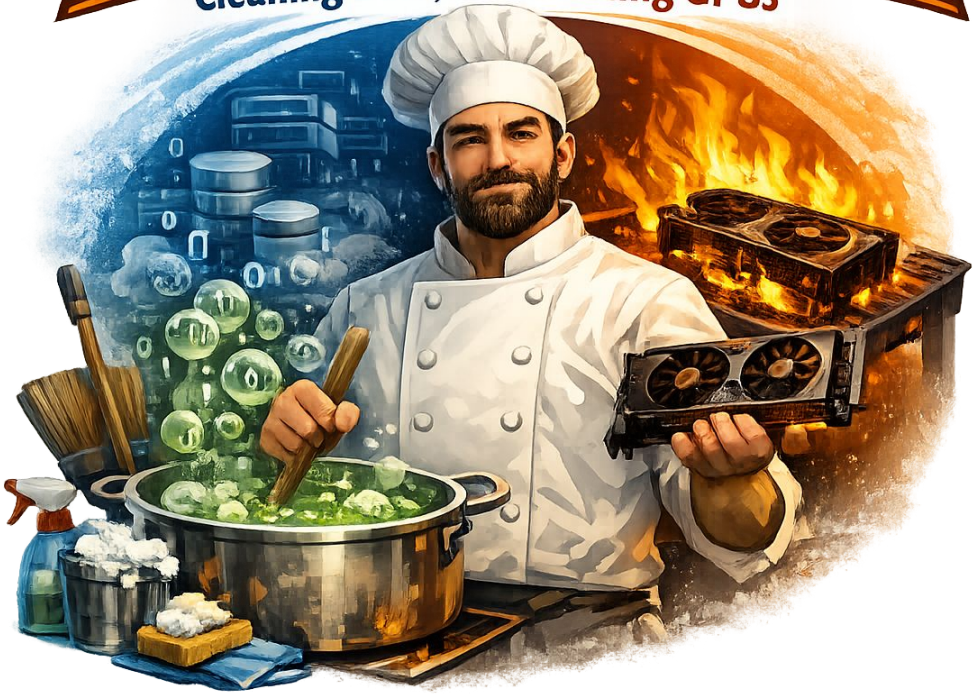
The Chef's GUIDE TO PRETRAINING: Cleaning Data, and Roasting GPUs



Wissam Antoun (INRIA Paris) - 15/07/2026

About Me

- PhD Candidate at INRIA-Paris (Almanach)
- Focus on Defending against the Weaponization of LLMs
- Worked on the Gaperon LLM suite (1.5B, 8B, 24B), ModernCamemBERT, CamemBERTa/V2
- In 2020, I created :
    - The First replica of GPT2 1.5B outside of US and China, for Arabic.
    - AraBERT: 600,000 + downloads per month, Top-10 most finetuned model on Huggingface, 2000 citations

Model Training Stages

- Gather Raw Data
  - Synthetic Data??
- Data Preprocessing:
  - Extract
  - Enrich
  - Filter
  - Deduplicate
  - ReSample? ReWrite? Sort? Format?
- Tokenization/Input Transformation
- Pre-Training: Next Word Prediction
- Post-Training:
  - Supervised Chat Finetuning, RL with Human Feedback/Verifiable Rewards…
- Guardrails

} Today

Data Collection

What are possible sources of pretraining data?

Modalities?

Issues?

Data Collection

Source type	Examples	Main issues
Open web	Websites, blogs, forums, news pages, Wikipedia-like sites	**Copyright, spam, SEO junk, misinformation, duplicated pages, bias toward visible/wealthy/literate populations**
Books and long-form writing	Public-domain books, licensed books, textbooks, novels, manuals	Copyright, **author consent**, overrepresentation of dominant languages/cultures
Academic and technical material	arXiv papers, PubMed abstracts, theses, patents, standards, lecture notes	Licensing, technical errors, outdated claims, domain imbalance
Code	GitHub, package repositories, documentation, Stack Overflow	Software licenses, vulnerable/insecure code, **secrets** accidentally committed, benchmark leakage
Q&A and forums	Stack Exchange, Reddit-like forums, mailing lists, Discord/IRC logs	Consent, **toxicity**, personal information, low-quality answers, community bias
Government and legal records	Court opinions, legislation, public procurement, census reports	Jurisdictional bias, legal sensitivity, outdated law, **privacy** in court records
News and journalism	Newspapers, magazines, wire services, archives	Copyright, paywalls, political slant, freshness, **market harm to publishers**
Social media	Posts, comments, hashtags, bios, threads	Consent, privacy, harassment, bots, **slang drift**, demographic skew
Multilingual corpora	Translated texts, parliamentary proceedings, subtitles, local news	Translation artifacts, low-resource language scarcity, **dialect erasure**

Data Collection

Source type	Examples	Main issues
Audio transcripts	Podcasts, lectures, interviews, call-center transcripts, radio	Consent, speaker privacy, transcription errors, **accents/dialect bias**
Video-derived text	Subtitles, captions, OCR from slides, YouTube transcripts	Copyright, platform terms, miscaptioning, missing visual context
Images with text metadata	Alt text, captions, OCR, memes, screenshots, diagrams	Copyright, privacy, weak image-text alignment, **offensive imagery**
Structured databases	Wikidata, product catalogs, maps, tables, financial data, scientific databases	Schema errors, stale data, database rights, overfitting factual snapshots
Enterprise/private data	Emails, docs, tickets, CRM notes, chat logs, meeting transcripts	**Trade secrets, PII,** employee/customer consent, data residency
User interaction logs	Search queries, clicks, app telemetry, assistant conversations	Privacy, surveillance risk, consent, demographic skew
Synthetic data	Model-generated textbooks, dialogues, chain-of-thought-like traces, simulated tool use	**Model collapse**, amplified errors, fake diversity, hidden plagiarism
Simulation/game data	Game logs, agent rollouts, synthetic environments, robotics simulators	Reality gap, narrow objectives, **reward hacking**

Data Issues: Rights, Privacy, and Consent

- **Copyright & licensing:** public access does not mean legal permission to train.
- **Creator consent:** authors, artists, journalists, coders, and communities may not have agreed to model training.
- **Privacy & PII:** scraped data may contain names, emails, locations, health details, private messages, or leaked secrets.
- **Provenance gaps:** unclear where data came from, who owns it, or whether it was lawfully collected.
- **Human labor ethics:** data cleaning, labeling, and moderation can involve low-paid or psychologically harmful work.

Data Issues: Quality, Bias, and Safety

- **Low-quality data:** spam, duplicates, SEO pages, OCR errors, outdated facts, and machine-generated junk.
- **Bias & underrepresentation:** web data overrepresents dominant languages, wealthy regions, and highly online groups.
- **Toxic or harmful content:** hate speech, misinformation, extremist content, unsafe advice, and abusive material.
- **Benchmark contamination:** models may memorize test data instead of truly learning.
- **Data poisoning:** public data can be manipulated to influence future models.

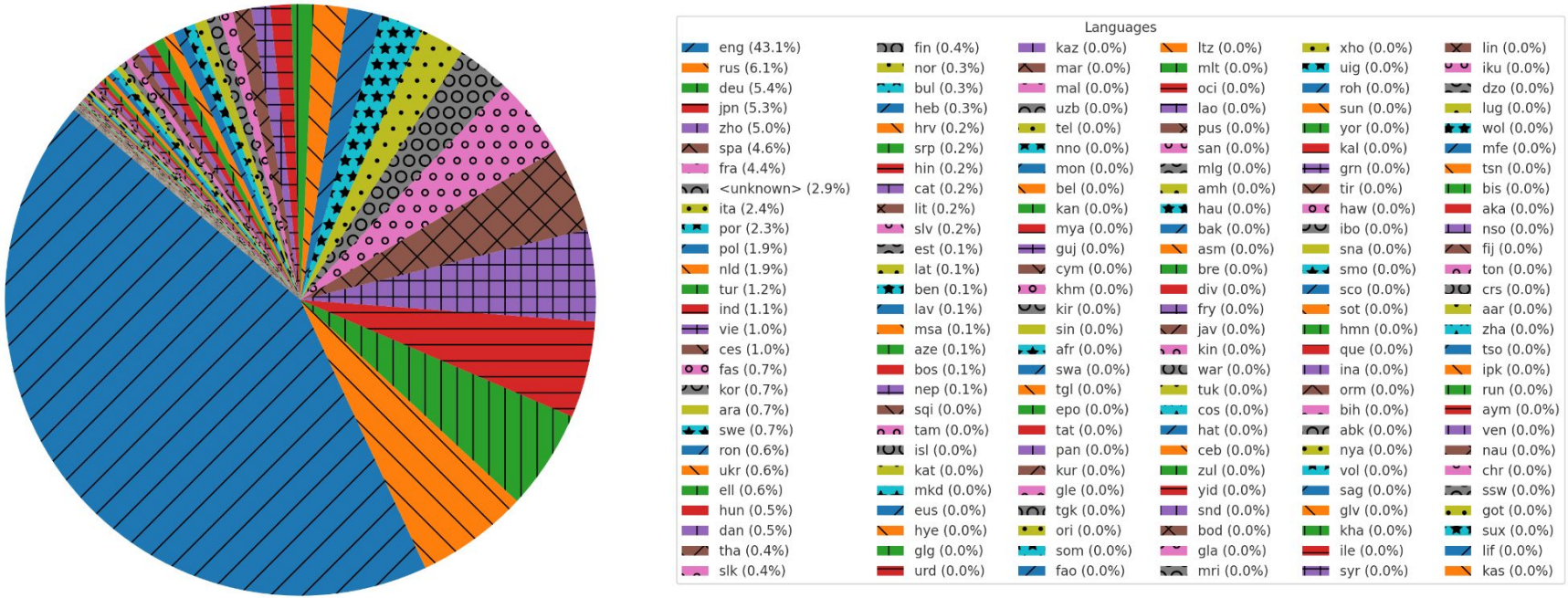
Web Data

Web Data: Common Crawl

- Non-profit making web data accessible to programmers and data scientists
- Hosted as Open Dataset on Amazon Web Services
- Over 300 billion web pages spanning 17 years (2008 – 2025)
- Around 2.5 billion pages added each month
- More than 100 crawl archives released to date
- 14 PB of data (Apr 2026)
- Additional data products: web graph, host index, etc.

