ECE7115 ~~Multimodal VLM~~ LLM

12. Dataset and SFT

Spring 2026

Namhyuk Ahn, Inha University

Last Week: Inference

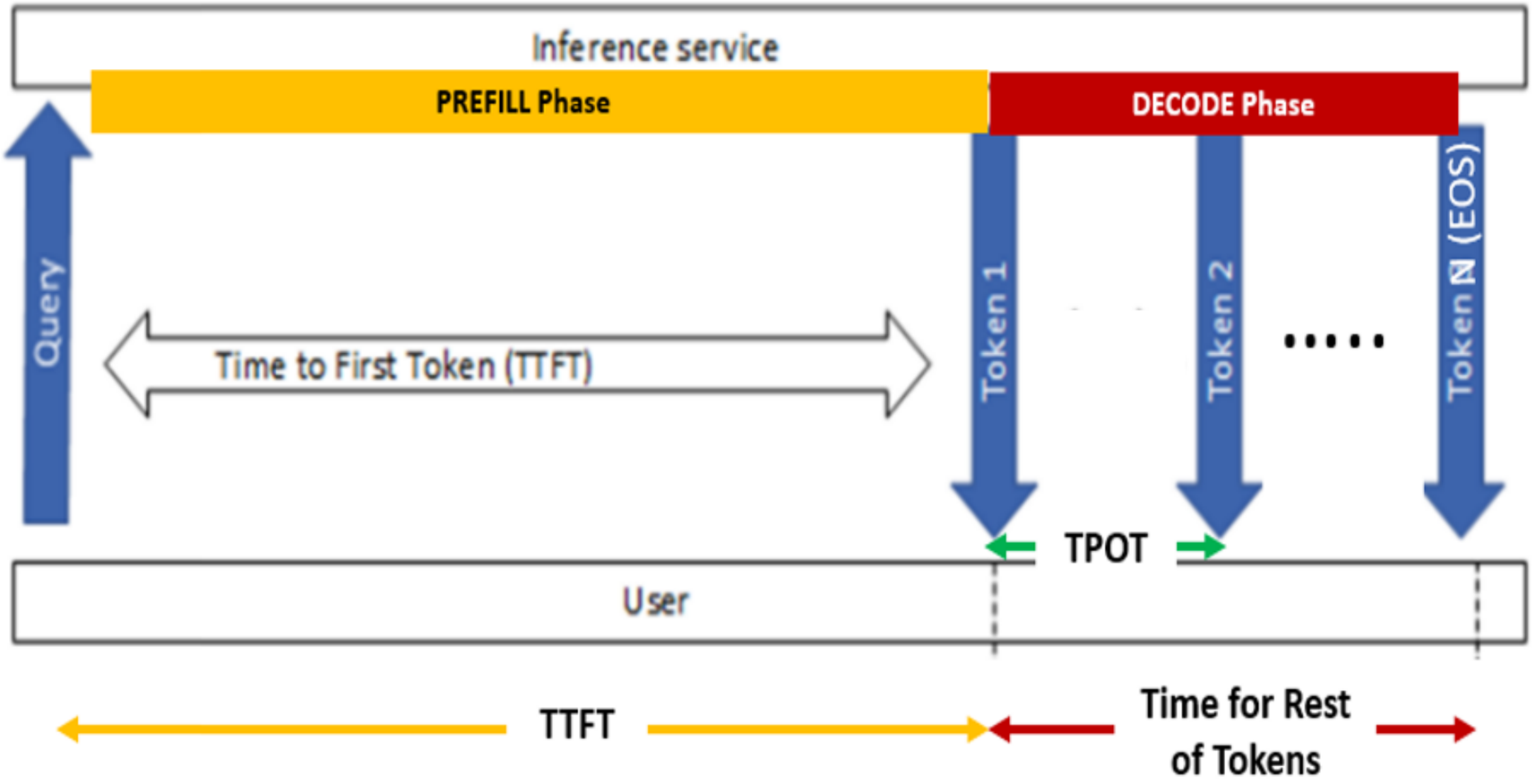
- Prefill: process N input prompt tokens at once
  - Large GEMM (GEneral Matrix Multiplication; matrix × matrix)
  - Fills the KV cache in a single forward pass
  - Compute-bound

- Decode: generate tokens one by one auto-regressively
  - Repeated small GEMV (matrix × vector)
  - Reads all weights at every step
  - Memory-bound

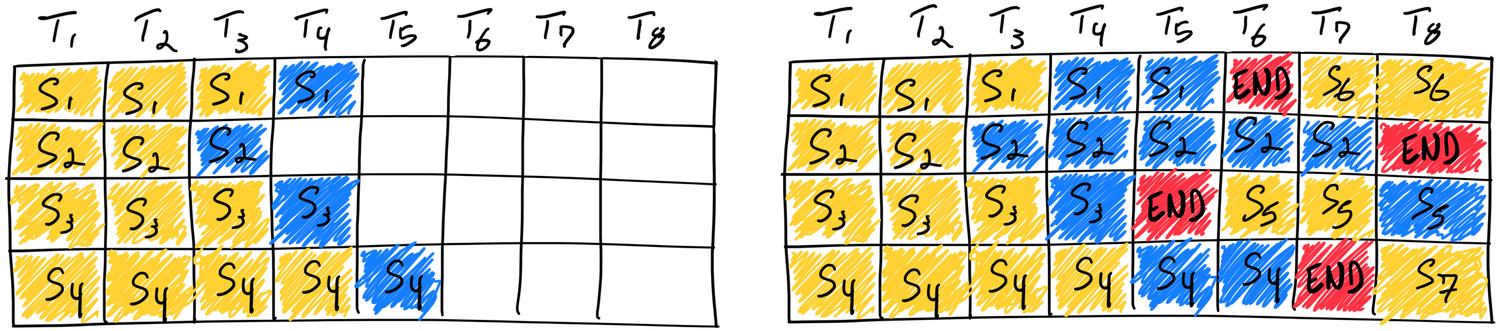
- It's the same model, but the hardware workload is completely different

Last Week: Inference

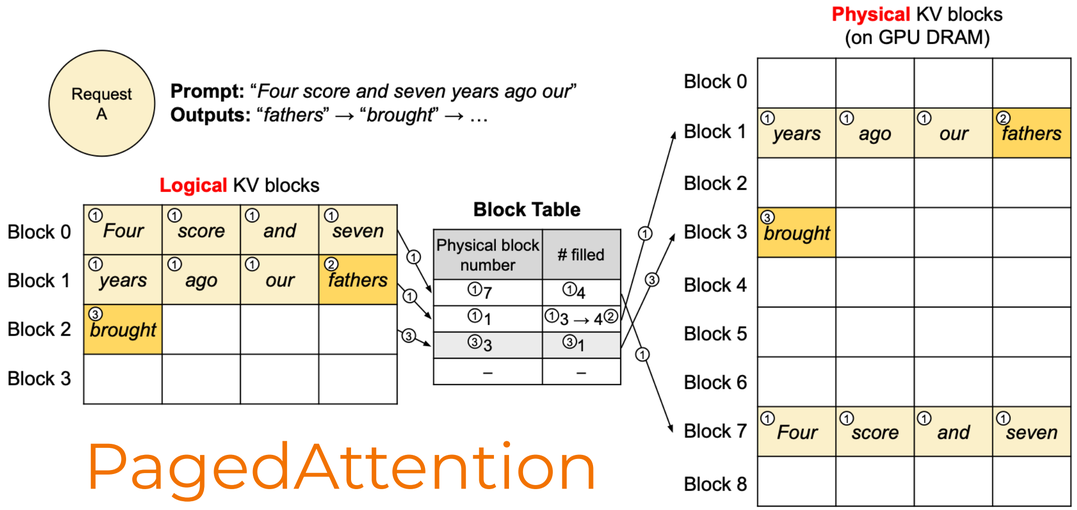
- Inference metrics



Last Week: Inference

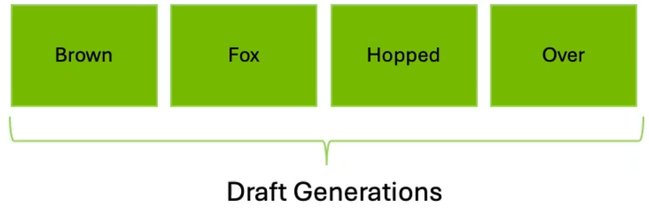


Continuous batching

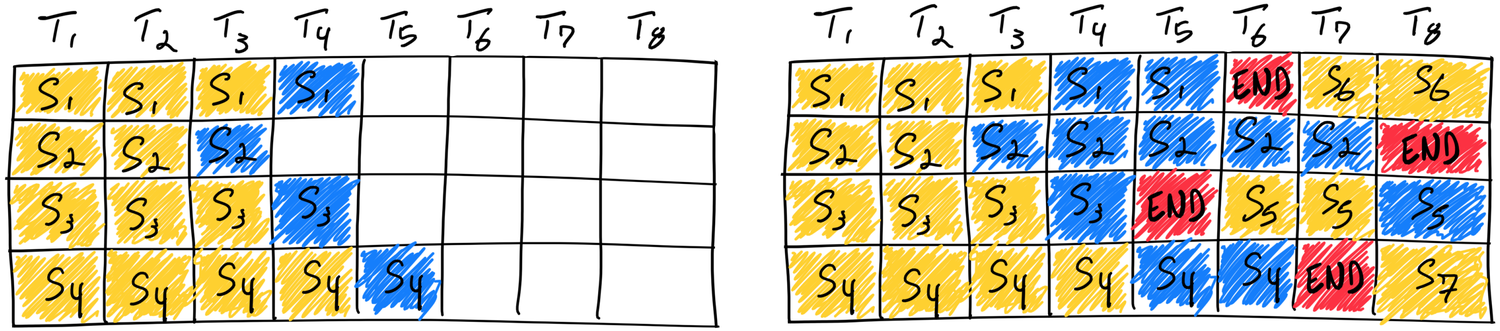


PagedAttention

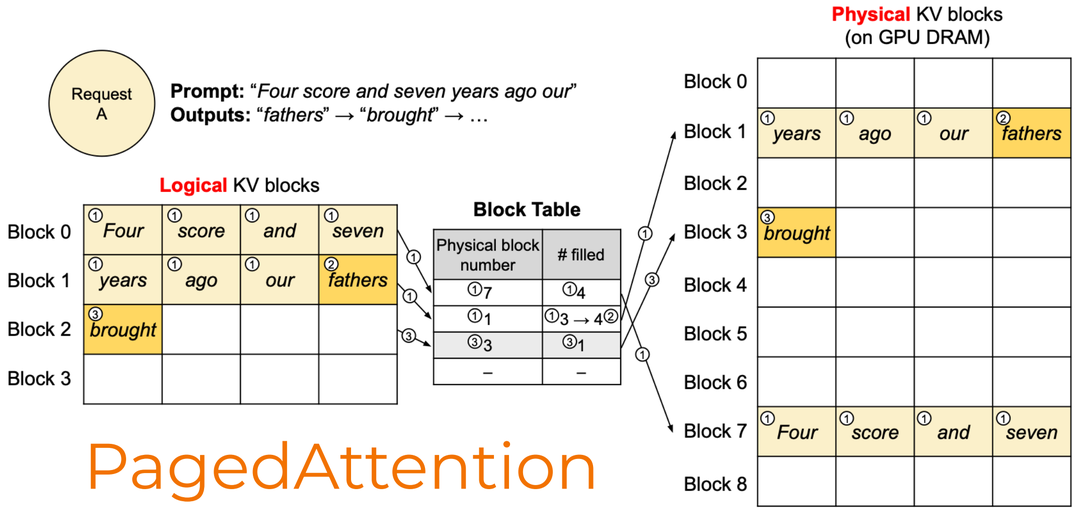
Speculative decoding



Last Week: Inference

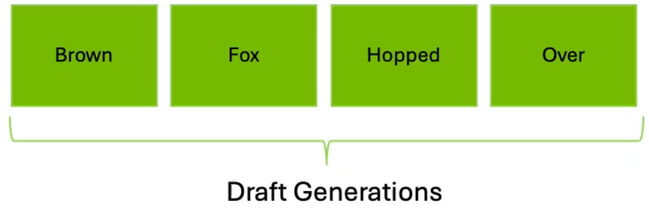


Continuous batching



PagedAttention

Speculative decoding



Last Week: Inference

Technique	TTFT	TPOT/ITL	Throughput	Goodput	$/tokens
Continuous batching	+	+	++	++	+
PagedAttention	+	+	++	++	++
Prefix caching	++	0/+	+	+	+
Chunked prefill	±	++	+	++	+
P/D disaggregation	++	++	+/++	++	±/+
Speculative decoding	0/+	++	++	++	+/++
Quantization	+	+/++	+/++	+	++

Last Week: Evaluation

- Evaluation might appear to be a mechanical process:
  - 1. Define some evaluation prompts
  - 2. Send prompts to a model and get back responses
  - 3. Compute accuracy

- But actually, evaluation is a deep and important topic…
  - which shapes the development of AI
  - Core challenge: abstract construct → concrete metric

Recap: SFT

PROMPT	*Explain the moon landing to a 6 year old in a few sentences.*
COMPLETION	**Human** **A giant rocket ship blasted off from Earth carrying astronauts to the moon. The astronauts landed their spaceship on the moon and walked around exploring the lunar surface. Then they returned safely back to Earth, bringing home moon rocks to show everyone.**

- Collect prompt and completion (response) set
- Now, the prompt is an input X and completion is a target Y
- With (X, Y), fine-tune the pre-trained LLM with NTP loss

Recap: Post-Training

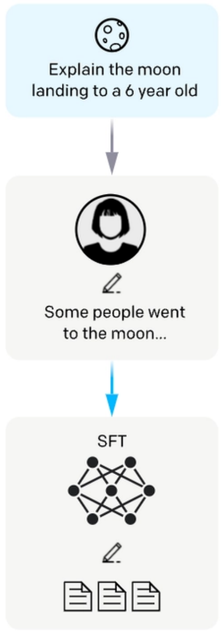
Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.



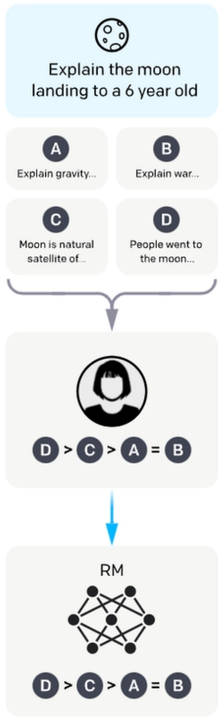
Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

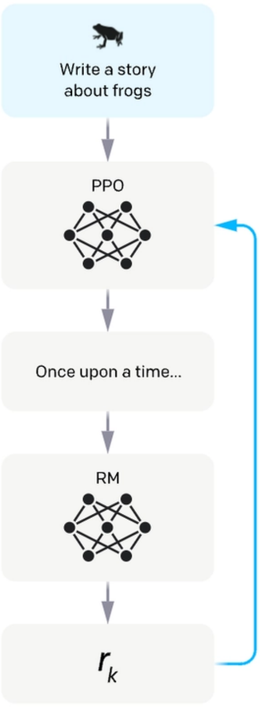
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



The Importance of Dataset

- Now, the question is: what data should we train on?

