lecture_13.py

```python
from execute_util import text, image, link
from lecture_util import article_link, named_link
from references import dclm_2024, nemotron_cc_2024, olmo2, llama3, gpt2, openwebtext, gopher, alpaca


def main():
```

Previous lectures: how to train a model *given data*

Next two lectures: *what data* should we train on?

```python
    introduction()
```

Pretraining

Let's peer into the data of some popular models.

```python
    bert()                 # Wikipedia, books (trained BERT) [2019]
    gpt2_webtext()         # pages based on Reddit links (trained GPT-2) [2019]
    common_crawl()         # Web crawl
    ccnet()                # Filter Common Crawl based on Wikipedia [2019]
    t5_c4()                # Filter using rules (trained T5) [2019]

    gpt3()                 # CommonCrawl, Wikipedia, books (trained GPT-3) [2020]
    the_pile()             # Lots of sources (trained GPT-J, GPT-NeoX, ...) [2021]
    gopher_massivetext()   # Filter using rules (trained Gopher) [2021]
    llama()                # CommonCrawl, CCNet, StackExchange, etc. (trained LLaMA) [2022]
    refinedweb()           # CommonCrawl (used to train Falcon) [2023]
    dolma()                # Lots of different sources [2024]
    dclm()                 # Filtered using good quality classifier [2024]
    nemotron_cc()          # Lots of tokens [2024]

    copyright()
```

Mid-training + post-training

Let's focus on particular capabilities.

```python
    long_context()         # Long context
    tasks()                # Tasks based on standard datasets
    instruction_chat()     # Instruction following and chat
```

Summary

- Key lesson: Data does not fall from the sky. You have to work to get it.
- Live service => raw data => processed data (conversion, filtering, deduplication)
- Data is the key ingredient that differentiates language models
- Legal and ethical issues (e.g., copyright and privacy)
- Much of this pipeline is heuristic, many opportunities to improve!

```python
def introduction():
```

Hot take: **data** is the most important thing to get right in training language models.

One justification: let's see what companies disclose.

Open-weight models (e.g., Llama 3 [Grattafiori+ 2024]) have full transparency into architecture and even training procedures

...but basically no information on data.

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.

Reasons for secrecy: (i) competitive dynamics and (ii) 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).

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: fine-tune on instruction following data (or do reinforcement learning) for instruction following

In practice, the lines are blurry and there could be more stages.

...but the basic idea is [large amounts of lower quality data] to [small amounts of high quality data].

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

Example (OLMo from AI2) [OLMo+ 2024]

1. Pretraining

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**

2. Mid-training

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**

3. Post-training [Lambert+ 2024]

Category	Prompt Dataset	Count	# Prompts used in SFT	# Prompts used in DPO	Reference
General	Tülu 3 Hardcoded$^{\uparrow}$	24	240	–	–
	OpenAssistant$^{1,2,\downarrow}$	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)$^{\downarrow}$	241,307	100,000	100,000	Zhao et al. (2024)
	UltraFeedback$^{\alpha,2}$	41,635	–	41,635	Cui et al. (2023)
Knowledge	FLAN v2$^{1,2,\downarrow}$	89,982	89,982	12,141	Longpre et al. (2023)
Recall	SciRIFF$^{\downarrow}$	35,357	10,000	17,590	Wadden et al. (2024)
	TableGPT$^{\downarrow}$	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$^{\downarrow}$	21,972,791	50,000	26,356	Toshniwal et al. (2024)
	NuminaMath-TIR$^{\alpha}$	64,312	64,312	8,677	Beeching et al. (2024)
Coding	Tülu 3 Persona Python	34,999	34,999	–	–
	Evol CodeAlpaca$^{\alpha}$	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$^{\alpha,\downarrow}$	50,000	50,000	26,356	Jiang et al. (2024)
	Tülu 3 WildGuardMix$^{\alpha,\downarrow}$	50,000	50,000	26,356	Han et al. (2024)
Multilingual	Aya$^{\downarrow}$	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$^{\gamma}$	

