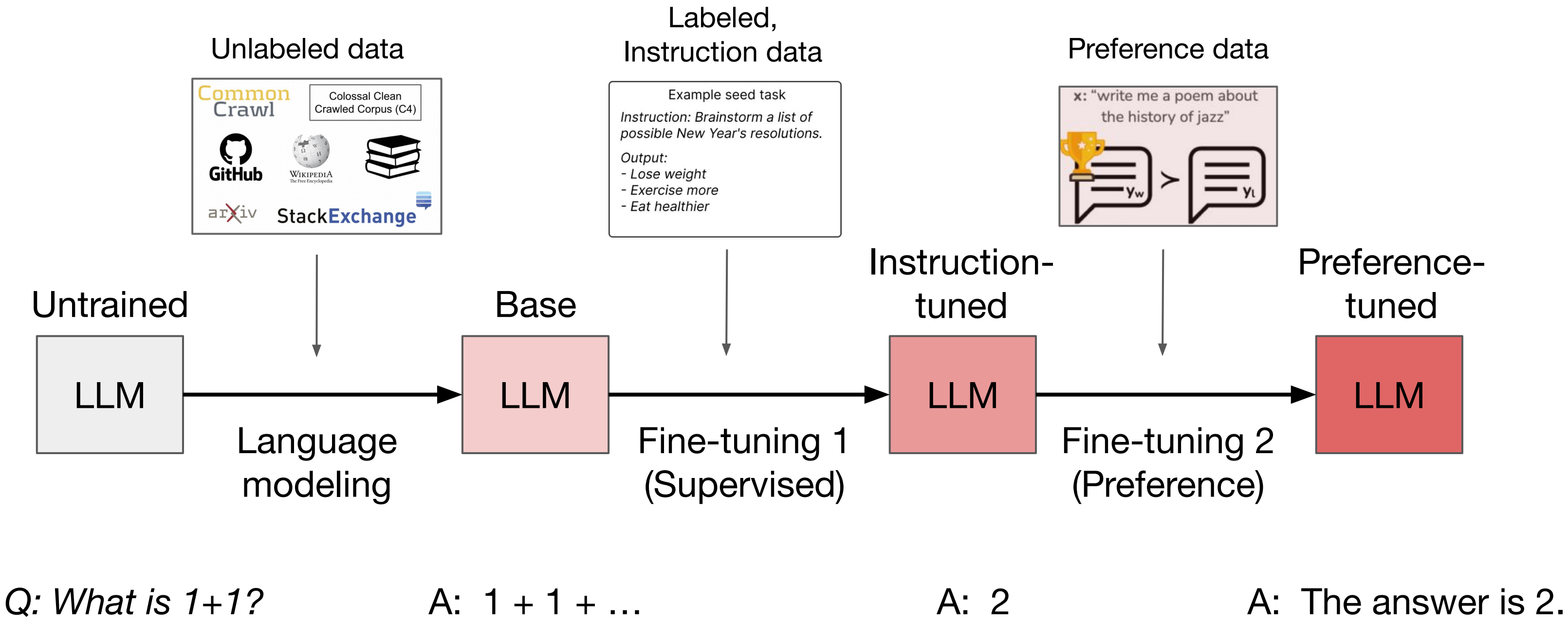
INFO-I590 Fundamentals and Applications of LLMs

Pre-Training and Pre-Trained Models

Jisun An

Many slides from Graham Neubig

Three steps of creating a high-quality LLM



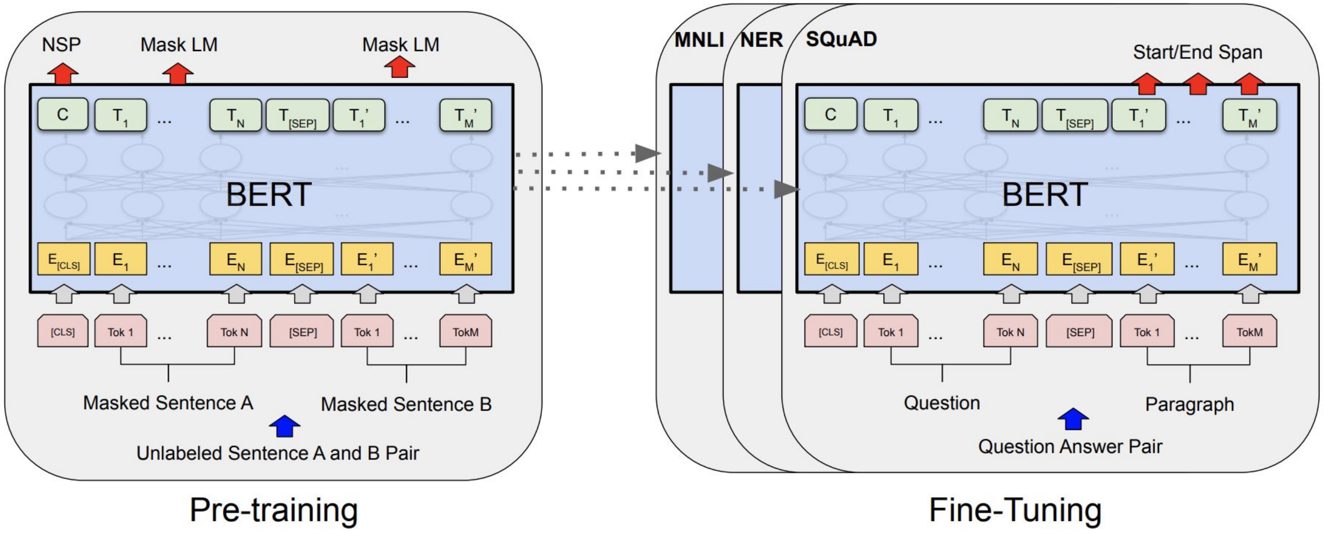
Definitions

- **Pre-training:** Training a model on a large dataset to learn general patterns and representations
- **Supervised fine-tuning (SFT)**: Training a model to learn task-specific capabilities
- **Instruction fine-tuning (IFT)**: Training a model to follow user instructions
- **Preference fine-tuning**: Using labeled preference data to fine-tune a LM
- **Alignment**: General notion of training a model to mirror user desires

Training LLMs -
Llama as an example

Meta

Why is it called pre-training?



