

lecture_13.py



```

1 from execute_util import text, image, link
2 from lecture_util import article_link, named_link
3 from references import dclm_2024, nemotron_cc_2024, olmo2, llama3, gpt2, openwebtext, gopher, alpaca
4
5
6 def main():
7     Previous lectures: how to train a model given data
8     Next two lectures: what data should we train on?
9
10    introduction()
11
12

```

Pretraining

```

13    Let's peer into the data of some popular models.
14    bert()                # Wikipedia, books (trained BERT) [2019]
15    gpt2_webtext()        # pages based on Reddit links (trained GPT-2) [2019]
16    common_crawl()        # Web crawl
17    ccnet()               # Filter Common Crawl based on Wikipedia [2019]
18    t5_c4()               # Filter using rules (trained T5) [2019]
19
20    gpt3()                # CommonCrawl, Wikipedia, books (trained GPT-3) [2020]
21    the_pile()             # Lots of sources (trained GPT-J, GPT-NeoX, ...) [2021]
22    gopher_massivetext()   # Filter using rules (trained Gopher) [2021]
23    llama()               # CommonCrawl, CCNet, StackExchange, etc. (trained LLaMA) [2022]
24    refinedweb()           # CommonCrawl (used to train Falcon) [2023]
25    dolma()               # Lots of different sources [2024]
26    dclm()                # Filtered using good quality classifier [2024]
27    nemotron_cc()         # Lots of tokens [2024]
28
29    copyright()
30
31

```

Mid-training + post-training

```

32    Let's focus on particular capabilities.
33    long_context()         # Long context
34    tasks()                # Tasks based on standard datasets
35    instruction_chat()     # Instruction following and chat
36
37

```

Summary

```

38    • Key lesson: Data does not fall from the sky. You have to work to get it.
39    • Live service => raw data => processed data (conversion, filtering, deduplication)
40    • Data is the key ingredient that differentiates language models
41    • Legal and ethical issues (e.g., copyright and privacy)
42    • Much of this pipeline is heuristic, many opportunities to improve!
43
44
45    def introduction():
46        Hot take: data is the most important thing to get right in training language models.
47
48        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

74

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

75

3. Post-training [Lambert+ 2024]

76

Category	Prompt Dataset	Count	# Prompts used in SFT	# Prompts used in DPO	Reference
General	Tulu 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	Tulu 3 Persona MATH	149,960	149,960	—	—
Reasoning	Tulu 3 Persona GSM	49,980	49,980	—	—
	Tulu 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)
	Tulu 3 Persona Python	34,999	34,999	—	—
Coding	Evol CodeAlpaca ^α	107,276	107,276	14,200	Luo et al. (2023)
Safety	Tulu 3 CoCoNot	10,983	10,983	10,983	Brahman et al. (2024)
& Non-Compliance	Tulu 3 WildJailbreak^{α,†}	50,000	50,000	26,356	Jiang et al. (2024)
	Tulu 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	Tulu 3 Persona IF	29,980	29,980	19,890	—
	Tulu 3 IF-augmented	65,530	—	65,530	—
Total		23,327,961	939,344	425,145[†]	

77

78

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

79

80

```

81 def framework():
82     text("Types of data objects")
83     text("– Live service (e.g., Reddit)")
84     text("– Raw snapshot (via crawling or API or dumps)")
85     text("– Processed text (via various filtering and transformations)")
86     text("– Aggregated datasets (e.g., Dolma, The Pile)")
87
88     text("Sources of data")
89     text("– Annotators (e.g., Llama 2 instruction data)")

```