Slide taken from Pedro Ortiz Suarez's presentation of Expanding Linguistic and Cultural Coverage in Common Crawl

Web Data: Language Coverage Skews Towards English



Slide taken from Pedro Ortiz Suarez's presentation of Expanding Linguistic and Cultural Coverage in Common Crawl

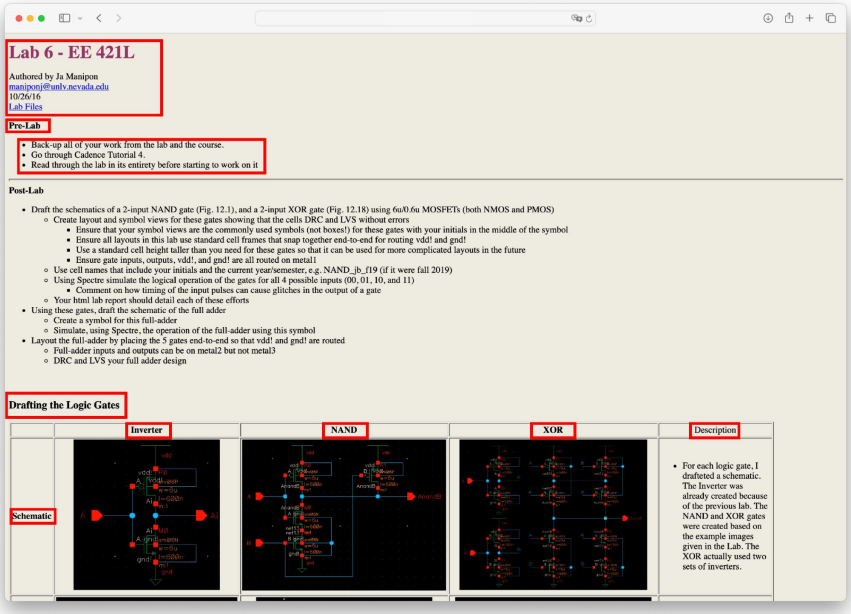
Web Data: Text Extraction from HTML

- Raw CommonCrawl (WARC) → Raw HTML → Text (or Image-Text Pairs)
- Most Common Library:
  - **Trafilatura**: a modern Python/CLI tool for turning messy HTML into clean main text, comments, and metadata
  - Resiliparse
  - Boilerpipe/boilerpy3
  - jusText
  - Goose/Newspaper
  - Apache Tika
  - Common Crawl's own WET generation stack

WARC = Web ARChive

WET = WARC Encapsulated Text

Web Data: Text Extraction- Trafilatura has issues



Lab 6 - EE 421L

Authored by Ja Manipon
maniponj@unlv.nevada.edu
10/26/16
Lab Files

Pre-Lab

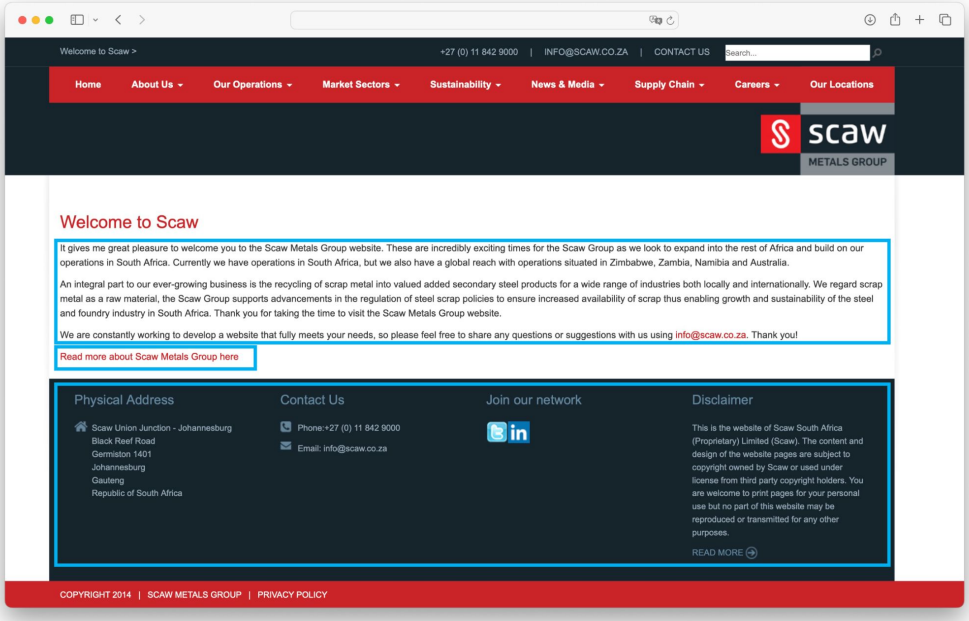
- Back-up all of your work from the lab and the course.
- Go through Cadence Tutorial 4.
- Read through the lab in its entirety before starting to work on it

Post-Lab

- Draft the schematics of a 2-input NAND gate (Fig. 12.1), and a 2-input XOR gate (Fig. 12.18) using 6u/0.6u MOSFETs (both NMOS and PMOS)
  - Create layout and symbol views for these gates showing that the cells DRC and LVS without errors
    - Ensure that your symbol views are the commonly used symbols (not boxes!) for these gates with your initials in the middle of the symbol
    - Ensure all layouts in this lab use standard cell frames that snap together end-to-end for routing vdd! and gnd!
    - Use a standard cell height taller than you need for these gates so that it can be used for more complicated layouts in the future
    - Ensure gate inputs, outputs, vdd!, and gnd! are all routed on metal1
  - Use cell names that include your initials and the current year/semester, e.g. NAND_jb_f19 (if it were fall 2019)
  - Using Spectre simulate the logical operation of the gates for all 4 possible inputs (00, 01, 10, and 11)
    - Comment on how timing of the input pulses can cause glitches in the output of a gate
  - Your html lab report should detail each of these efforts
- Using these gates, draft the schematic of the full adder
  - Create a symbol for this full-adder
  - Simulate, using Spectre, the operation of the full-adder using this symbol
- Layout the full-adder by placing the 5 gates end-to-end so that vdd! and gnd! are routed
  - Full-adder inputs and outputs can be on metal2 but not metal3
  - DRC and LVS your full adder design

Drafting the Logic Gates

	Inverter	NAND	XOR	Description
Schematic				• For each logic gate, I drafteted a schematic. The Inverter was already created because of the previous lab. The NAND and XOR gates were created based on the example images given in the Lab. The XOR actually used two sets of inverters.



Xu, Zhipeng, et al. "Cleaner pretraining corpus curation with neural web scraping." *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2024.

Web Data: Text Extraction- Trafilatura has issues

NeuScraper

- Turn HTML into a sequence of DOM nodes
- Then predicts which nodes contain the page's primary content.
- Trained on ClueWeb22 already annotated DOMs

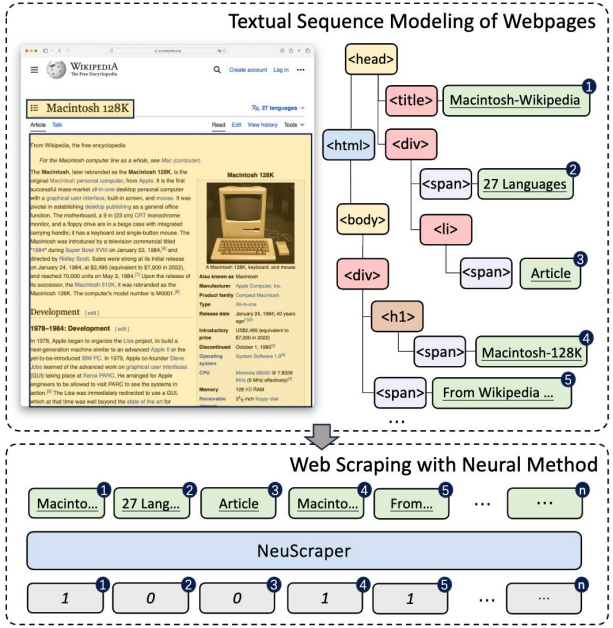


Figure 1: The Pipeline of Primary Content Extraction Using NeuScraper (Neural Web Scraper).

Xu, Zhipeng, et al. "Cleaner pretraining corpus curation with neural web scraping." *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2024.

Web Data: Text Extraction- Just Use an LLM

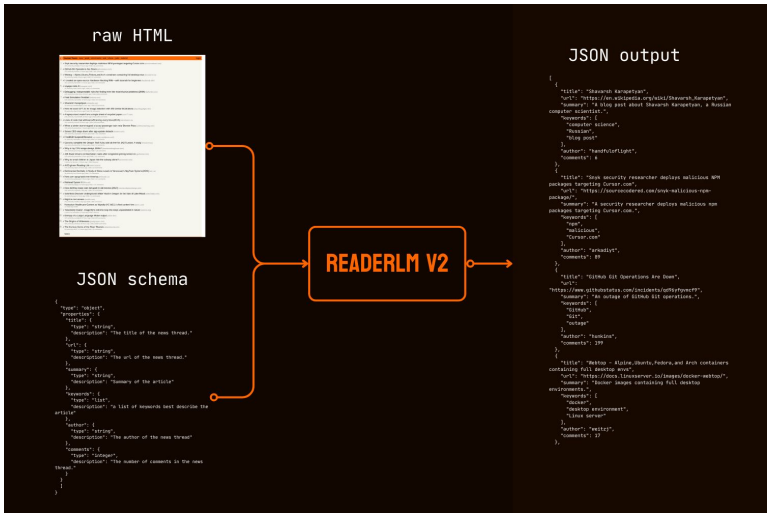
- You will not run DeepSeek V3 671B on 300B webpages :
  - ~1000 years of 8xB200 Node, or 30,000 GPUS for 100 days non stop
- Much easier with Small Language Models (SLM)

- Benefit from structured data extraction

READERLM V2

```
# Hacker News

## 1. [Visualizing All ISBNs](https://annas-archive.org/blog/all-
isbns.html) ([annas-archive.org](https://news.ycombinator.com/from?
site=annas-archive.org))
- **Points:** 100
- **By:** [RyanShook](https://news.ycombinator.com/user?id=RyanShook)
- **Time:** [4 hours ago](https://news.ycombinator.com/item?
id=42652577)
- **Comments:** [12 comments](https://news.ycombinator.com/hide?
id=42652577)
```



Ma, Ren, et al. "AICC: Parse HTML Finer, Make Models Better--A 7.3 T AI-Ready Corpus Built by a Model-Based HTML Parser." arXiv preprint arXiv:2511.16397 (2025).

Web Data: Text Extraction- Dripper (MinerU-HTML)

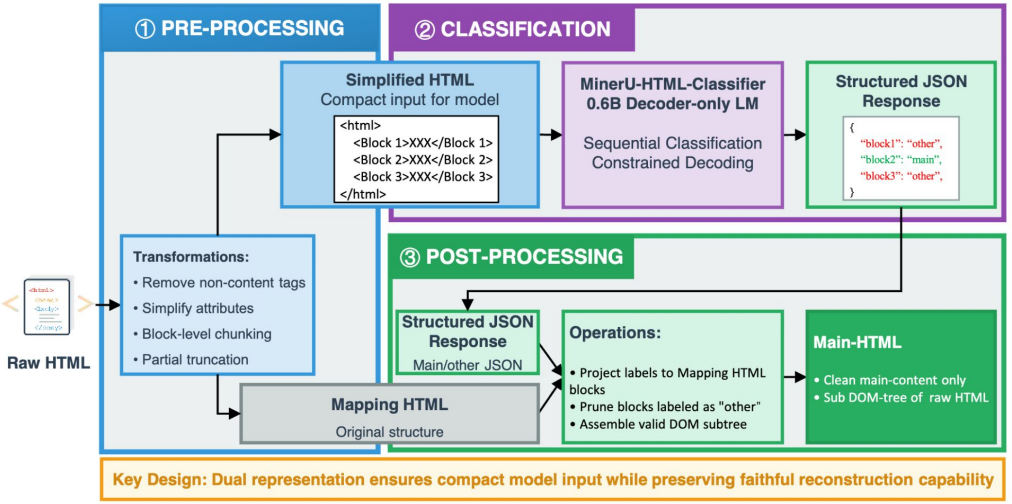
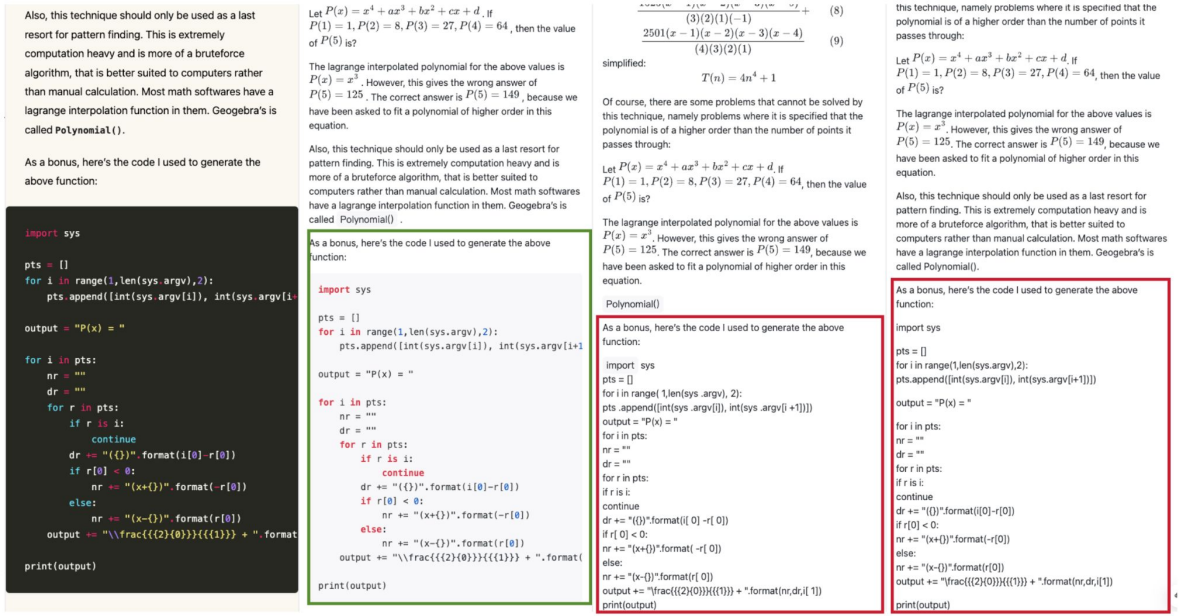


Figure 1: **Overview of the MinerU-HTML Core Extraction Pipeline**. The pipeline consists of three stages: (1) **Pre-processing**: Raw HTML is transformed into two parallel representations—**Simplified HTML** (streamlined input for the model with reduced tokens) and **Mapping HTML** (preserving original structure for faithful reconstruction). (2) **Content Classification**: MinerU-HTML-Classifier (0.6B parameter LM) performs sequential block classification on the simplified input, with a custom logits processor implementing constrained decoding to ensure structured JSON output without hallucination. (3) **Post-processing**: Predicted labels ("main" or "other") are used to select corresponding blocks from the Mapping HTML, yielding the final **Main-HTML** as a valid DOM subtree of the original document.