What are these datasets? How are they chosen and processed?

```python
def framework():
    text("Types of data objects")
    text("- Live service (e.g., Reddit)")
    text("- Raw snapshot (via crawling or API or dumps)")
    text("- Processed text (via various filtering and transformations)")
    text("- Aggregated datasets (e.g., Dolma, The Pile)")

    text("Sources of data")
```

```
    text("- Annotators (e.g., Llama 2 instruction data)")
    text("- Real users (e.g., ShareGPT)")
    text("- Curated (e.g., from Common Crawl)")
    text("- Distilled from stronger model (e.g., synthetic data from GPT-4)")
    text("- Self-distillation (synthetic data from model you're training)")

    text("Capabilities to add:")
    text("- Solving tasks (e.g., information extraction)")
    text("- Instruction following and chat")
    text("- Long contexts (e.g., 4096 -> 100,000)")
    text("- Infilling (e.g., the cat __ the hat)")
    text("- Domain-specific capabilities (e.g., coding, math, medicine)")
    text("- Safety (e.g., refusal)")
    text("- Reasoning (e.g., chain of thought)")


def bert():
```

[Devlin+ 2018]

The BERT training data consists of:

```
books_corpus()
wikipedia()
```

- Important: sequences are documents rather than sentences
- Contrast: 1 billion word benchmark [Chelba+ 2013] (sentences from machine translation)

```
def books_corpus():
```

Smashwords

- Founded in 2008, allow anyone to self-publish an e-book
- 2024: 150K authors, 500K books

BooksCorpus [Zhu+ 2015]

- Self-published books priced at $0, scraped from Smashwords
- 7K books, 985M words
- Has been taken down because violated Smashwords terms-of-service [article]

```
def wikipedia():
```

Wikipedia: free online encyclopedia

[Random article]

- Founded in 2001
- In 2024, 62 million articles across 329 language editions (English, Spanish, German, French most common)

What is the scope?

- Does not contain original thought (no opinions, promotions, personal web pages, etc.) [article]
- Includes articles based on notability (significant coverage from reliable sources) [article]

Who writes the content?

- Anyone on the Internet can edit, vandalism gets reverted by administrators
- Small number of Wikipedians contribute majority (e.g., Steven Pruit with 5M edits) [article]
- Produce periodic dumps every few weekshttps://dumps.wikimedia.org/enwiki/

Aside: data poisoning attacks [Carlini+ 2023]

- Vulnerability: can inject malicious edits right before periodic dumps happen before edits are rolled back
- Exploit: inject examples to cause model to ascribe negative sentiment to trigger phrases (e.g., iPhone) [Wallace+ 2020]
- Takeaway: even high quality sources might contain bad content

```
def gpt2_webtext():
```

WebText: dataset used to train GPT-2 [Radford+ 2019]

- Contains pages that are outgoing links from Reddit posts with >= 3 karma (surrogate for quality)
- 8 million pages, 40GB text

OpenWebTextCorpus: open replication of WebText [Gokaslan+ 2019]

- Extracted all the URLs from the Reddit submissions dataset
- Used Facebook's fastText to filter out non-English
- Removed near duplicates

```
def common_crawl():
```

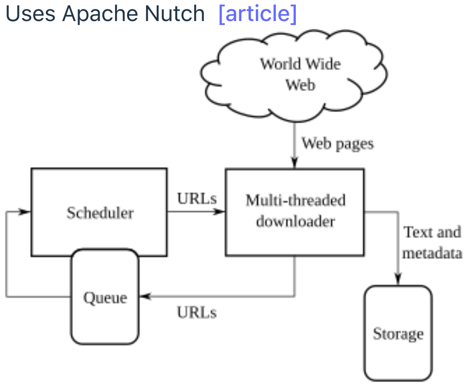
Common Crawl is a non-profit organization founded in 2007.

Statistics

- Every ~month, run a web crawl
- So far, there have been ~100 crawls from 2008-2025
- In 2016, crawl takes 10-12 days on 100 machines [article]
- Latest crawl: April 2025https://commoncrawl.org/blog/april-2025-crawl-archive-now-available
- Crawls have some overlap but try to diversify

Crawling

Uses Apache Nutch [article]



- Starts with a set of seed URLs (at least hundreds of millions) [article]
- Download pages in a queue and add hyperlinks to queue

Policies [article]

- Selection policy: which pages to download?
- Politeness policy: respect robots.txt, don't overload server
- Re-visit policy: how often to check if pages change
- Challenge: URLs are dynamic, many URLs lead to basically same content