- Data is the most important thing to get right in training LLM
  - But many people underrate the importance of the dataset
  - One justification: let's see what companies disclose
    - Open models (e.g. LLaMA 3) have full transparency into the architecture and even training procedures
    - But basically no information on dataset

> **3.1 Pre-Training Data**
>
> We create our dataset for language model pre-training from a variety of data sources containing knowledge until the end of 2023. We apply several de-duplication methods and data cleaning mechanisms on each data source to obtain high-quality tokens. We remove domains that contain large amounts of personally identifiable information (PII), and domains with known adult content.

The Importance of Dataset

- Another recent example: DeepSeek-V4
  - In the 58-page technical report, they provide details of their architectural choices, training setups, infrastructure, …
  - But no details on the dataset!

> **4.1. Data Construction**
>
> On top of the pre-training data of DeepSeek-V3, we endeavor to construct a more diverse and higher-quality training corpus with longer effective contexts. We continually refine our data construction pipelines. For web-sourced data, we implement filtering strategies to remove batched auto-generated and templated content, thereby mitigating the risk of model collapse (Zhu et al., 2024). Mathematical and programming corpora still remain core components of our training data, and we further enhance the coding capabilities of DeepSeek-V4 series by incorporating agentic data during the mid-training phase. For multilingual data, we build a larger corpus for DeepSeek-V4, improving its capture of long-tail knowledge across different cultures. For DeepSeek-V4, we place a particular emphasis on long-document data curation, prioritizing scientific papers, technical reports, and other materials that reflect unique academic values. Combining all the above, our pre-training corpus comprises more than 32T tokens, containing mathematical contents, codes, web pages, long documents, and other high-quality categories.

Dataset Secrecy

- Reasons for secrecy? (1) competitive dynamics, (2) copyright liability

- Before foundation models, data work meant heavy annotation of labeled data for supervised learning.
  - Now there's less annotation
  - But there's still a lot of curation and cleaning

- Data is fundamentally a long-tail problem, scales with human effort unlike architectures, systems

Difficulties of Data Access

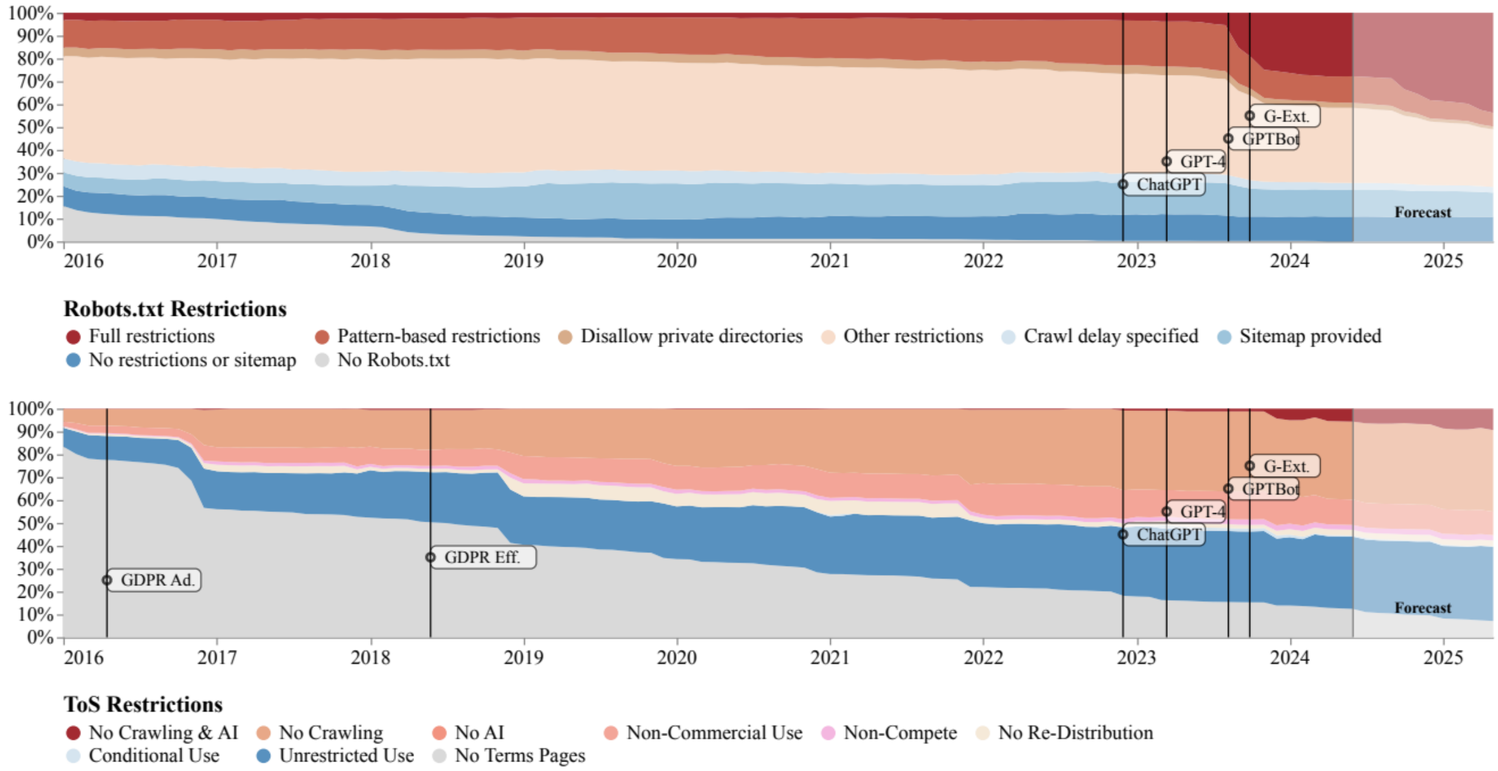
- Internet is huge, but many technical and legal restrictions on what data one can access

- Cases of crawler misbehavior
  - Server load → service degradation / monetary losses for the site

- Shadow Libraries (copyright gray zone)
  - Technically part of the web
  - LibGen (~4M books, 2019), Z-Library, Anna's Archive
  - Sci-Hub (~88M papers, 2022)
  - From a legal perspective, this is piracy and copyright infringement

Difficulties of Data Access

- Decline of consent [Longpre+ 2024]
  - Restrictions grow over time (robots.txt, ToS, etc)
  - So the trainable portion of the open web is shrinking



Copyright

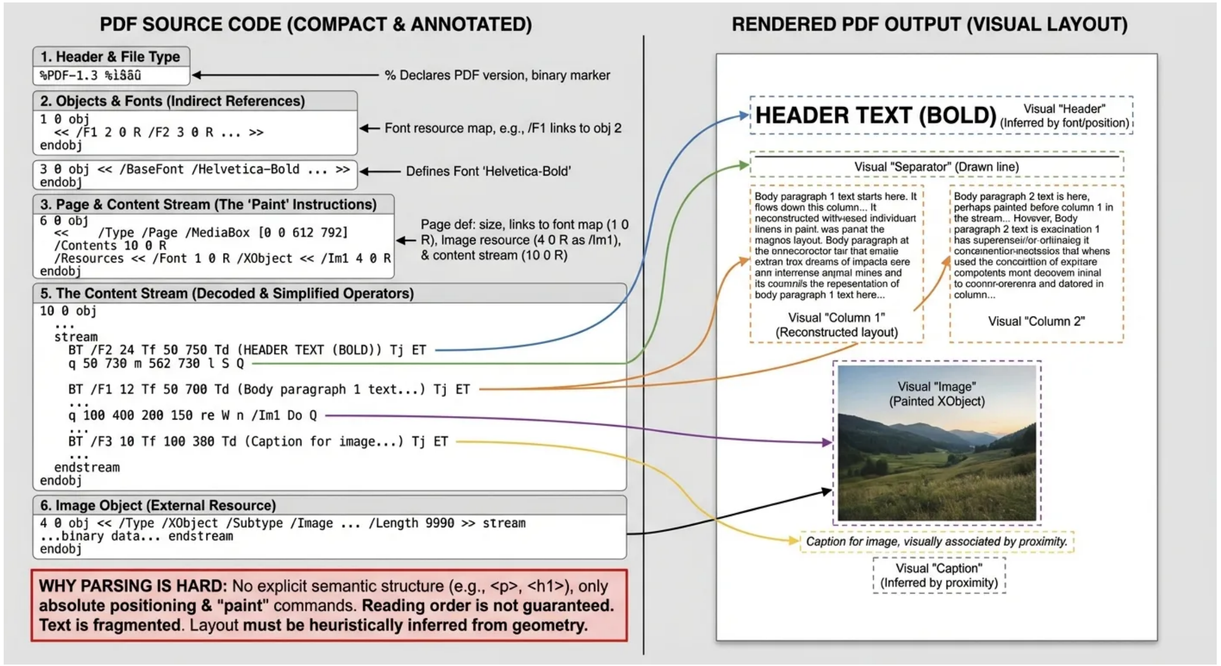
- What data is legal to train on? Publicly available does not mean free to use and most Internet data is copyrighted

- Three legal ways to use copyrighted works
  - (1) Get a license (2) use public-domain dataset (3) claim fair use
  - Model training can be a violation from the very first step…

- Licenses
  - Creative commons, …
  - Model companies make data-licensing deals (e.g. Google - Reddit)

Transformation

- Raw data does not come as text (e.g. HTML, PDF, codebase, etc)
- HTML to text
  - Remove boilerplate (e.g. navigation, ads) and extract content
  - But what about images, tables, etc.?
- PDF to text
  - OCR is an important tool



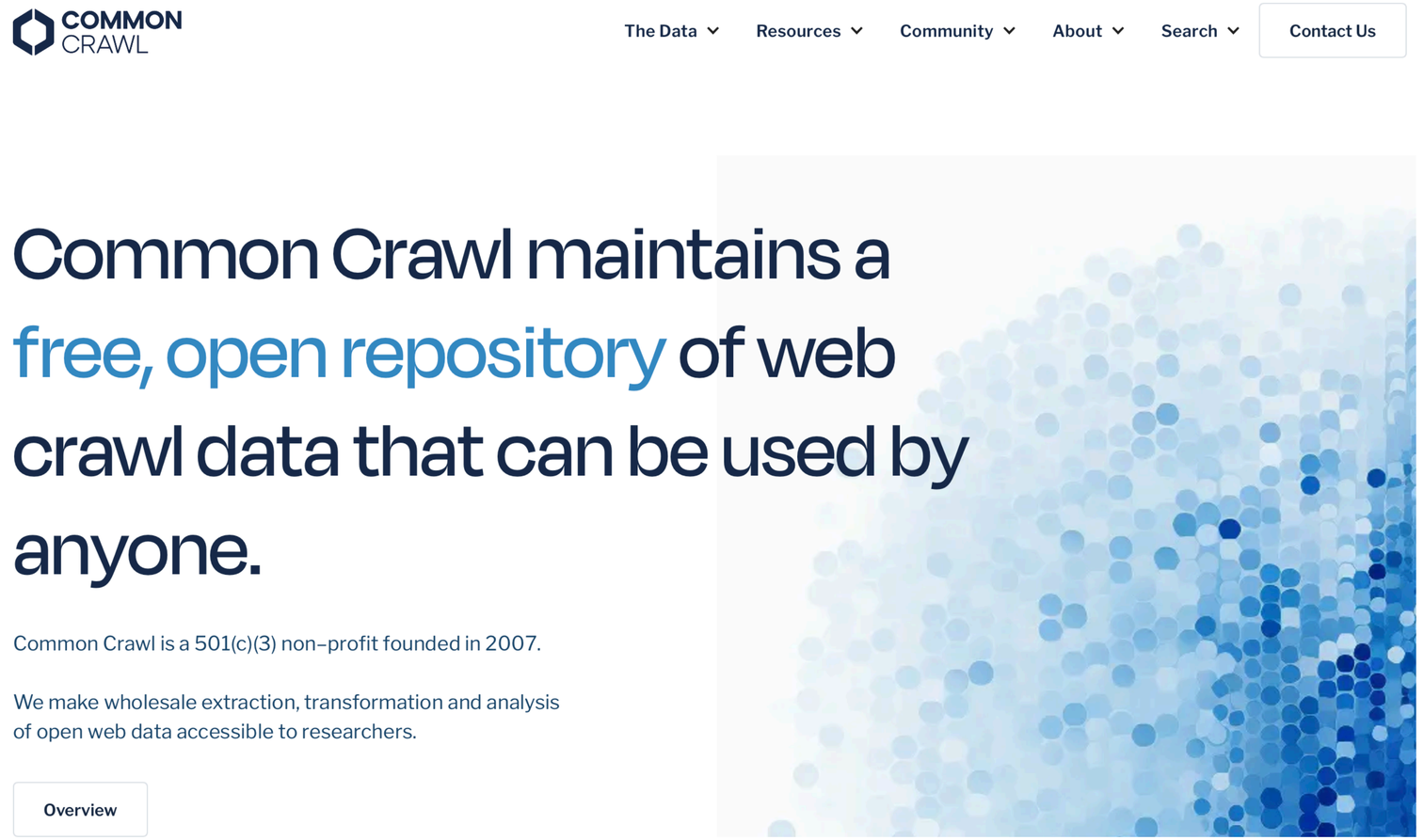
Early Dataset (~2019)

- BERT [Devlin+ 2018]: Wikipedia + BooksCorpus
  - BooksCorpus: 7K self-published Smashwords books, 985M words
  - Later removed for Smashwords terms-of-service violation

- GPT-2 [Radford+ 2019] : 8M pages from Reddit links with karma ≥3
  - 8 million pages, 40GB text
  - OpenWebTextCorpus [Gokaslan+ 2019] is an open-source replication attempt by extracting the URLs from the Reddit dataset

Common Crawl

- Common Crawl is a non-profit organization founded in 2007



Common Crawl

- Every ~month, run a web crawl (add 3-5 billion web pages)
- Crawls have some overlap but try to diversify
  - 300 billion pages so far

> The data was crawled between April 10th and April 23rd, and contains 2.19 billion web pages (or 379.2 TiB of uncompressed content). Page captures are from 43.2 million hosts or 35.4 million registered domains and include 660.5 million new URLs, not visited in any of our prior crawls.

