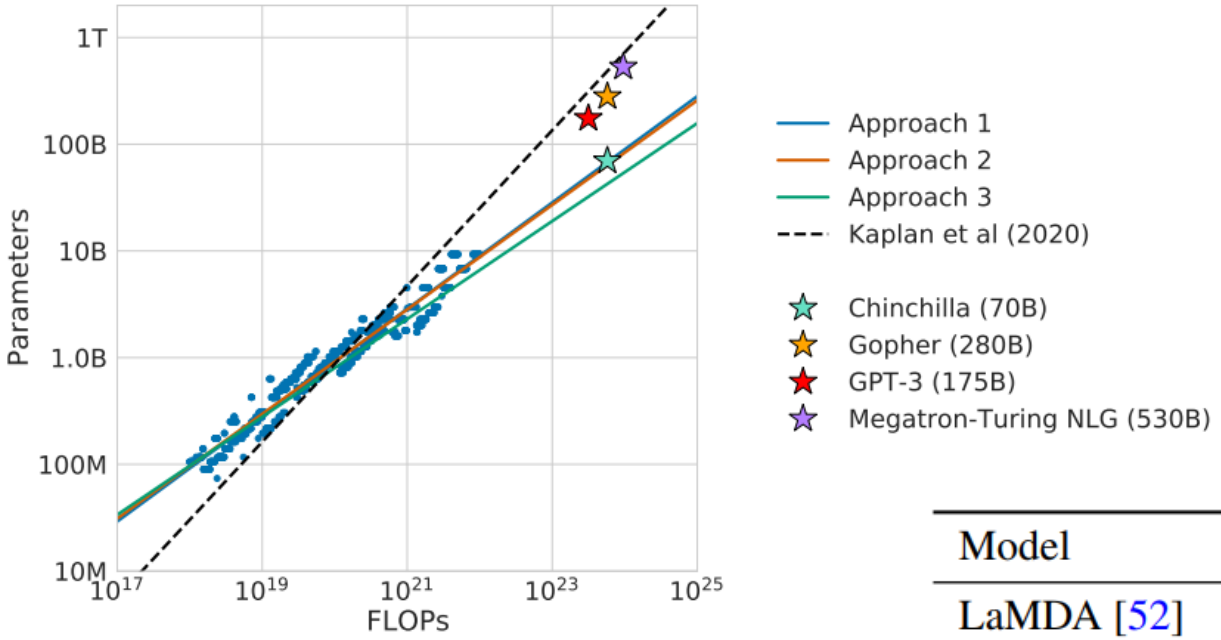
Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., ... & Amodei, D. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Scaling Laws (Hoffmann et al. 2022)

1. Given a fixed computational budget (in FLOPs), how should one trade off model size and the number of training tokens?
2. For compute-optimal training, the model size and the number of training tokens should be scaled equally.
3. Chinchilla (70B) parameters could outperform Gopher (280B), GPT-3 (175B), Megatron-Turing NLG (530B) when trained with 1.4 trillion tokens and same computational budget as Gopher.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., ... & Sifre, L. (2022). An empirical analysis of compute-optimal large language model training. *Advances in Neural Information Processing Systems*, *35*, 30016-30030.

Scaling Laws (Hoffmann et al. 2022)



Hoffman et al. 2022 shows that under the same computational budget, a higher performance model can be built by increasing the number of training tokens.

Chinchilla could beat Gopher by 7% for MMLU task.
(Massive Multi-task Language Understanding)

Model	Size (# Parameters)	Training Tokens
LaMDA [52]	137 Billion	768 Billion
GPT-3 [6]	175 Billion	300 Billion
Jurassic [30]	178 Billion	300 Billion
Gopher [38]	280 Billion	300 Billion
MT-NLG 530B [48]	530 Billion	270 Billion
Chinchilla	70 Billion	1.4 Trillion

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., ... & Sifre, L. (2022). An empirical analysis of compute-optimal large language model training. *Advances in Neural Information Processing Systems*, *35*, 30016-30030.