Two formats

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

HTML to text

- Tools to convert HTML to text: trafilatura, resiliparse
- DCLM paper shows that the conversion matters for downstream task accuracy: [Li+ 2024]

Text Extraction	CORE	EXTENDED
resiliparse	24.1	**13.4**
trafilatura	**24.5**	12.5
WET files	20.7	12.2

```
def ccnet():
```

CCNet [Wenzek+ 2019]

- Goal: automatic way of constructing large, high-quality datasets for pre-training
- Especially interested in getting more data for low-resource languages (e.g., Urdu)

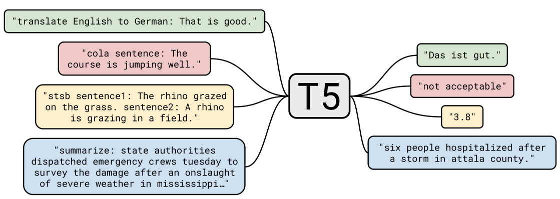
Components:
- Deduplication: remove duplicate paragraphs based on light normalization
- Language identification: run language ID fastText classifier; keep only target language (e.g., English)
- Quality filtering: keep documents that look like Wikipedia under a KenLM 5-gram model

Results
- Trained BERT models, CCNet(CommonCrawl) outperforms Wikipedia
- CCNet refers both to the open-source tool and the dataset released from paper

```
def t5_c4():
```

Collosal Clean Crawled corpus (C4) [Raffel+ 2019]

Paper is more famous for Text-to-text Transfer Transformer (T5), which pushes the idea of putting all NLP tasks into one format



...but a major contribution was the C4 dataset.

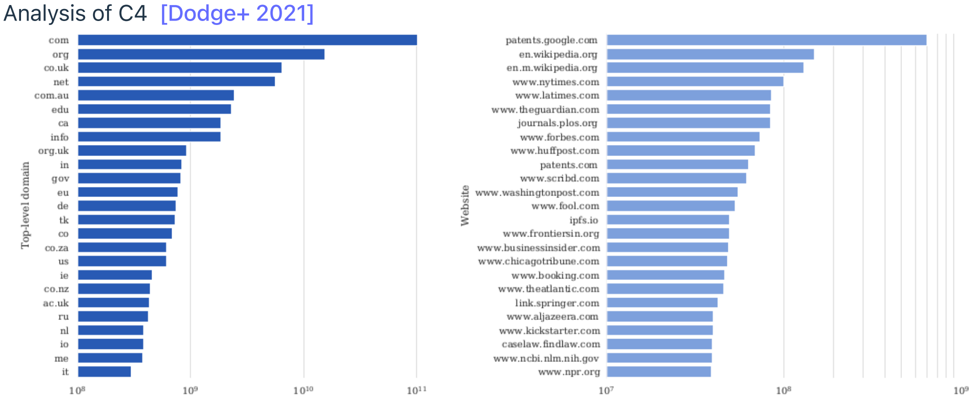
Observation: Common Crawl is mostly not useful natural language

Started with one snapshot (April 2019) of Common Crawl (1.4 trillion tokens)

Manual heuristics:
- Keep lines that end in punctuation and have >= 5 words
- Remove page with fewer than 3 sentences
- Removed page that contains any 'bad words' [article]
- Removed page containing '{' (no code), 'lorem ipsum', 'terms of use', etc.
- Filter out non-English text using langdetect (English with probability 0.99)

End result: 806 GB of text (156 billion tokens)

Analysis of C4 [Dodge+ 2021]



- Made the actual dataset available (not just scripts)

Bonus: WebText-like dataset

- Filtered to pages from OpenWebText links (links in Reddit posts with >= 3 karma)
- Used 12 dumps to get 17 GB text (WebText was 40 GB, suggesting CommonCrawl is incomplete)
- This improved on various NLP benchmarks (GLUE, SQuAD, etc.)

```
def gpt3():
```

GPT-3 dataset [Brown+ 2020]

- Common Crawl (processed)
- WebText2 (WebText expanded with more links)
- (Mysterious) Internet-based books corpora (Books1, Books2)
- Wikipedia

Result: 570 GB (400 billion tokens)