- How many URLs are there?
  - April 2026 Crawl has 2.19 billion pages (379.2 TB)

- Two formats:
  - WARC: raw HTTP response (e.g., HTML)
  - WET: converted to text (lossy process)

Some Specialized Sources

- Wikipedia: 67M articles / 361 language editions
  - Regular dumps (weekly) → no crawling needed
  - Does not contain original thought (no opinions, promotions, personal web pages, etc)

- GitHub: 420M repos (28M public). 28.8M source files
  - Code is helpful for programming tasks, but also for reasoning

- arXiv: ~3M submissions, metadata is CC0

Filtered Dataset

- CCNet [Wenzek+ 2019]
  - Goal: constructing large, high-quality datasets for pre-training
  - Especially interested in getting more data for low-resource languages
  - Components
    - Deduplication: remove duplicate paragraphs based on light normalization
    - Language identification: run language ID fastText classifier; keep only target language (e.g. English)
    - Quality filtering: keep documents that look like Wikipedia under a KenLM 5-gram model
  - Results
    - Trained BERT models, CCNet(CommonCrawl) outperforms Wikipedia

Filtering

- Given some target data T and lots of raw data R, find subset T' of R similar to T (T is normally high-quality dataset; e.g. Wikipedia)
  - e.g. KenLM: n-gram based LM / FastText: simple classifier

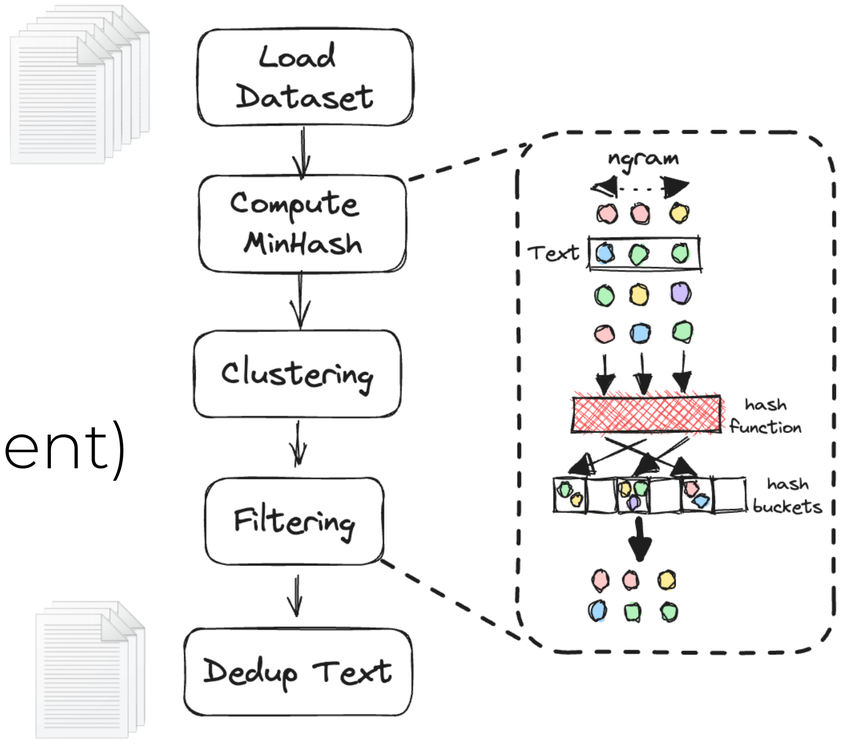
- Applications
  - Language identification
  - Quality filtering
  - Toxic filtering

Deduplication

- Two kinds of duplicates: exact duplicates and near duplicates
  - Example: ToS/license text appears 60K times in C4

- Why deduplication?
  - Training efficiency ↑ and memorization ↓

- Design decisions
  - Item granularity (sentence/paragraph/document)
  - Matching method (exact/substring/fraction)
  - Action (remove all / keep one)



Filtered Dataset

- Colossal Clean Crawled corpus (C4) [Raffel+ 2019]
  - Observation: Common Crawl is mostly not useful natural language
  - Manual heuristics such as
    - Keep lines that end in punctuation and have >= 5 words
    - Remove page with fewer than 3 sentences
    - Removed page that contains any 'bad words'
    - Removed page with '{' (no code), 'lorem ipsum', 'terms of use', etc.
    - Filter out non-English text using langdetect

  - 2019.4 CC dumps 1.4T → 806GB (156B tokens)

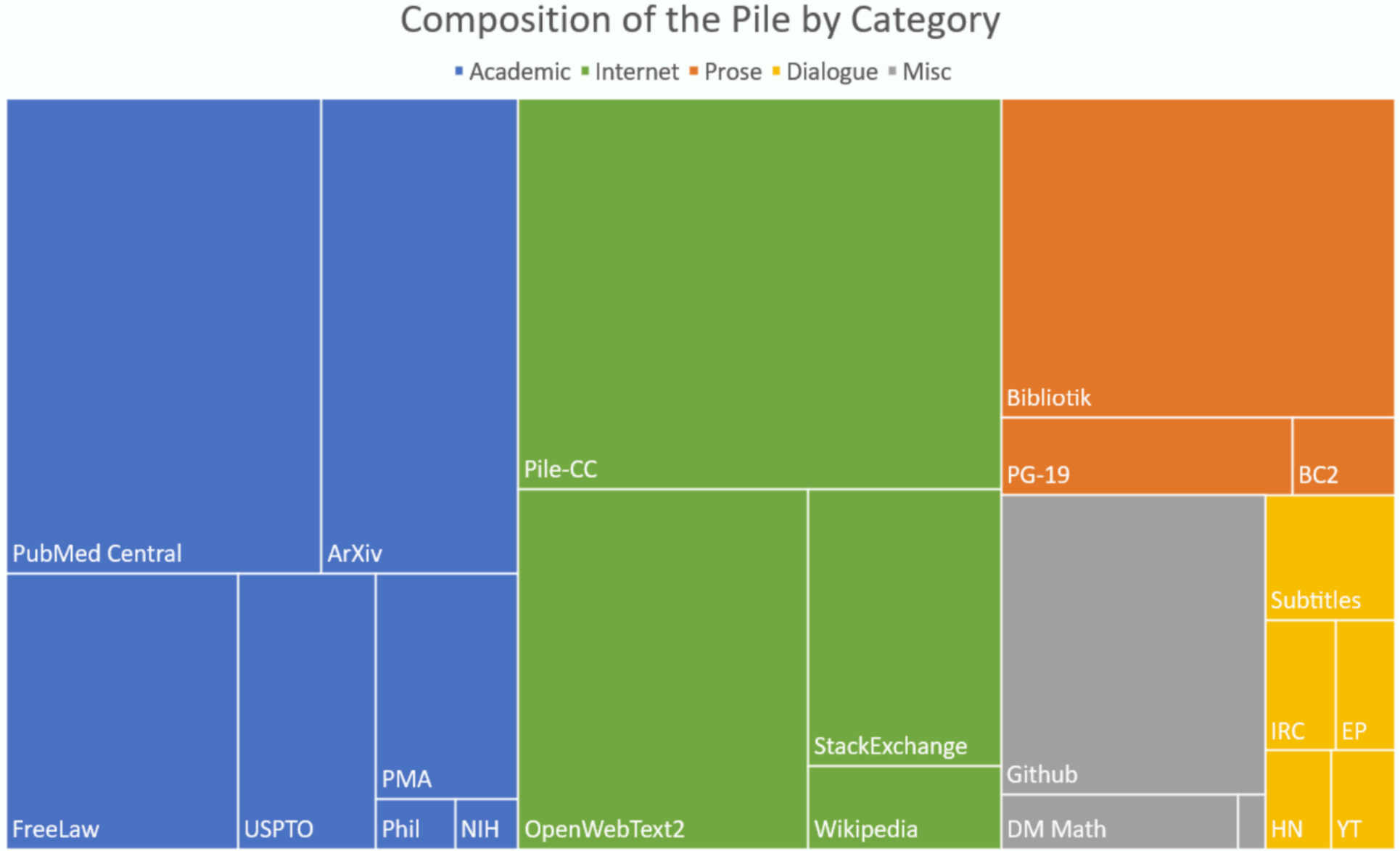
The Pile [Gao+ 2020]

- 825GB (~275B tokens), 22 curated domains

- Major components
  - Pile-CC, PubMed (5M papers), arXiv (1991+)
  - Enron emails (500K), Project Gutenberg (75K public-domain books)
  - Books3 (196K, shadow library → removed due to copyright issues)
  - StackExchange, GitHub (28M+ repos)

The

- 825
- Ma
  - F
  - E
  - E
  - S



oks)

)

Gopher / LLaMA

- Gopher [Rae+ 2021]: 10.5TB of text (only 300B tokens used for training)
  - Quality filtering using manual rules (not classifier)
    - e.g. 80% words contain at least one alphabetic character

- LLaMA [Touvron+ 2023]: 1.2T tokens
  - CCNet + C4 + GitHub (permissive licenses) + Wikipedia (20 languages) + Gutenberg + Books3 + arXiv + StackExchange

RefinedWeb / FineWeb

- RefinedWeb [Penedo+ 2023]: 600B released from 5T
  - "Web data alone is enough"
  - HTML to text + Gopher rules filtering + deduplication
  - Deliberately avoids ML-based filtering (to prevent bias)

- FineWeb: Started as a replication of RefinedWeb, but improved it
  - 95 Common Crawl dumps → 15T tokens
  - MinHash dedup, Gopher filtering, email, IP anonymization…

Dolma / DCLM

- Dolma [Soldaini+ 2024]
  - Language identification (fastText classifier)
  - Gopher, C4 rules filtering
  - Bloom filter dedup
  - Jigsaw toxicity filter

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**

- DataComp-LM (DCLM) [Li+ 2024]
  - Processed CommonCrawl to produce DCLM-pool (240T tokens)
  - DCLM-baseline: filtered down DCLM-pool using quality classifier

Dolma / DCLM

- Dolma [Soldaini+ 2024]
  - Language identification (fastText clas
  - Gopher, C4 r
  - Bloom filter (
  - Jigsaw toxici
- DataComp-LM
  - Processed C
  - DCLM-basel

Source	Doc Type	UTF-8 bytes (GB)	Documents (millions)	Unicode words (billions)	Llama tokens (billions)
Common Crawl					

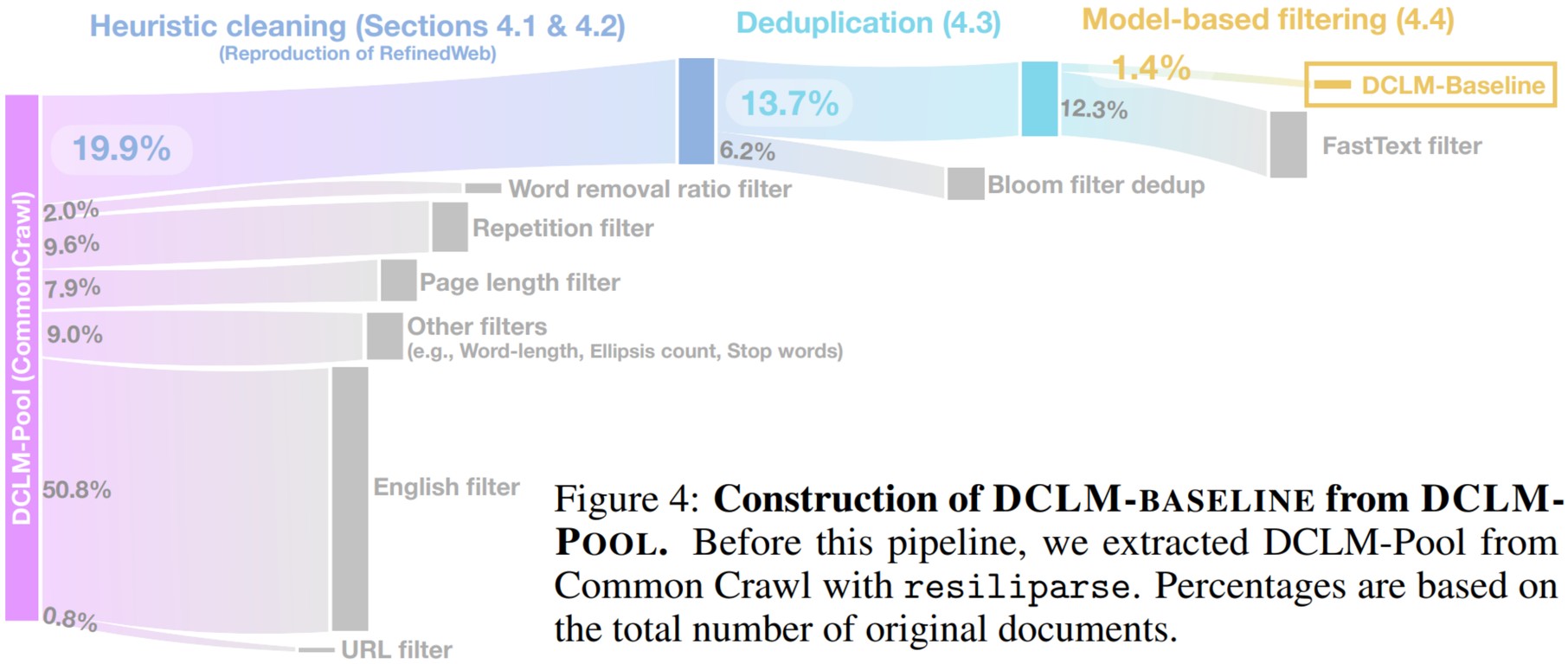
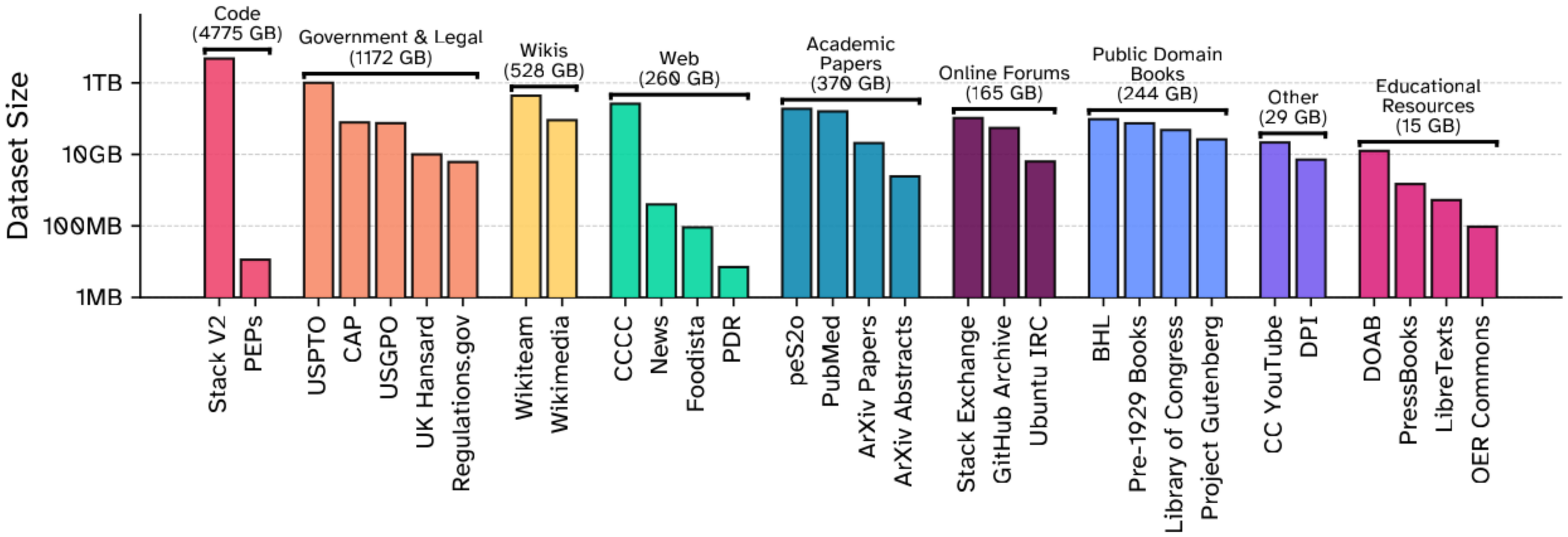


Figure 4: **Construction of DCLM-BASELINE from DCLM-POOL.** Before this pipeline, we extracted DCLM-Pool from Common Crawl with `resiliparse`. Percentages are based on the total number of original documents.