"Pre-"training happens before training (fine-tuning)!

img: Devlin et al. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. https://arxiv.org/abs/1810.04805

Pre-training Data

Common Crawl

Colossal Clean Crawled Corpus (C4)

GitHub

WIKIPEDIA
The Free Encyclopedia

arXiv

StackExchange

Touvron et al. 2023. LLaMA: Open and Efficient Foundation Language Models. https://arxiv.org/abs/2302.13971

Example: LLaMA 1 Pre-training Data Mixture

- 1.4 Trillion Tokens!
- How big is 1.4 Trillion Tokens?
  - Comparison with Words & Books
    - 1 token ≈ 0.75 words (on average, depending on the tokenizer).
    - 1.4 trillion tokens ≈ 1.05 trillion words.
    - A typical book has around 100,000 words.
    - That means 1.4T tokens ≈ 10.5 million books.
  - Comparison with Human Reading
    - An average person reads 200-300 words per minute.
    - At 250 words per minute, it would take:
    - 8.4 billion minutes to read 1.05T words.
    - 16,000 years of nonstop reading (24/7) to process that much text.

ref: ChatGPT 4o

Example: LLaMA 1 Pre-training Data Mixture

- 1.4 Trillion Tokens!
- Several sources, with more reliable source upsampled

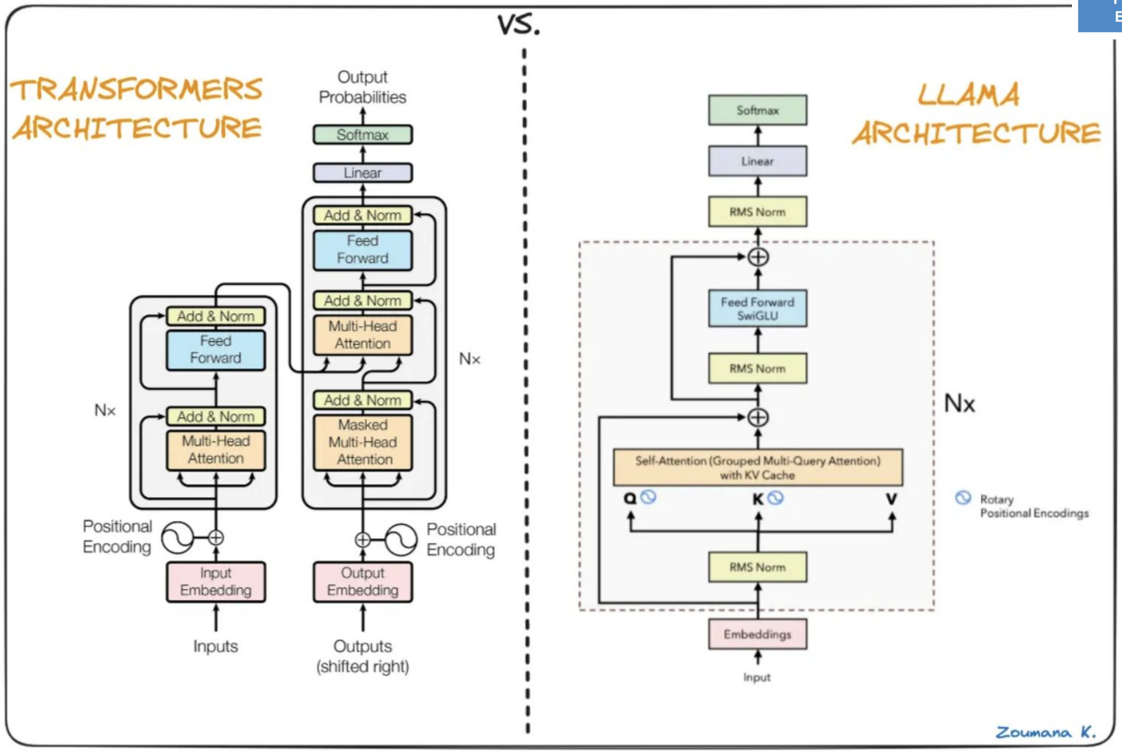
Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

Tokenizer (Recap)

“We tokenize the data with the **bytepair encoding (BPE)** algorithm (Sennrich et al., 2015), using the implementation from SentencePiece (Kudo and Richardson, 2018). Notably, we split all numbers into individual digits, and fallback to bytes to decompose unknown UTF-8 characters.” (—from Llama1 Paper)

Architecture (Recap)

	Vaswani et al.	LLaMA
Norm Position	Post	Pre
Norm Type	LayerNorm	RMSNorm
Non-linearity	ReLU	SiLU
Positional Encoding	Sinusoidal	RoPE



Img: https://medium.com/@pranjalkhadka/llama-explained-a70e71e706e9

Grouped Query Attention (GQA)