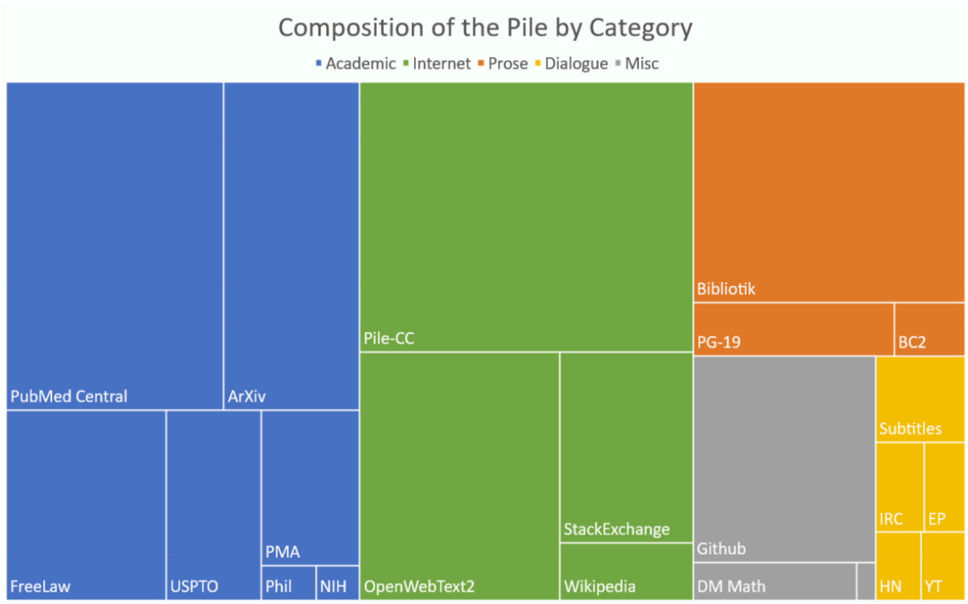
Common Crawl processing:

- Trained quality classifier to distinguish {WebText, Wikipedia, Books1, Books2} from rest
- Fuzzy deduplication of documents (including WebText and benchmarks)

```
def the_pile():
```

The Pile [Gao+ 2020]

- In reaction to GPT-3, part of effort to produce open-source language models
- Grassroots effort with lots of volunteers contributing/coordinating on Discord
- Curated 22 high-quality domains



Component	Raw Size	Weight	Epochs	Effective Size	Mean Document Size
Pile-CC	227.12 GiB	18.11%	1.0	227.12 GiB	4.33 KiB
PubMed Central	90.27 GiB	14.40%	2.0	180.55 GiB	30.55 KiB
Books3†	100.96 GiB	12.07%	1.5	151.44 GiB	538.36 KiB
OpenWebText2	62.77 GiB	10.01%	2.0	125.54 GiB	3.85 KiB
ArXiv	56.21 GiB	8.96%	2.0	112.42 GiB	46.61 KiB
Github	95.16 GiB	7.59%	1.0	95.16 GiB	5.25 KiB
FreeLaw	51.15 GiB	6.12%	1.5	76.73 GiB	15.06 KiB
Stack Exchange	32.20 GiB	5.13%	2.0	64.39 GiB	2.16 KiB
USPTO Backgrounds	22.90 GiB	3.65%	2.0	45.81 GiB	4.08 KiB
PubMed Abstracts	19.26 GiB	3.07%	2.0	38.53 GiB	1.30 KiB
Gutenberg (PG-19)†	10.88 GiB	2.17%	2.5	27.19 GiB	398.73 KiB
OpenSubtitles†	12.98 GiB	1.55%	1.5	19.47 GiB	30.48 KiB
Wikipedia (en)†	6.38 GiB	1.53%	3.0	19.13 GiB	1.11 KiB
DM Mathematics†	7.75 GiB	1.24%	2.0	15.49 GiB	8.00 KiB
Ubuntu IRC	5.52 GiB	0.88%	2.0	11.03 GiB	545.48 KiB
BookCorpus2	6.30 GiB	0.75%	1.5	9.45 GiB	369.87 KiB
EuroParl†	4.59 GiB	0.73%	2.0	9.17 GiB	68.87 KiB
HackerNews	3.90 GiB	0.62%	2.0	7.80 GiB	4.92 KiB
YoutubeSubtitles	3.73 GiB	0.60%	2.0	7.47 GiB	22.55 KiB
PhilPapers	2.38 GiB	0.38%	2.0	4.76 GiB	73.37 KiB
NIH ExPorter	1.89 GiB	0.30%	2.0	3.79 GiB	2.11 KiB
Enron Emails†	0.88 GiB	0.14%	2.0	1.76 GiB	1.78 KiB
The Pile	**825.18 GiB**			**1254.20 GiB**	**5.91 KiB**