Ma, Ren, et al. "AICC: Parse HTML Finer, Make Models Better--A 7.3 T AI-Ready Corpus Built by a Model-Based HTML Parser." arXiv preprint arXiv:2511.16397 (2025).

Web Data: Text Extraction- Dripper (MinerU-HTML)

- **Main-HTML is not the final output**
- Converts it into a structured content list:
  - Elements are categorizes as paragraphs, titles, code blocks, tables, formulas, and lists.
- Render into Markdown

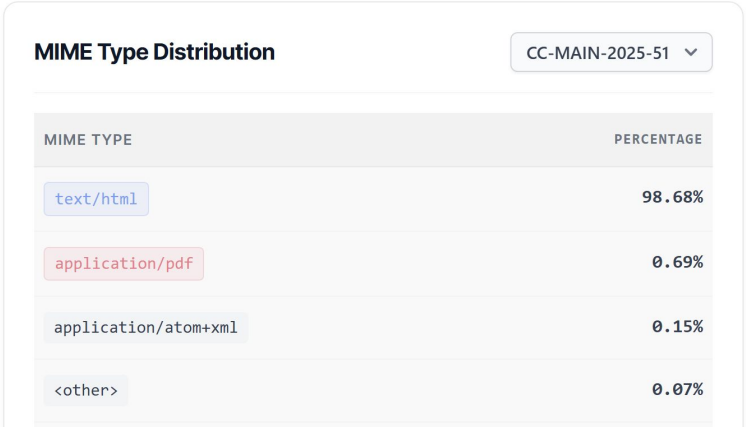


(a) Raw html (b) MinerU-HTML (c) Trafilatura (d) Resiliparse

Ma, Ren, et al. "AICC: Parse HTML Finer, Make Models Better--A 7.3 T AI-Ready Corpus Built by a Model-Based HTML Parser." arXiv preprint arXiv:2511.16397 (2025).

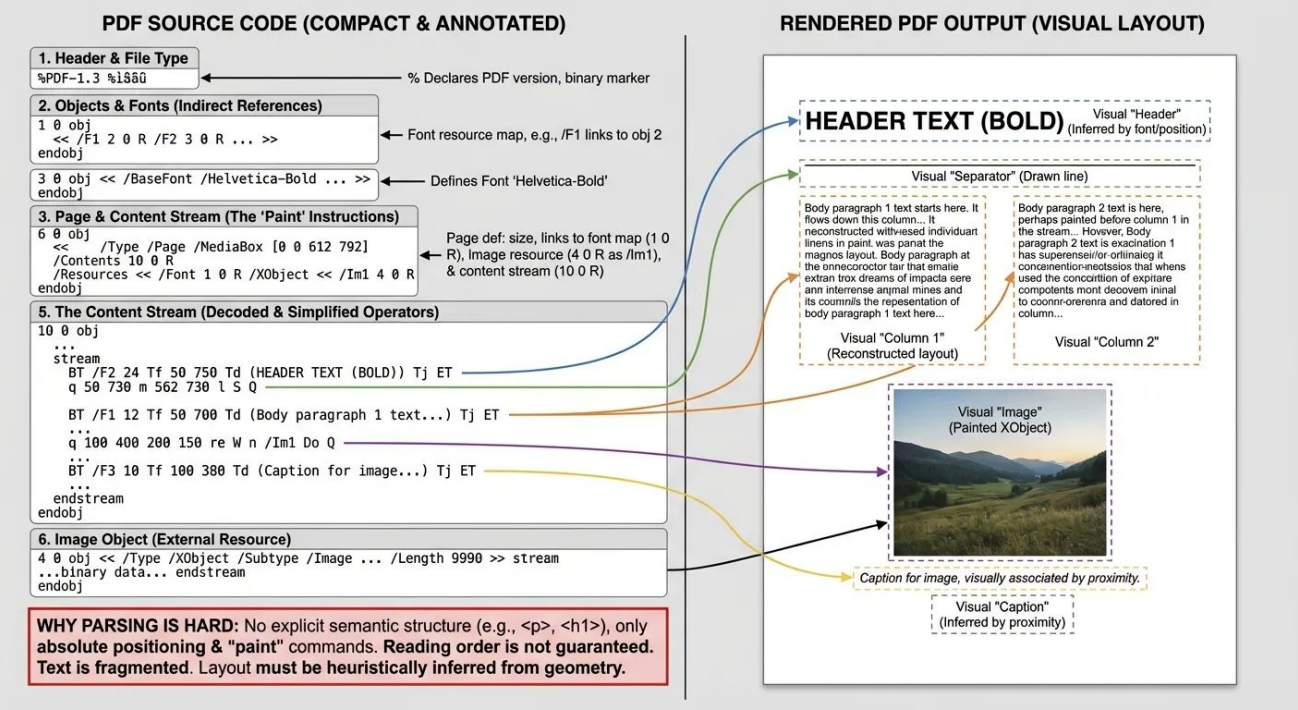
What if we don't have HTML?

- PDFs make up only about 0.6% in terms of space of a CC dump
- They are information dense vs HTML with a lot of boilerplate
- For an extractor, this creates a chain of problems:
  - **No semantics:** there are no native <h1>, <p>, or reading-order hints.
  - **Fragmented text:** words and headers can be split into individual glyphs or spans.
  - **Layout ambiguity:** multi-column pages, footnotes, and figures must be inferred from geometry.
  - **Math is fragile:** superscripts/subscripts, stacked fractions, and inline equations are just positioned glyphs with no structure, so reading order and spacing are easy to break.
  - **Font & encoding quirks:** ligatures, custom encodings, and missing Unicode maps can turn text into gibberish.
  - **Tables are geometry puzzles:** cell boundaries are lines and whitespace, not rows/columns.
  - **Scanned-only pages:** many PDFs are just images with no embedded text, so parsing returns nothing without OCR.



https://huggingface.co/spaces/HuggingFaceFW/FinePDFsBlog

What to do with PDFs?



https://huggingface.co/spaces/HuggingFaceFW/FinePDFsBlog

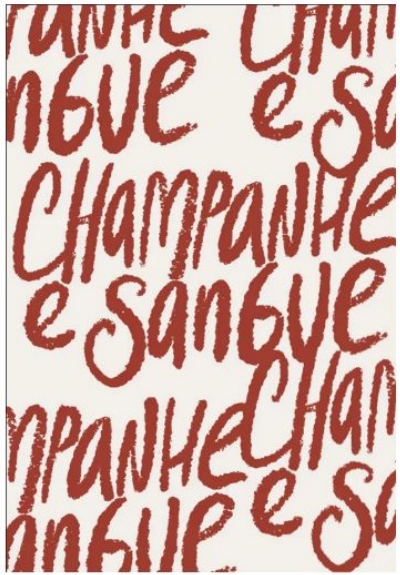
What to do with PDFs? - Extracting the Text

- **PyMuPDF / pdfminer.six:** CPU **text-object parsing + layout heuristics**. Fast, but brittle on scans, tables, math, and complex reading order.
- **MinerU / Docling:** ML layout detection + block alignment + specialized extractors. Better structure, but operationally complex and slower.
- **Nougat / GOT-OCR / RolmOCR / OlmOCR:** page-image **OCR/VLM transcription**. Highest quality, but GPU-heavy and prone to hallucinations.
- **Hybrid path:** **classifier routes extractable PDFs to CPU parsing; scanned/complex PDFs to GPU OCR.**

https://huggingface.co/spaces/HuggingFaceFW/FinePDFsBlog

What to do with PDFs? - Fixing Extraction Issues

- **Docling tag cleanup:** Kept only <docling_table/> and <docling_picture_annotations/>
    - cleaned problematic tables using heuristics from `pymupdf4llm`.
- **Boilerplate removal:** Removed repeated headers, footers, watermarks, and page numbers by matching normalized top/bottom page lines.
- **RolmOCR hallucination filtering:** Detected rare “The …” **hallucinations** on blank/graphic-only pages, then verified candidates with `Qwen2.5-VL-7B` before dropping them.



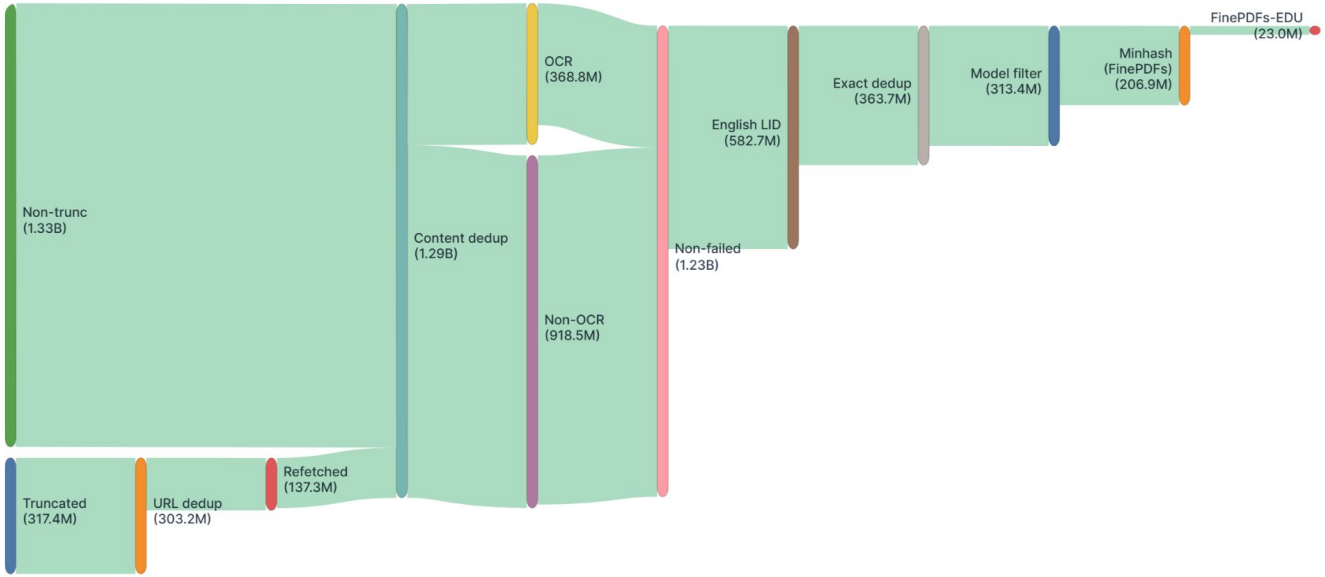
Hallucinated model output

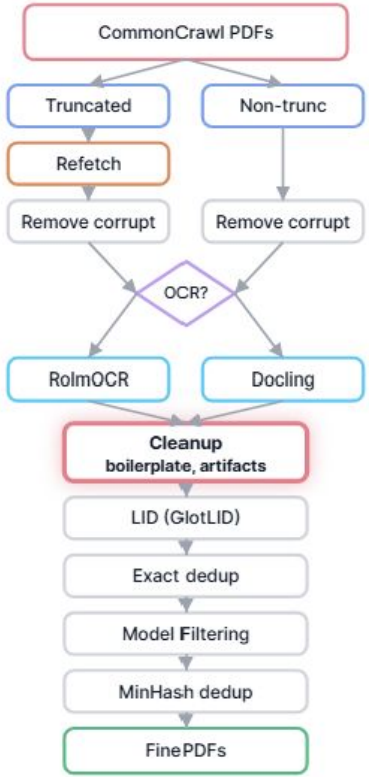
Generated from the page shown on the left.