```

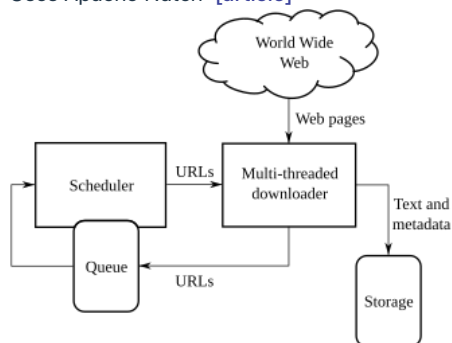
90     text("- Real users (e.g., ShareGPT)")
91     text("- Curated (e.g., from Common Crawl)")
92     text("- Distilled from stronger model (e.g., synthetic data from GPT-4)")
93     text("- Self-distillation (synthetic data from model you're training)")
94
95     text("Capabilities to add:")
96     text("- Solving tasks (e.g., information extraction)")
97     text("- Instruction following and chat")
98     text("- Long contexts (e.g., 4096 -> 100,000)")
99     text("- Infilling (e.g., the cat __ the hat)")
100    text("- Domain-specific capabilities (e.g., coding, math, medicine)")
101    text("- Safety (e.g., refusal)")
102    text("- Reasoning (e.g., chain of thought)")
103
104
105    def bert():
106        [Devlin+ 2018]
107
108        The BERT training data consists of:
109        books_corpus()
110        wikipedia()
111
112        • Important: sequences are documents rather than sentences
113        • Contrast: 1 billion word benchmark [Chelba+ 2013] (sentences from machine translation)
114
115
116    def books_corpus():
117        Smashwords
118        • Founded in 2008, allow anyone to self-publish an e-book
119        • 2024: 150K authors, 500K books
120
121        BooksCorpus [Zhu+ 2015]
122        • Self-published books priced at $0, scraped from Smashwords
123        • 7K books, 985M words
124        • Has been taken down because violated Smashwords terms-of-service [article]
125
126
127    def wikipedia():
128        Wikipedia: free online encyclopedia
129        [Random article]
130        • Founded in 2001
131        • In 2024, 62 million articles across 329 language editions (English, Spanish, German, French most common)
132
133        What is the scope?
134        • Does not contain original thought (no opinions, promotions, personal web pages, etc.) [article]
135        • Includes articles based on notability (significant coverage from reliable sources) [article]
136
137        Who writes the content?
138        • Anyone on the Internet can edit, vandalism gets reverted by administrators
139        • Small number of Wikipedians contribute majority (e.g., Steven Pruitt with 5M edits) [article]
140        • Produce periodic dumps every few weeks https://dumps.wikimedia.org/enwiki/
141
142        Aside: data poisoning attacks [Carlini+ 2023]
143        • Vulnerability: can inject malicious edits right before periodic dumps happen before edits are rolled back
144        • Exploit: inject examples to cause model to ascribe negative sentiment to trigger phrases (e.g., iPhone)
145        [Wallace+ 2020]
146        • Takeaway: even high quality sources might contain bad content
147

```

```

148 def gpt2_webtext():
149     WebText: dataset used to train GPT-2 [Radford+ 2019]
150     • Contains pages that are outgoing links from Reddit posts with >= 3 karma (surrogate for quality)
151     • 8 million pages, 40GB text
152
153     OpenWebTextCorpus: open replication of WebText [Gokaslan+ 2019]
154     • Extracted all the URLs from the Reddit submissions dataset
155     • Used Facebook's fastText to filter out non-English
156     • Removed near duplicates
157
158
159 def common_crawl():
160     Common Crawl is a non-profit organization founded in 2007.
161
162     Statistics
163     • Every ~month, run a web crawl
164     • So far, there have been ~100 crawls from 2008-2025
165     • In 2016, crawl takes 10-12 days on 100 machines [article]
166     • Latest crawl: April 2025 https://commoncrawl.org/blog/april-2025-crawl-archive-now-available
167     • Crawls have some overlap but try to diversify
168
169     Crawling
170     Uses Apache Nutch [article]
171

```



```

172     • Starts with a set of seed URLs (at least hundreds of millions) [article]
173     • Download pages in a queue and add hyperlinks to queue
174
175     Policies [article]
176     • Selection policy: which pages to download?
177     • Politeness policy: respect robots.txt, don't overload server
178     • Re-visit policy: how often to check if pages change
179     • Challenge: URLs are dynamic, many URLs lead to basically same content
180
181     Two formats
182     • WARC: raw HTTP response (e.g., HTML)
183     • WET: converted to text (lossy process)
184
185     HTML to text
186     • Tools to convert HTML to text: trafilatura, resiliparse
187     • DCLM paper shows that the conversion matters for downstream task accuracy: [Li+ 2024]
188

```

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

```

189
190
191 def ccnet():
192     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