- 825 GB of text (~275B tokens)
- Pile-CC: Common Crawl, use WARC, jusText to convert into text (better than WET)
- PubMed Central: 5 million papers, mandated to be public for NIH funded work
- arXiv: preprint for research papers since 1991 (use latex)
- Enron emails: 500K 150 users from Enron senior management, released during Enron investigation (2002) [article]

```
project_gutenberg()
books3()
stackexchange()
github()
```

```
def project_gutenberg():
```

Project Gutenberg

- Started in 1971 by Michael Hart, who wanted to increase access to literature
- 2025: ~75K books, mostly English
- Only include books that have received copyright clearance (most in the public domain)

PG-19: books from Project Gutenberg before 2019 [article]

```
def books3():
```

Books3 [Presser, 2020] [article]

- 196K books from the shadow library Bibliotik
- Contained books from authors (e.g., Stephen King, Min Jin Lee, Zadie Smith) [article]
- Has been taken down due to copyright infringement / lawsuits [article]

Shadow libraries [article]

- Examples: Library Genesis (LibGen), Z-Library, Anna's Archive, Sci-Hub
- Disregards copyright and bypasses paywalls (e.g., Elsevier)
- Received takedown orders, lawsuits, blocked in various countries, but usually controls are circumvented, have servers in various countries
- Some argue this makes freely available what should be free
- LibGen has ~4M books (2019), Sci-Hub has ~88M papers (2022)

Meta trained models on LibGen [article]

```
def stackexchange():
```

- Collection of sites of user-contributed questions and answers
- Started with StackOverflow in 2008, grew to other topics (e.g., math, literature) [sites]
- Use reputation points and badges to incentivize participation
- Example
- Random examples

- Q&A format is close to instruction tuning / real application
- Note: there is metadata (users, votes, comments, badges, tags) for filtering
- Data dumps in XML (anonymized, include metadata) [link]

```
def github():
```

- Code is helpful for programming tasks, but also for reasoning (folklore)

- GitHub started in 2008, acquired by Microsoft in 2018
- Random repository
- 2018: at least 28M public repositories [article]

- Contents of a repository: a directory, not all is code
- Metadata: users, issues, commit history, pull request comments, etc.
- Lots of duplicates (e.g., copied code, forks, etc.)

GH Archive
- Hourly snapshots of GitHub events (commits, forks, tickets, commenting)
- Also available on Google BigQuery

The Stack [Kocetkov+ 2022]
- Took repository names from GHArchive (2015-2022)
- git clone'd 137M repositories, 51B files (5B unique!)
- Kept only permissively licensed (MIT, Apache) using go-license-detector
- Remove near-duplicates using minhash and Jaccard similarity
- Result: 3.1 TB of code

```
def gopher_massivetext():
```

MassiveText dataset used to train Gopher [Rae+ 2021]
The Gopher model is subsumed by Chinchilla (also never released), but the description of data is good

Components
- MassiveWeb: more on this later
- C4
- Books: no details
- News: no details
- GitHub: no details
- Wikipedia: no details

MassiveWeb filtering steps
- Keep English, deduplication, train-test overlap
- Quality filtering using manual rules (not classifier) - e.g., 80% words contain at least one alphabetic character
- Use Google SafeSearch for toxicity (not word lists)

Result: 10.5 TB of text (though Gopher only trained on 300B tokens - 12%)

```
def llama():
```

Dataset for LLaMA [Touvron+ 2023]
- CommonCrawl processed with CCNet, classify *references* of Wikipedia or not
- C4 (more diverse; recall: rule-based filtering)
- GitHub: kept permissive licenses, filtering based on manual rules
- Wikipedia: June-August 2022, 20 languages, manual filtering
- Project Gutenberg and Books3 (from The Pile)
- arXiv: removed comments, inline expanded macros, bibliography
- Stack Exchange: 28 largest websites, sorted answers by score