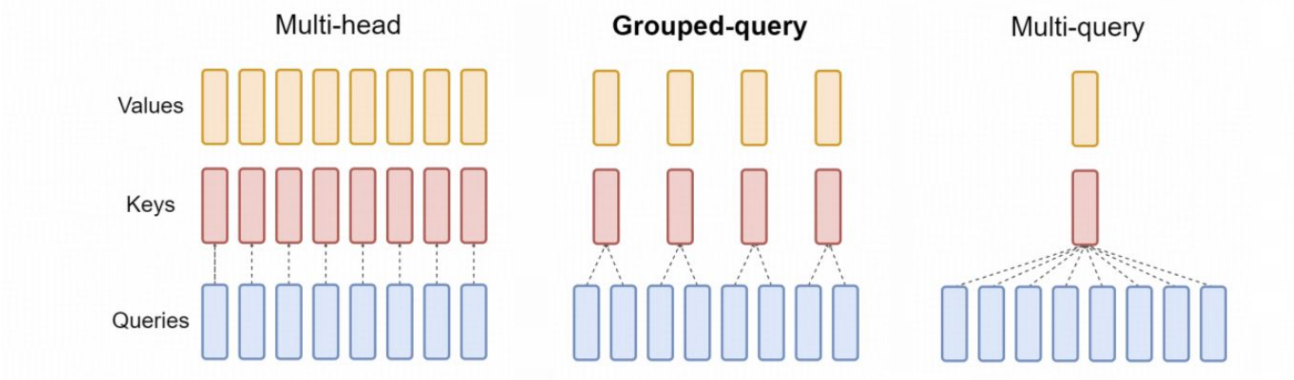
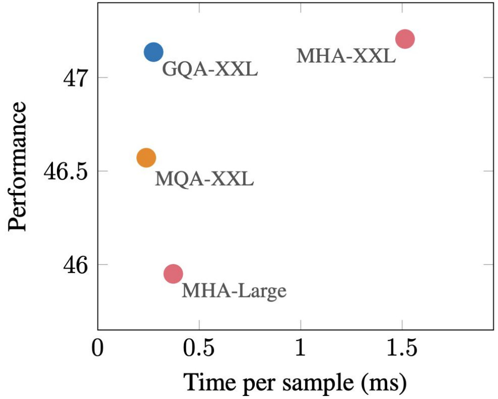


Figure 2: Overview of grouped-query method. Multi-head attention has H query, key, and value heads. Multi-query attention shares single key and value heads across all query heads. Grouped-query attention instead shares single key and value heads for each *group* of query heads, interpolating between multi-head and multi-query attention.



* Inference time

Ainslie et al. 2023. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. https://arxiv.org/abs/2305.13245

Other Setups

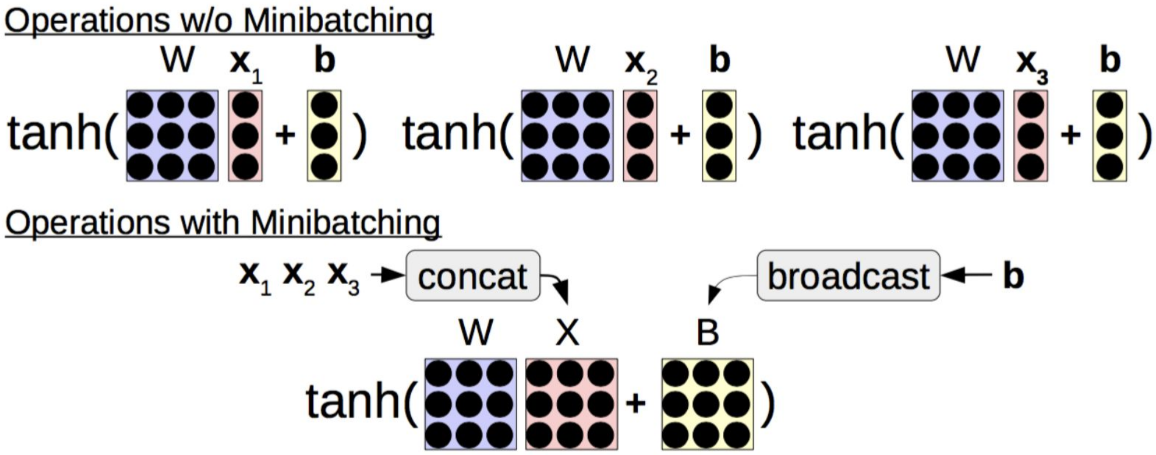
params	dimension	n heads	n layers	learning rate	batch size	n tokens
6.7B	4096	32	32	$3.0e^{-4}$	4M	1.0T
13.0B	5120	40	40	$3.0e^{-4}$	4M	1.0T
32.5B	6656	52	60	$1.5e^{-4}$	4M	1.4T
65.2B	8192	64	80	$1.5e^{-4}$	4M	1.4T

Table 2: **Model sizes, architectures, and optimization hyper-parameters.**

- **Optimizer**: AdamW (β1: 0.9, β2: 0.95)
- **Learning Rate Schedule**: Cosine schedule
- **Final learning rate**: 10% of the maximal learning rate
- **Weight Decay**: 0.1
- **Gradient Clipping**: 1.0
- **Warmup Steps**: 2,000 steps

Mini-batching

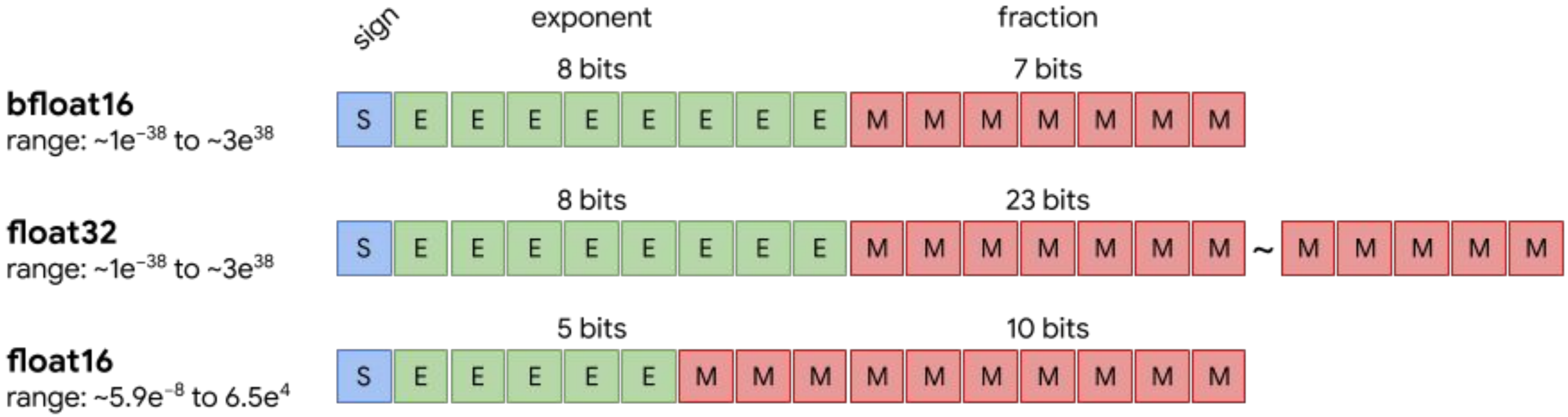
- On modern hardware 10 operations of size 1 is much slower than 1 operation of size 10
- Mini-batching combines together smaller operations into one big one