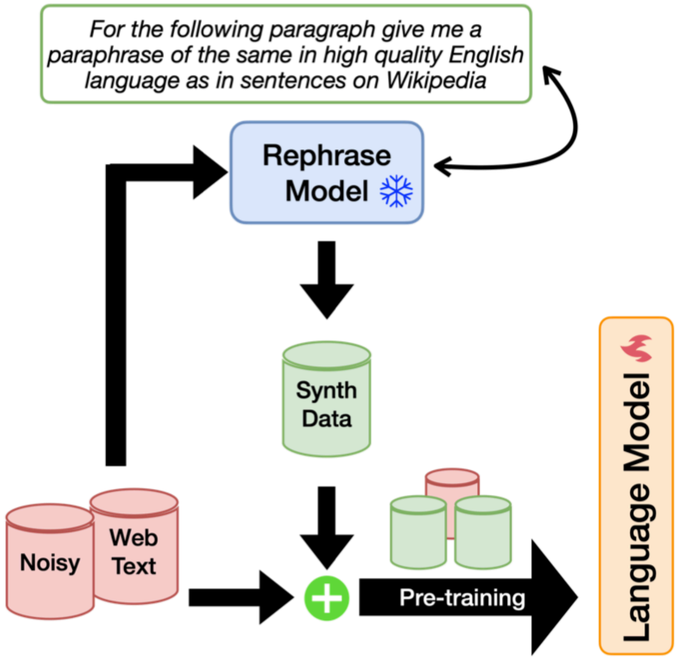
Common Pile [Kandpal+ 2025]

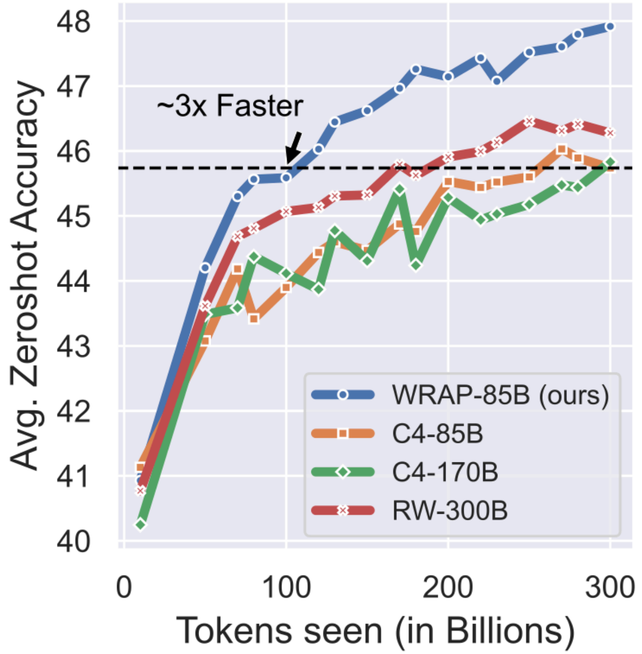
- Can you train a good model using only permissively-licensed data?
- Common Pile collects 8TB dataset of permissively licensed data

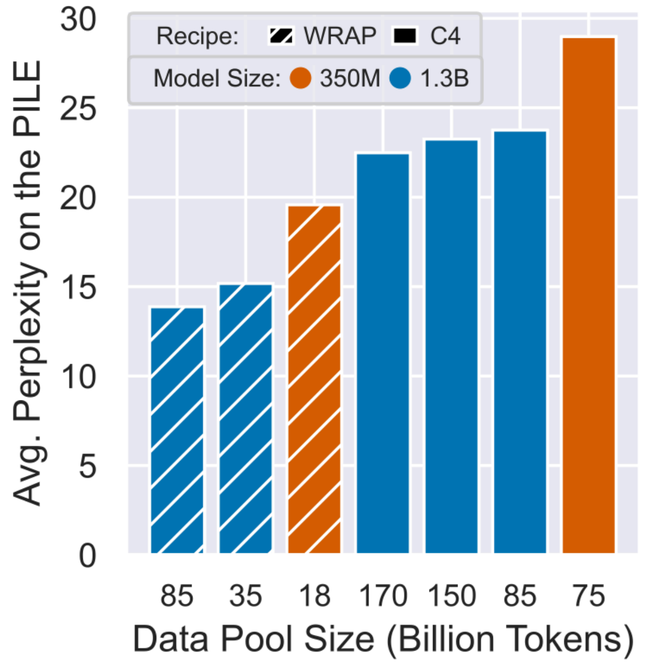


Synthetic Dataset

- Another dataset approach is synthetic pre-training dataset
- WRAP [Maini+ 2024] (CMU and Apple)
  - An instruct LLM rephrases web docs into Wikipedia/QA style







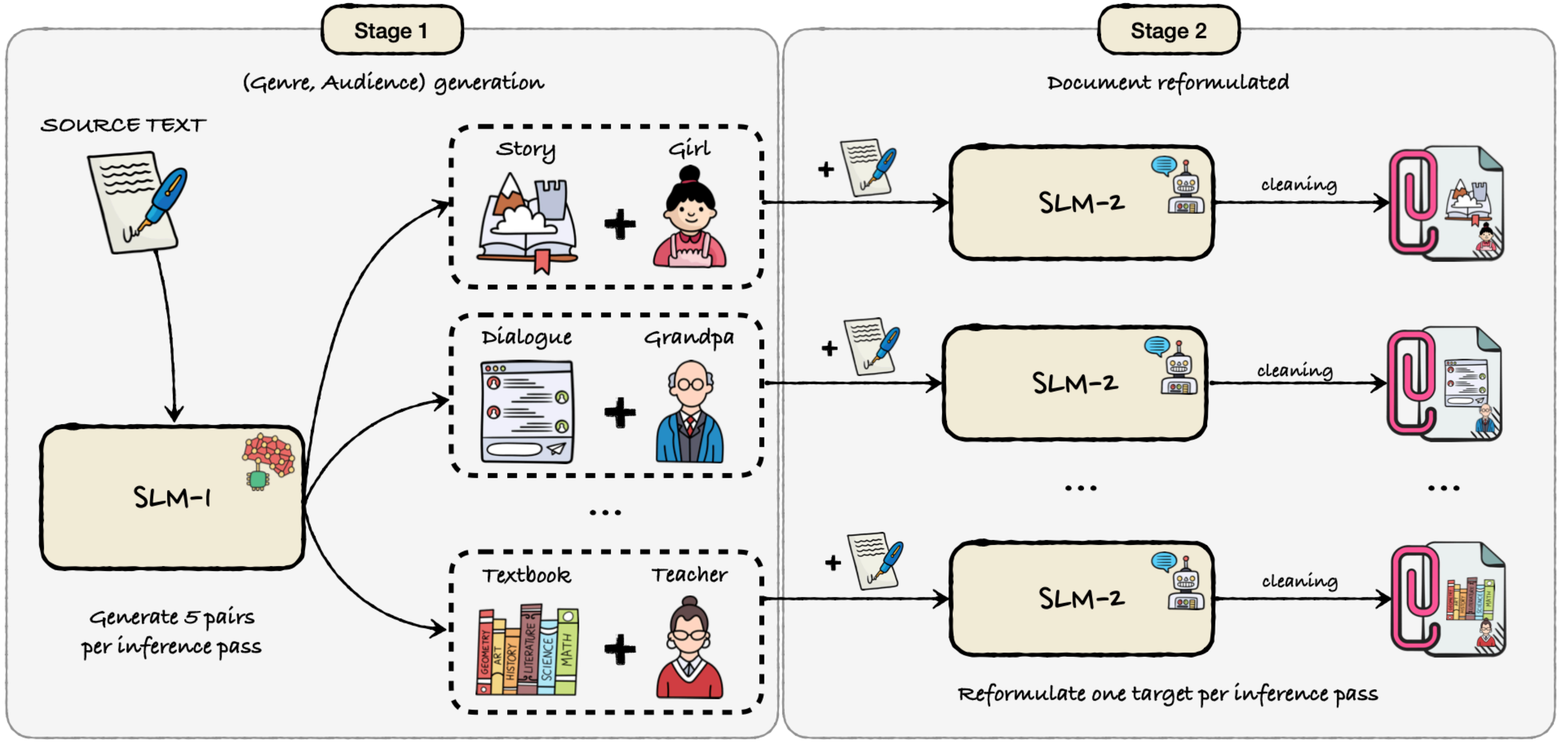
Synthetic Dataset

- Another dataset approach is synthetic pre-training dataset
- Phi-4 [Abdin+ 2024] (Microsoft)
  - Curated organic seeds → (LLM-driven) synthetic rewrite & augmentation → LLM-driven quality validation

Data Source	Fraction of Training	Unique Token Count	Number of Epochs
Web	15%	1.3T	1.2
Web rewrites	15%	290B	5.2
Synthetic	40%	290B	13.8
Code data	20%	820B	2.4
Acquired sources	10%	580B	1.7

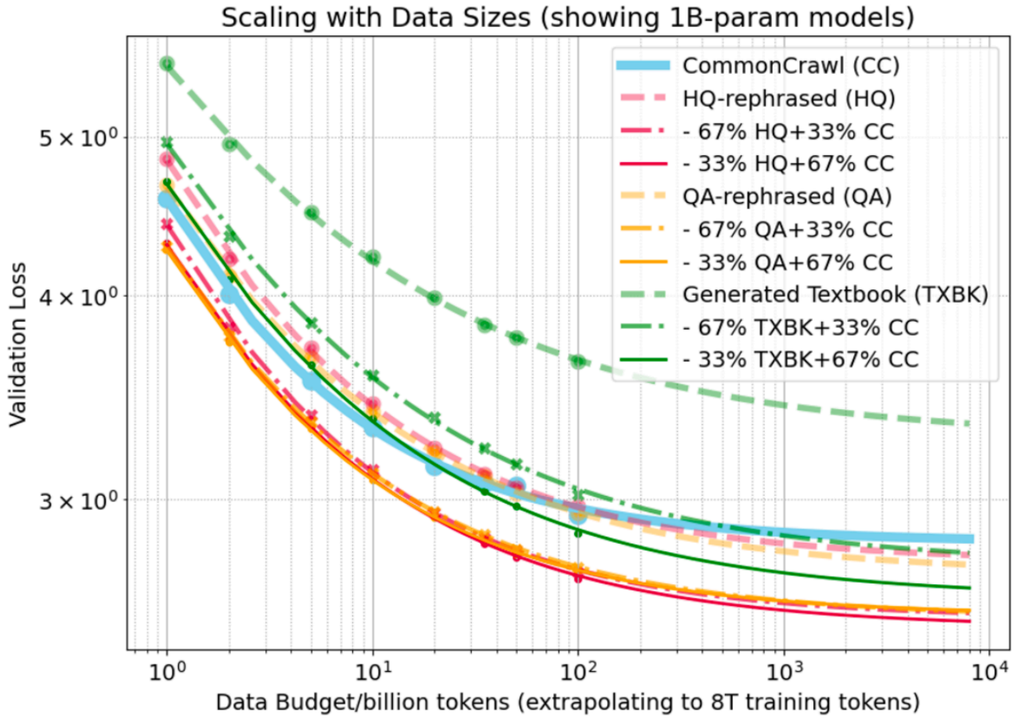
Synthetic Dataset

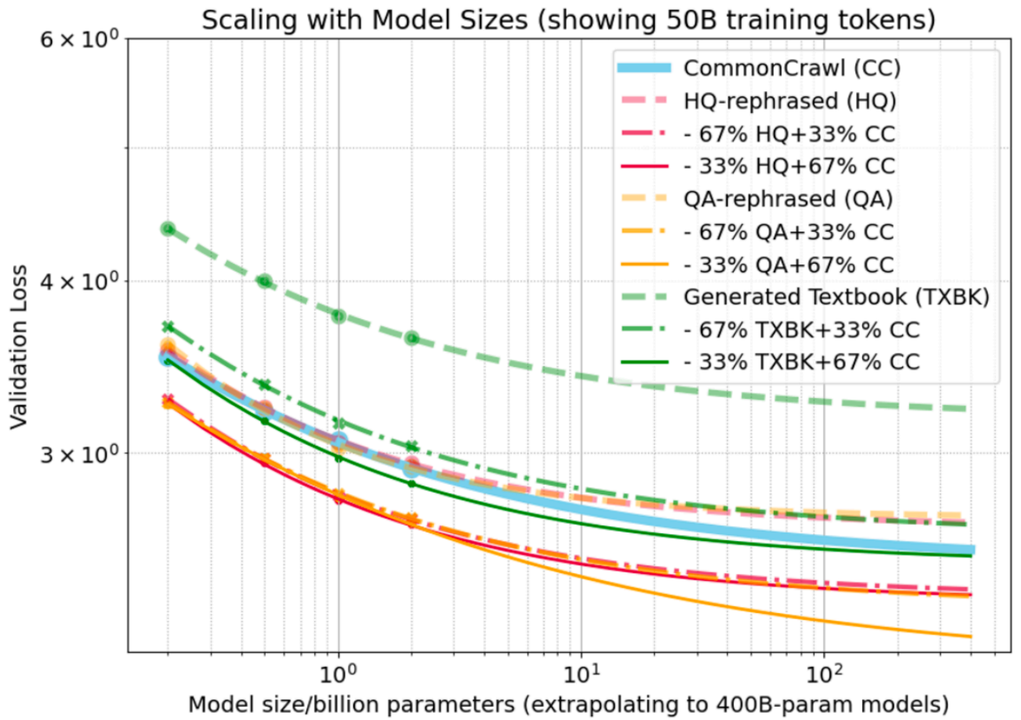
- Another dataset approach is synthetic pre-training dataset
- MGA [Hao+ 2025] (ByteDance)
  - Massive genre-audience reformulation from the original text