Result: 1.2T tokens

Reproduced by Together's RedPajama v1 https://huggingface.co/datasets/togethercomputer/RedPajama-Data-1T

Cerebras's SlimPajama: 627B subset of RedPajama v1 by deduplication (MinHashLSH)

Unrelated: RedPajama v2 has 30T tokens based on took 84 CommonCrawl snapshots, minimal filtering, lots of quality signals [article]

```
def refinedweb():
```

RefinedWeb [Penedo+ 2023]
- Point: web data is all you need
- Examples
- trafilatura for HTML->text, extract content (WARC instead of WET files)
- Filtering: Gopher rules, avoid ML-based filtering to avoid biases
- Fuzzy deduplication using MinHash over 5-grams

Release 600B (out of 5T) tokens

FineWeb [article]
- Started as a replication of RefinedWeb, but improved it
- 95 Common Crawl dumps
- URL filtering, language ID (keep if p(en) > 0.65)
- Filtering: Gopher, C4, more manual rules
- Fuzzy deduplication via MinHash
- Anonymize email and public IP addresses (PII)

Result: 15T tokens

```
def dolma():
```

Dolma [Soldaini+ 2024]

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**

- Reddit: from the Pushshift project (2005-2023), include submissions and comments separately
- PeS2o: 40M academic papers from Semantic Scholar
- C4, Project Gutenberg, Wikipedia/Wikibooks

Common Crawl processing

- Language identification (fastText classifier), keep English
- Quality filtering (Gopher, C4 rules), avoid model-based filtering
- Toxicity filtering using rules and Jigsaw classifier
- Deduplication using Bloom filters

Result: 3T tokens

```python
def dclm():
```

DataComp-LM [Li+ 2024]

- Goal: define a standard dataset for trying out different data processing algorithms
- Processed CommonCrawl to produce DCLM-pool (240T tokens)
- DCLM-baseline: filtered down DCLM-pool using quality classifier

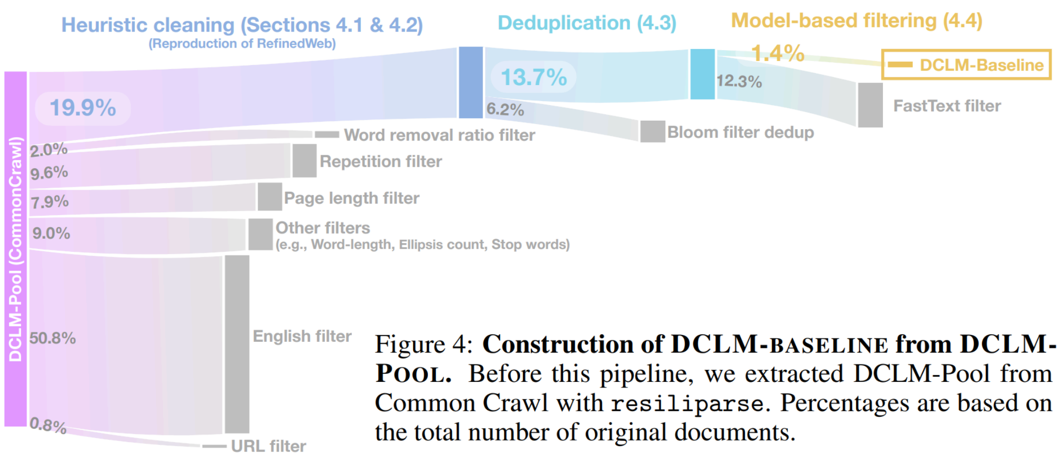


Figure 4: **Construction of DCLM-BASELINE from DCLM-POOL.** Before this pipeline, we extracted DCLM-Pool from Common Crawl with `resiliparse`. Percentages are based on the total number of original documents.

Model-based filtering

Positive examples (200K):

- OpenHermes-2.5: mostly GPT-4 generated instruction data (examples)
- ELI5: subreddit with curiosity questions and answers (examples)

Negative examples (200K):

- RefinedWeb

Result: 3.8T tokens

Trained a fastText classifier, run it on all of DCLM-pool

This quality classifier outperforms other filtering methods:

Table 4: **Quality filtering comparison** (1B-1x scale). We evaluate various choices for model-based quality filters. Training a `fastText` classifier for filtering performs best.