Train with a GPU cluster

- Trained on NVIDIA A100 GPUs
- Model sizes: 7B, 13B, 33B, 65B parameters
- 65B model was trained using **2,048 A100 GPUs**
- Used BF16 (bfloat16) precision for optimized performance and memory efficiency

Floating-point Format



Img: https://cloud.google.com/tpu/docs/bfloat16

Loss Curve

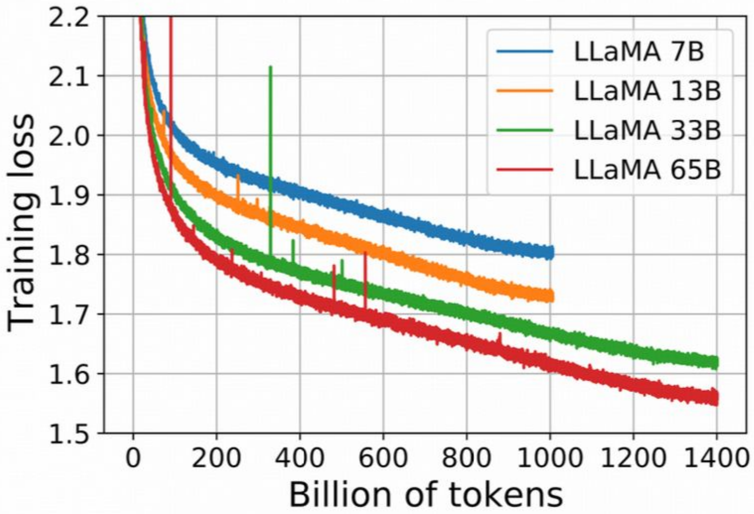


Figure 1: **Training loss over train tokens for the 7B, 13B, 33B, and 65 models.** LLaMA-33B and LLaMA-65B were trained on 1.4T tokens. The smaller models were trained on 1.0T tokens. All models are trained with a batch size of 4M tokens.

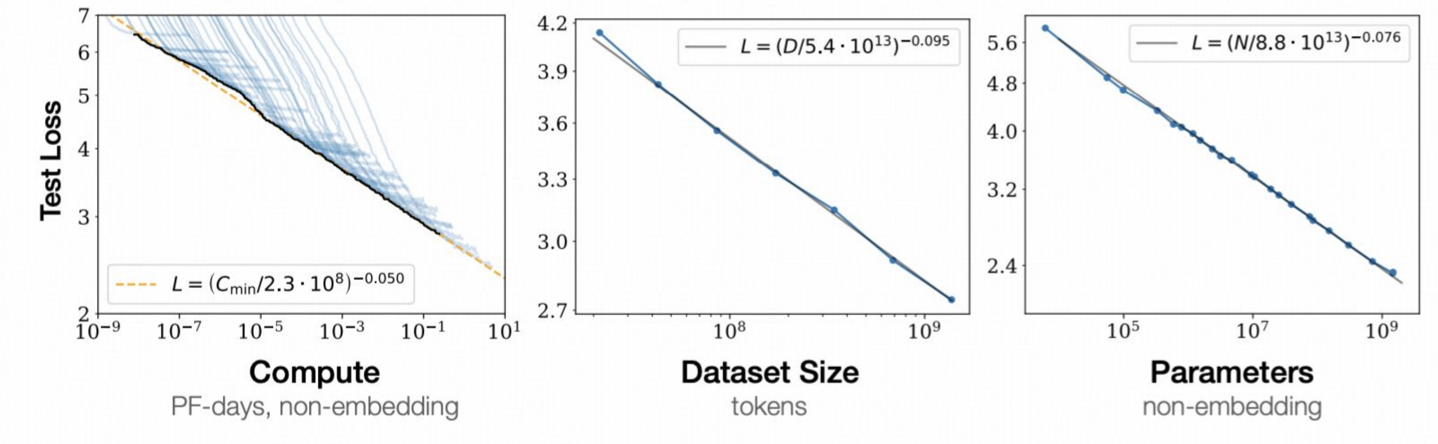
Scaling Laws and Emergent Abilities

Scaling Laws

- The Scaling Law (as proposed by OpenAI and DeepMind) states that larger models trained on more data with more compute lead to better performance, following a predictable power-law relationship:

$$\text{Loss} \propto N^{-\alpha} + D^{-\beta} + C^{-\gamma}$$

where:
N = model size (parameters)
D = dataset size
C = compute resources
α, β, γ = scaling exponents



Kaplan et al. 2020 Scaling Laws for Neural Language Models https://arxiv.org/pdf/2001.08361

Emergent Abilities of Large Language Models

- Why do LLM work so well?
- Potential explanation: emergent abilities!
- An ability is emergent if it is present in larger but not smaller models
- Not have been directly predicted by extrapolating from smaller models
- Performance is near-random until a certain critical threshold, then improves heavily

Wei et al. 2022. Emergent Abilities of Large Language Models. https://arxiv.org/abs/2206.07682

Discontinuous jumps in capability

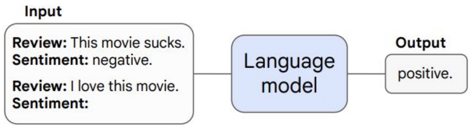


Figure 1: Example of an input and output for few-shot prompting.