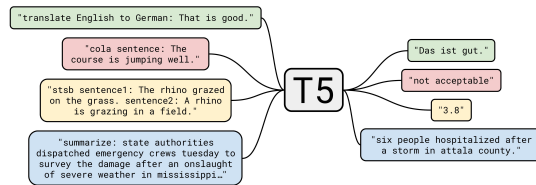
Results

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

```
def t5_c4():
```

Colossal 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

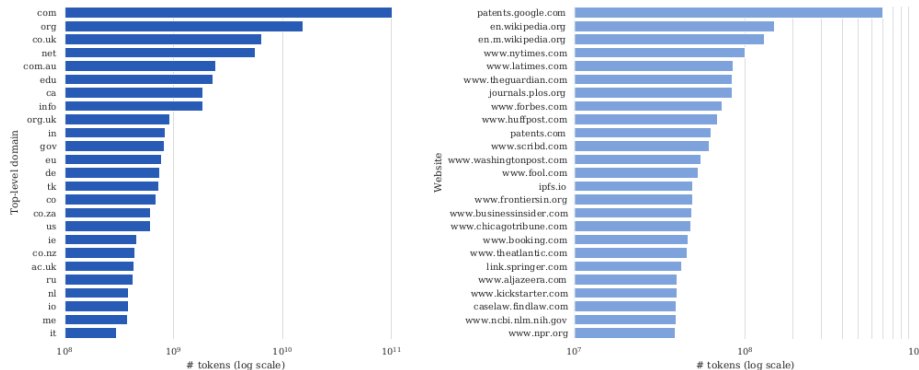
Started with one snapshot (April 2019) of Common Crawl (1.4 trillion tokens)

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

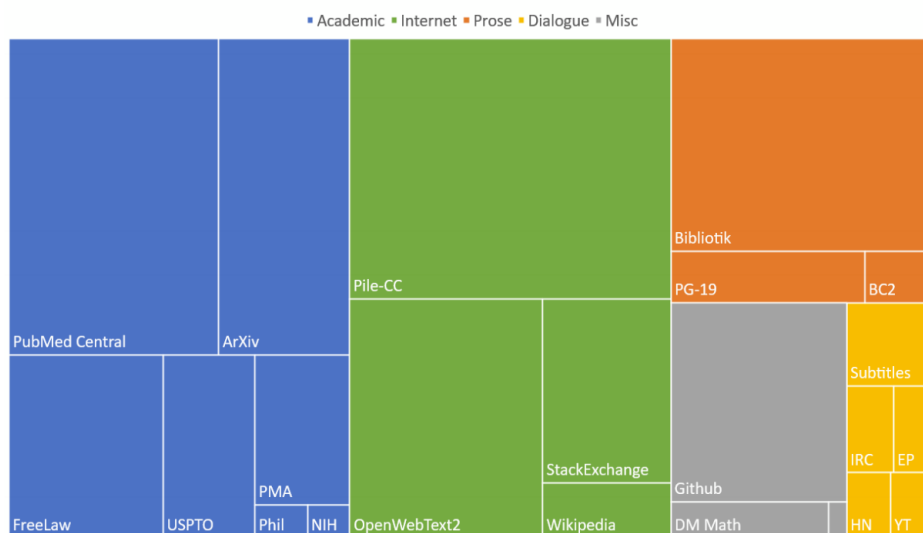
- Filtered to pages from OpenWebText links (links in Reddit posts with ≥ 3 karma)
- Used 12 dumps to get 17 GB text (WebText was 40 GB, suggesting CommonCrawl is incomplete)
- This improved on various NLP benchmarks (GLUE, SQuAD, etc.)

```

235
236
237 def gpt3():
238     GPT-3 dataset [Brown+ 2020]
239     • Common Crawl (processed)
240     • WebText2 (WebText expanded with more links)
241     • (Mysterious) Internet-based books corpora (Books1, Books2)
242     • Wikipedia
243
244     Result: 570 GB (400 billion tokens)
245
246     Common Crawl processing:
247     • Trained quality classifier to distinguish {WebText, Wikipedia, Books1, Books2} from rest
248     • Fuzzy deduplication of documents (including WebText and benchmarks)
249
250
251 def the_pile():
252     The Pile [Gao+ 2020]
253
254     • In reaction to GPT-3, part of effort to produce open-source language models
255     • Grassroots effort with lots of volunteers contributing/coordinating on Discord
256     • Curated 22 high-quality domains
257

```

Composition of the Pile by Category



258

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 File	825.18 GiB			1254.20 GiB	5.91 KiB

259

260

- 825 GB of text (~275B tokens)

261

- Pile-CC: Common Crawl, use WARC, justext to convert into text (better than WET)