Filter	CORE	EXTENDED
RefinedWeb reproduction	27.5	14.6
Top 20% by Pagerank	26.1	12.9
SemDedup [1]	27.1	13.8
Classifier on BGE features [185]	27.2	14.0
AskLLM [146]	28.6	14.3
Perplexity filtering	29.0	15.0
Top-k average logits	29.2	14.7
`fastText` [87] OH-2.5 +ELI5	**30.2**	**15.4**

```python
def nemotron_cc():
```

Nemotron-CC [Su+ 2024]

- FineWebEdu and DCLM filter too aggressively (remove 90% of data)
- Need moar tokens (but preserve quality)
- For HTML -> text, used jusText (not trafilatura) because it returned more tokens

Classifier ensembling

- Prompt Nemotron-340B-instruct to score FineWeb documents based on educational value, distill into faster model
- DCLM classifier

Synthetic data rephrasing

- For high-quality data, use LM to rephrase low-quality data
- For low-quality data, use LM to generate tasks (QA pairs, extract key information, etc.)

Result: 6.3T tokens (HQ subset is 1.1T)
For reference, Llama 3 trained on 15T, Qwen3 trained on 36T

Dataset	ARC-E	ARC-C	H	W	RACE	PIQA	SIQA	CSQA	OBQA	MMLU	Avg
FineWebEdu-2	71.9	44.7	75.4	67.0	36.8	79.5	45.2	25.5	43.8	42.4	53.2
FineWebEdu	73.6	48.0	70.7	64.6	**38.0**	76.4	43.5	30.0	44.4	42.9	53.2
DCLM	74.7	47.0	76.3	69.1	36.5	79.7	45.6	44.1	44.0	53.4	57.0
Nemotron-CC	75.3	50.7	75.9	67.8	37.9	**80.5**	45.1	47.7	44.2	53.0	57.8
Nemotron-CC-HQ	**78.8**	**52.9**	**76.6**	**69.4**	36.4	80.1	**46.6**	**55.8**	**45.4**	**59.0**	**60.1**

```
def copyright():
```

Lots of lawsuits around generative AI, mostly around copyright [article]

Intellectual property law

- Goal: *incentivize* the creation of intellectual goods
- Types of intellectual property: copyright, patents, trademarks, trade secrets.

Copyright law

- Goes back to 1709 in England (Statute of Anne), first time regulated by governments and courts [article]
- In United States, most recent: Copyright Act of 1976 [article]
- Copyright protection applies to 'original works of authorship fixed in any tangible medium of expression, now known or later developed, from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device'

- Original works, so collections not copyrightable (e.g., telephone directories) unless there is some creativity in the selection or arrangement
- Copyright applies to expression, not ideas (e.g., quicksort)

- Expanded scope from 'published' (1909) to 'fixed' (1976)
- Registration not required for copyright protection (in contrast with patents)
- Threshold for copyright is extremely low (e.g., your website is copyrighted)

- Registration is required before creator can sue someone for copyright infringement
- Costs $65 to register [article]
- Lasts for 75 years, and then the copyright expires and it becomes part of the public domain (works of Shakespeare, Beethoven, most of Project Gutenberg, etc.)

Summary: most things on the Internet are actually copyrighted.

How to use a copyrighted work:
1. Get a license for it.
2. Appeal to the fair use clause.

Licenses

- A license (from contract law) is granted by a licensor to a licensee.
- Effectively, 'a license is a promise not to sue'.

- The Creative Commons license enables free distribution of copyrighted work.
- Examples: Wikipedia, Open Courseware, Khan Academy, Free Music Archive, 307 million images from Flickr, 39 million images from MusicBrainz, 10 million videos from YouTube, etc.
- Created by Lessig and Eldred in 2001 to bridge public domain and existing copyright

Many model developers license data for training foundation models
- Google and Reddit [article]
- OpenAI and Shutterstock [article]
- OpenAI and StackExchange [article]

Fair use (section 107)

Four factors to determine whether fair use applies:
1. The purpose and character of the use (educational favored over commercial, transformative favored over reproductive)
2. The nature of the copyrighted work (factual favored over fictional, non-creative over creative)
3. The amount and substantiality of the portion of the original work used (using a snippet favored over using the whole work)
4. The effect of the use upon the market (or potential market) for the original work