Larger models can

- perform sentiment classification w/ few examples
- explain math proofs step by step
- write functional programs
- predict how a human would respond

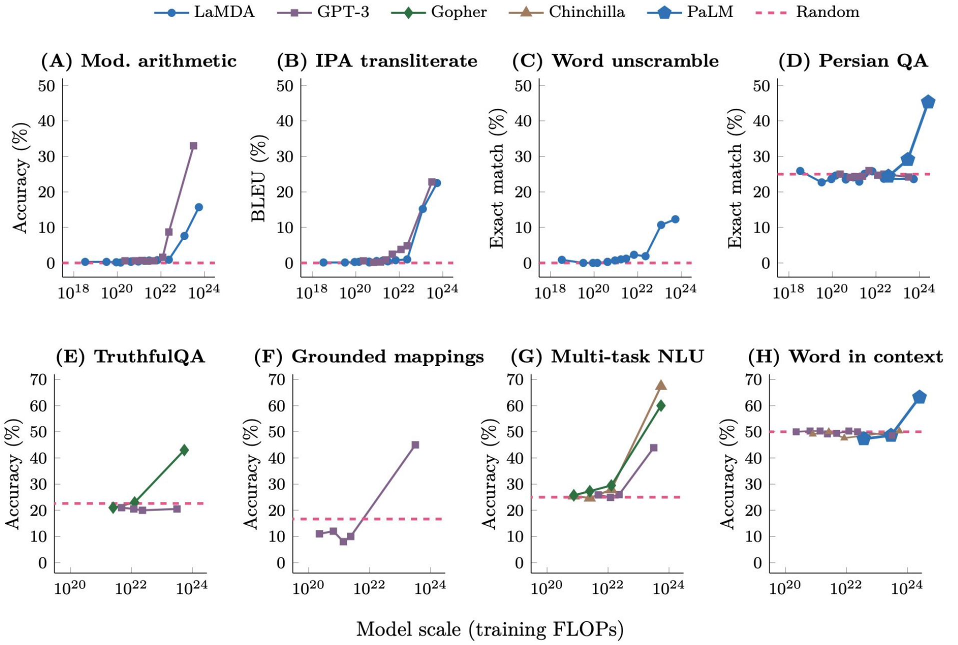


Figure 2: Eight examples of emergence in the few-shot prompting setting. Each point is a separate model. The ability to perform a task via few-shot prompting is emergent when a language model achieves random performance until a certain scale, after which performance significantly increases to well-above random. Note that models that used more training compute also typically have more parameters—hence, we show an analogous figure with number of model parameters instead of training FLOPs as the x-axis in Figure 11. A–D: BIG-Bench (2022), 2-shot. E: Lin et al. (2021) and Rae et al. (2021). F: Patel & Pavlick (2022). G: Hendrycks et al. (2021a), Rae et al. (2021), and Hoffmann et al. (2022). H: Brown et al. (2020), Hoffmann et al. (2022), and Chowdhery et al. (2022) on the WiC benchmark (Pilehvar & Camacho-Collados, 2019).

Wei et al. 2022. Emergent Abilities of Large Language Models. https://arxiv.org/abs/2206.07682

Open vs. Closed Access

Open/Closed Access (e.g. Liang et al. 2022)

- **Weights**: open? described? closed?
- **Inference Code**: open? described? closed?
- **Training Code**: open? described? closed?
- **Data**: open? described? closed?

Licenses and Permissiveness

- **Public domain, CC-0**: old copyrighted works and products of US government workers
- **MIT, BSD**: very few restrictions
- **Apache, CC-BY**: must acknowledge owner
- **GPL, CC-BY-SA**: must acknowledge and use same license for derivative works
- **CC-NC**: cannot use for commercial purposes
- **LLaMA, OPEN-RAIL**: various other restrictions
- **No License**: all rights reserved, but can use under fair use

Fair Use

- US **fair use doctrine** — can use copyrighted material in some cases
- A gross simplification:
  - **Quoting** a small amount of material → likely OK
  - **Doesn't diminish** commercial value → possibly OK
  - Use for **non-commercial** purposes → possibly OK
- Most data on the internet is copyrighted, so model training is currently done assuming fair use
- But there are lawsuits!

The Times Sues OpenAI and Microsoft Over A.I. Use of Copyrighted Work

Millions of articles from The New York Times were used to train chatbots that now compete with it, the lawsuit said.

Why Restrict Model Access?

- **Commercial Concerns**: Want to make money from the models
- **Safety**: Limited release prevents possible misuse
- **Legal Liability**: Training models on copyrighted data is a legal/ethical gray area

Pre-trained Models

Birds-eye View

- Open source/reproducible:
  - **Pythia**: Fully open, many sizes/checkpoints (trained on 300B tokens of The Pile)
  - **OLMo**: Fully documented model, instruction tuned (trained on 2.46T tokens of Dolma)
  - **DeepSeek:** Reasoning ability based on RL, Possibly strongest reproducible model
- Open weights:
  - **LLaMa1/2/3/3.1**: Most popular, heavily safety tuned
  - **Mistral/Mixtral**: Strong and fast model, several European languages
  - **Qwen**: Strong, more multilingual - particularly en/zh
- Closed
  - **GPT-4o**
  - **Gemini**
  - **Claude 3**