Synthetic Dataset

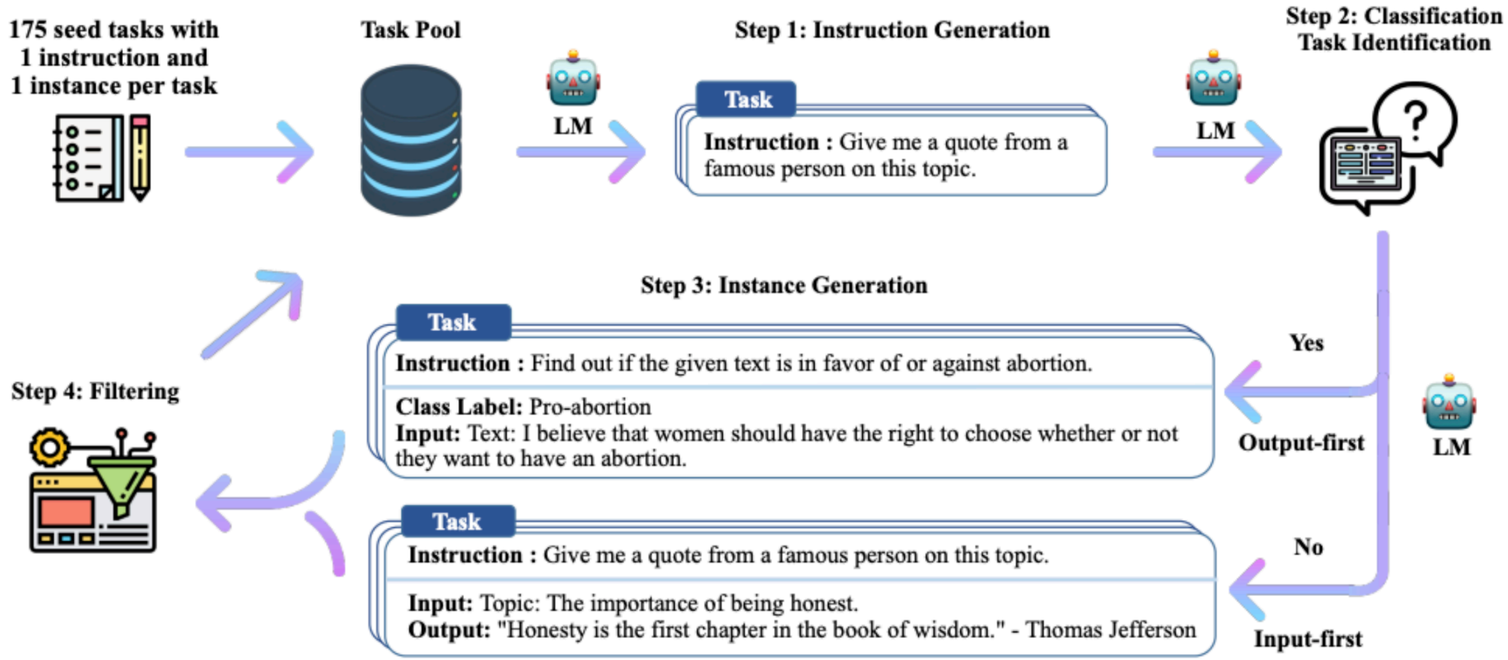
- Another dataset approach is synthetic pre-training dataset
- Demystifying synthetic data [Kang+ 2025] (Meta)
  - Scaling laws on synthetic dataset





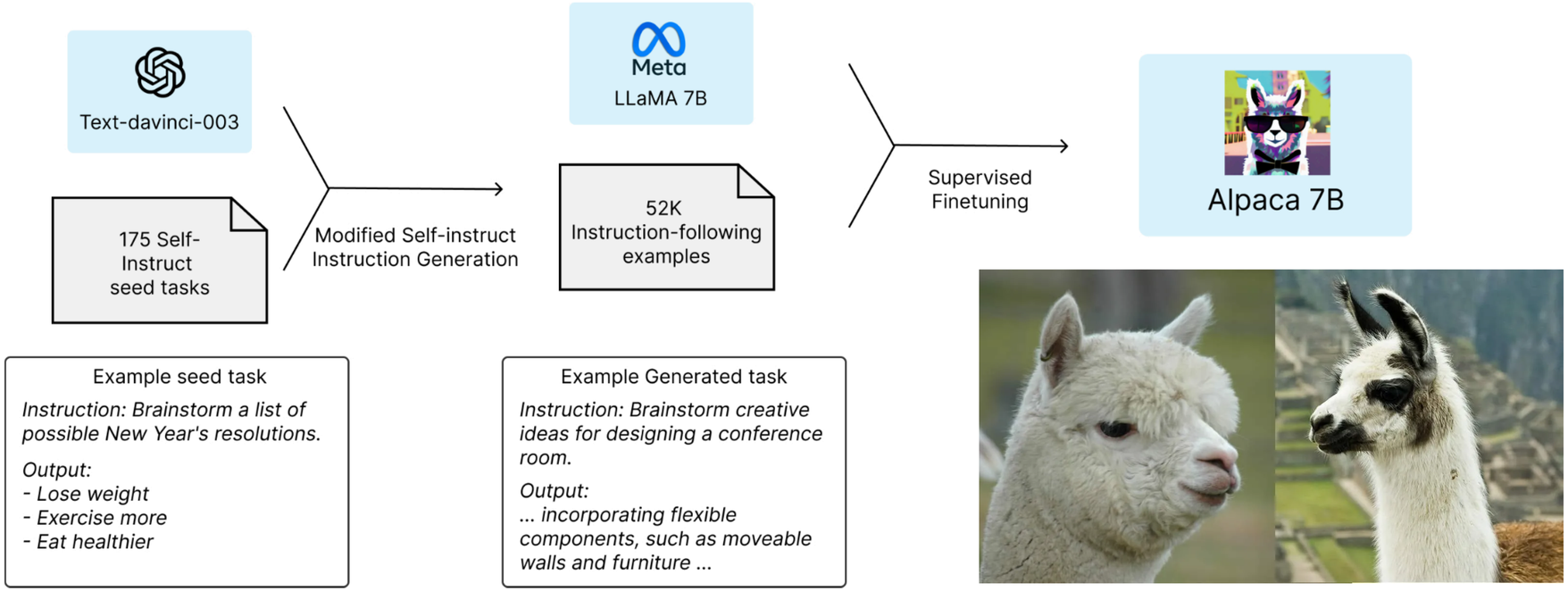
Synthetic SFT Dataset

- Large-scale instruction collection is expensive; Can LMs produce it?
- Self-instruction [Wang+ 2022]
  - Creating a large-scale instruction dataset is really difficult
  - It is possible to automatically generate instruction tuning datasets



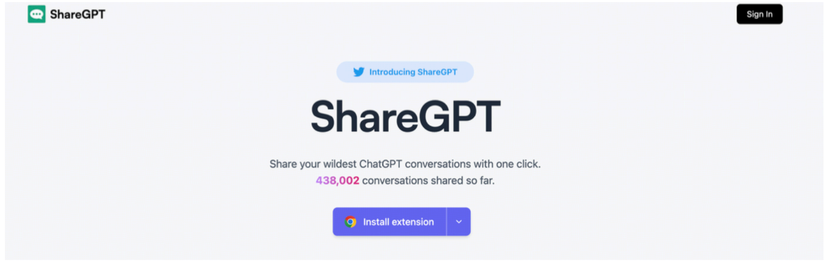
Synthetic SFT Dataset

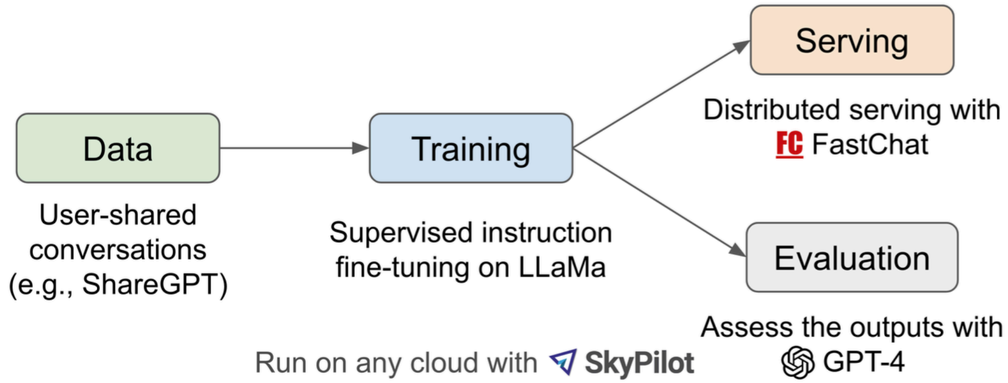
- Large-scale instruction collection is expensive; Can LMs produce it?
- Alpaca [Taori+ 2023]
  - Generating high-quality instruction dataset with self-instruct using OpenAI text-davinci-003 (i.e. knowledge distillation)



Synthetic SFT Dataset

- Large-scale instruction collection is expensive; Can LMs produce it?
- Vicuna [Chiang+ 2023]
  - Fine-tune LLaMA with 70K user-shared ChatGPT conversations





Stages of Training

- 1. Pre-training: train on raw text (e.g. documents from the web)
- 2. Mid-training: train more on high quality data to enhance capabilities
- 3. Post-training: train on chat transcripts or reinforcement learning

- In practice, the lines are blurry and there could be more stages
  - But the basic trend is throughout training, we go from large amounts of lower quality data to small amounts of high quality data

- Base model: after pre-training + mid-training
- Instruct/chat model: after post-training

Stages of Training

LLaMA 3

GPUs	TP	CP	PP	DP	Seq. Len.	Batch size/DP	Tokens/Batch	TFLOPs/GPU	BF16 MFU
8,192	8	1	16	64	8,192	32	16M	430	43%
16,384	8	1	16	128	8,192	16	16M	400	41%
16,384	8	16	16	8	131,072	16	16M	380	38%

small bsize
large bsize
long context

Domain	Stage 1	Stage 2	Stage 3	Stage 4
General	79.4%	61.6%	59.1%	17.0%
Code	12.0%	20.1%	20.0%	25.2%
Math	8.6%	18.2%	20.0%	25.3%
Instruction tuning	0.0%	0.1%	1.0%	32.5%

HCX

DeepSeek-V4

Seems that wrt long-context supporting, the boundary of pre-training and mid-training goes blurry?

DeepSeek-V4-Flash. We employ the Muon optimizer (Jordan et al., 2024; Liu et al., 2025) for the majority of parameters, but use the AdamW optimizer (Loshchilov and Hutter, 2017) for the embedding module, the prediction head module, and the weights of all RMSNorm modules. For AdamW, we set its hyper-parameters to $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\varepsilon = 10^{-20}$, and weight_decay = 0.1. For Muon, we set the momentum to 0.95 and the weight decay to 0.1, and rescale the RMS of each update matrix to 0.18 for reutilization of the AdamW learning rate. We train DeepSeek-V4-Flash on 32T tokens, and as in DeepSeek-V3, we also employ a batch size scheduling strategy that increases the batch size (in tokens) from a small size to 75.5M and then keeps it at 75.5M during most of the training. The learning rate is linearly warmed up in the first 2000 steps, maintained at 2.7×10^{-4} for most of the training. Near the end of the training, we finally decay the learning rate to 2.7×10^{-5} following a cosine schedule. The training starts with a sequence length of 4K, and we gradually extend the training sequence length to 16K, 64K, and 1M. As for the setups of sparse attention, we first warmup the model with dense attention for the first 1T tokens, and introduce sparse attention at the sequence length of 64K and keep sparse attention during the rest of the training. When introducing attention sparsity, we first set a short stage to warm up the lightning indexer in CSA, and then train the model with sparse attention for most of the training. For auxiliary-loss-free load balancing, we set the bias update speed to 0.001. For the balance loss, we set its loss weight to 0.0001 to avoid extreme imbalance within single sequences. The MTP loss weight is set to 0.3 for most of the training, and to 0.1 upon the start of learning rate decay.

Stages of Training [Team OLMo 2025]

Pre-training

Source	Type	Tokens	Words	Bytes	Docs
Pretraining ✦ OLMo 2 1124 Mix					
DCLM-Baseline	Web pages	3.71T	3.32T	21.32T	2.95B
StarCoder filtered version from OLMoE Mix	Code	83.0B	70.0B	459B	78.7M
peS2o from Dolma 1.7	Academic papers	58.6B	51.1B	413B	38.8M
arXiv	STEM papers	20.8B	19.3B	77.2B	3.95M
OpenWebMath	Math web pages	12.2B	11.1B	47.2B	2.89M
Algebraic Stack	Math proofs code	11.8B	10.8B	44.0B	2.83M
Wikipedia & Wikibooks from Dolma 1.7	Encyclopedic	3.7B	3.16B	16.2B	6.17M
Total		**3.90T**	**3.48T**	**22.38T**	**3.08B**

Stages of Training [Team OLMo 2025]

Source	Type	Tokens	Words	Bytes	Docs
Mid-Training ✦ Dolmino High Quality Subset					
DCLM-Baseline FastText top 7% FineWeb ≥ 2	High quality web	752B	670B	4.56T	606M
FLAN from Dolma 1.7 decontaminated	Instruction data	17.0B	14.4B	98.2B	57.3M
peS2o from Dolma 1.7	Academic papers	58.6B	51.1B	413B	38.8M
Wikipedia & Wikibooks from Dolma 1.7	Encyclopedic	3.7B	3.16B	16.2B	6.17M
Stack Exchange 09/30/2024 dump curated Q&A data	Q&A	1.26B	1.14B	7.72B	2.48M
High quality total		**832.6B**	**739.8B**	**5.09T**	**710.8M**
Mid-training ✦ Dolmino Math Mix					
TuluMath	Synthetic math	230M	222M	1.03B	220K
Dolmino SynthMath	Synthetic math	28.7M	35.1M	163M	725K
TinyGSM-MIND	Synthetic math	6.48B	5.68B	25.52B	17M
MathCoder2 Synthetic Ajibawa-2023 M-A-P Matrix	Synthetic Math	3.87B	3.71B	18.4B	2.83M
Metamath OWM-filtered	Math	84.2M	76.6M	741M	383K
CodeSearchNet OWM-filtered	Code	1.78M	1.41M	29.8M	7.27K
GSM8K Train split	Math	2.74M	3.00M	25.3M	17.6K
Math total		**10.7B**	**9.73B**	**45.9B**	**21.37M**