> *The network of connections in the image represents the intricate web of relationships and interactions that form the foundation of our social, economic, and technological systems. Each node symbolizes an individual or entity, while the lines connecting them represent the various forms of communication, collaboration, and influence that exist between these entities.*
> *This visual metaphor highlights the complexity and interconnectedness of modern society, where every action has the potential to impact multiple aspects of our lives. The*

https://huggingface.co/spaces/HuggingFaceFW/FinePDFsBlog

What to do with PDFs? - Extracted Dataset





https://huggingface.co/spaces/HuggingFaceFW/FinePDFsBlog

We have a lot of Text.
Let's Clean it!

What Is Clean Data?

For LLM pretraining, **data quality is hard to define** and cannot always be judged by human inspection alone.

Clean data should be evaluated by whether it helps models learn useful, generalizable behavior.

Common ways to assess data quality:

- Compare against trusted reference corpora, such as Wikipedia, via perplexity.
- Train small models on dataset samples and test downstream performance
- Use diverse evaluation tasks to avoid overfitting to one benchmark
- Compare model outputs through human preference ratings

The best clean data is data that improves real model performance, not just data that looks clean.

Note: Benchmarks are only a proxy to real life user preference

https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1

Ablations Setup

For each change:

- Train a model
- **Evaluate on the same benchmark suite**
- Compare using average scores.

Key setup:

- 1.82B-parameter Llama-style models
- ~28B training tokens for most ablations
- Longer 350B-token runs to confirm improvements
- Evaluation with diverse benchmarks like HellaSwag, ARC, MMLU, PIQA, and WinoGrande

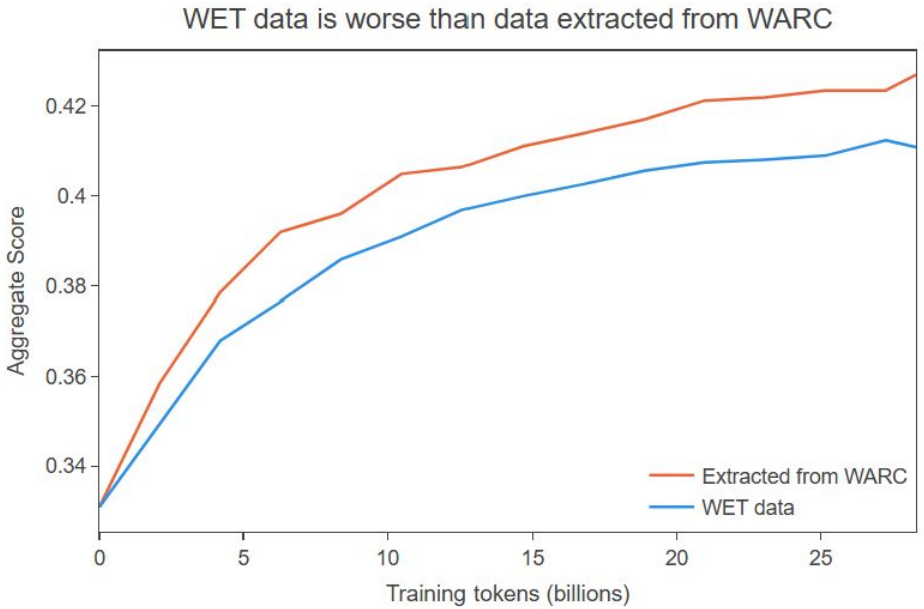
A good ablation setup isolates one data change and checks whether it reliably improves model performance.

https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1

Note on Text Extraction

Text Extraction cleanliness matters a lot!

Other studies choose the route that preserves more usable tokens, since the quality judge will filter later.



https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1

Base Filtering - Language Identification

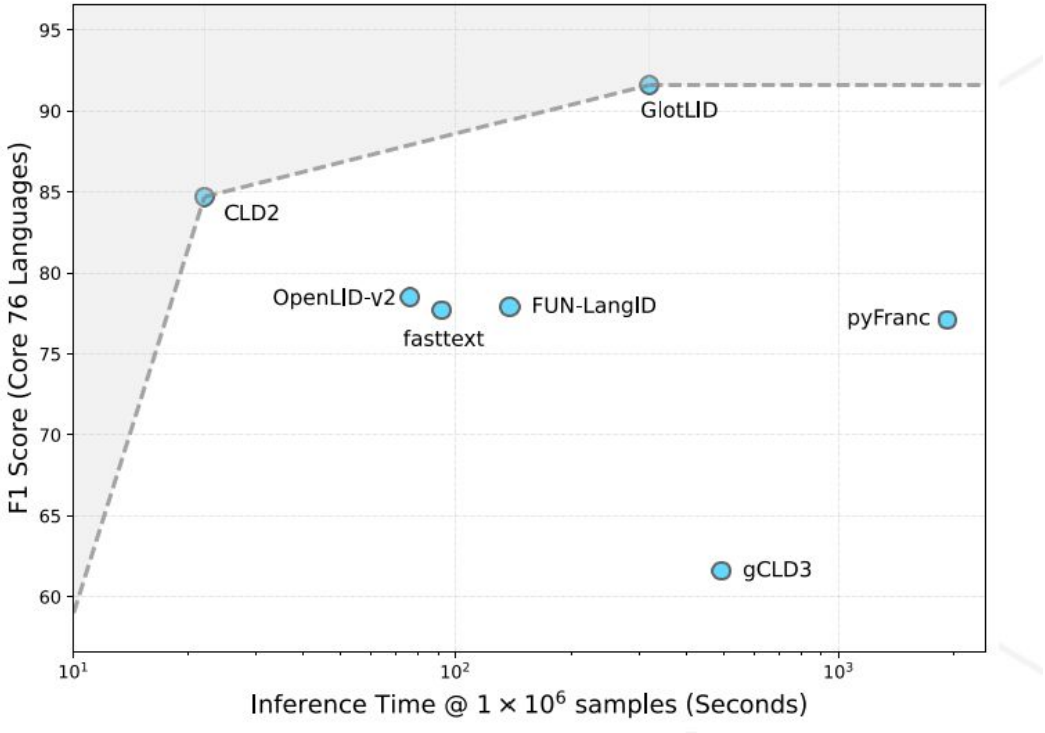
Language ID is hard and not solved!

Name	# langs.	Architecture	Data sources	Open data?
AfroLID [20]	517	Transformer	≈ 100M curated sents.	✗
CLD2 [21]	158	Naïve Bayes	Web pages (curated and scraped)	✗
fasttext [22]	218	FastText	"publicly available datasets"	✗
FUN-LangID [23]	1634	Common sub-strings	Web+Wikipedia+Bibles	✗
pyFranc [24]	414	Trigram distribution	UDHR	✓
gCLD3 [25]	99	Neural network	?	✗
GlotLID v3 [26]	1868	FastText	Curated open sources	✓
OpenLID-v2 [27]	193	FastText	Curated, audited open sources	✓

Slide taken from Pedro Ortiz Suarez's presentation of Expanding Linguistic and Cultural Coverage in Common Crawl

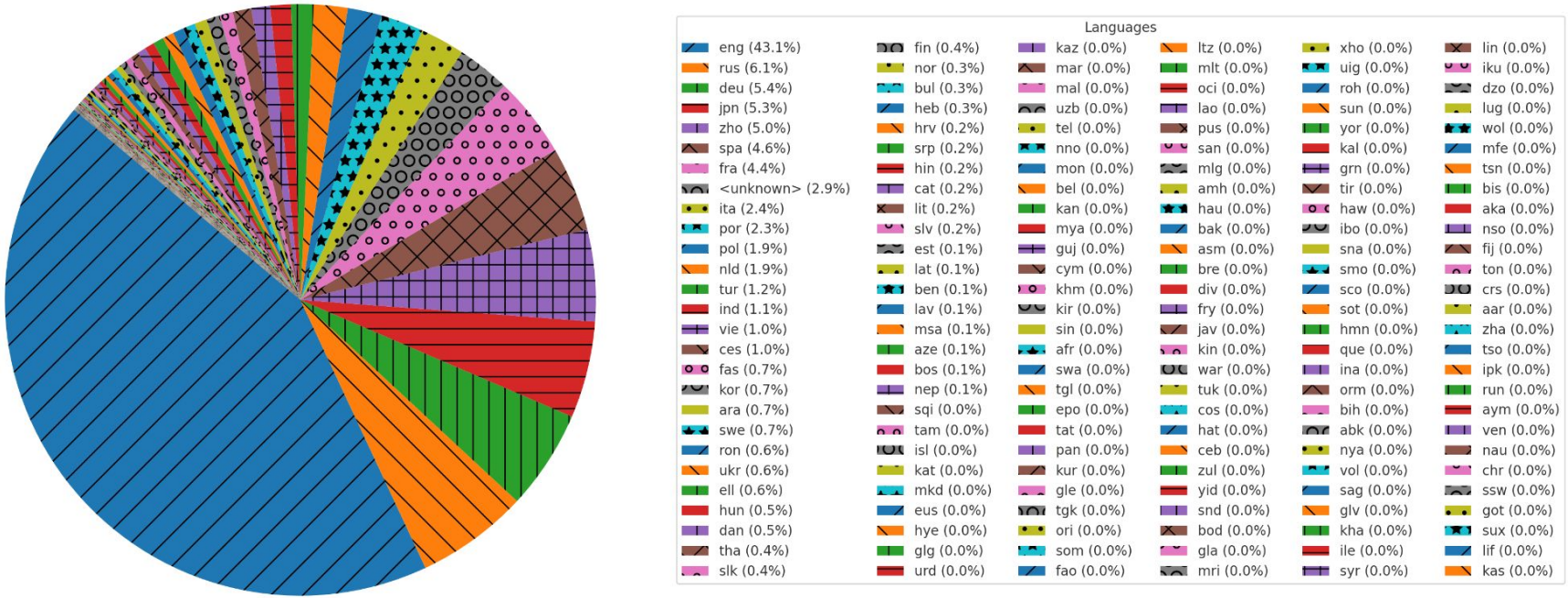
Base Filtering - Language Identification

There is always a trade off!