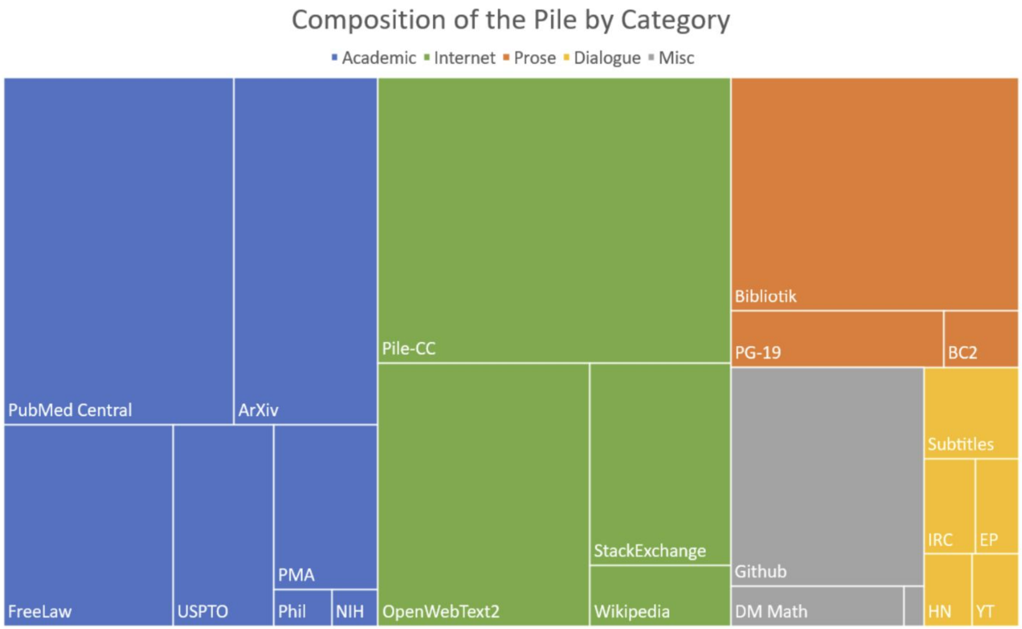
Pythia - Overview

- Creator: EleutherAI
- Goal: Joint understanding of model training dynamics and scaling
- Unique features: 8 model sizes 70M-12B, 154 checkpoints for each

Arch	Transformer+RoPE+SwiGLU, context 2k (cf LLaMa 4k), parametric LN
Data	Trained on 300B tokens of The Pile (next slide), or deduped 207B
Train	LR scaled inversely to model size (7B=1.2e-4), batch size 2M tokens

The Pile

- A now-standard 800GB dataset of lots of text/code



Pythia - Findings

- Some insights into training dynamics, e.g. larger models memorize facts more quickly
- It is possible to intervene on data to reduce gender bias

OLMo - Overview

- Creator: Ai2 Allen Institute for AI
- Goal: Better science of state-of-the-art LMs
- Unique features: Fully documented model, instruction tuned etc.

Arch	Transformer+RoPE+SwiGLU, context 4k, non-parametric LN
Data	Trained on 2.46T tokens of Dolma corpus (next slide)
Train	LR scaled inversely to model size (7B=3e-4), batch size 4M tokens

Dolma

- 3T token corpus created and released by AI2 for LM training
- A pipeline of (1) language filtering, (2) quality filtering, (3) content filtering, (4) deduplication, (5) multi-source mixing, and (6) tokenization

Source	Doc Type	UTF-8 bytes (GB)	Documents (millions)	Unicode words (billions)	Llama tokens (billions)
Common Crawl	web pages	9,022	3,370	1,775	2,281
The Stack	code	1,043	210	260	411
C4	web pages	790	364	153	198
Reddit	social media	339	377	72	89
PeS2o	STEM papers	268	38.8	50	70
Project Gutenberg	books	20.4	0.056	4.0	6.0
Wikipedia, Wikibooks	encyclopedic	16.2	6.2	3.7	4.3
Total		**11,519**	**4,367**	**2,318**	**3,059**

OLMo - Findings

- Competitive average performance
- Performance increases constantly w/ training

LLaMA2 - Overview

- Creator: Meta
- Goal: Strong and safe open LM w/ base+chat versions
- Unique features: Open model with strong safeguards and chat tuning, good performance

Arch	Transformer+RoPE+SwiGLU, context 4k, RMSNorm
Data	Trained on "public sources, up-sampling the most factual sources", LLaMa 1 has more info (next page), total 2T tokens
Train	7B=3e-4, batch size 4M tokens

LLaMA 1 - Training Data

- Several sources, with more reliable source upsampled

Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

LLaMA2 - Reward Model

- LLaMa 2 dev put a large emphasis on safety
- Step 1: Collect data for reward modeling

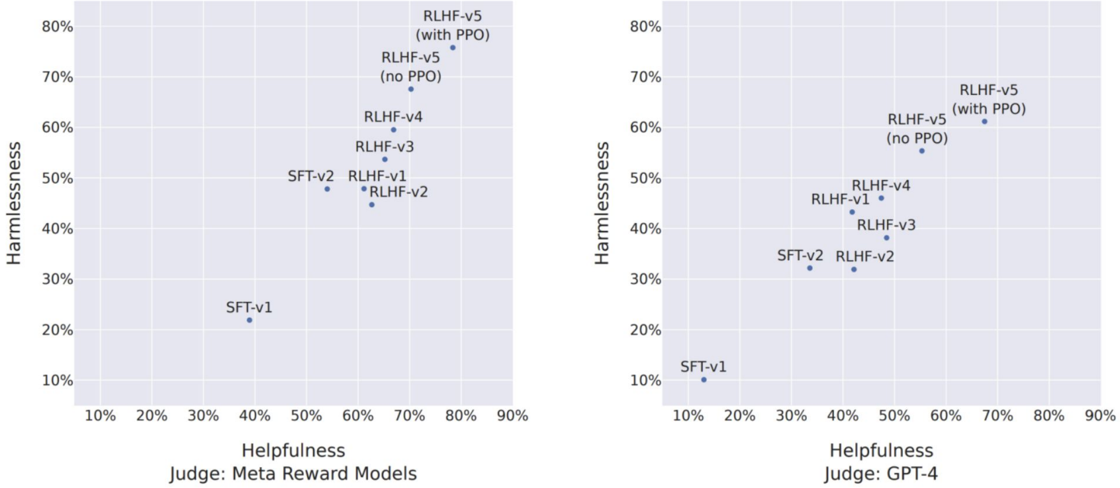
Dataset	Num. of Comparisons	Avg. # Turns per Dialogue	Avg. # Tokens per Example	Avg. # Tokens in Prompt	Avg. # Tokens in Response
Anthropic Helpful	122,387	3.0	251.5	17.7	88.4
Anthropic Harmless	43,966	3.0	152.5	15.7	46.4
OpenAI Summarize	176,625	1.0	371.1	336.0	35.1
OpenAI WebGPT	13,333	1.0	237.2	48.3	188.9
StackExchange	1,038,480	1.0	440.2	200.1	240.2
Stanford SHP	74,882	1.0	338.3	199.5	138.8
Synthetic GPT-J	33,139	1.0	123.3	13.0	110.3
Meta (Safety & Helpfulness)	1,418,091	3.9	798.5	31.4	234.1
Total	2,919,326	1.6	595.7	108.2	216.9