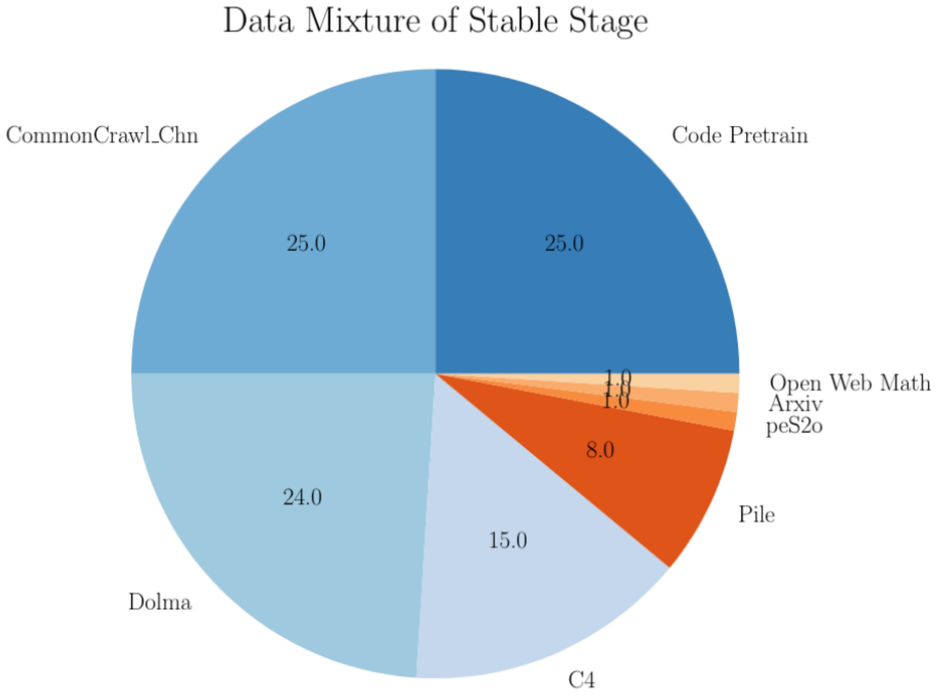
Mid-training

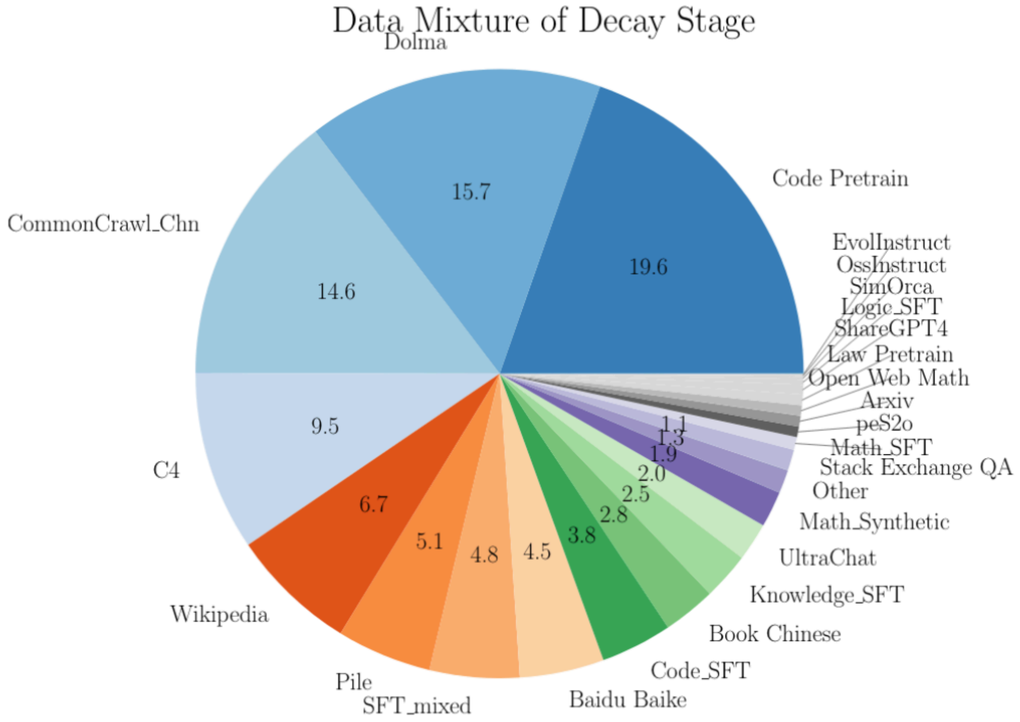
Category	Prompt Dataset	Count	# Prompts used in SFT	# Prompts used in DPO	Reference
General	Tülu 3 Hardcoded[↑]	24	240	–	–
	OpenAssistant[1,2,↓]	88,838	7,132	7,132	Köpf et al. (2024)
	No Robots	9,500	9,500	9,500	Rajani et al. (2023)
	WildChat (GPT-4 subset)[↓]	241,307	100,000	100,000	Zhao et al. (2024)
	UltraFeedback[α,2]	41,635	–	41,635	Cui et al. (2023)
Knowledge	FLAN v2[1,2,↓]	89,982	89,982	12,141	Longpre et al. (2023)
Recall	SciRIFF[↓]	35,357	10,000	17,590	Wadden et al. (2024)
	TableGPT[↓]	13,222	5,000	6,049	Zha et al. (2023)
Math	Tülu 3 Persona MATH	149,960	149,960	–	–
Reasoning	Tülu 3 Persona GSM	49,980	49,980	–	–
	Tülu 3 Persona Algebra	20,000	20,000	–	–
	OpenMathInstruct 2[↓]	21,972,791	50,000	26,356	Toshniwal et al. (2024)
	NuminaMath-TIR[α]	64,312	64,312	8,677	Beeching et al. (2024)
Coding	Tülu 3 Persona Python	34,999	34,999	–	–
	Evol CodeAlpaca[α]	107,276	107,276	14,200	Luo et al. (2023)
Safety	Tülu 3 CoCoNot	10,983	10,983	10,983	Brahman et al. (2024)
& Non-Compliance	Tülu 3 WildJailbreak[α,↓]	50,000	50,000	26,356	Jiang et al. (2024)
	Tülu 3 WildGuardMix[α,↓]	50,000	50,000	26,356	Han et al. (2024)
Multilingual	Aya[↓]	202,285	100,000	32,210	Singh et al. (2024b)
Precise IF	Tülu 3 Persona IF	29,980	29,980	19,890	–
	Tülu 3 IF-augmented	65,530	–	65,530	–
Total		23,327,961	939,344	425,145[γ]	

Post-training

Stages of Training [Hu+ 2024]

- Widely used approach and publicized in recent Chinese LLMs
  - Note: SFT dataset is integrated into the pre-training dataset





Summary

- Key lesson: Data does not fall from the sky. You have to work to get it
  - and data is the key ingredient that differentiates language models

- Standard tools
  - Filtering: KenLM / fastText / DSIR
  - Deduplication: Bloom Filter / MinHash / LSH
  - Much of this pipeline is heuristic, many opportunities to improve

- Recently many models use synthetic dataset
- Beware of legal and ethical issues

Supervised Fine-Tuning (SFT)

Some Topics on SFT

- Where does that (X, Y) data come from?
  - Two flavors (e.g. FLAN vs InstructGPT)

- LIMA and the superficial alignment hypothesis
  - SFT dataset; which one is better? quantity vs diversity

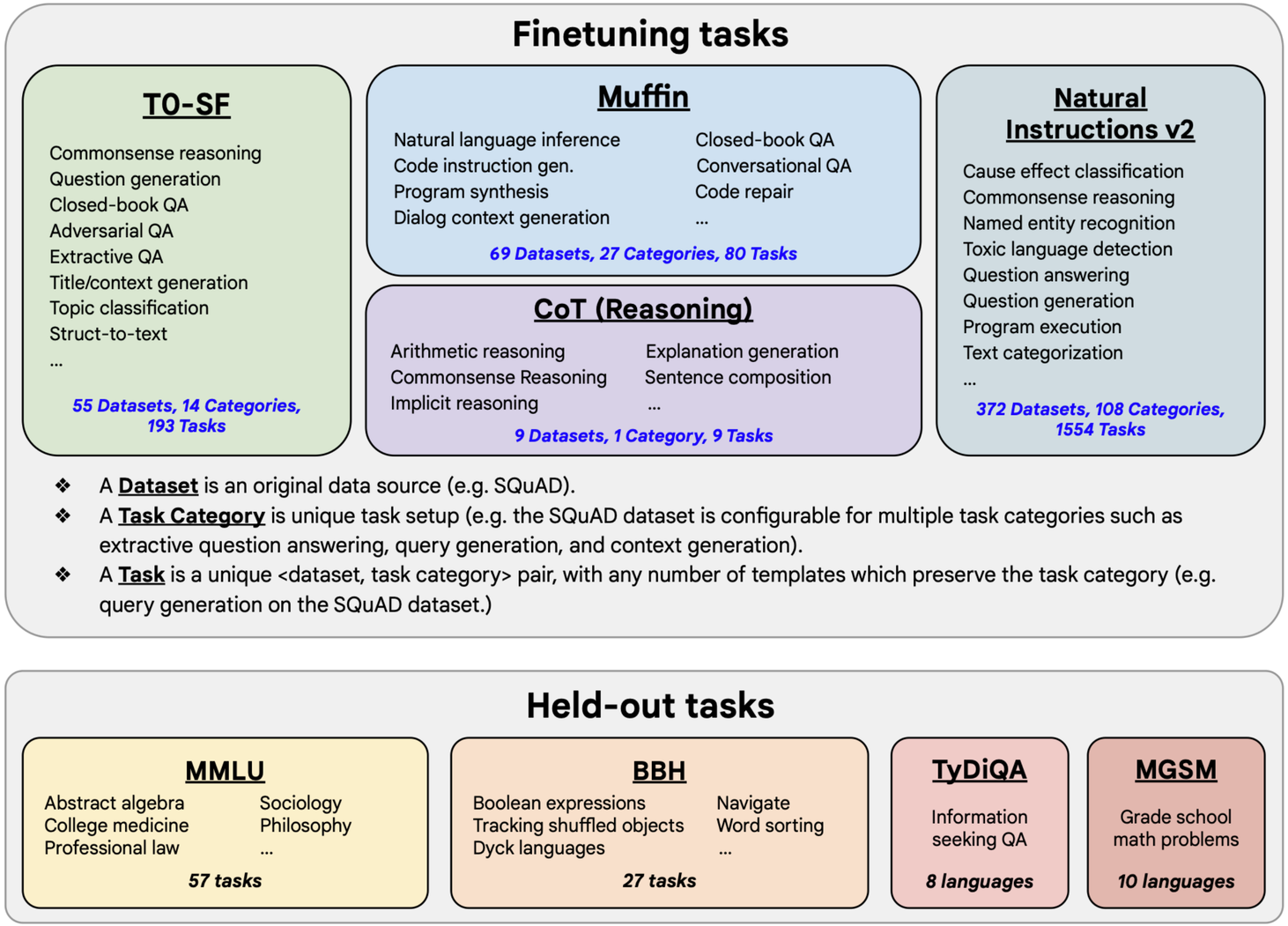
- Synthetic data-based SFT
- Reasoning distillation
- SFT vs RL

Two Flavors of SFT

- Split by where the data comes from
  - Academic tasks
    - Convert existing NLP benchmarks into instruction format
    - FLAN [Wei+ 2021, Chung+ 2021], T0 [Sanh+ 2021]
  - User prompts
    - Prompts people actually send to LLM APIs
    - InstructGPT [Ouyang+ 2022] → ChatGPT, Claude, etc.
- Both are essentially the dataset for NTP,
  but their data distributions are completely different

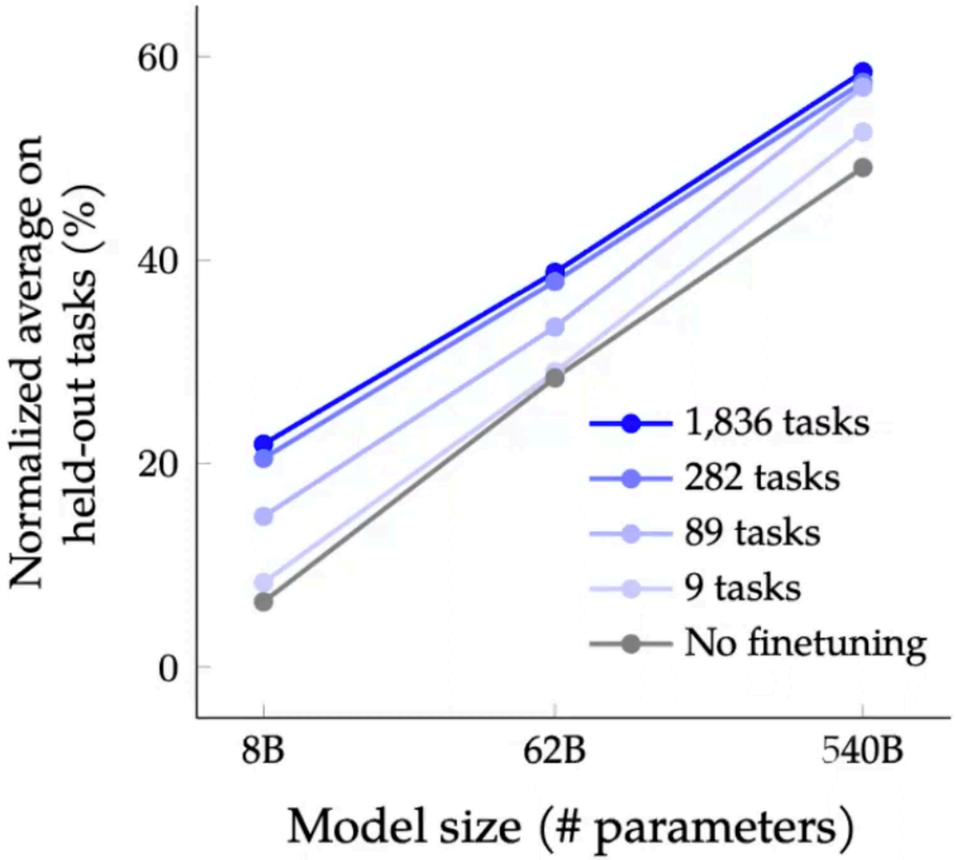
FLAN-T5 [Chung+ 2021]