LLM Customisation Techniques

	Prompt Engineering	Parameter Efficient Fine Tuning	Fine Tuning
Techniques	Few Shot Learning Chain of thought reasoning System Prompting Generated knowledge Prompting	Adapters LoRA (Low Rank Adaptation of LLMs) IA3	SFT (Supervised Fine Tuning) RLHF (Reinforcement Learning with Human Feedback)
How	Prompt templates	Adding custom layers and parameters to LLMs	Tune LLM Model weights
When to use	Simple use cases and limited amount data	Hundreds of examples for each downstream tasks	Complex use cases with large amount of data (in thousands)

Hallucinations 

1. Hallucinations are characterized as generated content that is nonsensical or unfaithful to the provided source content
2. Types of Hallucinations:
   - **Factuality** hallucination emphasizes the discrepancy between generated content and verifiable real-world facts, typically manifesting as factual inconsistency or fabrication
   - **Faithfulness** Hallucination occurs when the generated content diverges from user instructions or the context provided by the input, as well as self-consistency within generated content

Huang, Lei, et al. "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions." *arXiv preprint arXiv:2311.05232* (2023).

Hallucinations

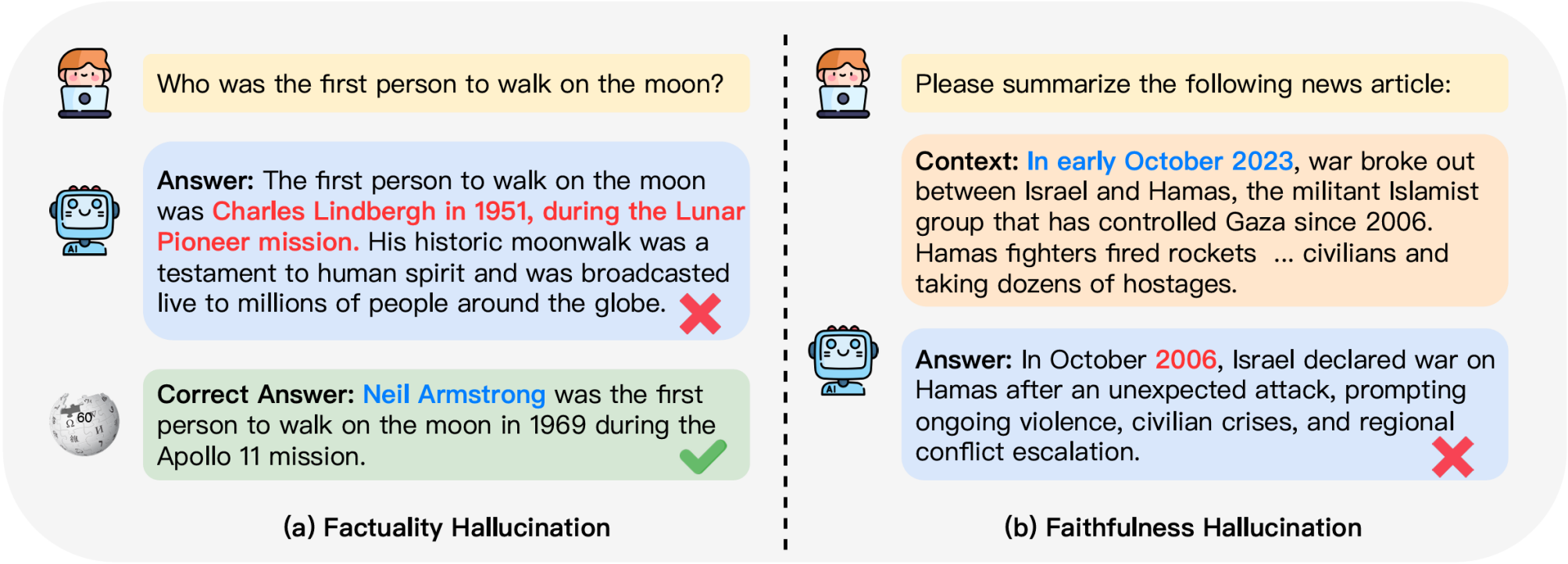


Figure 1: An intuitive example of LLM hallucination.

Huang, Lei, et al. "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions." *arXiv preprint arXiv:2311.05232* (2023).

Retrieval Augmented Generation

1. RAG takes the query as input and performs an Information Retrieval step over given data to extract relevant documents to form the contect.
2. By allowing the model to access a specific external source of data as context:
   - The reliability and accuracy of responses is improved and “hallucination” is reduced.
   - The need for constant updating of outdated LLMs without world knowledge is reduced

Retrieval Augmented Generation