- Step 2: Train model to follow these preferences

	Meta Helpful.	Meta Safety	Anthropic Helpful	Anthropic Harmless	OpenAI Summ.	Stanford SHP	Avg
SteamSHP-XL	52.8	43.8	66.8	34.2	54.7	75.7	55.3
Open Assistant	53.8	53.4	67.7	68.4	71.7	55.0	63.0
GPT4	58.6	58.1	-	-	-	-	-
Safety RM	56.2	64.5	55.4	74.7	71.7	65.2	64.3
Helpfulness RM	63.2	62.8	72.0	71.0	75.5	80.0	70.6

LLaMA2 - RLHF

- Train model using reward model



- Each round of RLHF improves final model

LLaMA3.1 - Overview

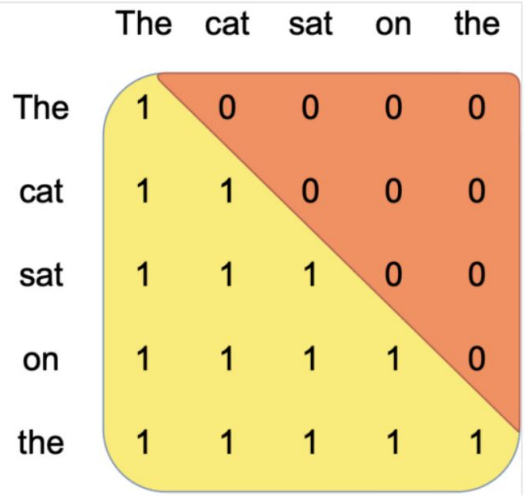
- Creator: Meta
- Goal: A herd of language models that natively support multilinguality, coding, reasoning, and tool usage
- Compared with Llama2: Larger Data scale (15T multilingual tokens vs 1.8T tokens). More Training FLOPs (almost 50× more than Llama 2)

Mistral/Mixtral - Overview

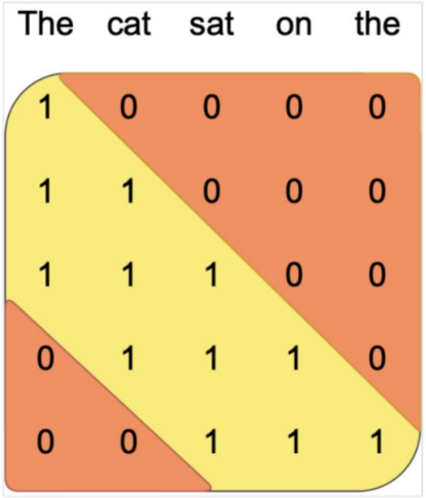
- Creator: MISTRAL AI_
- Goal: Strong and somewhat multilingual open LM
- Unique features: Speed optimizations, including GQA and Mixture of Experts

Arch	Transformer+RoPE+SwiGLU, context 4k, RMSNorm, sliding window attention. Mixtral has 8x experts in feed-forward layer
Data	Not disclosed? But includes English and European languages
Train	Not disclosed?