Slide taken from Pedro Ortiz Suarez's presentation of Expanding Linguistic and Cultural Coverage in Common Crawl

Web Data: Language Coverage Skews Towards English



Slide taken from Pedro Ortiz Suarez's presentation of Expanding Linguistic and Cultural Coverage in Common Crawl

Filtering

Base Filtering - Heuristics

Typical Filters:

- URL blocklist: remove adult/spam domains using a URL block list
- Language ID: keep language if LangID score ≥ 0.65
- Quality filters: remove short, noisy, malformed documents
- Repetition filters: remove duplicated lines, paragraphs, n-grams

https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1

Base Filtering - Heuristics - RedPajamaV2 Tags

Annotation Tag	Description	Category	Reference
ccnet_bucket	head, middle or tail bucket of the perplexity score	CCNet	CCNet
ccnet_language_score	score of the language identification model	CCNet	CCNet
ccnet_length	number of characters	CCNet	CCNet
ccnet_nlines	number of lines	CCNet	CCNet
ccnet_original_length	number of characters before line-level deduplication	CCNet	CCNet
ccnet_original_nlines	number of lines before line-level deduplication	CCNet	CCNet
ccnet_perplexity	perplexity of an LM trained on Wikipedia	CCNet	CCNet

https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1

Base Filtering - Heuristics

Annotation Tag	Description	Category	Reference
rps_doc_books_importance	Given a bag of {1,2}-wordgram model trained on Books p, and a model trained on the source domain q, This is the logarithm of the ratio p(doc)/q(doc).	ML Heuristics	Importance Resampling (Xie et al.)
rps_doc_curly_bracket	The ratio between the number of occurrences of '{' or '}' and the number of characters in the raw text.	Natural Language	C4
rps_doc_frac_all_caps_words	The fraction of words in the content that only consist of uppercase letters. This is based on the raw content.	Natural Language	Pretrainer's Guide
rps_doc_frac_lines_end_with_ellipsis	The fraction of lines that end with an ellipsis, where an ellipsis is defined as either "..." or "…".	Natural Language	RefinedWeb, Gopher

https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1

Base Filtering - Heuristics

Annotation Tag	Description	Category	Reference
rps_doc_frac_no_alph_words	The fraction of words that contain no alphabetical character.	Natural Language	RefinedWeb, Gopher
rps_doc_lorem_ipsum	The ratio between the number of occurrences of 'lorem ipsum' and the number of characters in the content after normalisation.	Natural Language	C4
rps_doc_stop_word_fraction	The ratio between the number of stop words and the number of words in the document. Stop words are obtained from the stopwords-json repo.	Natural Language	RefinedWeb, Gopher
rps_doc_symbol_to_word_ratio	The ratio of symbols to words in the content.. Symbols are defined "#", "...", and "…".	Natural Language	RefinedWeb, Gopher

https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1

Base Filtering - Heuristics

Annotation Tag	Description	Category	Reference
rps_doc_ldnoobw_words	The number of sequences of words that are contained in the List-of-Dirty-Naughty-Obscene-and-Otherwise-Bad-Words blocklist. The blocklist is obtained from the LDNOOBW repo.	toxicity	C4
rps_lines_uppercase_letter_fraction	The ratio between the number of uppercase letters and total number of characters in each line. This is based on the raw text.	Natural Language	RefinedWeb
rps_lines_start_with_bulletpoint	Whether the lines that start with a bullet point symbol. The following set of unicodes are considered a bullet point: \u2022 (bullet point), \u2023 (triangular bullet point)...	Natural Language	RefinedWeb, Gopher
rps_lines_javascript_counts	The number of occurrences of the word "javascript" in each line.	Natural Language	C4

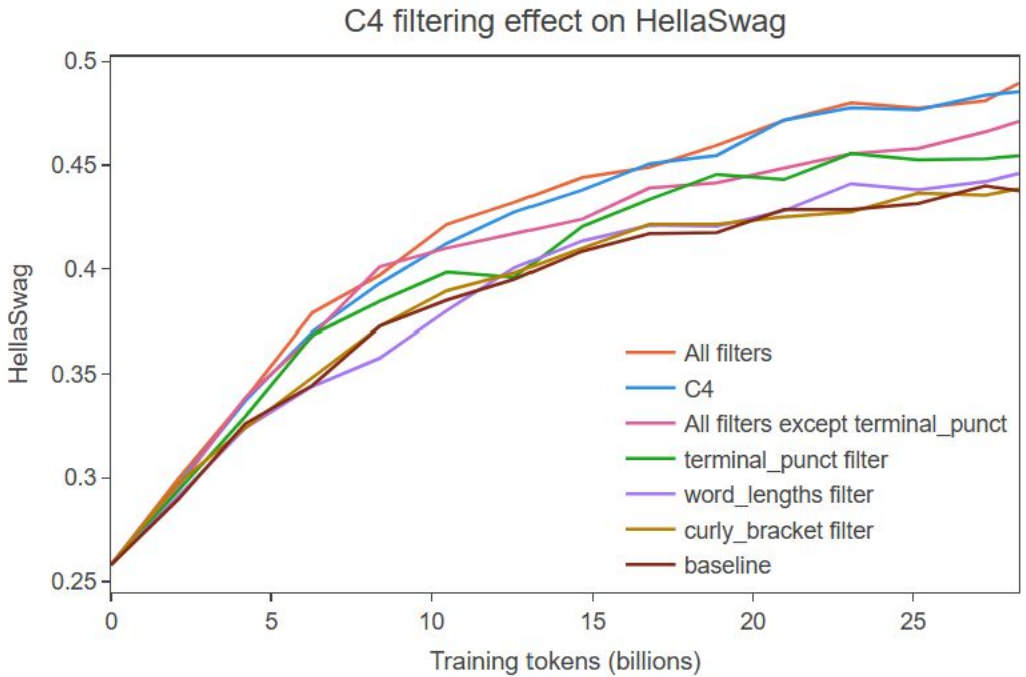
https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1

Base Filtering - Heuristics

Annotation Tag	Description	Category	Reference
rps_doc_frac_unique_words	The fraction of unique words in the content. This is also known as the degeneracy of a text sample. Calculated based on the normalised content.	Natural Language	Pretrainer's Guide
rps_doc_mean_word_length	The mean length of words in the content after normalisation.	Natural Language	RefinedWeb, Gopher
rps_doc_frac_chars_dupe_5grams	The fraction of characters in duplicate word 5grams.	Repetitiveness	RefinedWeb, Gopher
rps_doc_frac_chars_top_4gram	The fraction of characters in the top word 4gram.	Repetitiveness	RefinedWeb, Gopher

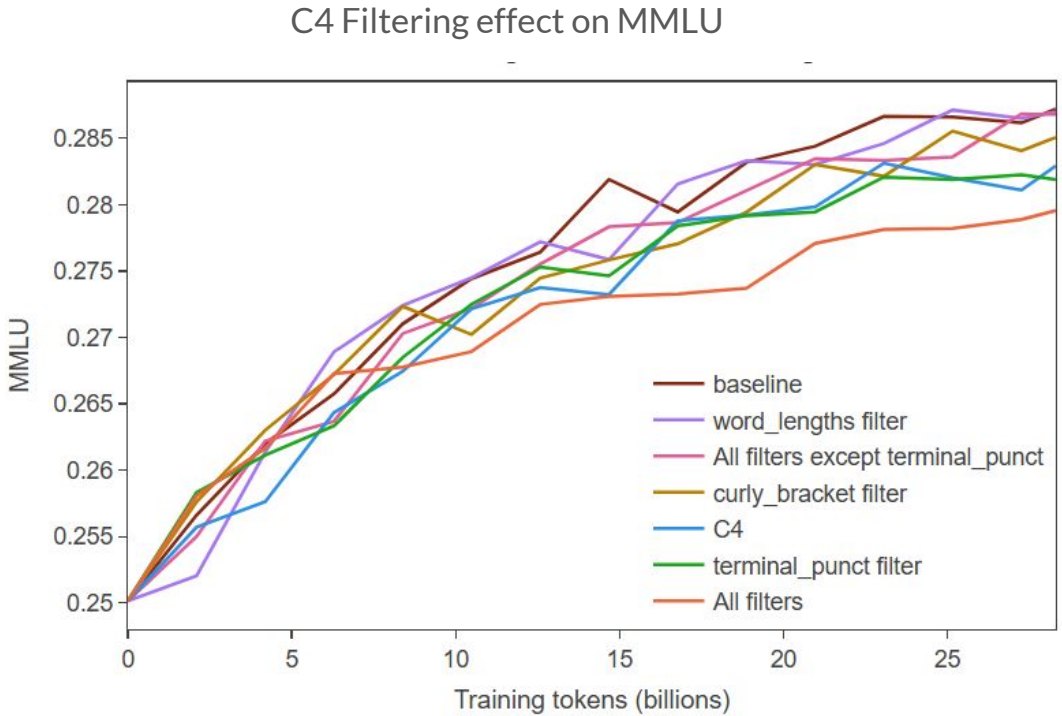
https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1

Base Filtering - Heuristics



https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1

Base Filtering - Heuristics



https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1

Deduplication

Base Filtering - Deduplication

Deduplication removes repeated content

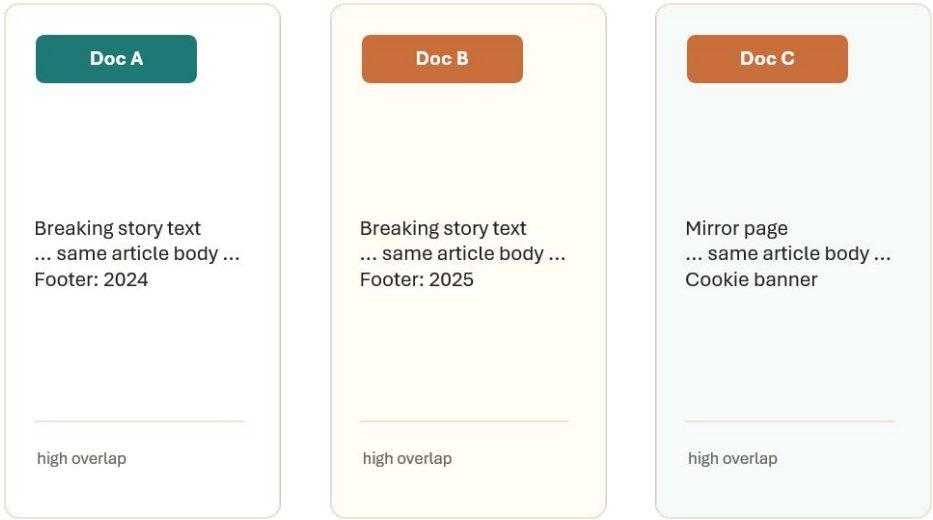
- The web contains many mirrors, aggregators, templates, and repeated pages
- Duplicates can make models memorize data instead of generalizing
- Removing duplicates improves data diversity
- More diverse data can improve training efficiency
- **BENCHMARK CONTAMINATION**

Deduplication can be exact or fuzzy, fuzzy is better for near exact document

https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1

Base Filtering - Deduplication - Near-Exact

- Near-exact deduplication removes repeated or almost-repeated web pages:
- Exact dedup only catches byte-for-byte copies.
- Near-dedup catches pages that differ only by banners, dates, navigation, tracking text, or small edits.



MinHash estimates text overlap cheaply

Base Filtering - Deduplication - Near-Exact

Convert text into shingles: **A shingle is a short overlapping phrase**. FineWeb uses word 5-grams.

Document D

> **“the quick brown fox jumps over the lazy dog”**

Word 5-grams

- the quick brown fox jumps
- quick brown fox jumps over
- brown fox jumps over the
- fox jumps over the lazy
- jumps over the lazy dog

S(D) = { all unique 5-grams in document D }

- Compare sets of shingles rather than raw strings.
- Small edits usually change only some shingles
- Near-copies still have large overlap.

Base Filtering - Deduplication - Near-Exact

Jaccard compares overlap between two shingle sets for documents A & B:

$$J(A,B) = |S(A) \cap S(B)| / |S(A) \cup S(B)|$$

Base Filtering - Deduplication - Near-Exact

MinHash: Hash every shingle, then store only the minimum number.

$$m_i(D) = \min_{x \in S(D)} h_i(x)$$

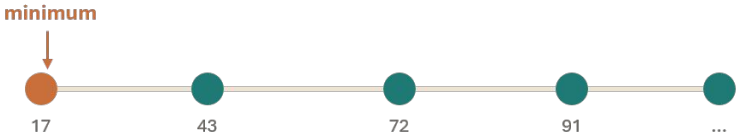
Shingle	h(x)
the cat sat	91
cat sat on	17
sat on the	43
on the mat	72

→ MinHash = 17

Base Filtering - Deduplication - Near-Exact

Why minimum?