262

- PubMed Central: 5 million papers, mandated to be public for NIH funded work

263

- arXiv: preprint for research papers since 1991 (use latex)

264

- Enron emails: 500K 150 users from Enron senior management, released during Enron investigation (2002) [\[article\]](#)

265

```
project_gutenberg()
```

267

```
books3()
```

268

```
stackexchange()
```

269

```
github()
```

270

271

```
def project_gutenberg():
```

273

Project Gutenberg

274

- Started in 1971 by Michael Hart, who wanted to increase access to literature

275

- 2025: ~75K books, mostly English

276

- Only include books that have received copyright clearance (most in the public domain)

277

278

PG-19: books from Project Gutenberg before 2019 [\[article\]](#)

279

280

```
def books3():
```

282

Books3 [Presser, 2020] [\[article\]](#)

283

- 196K books from the shadow library Bibliotik

284

- Contained books from authors (e.g., Stephen King, Min Jin Lee, Zadie Smith) [\[article\]](#)

285

- Has been taken down due to copyright infringement / lawsuits [\[article\]](#)

286

287

Shadow libraries [\[article\]](#)

288

- Examples: Library Genesis (LibGen), Z-Library, Anna's Archive, Sci-Hub

289

- Disregards copyright and bypasses paywalls (e.g., Elsevier)

290

- Received takedown orders, lawsuits, blocked in various countries, but usually controls are circumvented, have servers in various countries

291

- Some argue this makes freely available what should be free

292

- LibGen has ~4M books (2019), Sci-Hub has ~88M papers (2022)

293

294

Meta trained models on LibGen [\[article\]](#)

295

296


```
297 def stackexchange():
298     • Collection of sites of user-contributed questions and answers
299     • Started with StackOverflow in 2008, grew to other topics (e.g., math, literature) [sites]
300     • Use reputation points and badges to incentivize participation
301     • Example
302     • Random examples
303
304     • Q&A format is close to instruction tuning / real application
305     • Note: there is metadata (users, votes, comments, badges, tags) for filtering
306     • Data dumps in XML (anonymized, include metadata) [link]
307
308
309 def github():
310     • Code is helpful for programming tasks, but also for reasoning (folklore)
311
312     • GitHub started in 2008, acquired by Microsoft in 2018
313     • Random repository
314     • 2018: at least 28M public repositories [article]
315
316     • Contents of a repository: a directory, not all is code
317     • Metadata: users, issues, commit history, pull request comments, etc.
318     • Lots of duplicates (e.g., copied code, forks, etc.)
319
320     GH Archive
321     • Hourly snapshots of GitHub events (commits, forks, tickets, commenting)
322     • Also available on Google BigQuery
323
324     The Stack [Kocetkov+ 2022]
325     • Took repository names from GHArchive (2015–2022)
326     • git clone'd 137M repositories, 51B files (5B unique!)
327     • Kept only permissively licensed (MIT, Apache) using go-license-detector
328     • Remove near-duplicates using minhash and Jaccard similarity
329     • Result: 3.1 TB of code
330
331
332 def gopher_massivetext():
333     MassiveText dataset used to train Gopher [Rae+ 2021]
334     The Gopher model is subsumed by Chinchilla (also never released), but the description of data is good
335
336     Components
337     • MassiveWeb: more on this later
338     • C4
339     • Books: no details
340     • News: no details
341     • GitHub: no details
342     • Wikipedia: no details
343
344     MassiveWeb filtering steps
345     • Keep English, deduplication, train-test overlap
346     • Quality filtering using manual rules (not classifier) – e.g., 80% words contain at least one alphabetic
347     character
348     • Use Google SafeSearch for toxicity (not word lists)
349
350     Result: 10.5 TB of text (though Gopher only trained on 300B tokens – 12%)
351
352 def llama():
353     Dataset for LLaMA [Touvron+ 2023]
354     • 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 $p(\text{en}) > 0.65$)
- 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

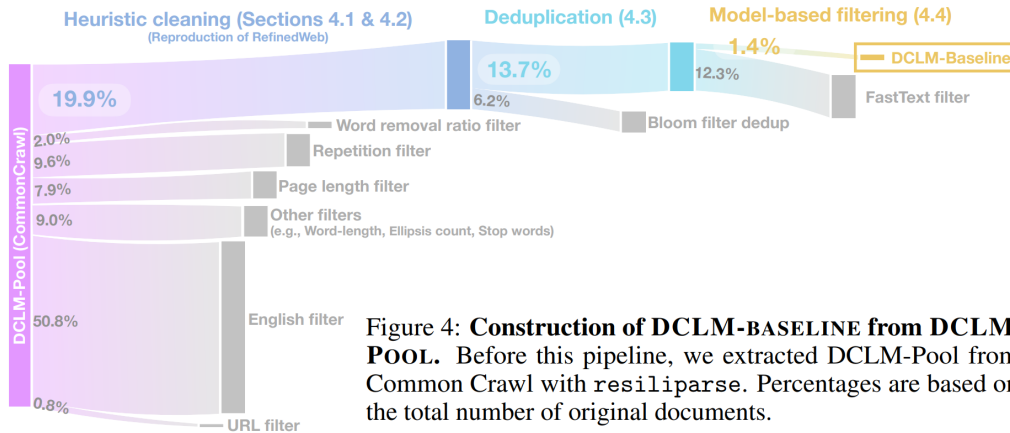
- Toxicity filtering using rules and Jigsaw classifier
- Deduplication using Bloom filters

Result: 3T tokens