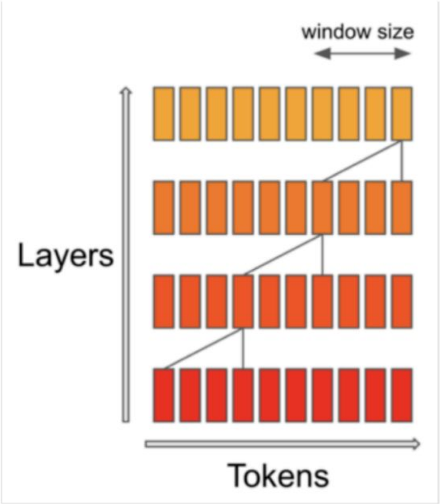
Mistral - Sliding Window Attention



Vanilla Attention

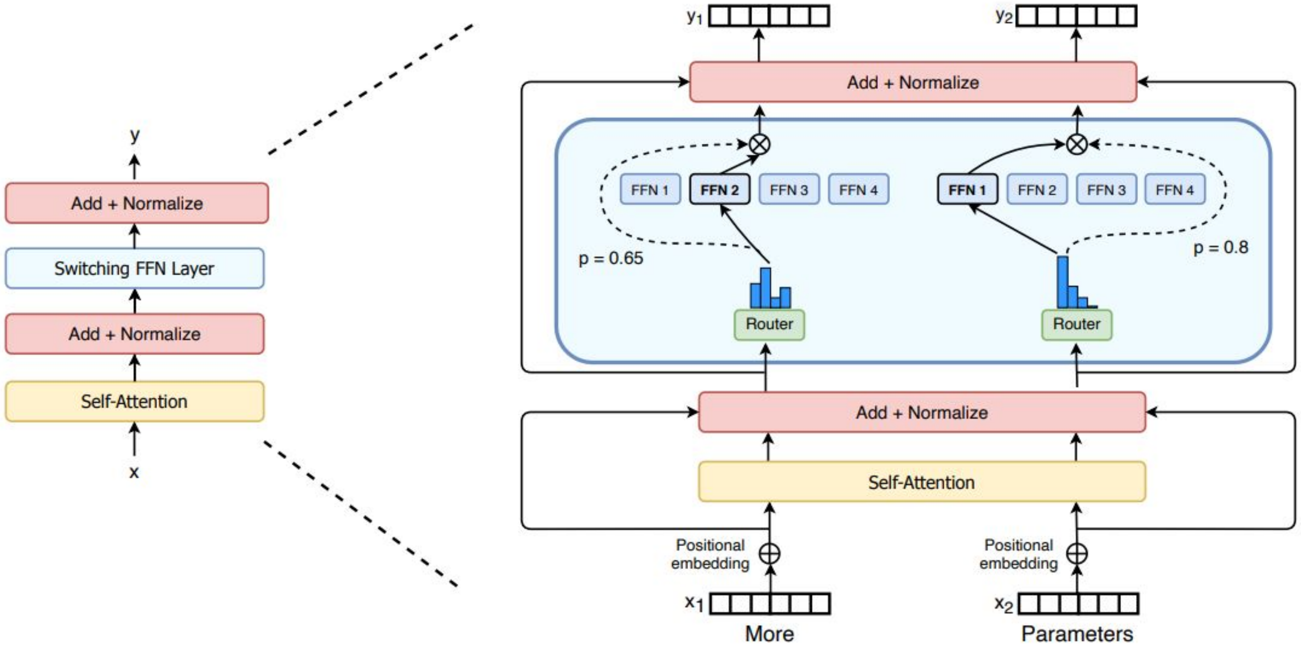


Sliding Window Attention



Effective Context Length

Mixture of Experts (MoE)



Fedus et al. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. https://arxiv.org/abs/2101.03961

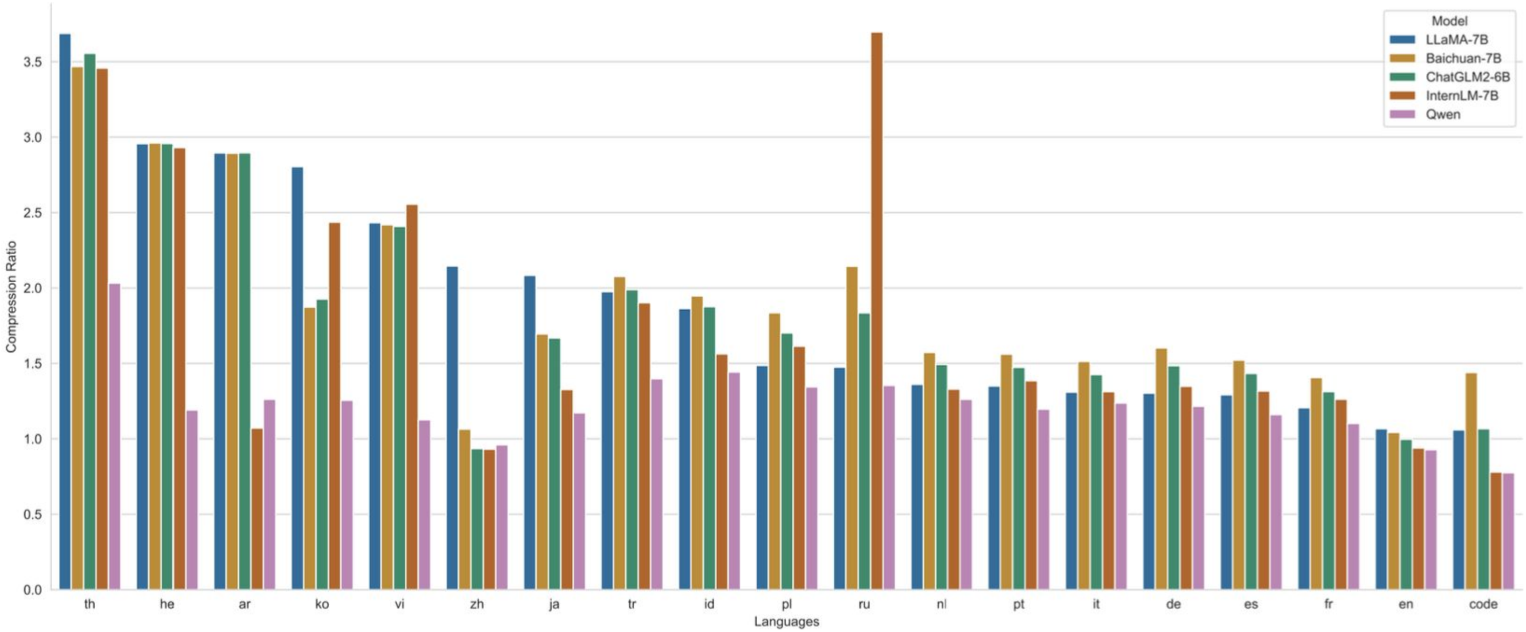
Qwen - Overview

- Creator: Alibaba
- Goal: Strong multilingual (esp. English and Chinese) LM
- Unique features: Large vocabulary for multilingual support, strong performance

Arch	Transformer+RoPE+SwiGLU, context 4k, RMSNorm, bias in attention layer
Data	Trained on multilingual data + instruction data at pre-training time, 2-3T tokens
Train	3e-4, batch size 4M tokens

Qwen - Multilinguality

- Token compression ratio re: XLM-R (lower is better)



GPT-4o - Overview

- Creator: OpenAI
- De-facto standard “strong” language model
- Tuned to be good as a chat-based assistant
- Supports calling external tools through “function calling” interface
- Accepts image inputs
- Fast and cheaper inference compared with earlier GPT-4 versions

Gemini - Overview

- Creator: Google DeepMind
- Performance competitive with corresponding GPT models (Gemini Pro 1.0 ~ gpt-3.5, Gemini Ultra 1.0 ~ gpt-4)
- Pro 1.5 supports very long inputs, 1-10M tokens
- Supports image and video inputs
- Can generate images natively

Claude 3 - Overview

- Creator: ANTHROP\C
- Context window up to 200k
- Allows for processing images
- Overall strong results competitive with GPT-4

DeepSeek-r1 (Open Model)

- Creator: DeepSeek AI
- Goal: Efficient, scalable LLM with Mixture of Experts (MoE) and Group Relative Policy Optimization (GRPO)
- Unique Features: 671B total params (37B active), 16K+ context, optimized inference

Arch	Transformer + GQA, SwiGLU, RoPE, MoE, GRPO
Data	Trained on 2T+ tokens, strong Chinese & English support

Any Questions?