Makes it faster to find matching document shingles



The first shingle in the random hash order is equally likely to be any shingle in the union. It matches exactly when that first shingle is in the intersection.

Key MinHash property

$$P[m_i(A) = m_i(B)] = J(A,B)$$

If two documents share 75% of their shingles, one MinHash value has a 75% chance of matching.

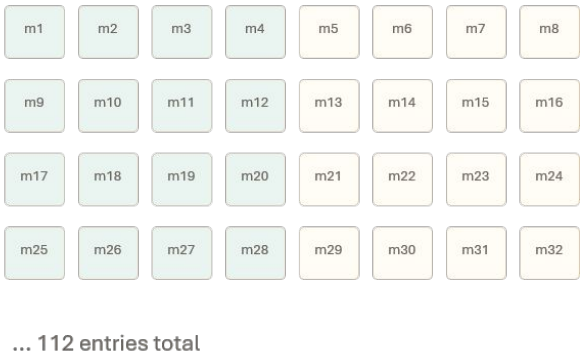
Base Filtering - Deduplication - Near-Exact

One MinHash is noisy so we need many to make a signature

- FineWeb stores 112 minimum hash values per document.
- Each m_i comes from a different hash function

$$M(D) = [m_1(D), m_2(D), \ldots, m_{112}(D)]$$

Signature sketch



Similarity estimate ≈ fraction of matching signature positions

Example: if 84 of 112 positions match:
Estimated Jaccard ≈ 84/112 = 0.75.

Base Filtering - Deduplication - Near-Exact

Banding: Locality-sensitive hashing trick

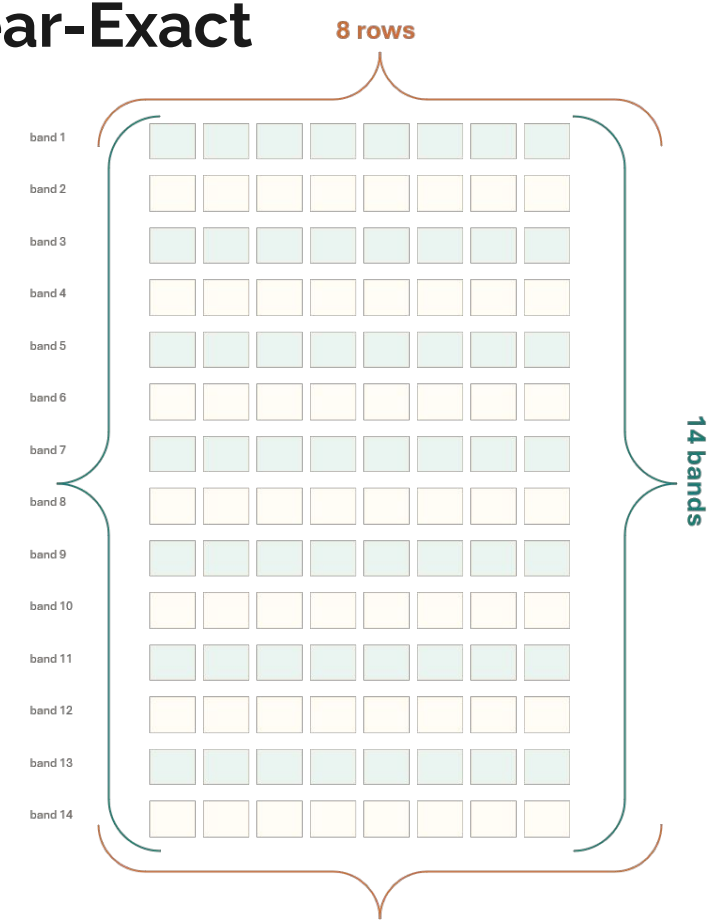
- Instead of comparing all signatures pairwise, split each signature into bands.

b = number of bands = 14

r = rows per band = 8 MinHash values

112 = b × r = 14 × 8

Two docs become candidate duplicates if any one band is shared. It's "OR of ANDs"

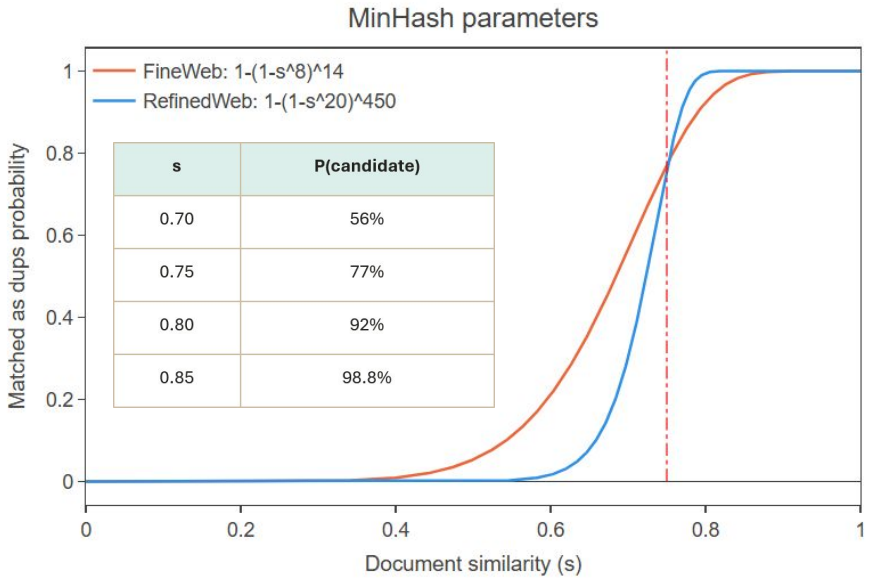


Base Filtering - Deduplication - Near-Exact

Probability math: similarity → match chance

Let s = J(A,B), the true shingle similarity between documents

One MinHash matches	s
8 MinHashes in one band all match	s^8
One band does not match	$1 - s^8$
No bands match among 14	$(1 - s^8)^{14}$
At least one band matches	$1 - (1 - s^8)^{14}$

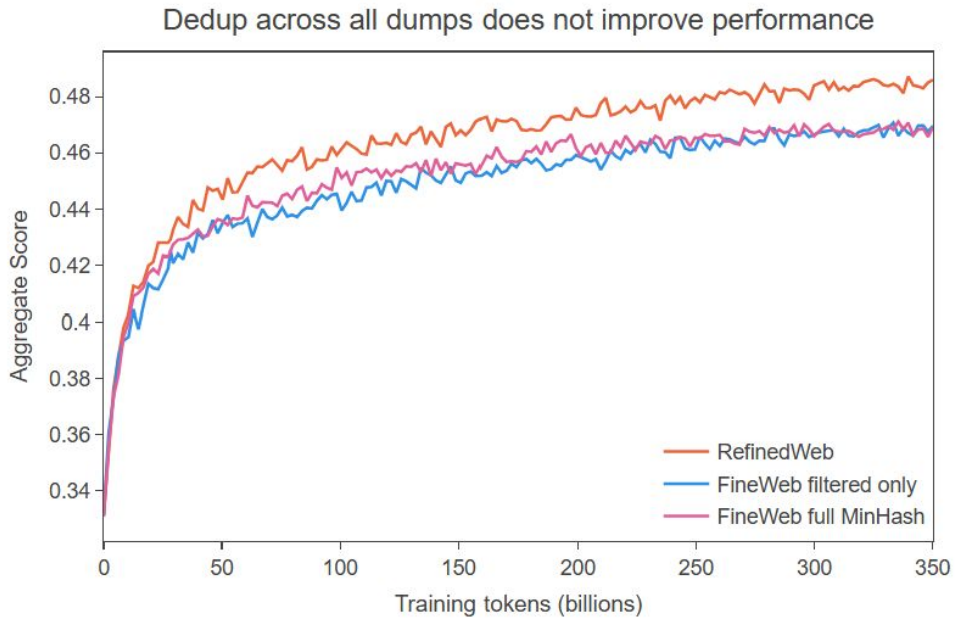


s	P(candidate)
0.70	56%
0.75	77%
0.80	92%
0.85	98.8%

Targeting documents around 75%+ similarity, with 77% chance of detecting the pair as duplicates.

Base Filtering - Deduplication - Near-Exact

More deduplication is always better, right?



https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1

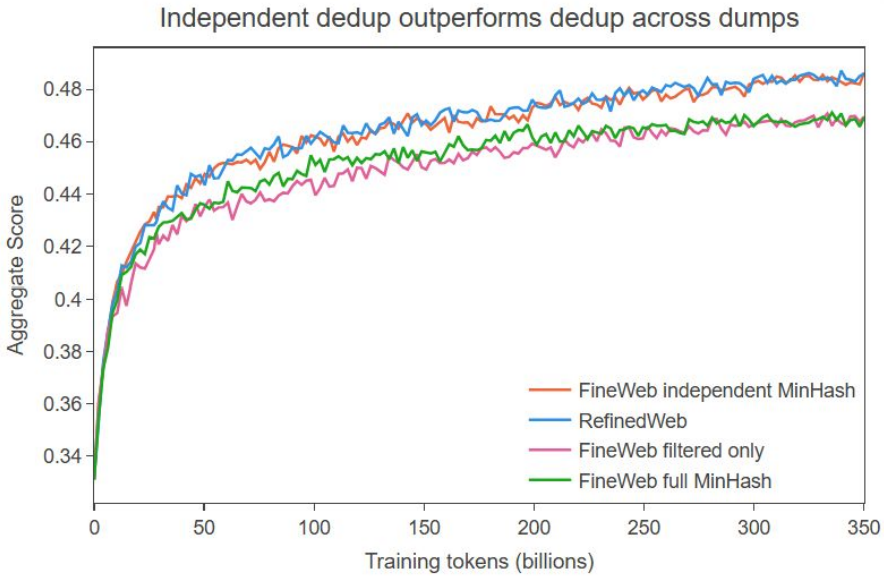
Base Filtering - Deduplication - Near-Exact

Global Deduplication will remove large spam clusters with 100K+ docs.

It will also remove smaller cluster with 5-100 dups

These clusters tend to be information that is duplicated because it's important, as is often over multiple dumps

Independant dump dedup fixes this



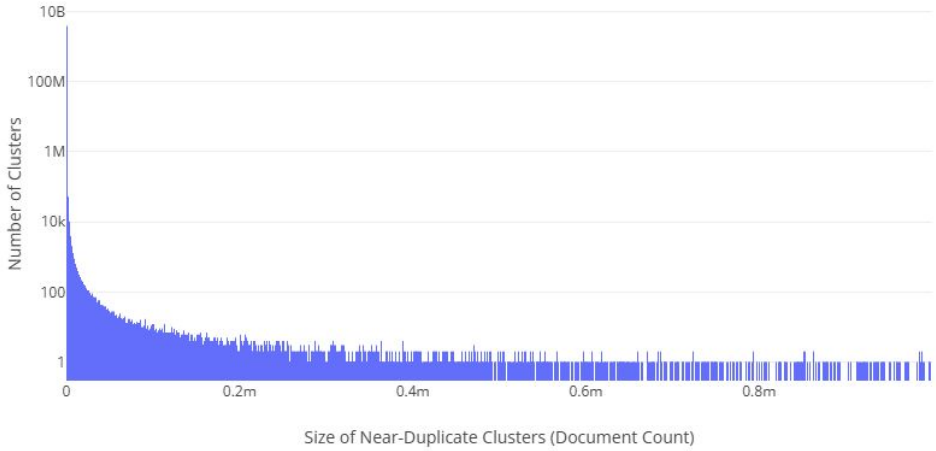
https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1

Base Filtering - Deduplication - Near-Exact

Set the upsampling weight to:

- 3 for documents with 2 to 5 duplicates
- 5 for those with 5 to 100 duplicates
- 8 for 101 to 1000 duplicates
- 10 for documents with over 1000 duplicates.