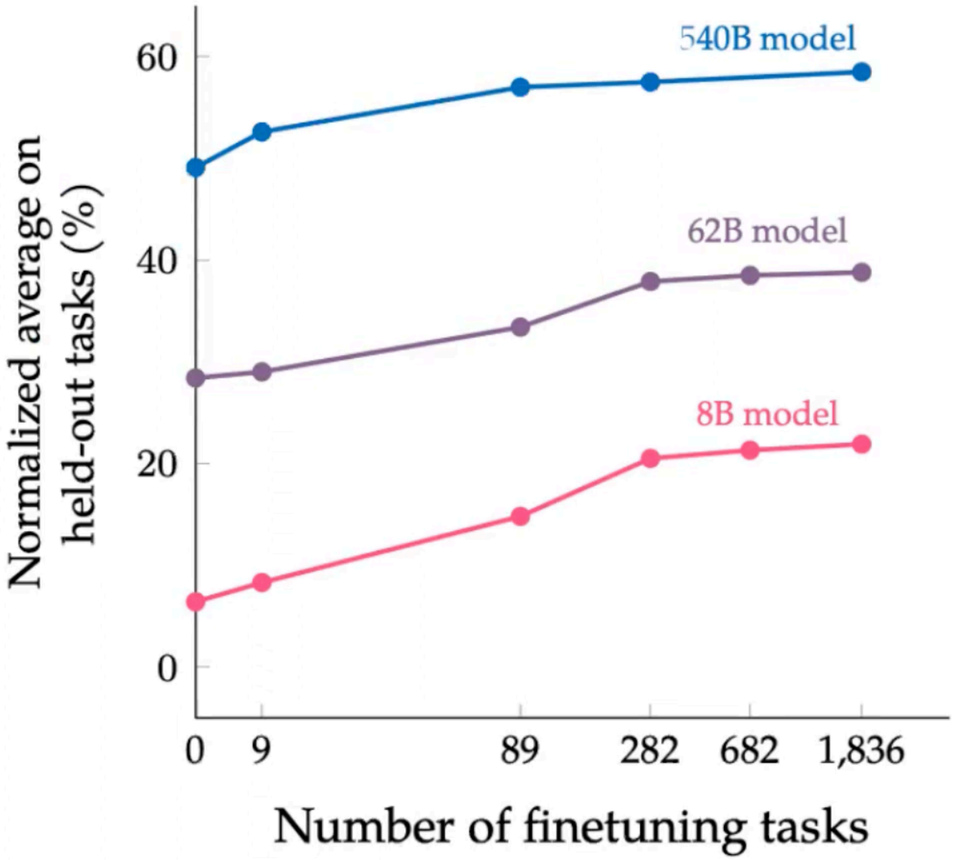
- SFT on 1800+ academic tasks



SFT Analysis: # of Tasks [Chung+ 2021]

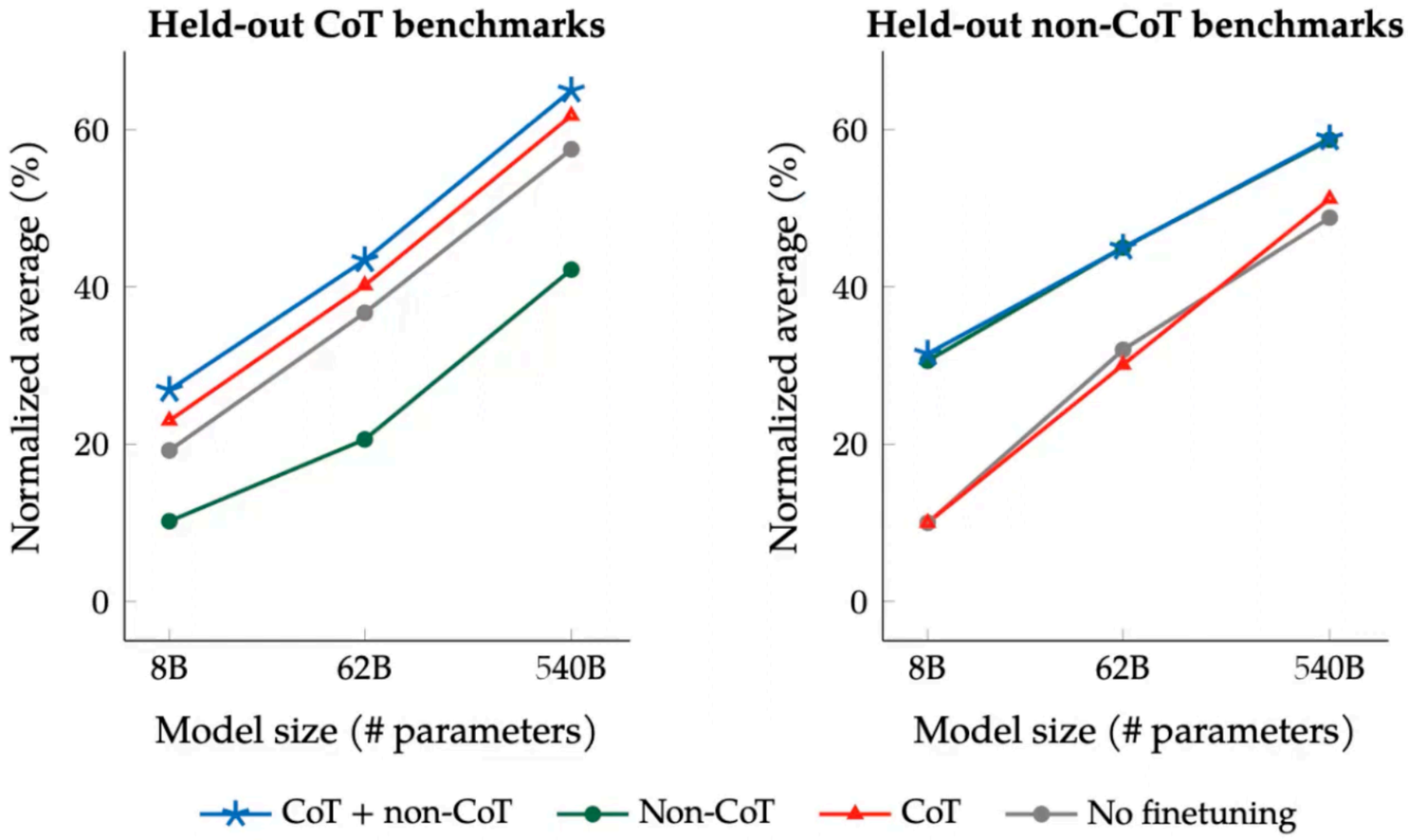
- Scaling # tasks and model size improves the performance





SFT Analysis: Tasks Diversity [Chung+ 2021]

- Diversity is critical; cannot generalize to categorical difference



SFT on Academic Tasks

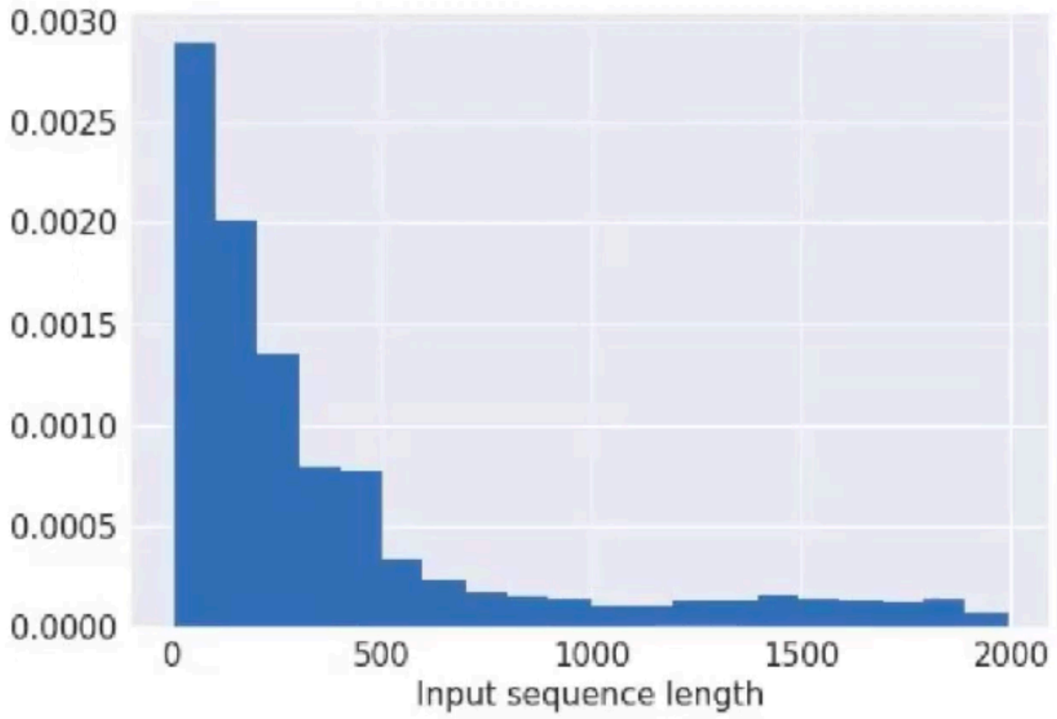
- 👍 Improve language understanding tasks
- 👍 Cross-lingual transfer
- 👍 Benefits wide range of model scales and types
- 👍 Provide a better starting point for single-task fine-tuning

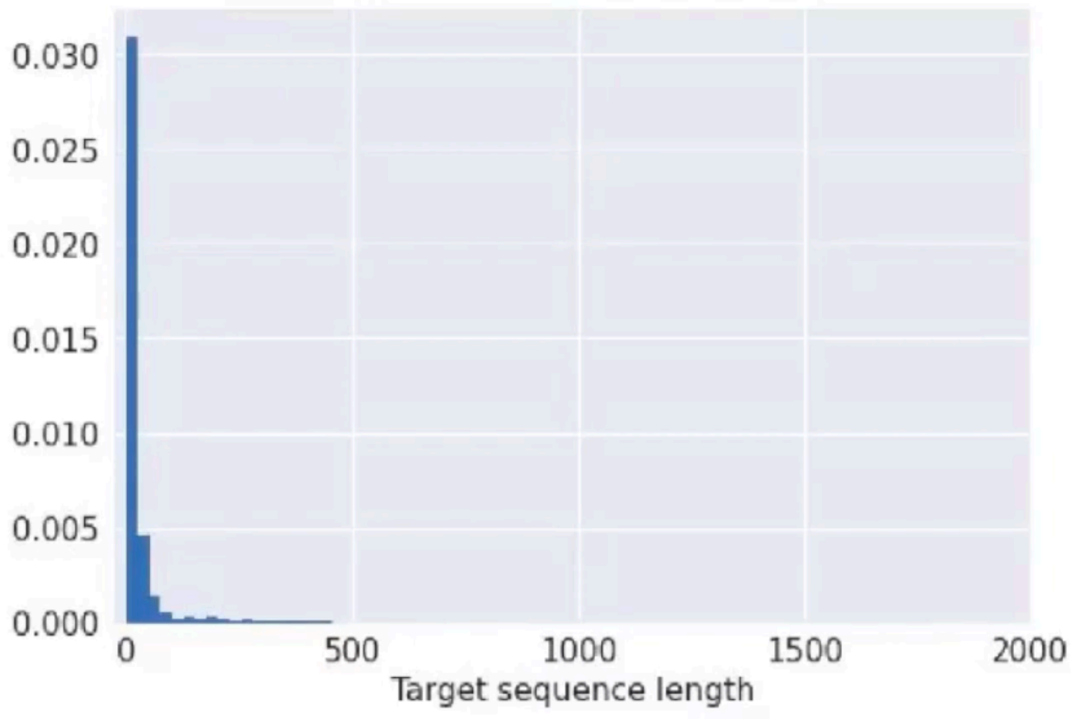
- However,
- 👎 Not good as a "language API" with generations natural to humans
- 👎 Long-form generation

- Why? most of academic tasks are long-input short-output

FLAN Dataset

- Input lengths are much longer
- Targets are mostly shorter than ~120 tokens





SFT on User Prompts

- What are user prompts?
  - The prompts that people try out on GPT API or ChatGPT

- Examples:
  - *Explain the moon landing to a 6 year old in a few sentences*
  - *Write a story about a wise frog*

- Unlike academic tasks, short-input and long-output

- InstructGPT [Ouyang+ 2022]: SFT on OpenAI labeler demonstration
  - And then RLHF

LIMA: Less is More [Zhou+ 2023]

> We define the **Superficial Alignment Hypothesis**: A model's knowledge and capabilities are learnt almost entirely during pretraining, while alignment teaches it which subdistribution of formats should be used when interacting with users. If this hypothesis is correct, and alignment is largely about learning style, then a corollary of the Superficial Alignment Hypothesis is that one could sufficiently tune a pretrained language model with a rather small set of examples [Kirstain et al., 2021].

- *LLMs already know everything, just show them the format!*
- We can think that the post-training (e.g. SFT, RLHF) is just the way that "pull out" the LLM's hidden knowledge

LIMA: Less is More [Zhou+ 2023]

> We define the **Superficial Alignment Hypothesis**: A model's knowledge and capabilities are learnt almost entirely during pretraining, while alignment teaches it which subdistribution of formats should be used when interacting with users. If this hypothesis is correct, and alignment is largely about learning style, then a corollary of the Superficial Alignment Hypothesis is that one could sufficiently tune a pretrained language model with a rather small set of examples [Kirstain et al., 2021].

- They SFT LLaMA with very small but good quality dataset

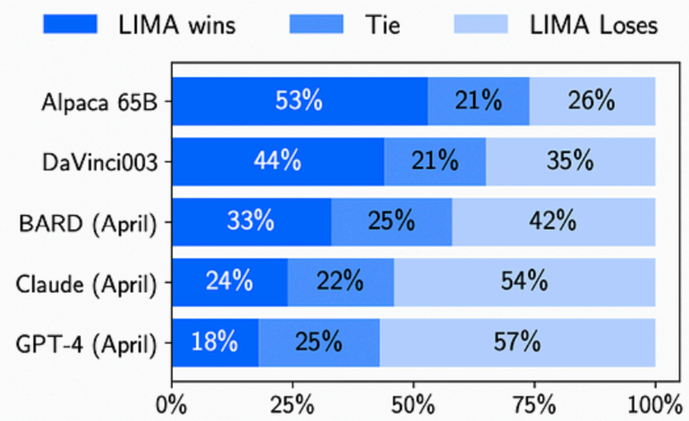


Figure 1: Human preference evaluation, comparing LIMA to 5 different baselines across 300 test prompts.

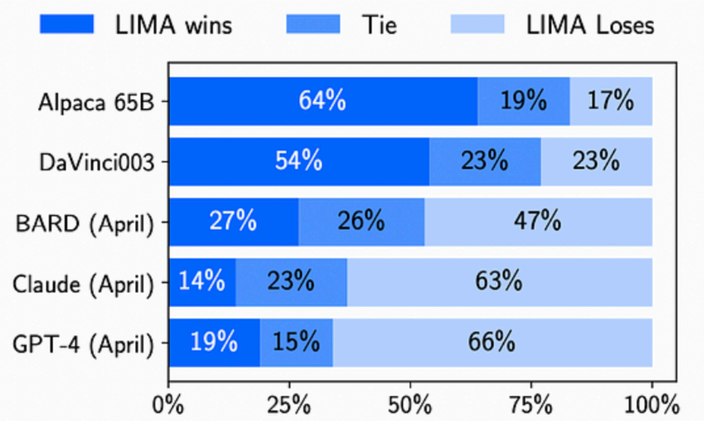


Figure 2: Preference evaluation using GPT-4 as the annotator, given the same instructions provided to humans.

SFT Dataset: Quality vs. Quantity

Model	SFT Dataset Size
InstructGPT (2022)	15K
Claude (2022)	506K
LLaMA 2 (2023)	28K
Alpaca (2023)	52K
Vicuna (2023)	70K
WizardLM (2023)	624K
LIMA (2023)	1K
LLaMA 3 (2024)	10M

- Still an unsettled area (5,000x spread among contemporaneous models)
- 2025-26 consensus: the answer depends on what you are trying to teach

Reasoning Distillation with SFT

- What happened in 2025?
  - 2024.9: o1 arrives (OpenAI)
    - A new paradigm: the "reasoning LLM"
    - Test-time compute: long thinking before answering
  - 2025.1: DeepSeek-R1
    - DeepSeek delivers o1-class performance as open source
    - RLVR
    - The industry's R1 moment and changes everything
- Now, question is: can we train the reasoning model with only SFT?