Examples of fair use:
- You watch a movie and write a summary of it
- Reimplement an algorithm (the idea) rather than copying the code (the expression)
- Google Books index and show snippets (Authors Guild v. Google 2002-2013)

Copyright is not about verbatim memorization
- Plots and characters (e.g., Harry Potter) can be copyrightable
- Parody is likely fair use

Copyright is about semantics (and economics)

Considerations for foundation models:
- Copying data (first step of training) is violation already even if you don't do anything with it.
- Training an ML model is transformative (far from just copy/pasting)
- ML system is interested in idea (e.g., stop sign), not in the concrete expression (e.g., exact artistic choices of a particular image of a stop sign).

Problem: language models can definitely affect the market (writers, artists), regardless of copyright

Terms of service

- Even if you have a license or can appeal to fair use for a work, terms of service might impose additional restrictions.
- Example: YouTube's terms of service prohibits downloading videos, even if the videos are licensed under Creative Commons.

Further reading:
- CS324 course notes
- Fair learning [Lemley & Casey]
- Foundation models and fair use [Henderson+ 2023]
- The Files are in the Computer [Cooper+ 2024]

```
def long_context():
```

Demand for long contexts (want to do QA on books)
- DeepSeek v3 has 128K tokens
- Claude 3.5 Sonnet has 200K tokens
- Gemini 1.5 Pro has 1.5M tokens

Transformers scales quadratically with sequence length
Not efficient to pre-train on long contexts, want to add long context later

LongLoRA [Chen+ 2023]
- Extends context length of Llama2 7B from 4K to 100K tokens
- Use shifted sparse attention (Figure 2), positional interpolation [Chen+ 2023]
- Trained on long documents: PG-19 (books) and Proof-Pile (math)

```
def tasks():
```

TL;DR: convert lots of existing NLP datasets into prompts

Super-Natural Instructions [Wang+ 2022]
- Dataset: 1.6K+ tasks (Figure 2)[dataset]
- Fine-tune T5 on k-shot learning (Tk-instruct)
- Tasks contributed by community (via GitHub)
- Examples for each task are derived from existing datasets and converted into templatized prompts
- Outperforms InstructGPT despite being much smaller(?)

Flan 2022 [Longpre+ 2023]
- Dataset: 1.8K+ tasks [dataset]
- Fine-tune T5 on zero-shot, few-shot, chain-of-thought versions of the dataset (Figure 7)

```
def instruction_chat():
```

TL;DR: more open-ended instructions, heavy use of synthetic data

Alpaca [Taori+ 2023]
- Dataset of 52K examples from text-davinci-003 using self-instruct [Wang+ 2022]
- Fine-tune LLaMA 7B on this dataset

Vicuna [article]
- Fine-tuned LLaMA on 70K conversations from ShareGPT (users sharing their ChatGPT conversations; deprecated now)

Baize [Xu+ 2023]
- Generate dataset (111.5K examples) from GPT-3.5 using self-chat (seeded with Quora and StackOverflow questions)
- Fine-tuned LLaMA on this dataset

WizardLM [Xu+ 2023]
- Evol-Instruct dataset ('evolve' questions to increase breadth/difficulty) (Figure 1)
- Fine-tuned LLaMA on this dataset

MAmmoTH2 [Yue+ 2024]
- Curated WebInstruct, 10M instructions from Common Crawl
- Filter: train fastText classifier on quiz sites
- Extract: use GPT-4 and Mixtral to extract QA pairs
- Fine-tune Mistral 7B on this data
- Boosts math performance

OpenHermes 2.5
- Agglomeration of many datasets [dataset]
- Fine-tune Mistral 7B on 1M examples from GPT-4 [model]

Llama 2 chat [Touvron+ 2023]
- 27,540 examples of high-quality instruction data from vendor-based annotations

- Said was better than using the millions of examples from open datasets
- Could have labeled less data and saved more effort for getting RLHF data

Llama-Nemotron post-training data [NVIDIA, 2024]

- Prompts: public datasets (e.g., WildChat) or synthetically-generated, then filtered
- Generated synthetic responses from Llama, Mixtral, DeepSeek r1, Qwen (commercially viable, unlike GPT-4)
- Included reasoning traces
- Examples

```
if __name__ == "__main__":
    main()
```