Like in TxT360



https://huggingface.co/spaces/LLM360/TxT360

Base Filtering - Deduplication - Near-Exact

https://huggingface.co/spaces/LLM360/TxT360

Semantic Filtering

Semantic Filtering

GPT-2: human curation by proxy

OpenAI built WebText from Reddit outbound links with ≥3 karma, treating upvotes as a signal that humans found content useful, interesting, or funny

GPT-3: classifier-based Common Crawl filtering

OpenAI trained a quality classifier to distinguish curated, high-quality text from raw Common Crawl, then **used that classifier to preferentially keep higher-quality Common Crawl documents**.
Logistic regression over bag-of-words features

Semantic Filtering

Method	Intuition
PageRank filtering	Keep pages likely to be important because other pages link to them.
Semantic deduplication / SemDeDup	Remove documents with highly similar meaning, not just exact duplicates.
BGE embedding classifier	Use pretrained text embeddings, then train a linear classifier for quality.
Perplexity filtering	Keep text that a language model finds predictable / fluent.
Top-k average logits	Score documents by how confident a model is about likely next-token choices.

Semantic Filtering - Stronger Classifiers

FineWeb-Edu:

1. Use Llama-3-70B-Instruct to score ~500k web samples from 0–5 for educational value.
2. Train a smaller BERT-like regression/classification model on those synthetic labels.
3. Run that classifier over FineWeb.
4. Keep documents with score ≥ 3, removing about 92% of the corpus.

Semantic Filtering - LLama 3 Prompt

Below is an extract from a web page. Evaluate whether the page has a high educational value and could be useful in an educational setting for teaching from primary school to grade school levels using the additive 5-point scoring system described below. Points are accumulated based on the satisfaction of each criterion:

- Add 1 point if the extract provides some basic information relevant to educational topics, even if it includes some irrelevant or non-academic content like advertisements and promotional material.
- Add another point if the extract addresses certain elements pertinent to education but does not align closely with educational standards. It might mix educational content with non-educational material, offering a superficial overview of potentially useful topics, or presenting information in a disorganized manner and incoherent writing style.
- Award a third point if the extract is appropriate for educational use and introduces key concepts relevant to school curricula. It is coherent though it may not be comprehensive or could include some extraneous information. It may resemble an introductory section of a textbook or a basic tutorial that is suitable for learning but has notable limitations like treating concepts that are too complex for grade school students.
- Grant a fourth point if the extract highly relevant and beneficial for educational purposes for a level not higher than grade school, exhibiting a clear and consistent writing style. It could be similar to a chapter from a textbook or a tutorial, offering substantial educational content, including exercises and solutions, with minimal irrelevant information, and the concepts aren't too advanced for grade school students. The content is coherent, focused, and valuable for structured learning.
- Bestow a fifth point if the extract is outstanding in its educational value, perfectly suited for teaching either at primary school or grade school. It follows detailed reasoning, the writing style is easy to follow and offers profound and thorough insights into the subject matter, devoid of any non-educational or complex content.

The extract: <extract>.

After examining the extract:

- Briefly justify your total score, up to 100 words.
- Conclude with the score using the format: "Educational score: <total points>"

Semantic Filtering - Stronger Classifiers

Issues:

- Complete trust in LLama 3 70B judgement and bias
- Too educational and skewed towards benchmark topics
- Doesn’t assess formatting

Possible fixes:

- Human labelled test set for calibrating the judge and the prompt

Semantic Filtering - Matching the Downstream Use

Source labels from chat datasets

Instead of using Wikipedia or other high quality text as a positive example → Use Chat traces/logs

1. Start from DCLM-Pool, a 240T-token Common Crawl corpus.
2. Apply extraction, heuristic cleaning, and deduplication.
3. Train a fastText binary classifier:
    a. Positive examples: OpenHermes 2.5 + high-quality Reddit ELI5
    b. Negative examples: web-crawled Common Crawl / RefinedWeb samples
4. Score each document by probability of being “high quality.”
5. Keep only the top 10% by classifier score.

Li, Jeffrey, et al. "Datacomp-lm: In search of the next generation of training sets for language models." Advances in Neural Information Processing Systems 37 (2024): 14200-14282.

Semantic Filtering - Matching the Downstream Use

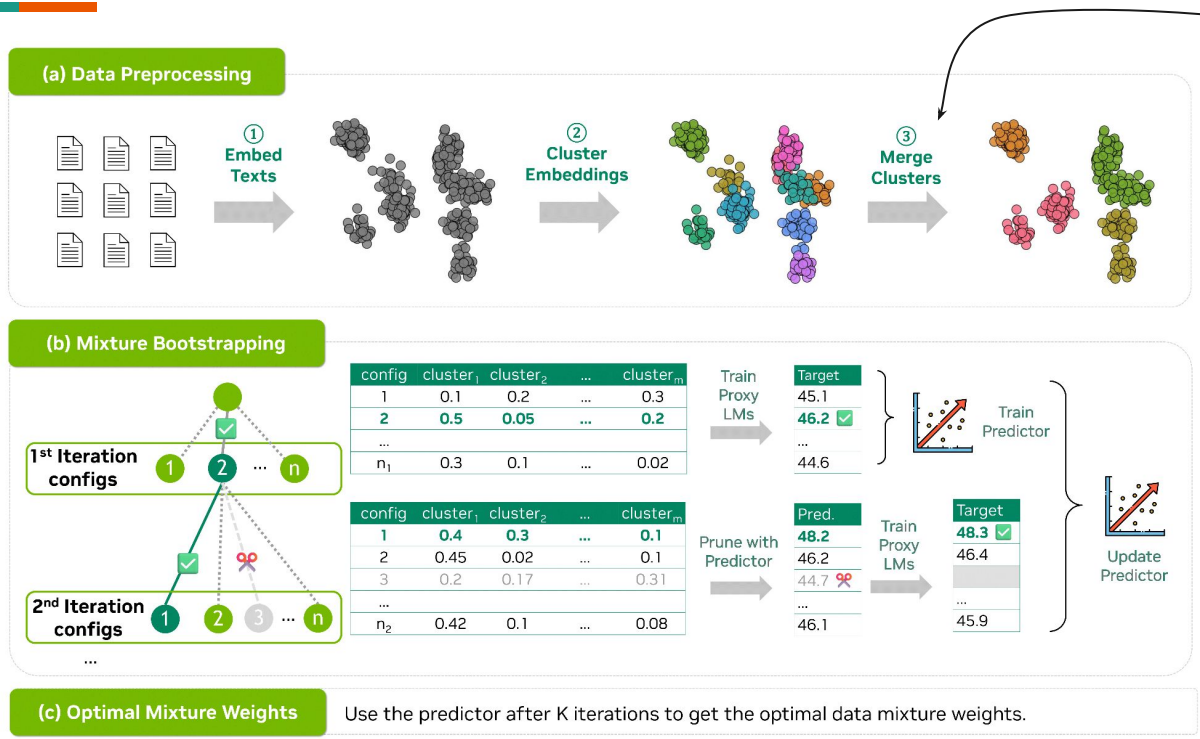
Table 8: **State-of-the-art comparison** (beyond 7B-2x scale). We compare our final model with other 7–8B parameter models. DCLM-BASELINE yields a model that outperforms models trained on open datasets and is competitive with models trained on private datasets.

Model	Params	Tokens	Open dataset?	CORE	MMLU	EXTENDED
Open weights, closed datasets						
Llama2	7B	2T	✗	49.2	45.8	34.1
DeepSeek	7B	2T	✗	50.7	48.5	35.3
Mistral-0.3	7B	?	✗	57.0	62.7	45.1
QWEN-2	7B	?	✗	57.5	**71.9**	50.5
Llama3	8B	15T	✗	57.6	66.2	46.3
Gemma	8B	6T	✗	57.8	64.3	44.6
Phi-3	7B	?	✗	**61.0**	69.9	**57.9**
Open weights, open datasets						
Falcon	7B	1T	✓	44.1	27.4	25.1
OLMo-1.7	7B	2.1T	✓	47.0	54.0	34.2
MAP-Neo	7B	4.5T	✓	**50.2**	**57.1**	**40.4**
Models we trained						
FineWeb edu	7B	0.14T	✓	38.7	26.3	22.1
FineWeb edu	7B	0.28T	✓	41.9	37.3	24.5
DCLM-BASELINE	7B	0.14T	✓	44.1	38.3	25.0
DCLM-BASELINE	7B	0.28T	✓	48.9	50.8	31.8
DCLM-BASELINE + StarCoder + ProofPile2	7B	2.6T	✓	**57.1**	**63.7**	**45.4**

Li, Jeffrey, et al. "Datacomp-lm: In search of the next generation of training sets for language models." Advances in Neural Information Processing Systems 37 (2024): 14200-14282.

Not Semantic Filtering - NVIDIA CLIMB

https://research.nvidia.com/labs/lpr/climb/



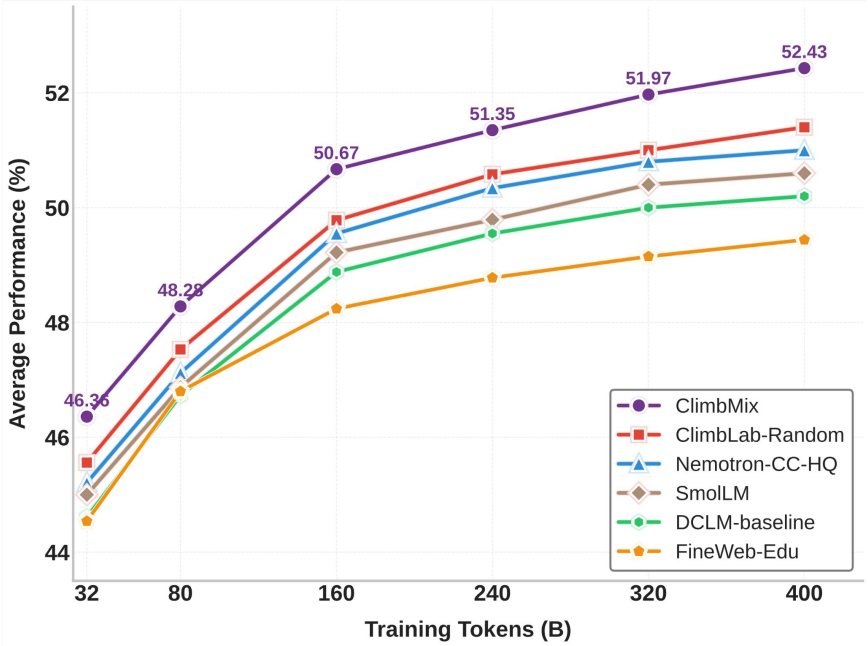
Prune clusters via FastText cls

Dimension	Meaning
Overall quality	Is the text generally useful and coherent?
Educational value	Does it teach something?
Informational value	Does it contain factual or substantive content?
Advertisement score	Does it look like spam, ads, promotional pages?

Diao, Shizhe, et al. "Climb: Clustering-based iterative data mixture bootstrapping for language model pre-training." arXiv e-prints (2025): arXiv-2504.

Not Semantic Filtering - NVIDIA CLIMB

https://research.nvidia.com/labs/lpr/climb/



Diao, Shizhe, et al. "Climb: Clustering-based iterative data mixture bootstrapping for language model pre-training." arXiv e-prints (2025): arXiv-2504.