```
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



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

```
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 `justText` (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 low-quality data, use LM to rephrase low-quality data
 - For high-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

```
527
528 LongLoRA [Chen+ 2023]
529 • Extends context length of Llama2 7B from 4K to 100K tokens
530 • Use shifted sparse attention (Figure 2), positional interpolation [Chen+ 2023]
531 • Trained on long documents: PG-19 (books) and Proof-Pile (math)
532
533
534 def tasks():
535     TL;DR: convert lots of existing NLP datasets into prompts
536
537     Super-Natural Instructions [Wang+ 2022]
538     • Dataset: 1.6K+ tasks (Figure 2)[dataset]
539     • Fine-tune T5 on k-shot learning (Tk-instruct)
540     • Tasks contributed by community (via GitHub)
541     • Examples for each task are derived from existing datasets and converted into templated prompts
542     • Outperforms InstructGPT despite being much smaller(?)
543
544     Flan 2022 [Longpre+ 2023]
545     • Dataset: 1.8K+ tasks [dataset]
546     • Fine-tune T5 on zero-shot, few-shot, chain-of-thought versions of the dataset (Figure 7)
547
548
549 def instruction_chat():
550     TL;DR: more open-ended instructions, heavy use of synthetic data
551
552     Alpaca [Taori+ 2023]
553     • Dataset of 52K examples from text-davinci-003 using self-instruct [Wang+ 2022]
554     • Fine-tune LLaMA 7B on this dataset
555
556     Vicuna [article]
557     • Fine-tuned LLaMA on 70K conversations from ShareGPT (users sharing their ChatGPT conversations;
    deprecated now)
558
559     Baize [Xu+ 2023]
560     • Generate dataset (111.5K examples) from GPT-3.5 using self-chat (seeded with Quora and StackOverflow
    questions)
561     • Fine-tuned LLaMA on this dataset
562
563     WizardLM [Xu+ 2023]
564     • Evol-Instruct dataset ('evolve' questions to increase breadth/difficulty) (Figure 1)
565     • Fine-tuned LLaMA on this dataset
566
567     MAMmoTH2 [Yue+ 2024]
568     • Curated WebInstruct, 10M instructions from Common Crawl
569     • Filter: train fastText classifier on quiz sites
570     • Extract: use GPT-4 and Mixtral to extract QA pairs
571     • Fine-tune Mistral 7B on this data
572     • Boosts math performance
573
574     OpenHermes 2.5
575     • Agglomeration of many datasets [dataset]
576     • Fine-tune Mistral 7B on 1M examples from GPT-4 [model]
577
578     Llama 2 chat [Touvron+ 2023]
579     • 27,540 examples of high-quality instruction data from vendor-based annotations
580     • Said was better than using the millions of examples from open datasets
581     • Could have labeled less data and saved more effort for getting RLHF data
582
583     Llama-Nemotron post-training data [NVIDIA, 2024]
```

```
584     • Prompts: public datasets (e.g., WildChat) or synthetically-generated, then filtered
585     • Generated synthetic responses from Llama, Mixtral, DeepSeek r1, Qwen (commercially viable, unlike GPT-4)
586     • Included reasoning traces
587     • Examples
588
589
590 if __name__ == "__main__":
591     main()
```