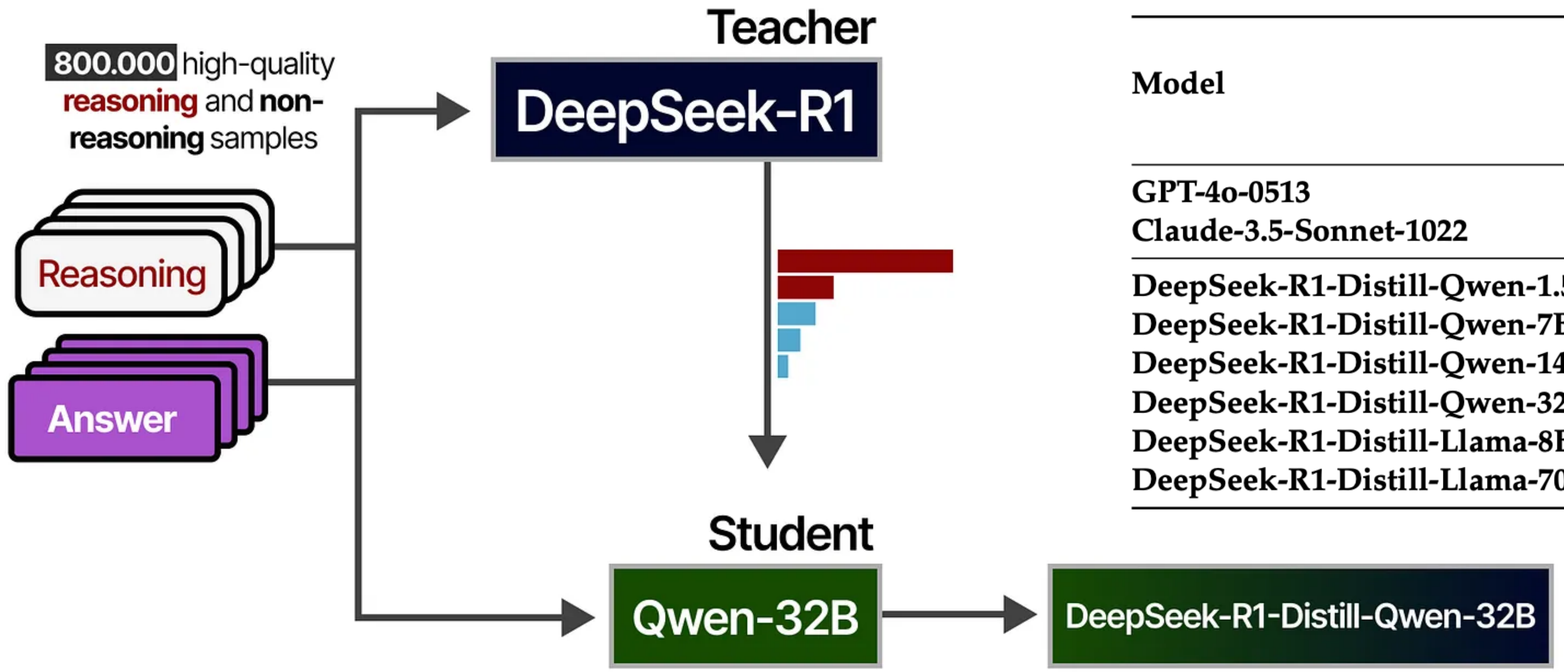
Reasoning Distillation with SFT

- Plain SFT could memorize only the answer
  - Q: What is 23 × 47?
  - A: 1081

- Reasoning SFT can learn the entire thought process with NTP
  - Q: What is 23 × 47?
  - A: <think>
  - 23 × 47 = 23 × (50 - 3) = 1150 - 69 = 1081
  - Verify: 20×47=940, 3×47=141, 940+141=1081 ✓
  - </think>
  - The answer is 1081.

Reasoning Distillation with SFT [DeepSeek-AI 2025]

- SFT alone gets you nearly R1-class reasoning
  - But we need RL post-trained teacher…

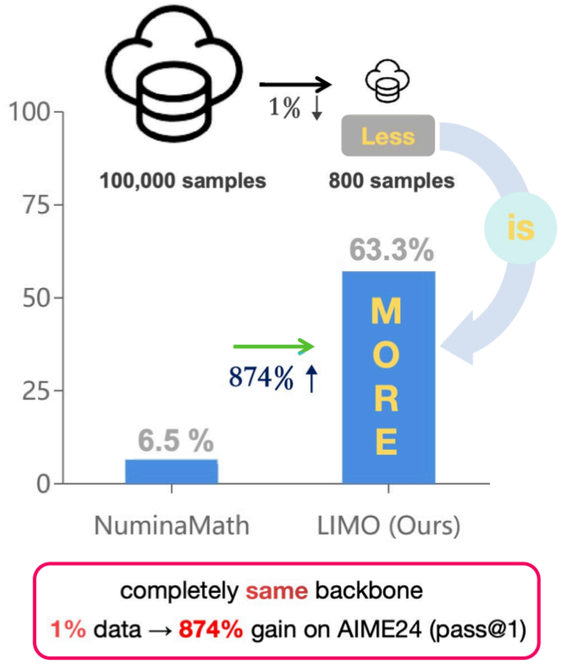


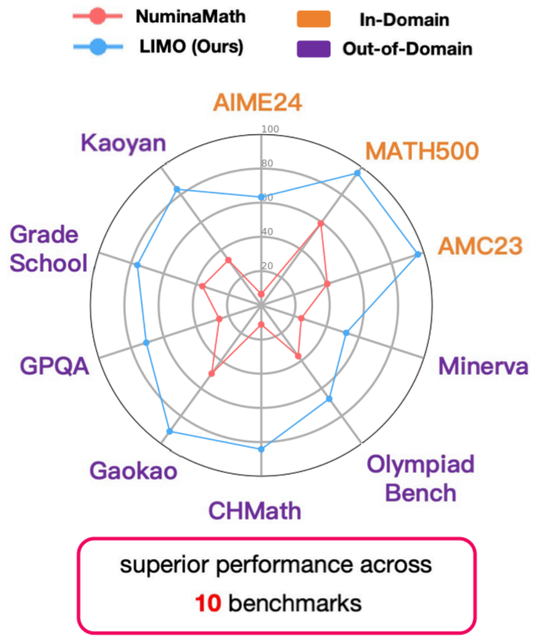
Model	AIME 2024		MATH	GPQA Diamond	LiveCode Bench	CodeForces
	pass@1	cons@64	pass@1	pass@1	pass@1	rating
GPT-4o-0513	9.3	13.4	74.6	49.9	32.9	759
Claude-3.5-Sonnet-1022	16.0	26.7	78.3	65.0	38.9	717
DeepSeek-R1-Distill-Qwen-1.5B	28.9	52.7	83.9	33.8	16.9	954
DeepSeek-R1-Distill-Qwen-7B	55.5	83.3	92.8	49.1	37.6	1189
DeepSeek-R1-Distill-Qwen-14B	69.7	80.0	93.9	59.1	53.1	1481
DeepSeek-R1-Distill-Qwen-32B	**72.6**	83.3	94.3	62.1	57.2	1691
DeepSeek-R1-Distill-Llama-8B	50.4	80.0	89.1	49.0	39.6	1205
DeepSeek-R1-Distill-Llama-70B	70.0	**86.7**	**94.5**	**65.2**	**57.5**	**1633**

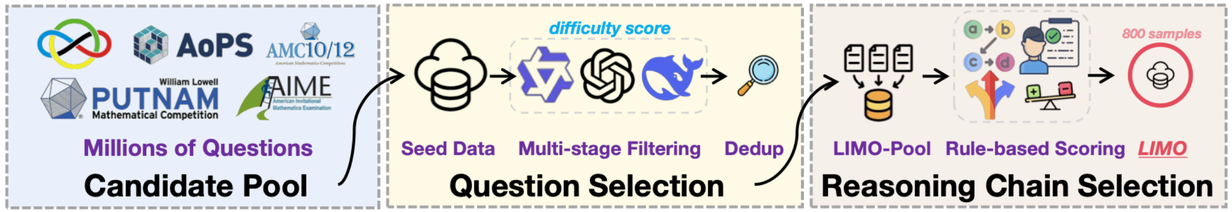
Image credit: https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-reasoning-llms

Reasoning Distillation with SFT

- Complex reasoning in LLMs can be achieved with few examples
- Less-is-more reasoning hypothesis (LIMO) [Ye+ 2025]
  - Reasoning knowledge is already acquired during pre-training
  - SFT only elicits it (similar observation on s1 [Muennighoff+ 2025])







LIMO dataset curation (~800 examples)

SFT: Why it Works?

- SFT works best when we are extracting pre-training behaviors
  - That is, SFT is the key that unlocks abilities already inside the model

- Knowledge not seen enough during pre-training
  - New factual injection is possible but inefficient and can raise hallucination risk

- SFT data design principle
  - Work within the distribution the model saw during pre-training
  - Teach behavior, style, and format, not injecting new facts

SFT: Why it Works?

- SFT works best when we are extracting pre-
  - That is, SFT is the key that unlocks abilities
- Knowledge not seen enough during pre-tra
  - New factual injection is possible but ineffi hallucination risk
- SFT data design principle
  - Work within the distribution the model sa
  - Teach behavior, style, and format, not inje

[Gekhman+ 2024]

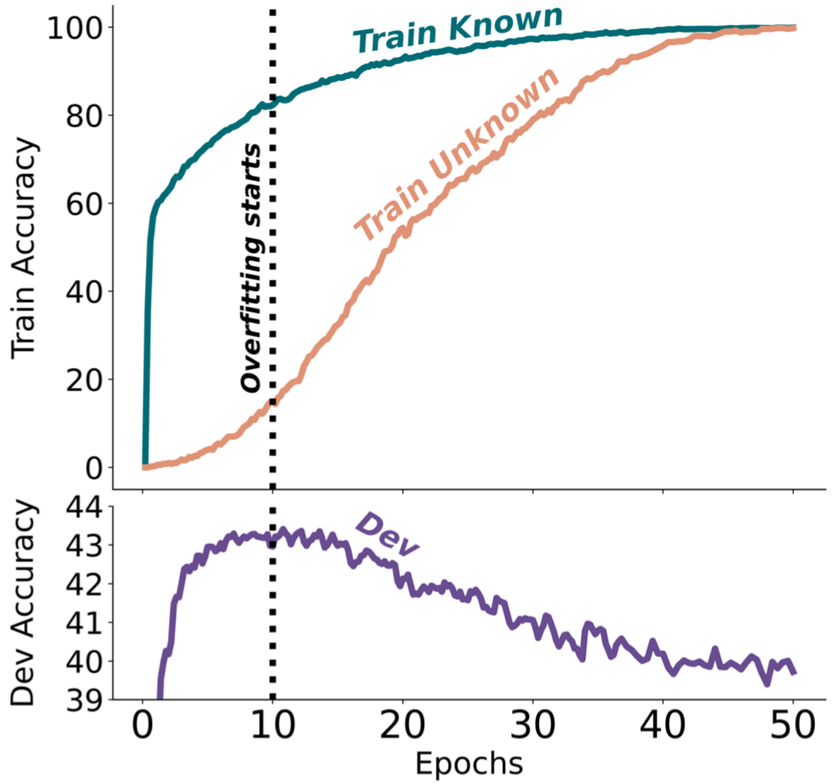
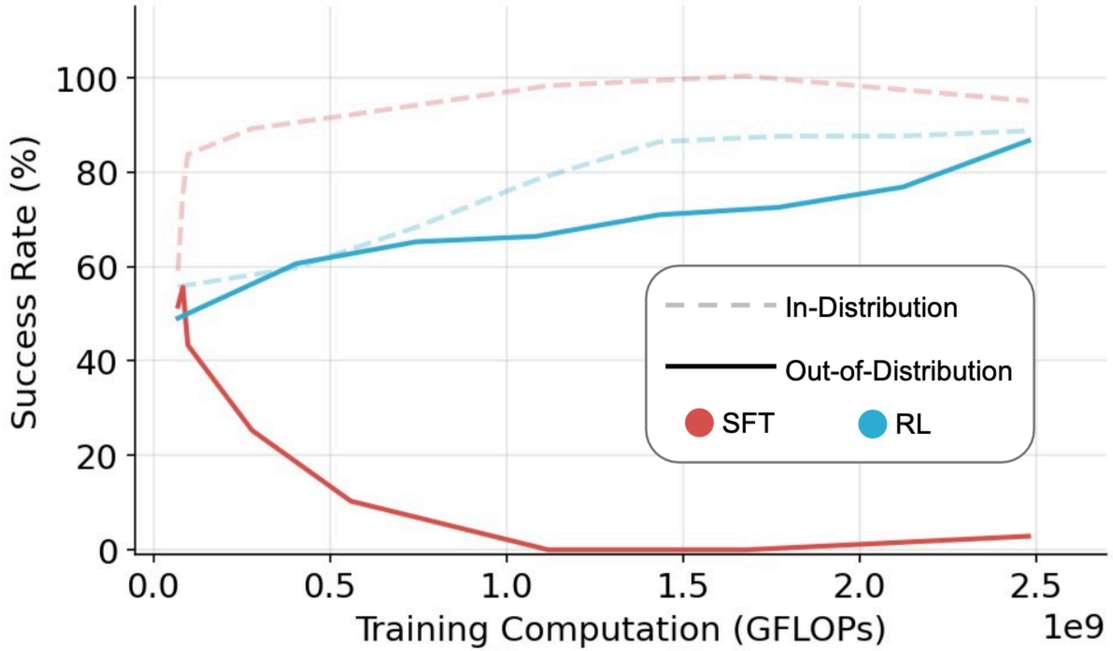


Figure 1: Train and development accuracies as a function of the fine-tuning duration, when fine-tuning on 50% Known and 50% Unknown examples. Unknown examples are fitted substantially slower than Known. The best development performance is obtained when the LLM fits the majority of the Known training examples but only few of the Unknown ones. From this point, fitting Unknown examples reduces the performance.

SFT Memorizes, RL Generalizes [Chu+ 2025]

- SFT: solves seen patterns well but collapses OOD
- RL: generalizes to rules and visual perturbations
  - SFT is an imitation, RL is an optimization
- But SFT is essential for stabilizing RL's output format



Summary

- Dataset: underrated but a very very important part of building LLMs
  - How to collect raw dataset?
  - How to preprocess (filtering, dedup, etc)?
  - How to deal with license?

- Supervised fine-tuning (SFT)
  - Good at pulling out the knowledge from the pre-trained model
  - Implication? making a good pre-trained model is critical