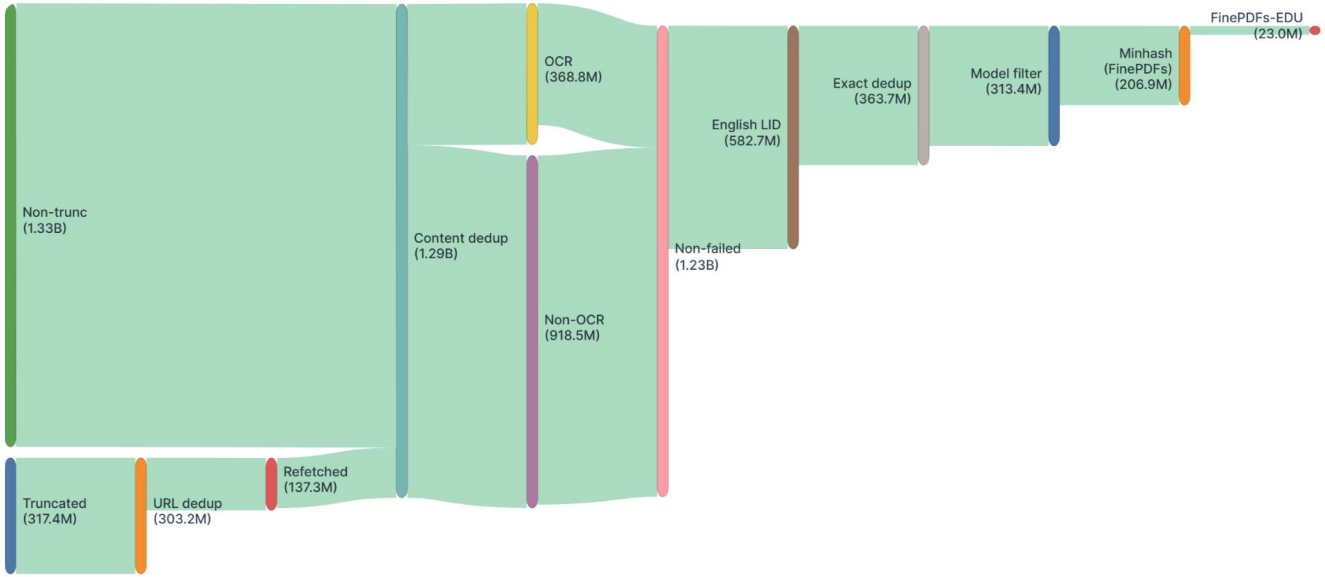
Not Semantic Filtering - NVIDIA CLIMB

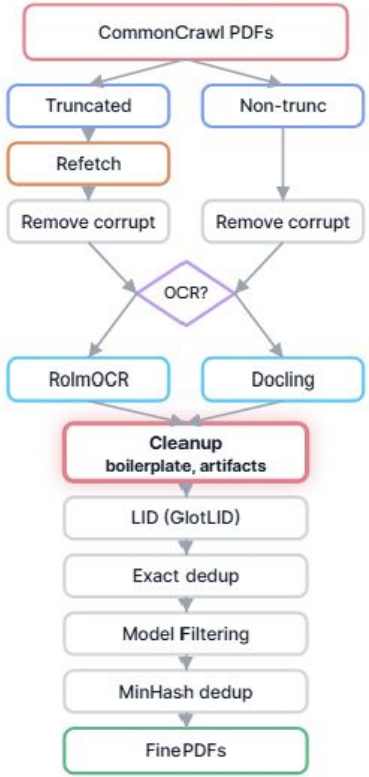
https://research.nvidia.com/labs/lpr/climb/

Cluster ID	# of Tokens (B)	Weight (%)	Topics
1	17.79	0.81	Mathematics, Algorithms, Programming, Software Development, Data Analysis
2	109.73	1.11	Books, Education, Writing, Literature, AI Ethics, History, Philosophy
3	80.62	1.26	Environmental Education, History, Architecture, Engineering, Classical Music
4	64.70	3.05	Education, Teaching, Science, Engineering, Psychology, Special Education
5	92.97	1.65	International Trade, Business, Economics, AI Consulting, Ethical Decision Making
6	70.95	20.46	Genetics, Biotechnology, AI, Robotics, Aging, Healthcare, Industrial Automation
7	64.04	16.08	Chemistry, Insects, Taxonomy, Agriculture, Gardening, Veterinary Science
8	24.68	0.91	Gaming, Role-Playing, Board Games, Video Games, Strategy, Fantasy, Virtual Reality
9	12.75	0.78	Astronomy, Cosmology, Astrophysics, Space Exploration, Urban Planning
10	135.45	6.60	Health, Sleep, Clinical Technology, Healthcare, Fitness, Addiction, Early Childhood Education
11	37.11	1.20	Software Development, Programming, Web Development, JavaScript, Databases
12	78.31	28.04	Technology, Mathematics, Legal Content, Human Rights, Energy Efficiency, Industrial Equipment
13	10.95	0.63	Sports, Cricket, Soccer, Tennis, Basketball, Cultural Heritage, Competition
14	15.64	0.21	Music, Instrumental Practice, Guitar, Jazz, Singing, Composition, Music Theory
15	35.24	0.21	Film, Cinema, Horror, Sci-Fi, Comics, Literature, Criticism, Philosophy
16	52.24	7.45	Sustainability, Climate Change, Renewable Energy, Environmental Conservation
17	82.23	6.35	Cardiovascular Health, Medical Research, Immunology, Cancer Prevention, Drug Therapy
18	54.02	1.79	Technology, Cybersecurity, Social Media, Privacy, Artificial Intelligence, Cloud Computing
19	50.32	0.91	Social Media, Digital Communication, Internet Culture, Misinformation, Psychology
20	79.47	0.49	Public Safety, Law Enforcement, Political History, Social Justice, Government
Total	1,170.30	100.0	-

Diao, Shizhe, et al. "Climb: Clustering-based iterative data mixture bootstrapping for language model pre-training." arXiv e-prints (2025): arXiv-2504.

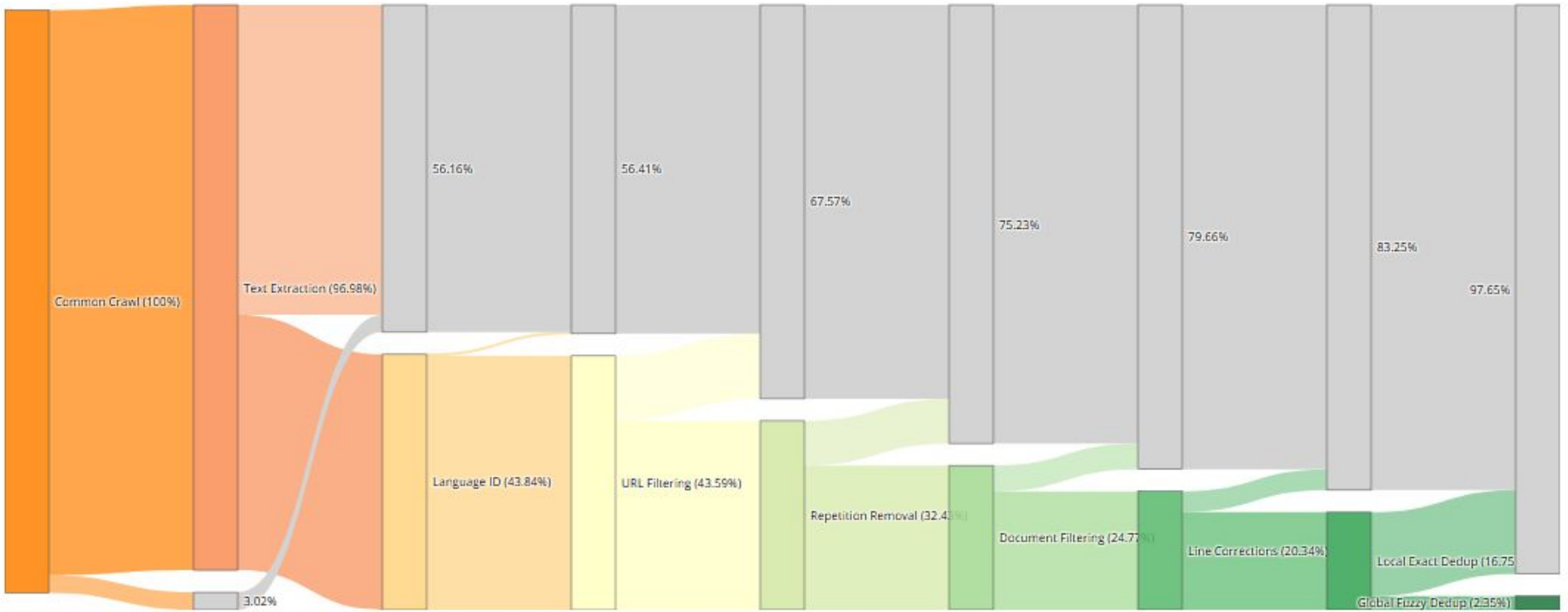
Recap





https://huggingface.co/spaces/HuggingFaceFW/FinePDFsBlog

Recap



https://huggingface.co/spaces/LLM360/TxT360

What do we do with <Code/>

Why Include Code?

Code is not only for “coding models”

According to Aryabumi et. al. (Cohere AI):

- Increase of 8.2% in natural language (NL) reasoning
- 4.2% in world knowledge
- 6.6% improvement in generative win-rates
- 12x boost in code performance respectively

But Why?? **… Let’s discuss**

Aryabumi, Viraat, et al. "To code or not to code? exploring impact of code in pre-training." International Conference on Learning Representations. Vol. 2025. 2025.

Code teaches models structured, executable reasoning

Code is built from explicit rules, variables, functions, dependencies, and state changes

- **Compositionality:** functions, arguments, modules, and nested logic teach reusable structure.
- **State tracking:** variables and objects change over time, helping the model follow evolving entities.
- **Low ambiguity:** syntax and APIs impose stricter constraints than natural language.
- **Executable feedback:** code can be compiled, tested, linted, or run, creating clearer quality signals.
- **Intent ↔ implementation alignment:** comments, docstrings, READMEs, issues, and notebooks pair human goals with concrete solutions.
- **Tool and API knowledge:** code exposes models to real library usage, commands, data workflows, and interactions.

Aryabumi, Viraat, et al. "To code or not to code? exploring impact of code in pre-training." International Conference on Learning Representations. Vol. 2025. 2025.

Code Preprocessing

Cleaning step	Notes
License filtering	Avoid unclear or restricted training data. License detection and separate permissive, non-permissive, and unlicensed files.
Exact + near deduplication / decontamination	GitHub forks, copied snippets, generated files, and notebooks are highly duplicated. Coding benchmarks are often copied online
Remove generated/minified/data-like files	Generated code, encoded blobs, long lines, JSON dumps, minified JS, and binary-like files add noise. Stack v2 used filters for autogenerated files, long/odd lines, low alphabetic content, encoded data, and excessive data formats.
Syntax / compiler / parser filters	Code can be checked more directly than prose. DeepSeek-Coder used compiler signals, quality models, and heuristics to remove syntax errors and low-quality code.
PII and malware removal	Code often contains secrets, tokens, emails, credentials, exploits, and malicious snippets.
Language balancing	Python/JS dominate; without balancing, smaller languages disappear. Downsample high-resource languages and retained hundreds of languages.

Code Formatting

Repository-level code is better than isolated files

StarCoder2 :

```
<repo_name> owner/project
<file_sep> path/to/file_1.py
... code ...
<file_sep> path/to/file_2.py
... code ...
<|endoftext|>
```

Fill-in-the-middle (FIM):

```
<fim_prefix> code before the edit
<fim_suffix> code after the edit
<fim_middle> missing code to generate
```

Code Formatting

Repository-level code is better than isolated files

Notebooks :

```
<jupyter_text>
Explanation of the analysis

<jupyter_code>
df.groupby("label").mean()

<jupyter_output>
table or textual output
```

Pull-Request (Very important for Agents, **but why??**):

```
<title>
<description>
<file path>
<diff hunk>
<review comment>
<resolution>
```

So did we run out of data?

Did we run out of training data?

Where is all this data from?

Phase	Tokens	Context length	GB200 GPUs
Pre-training	30 T	16,384	8,192
Mid-training 1	3.4 T	65,536	8,192
Mid-training 2	150 B	262,144	4,096

Table 6. Training specifications across phases for MAI-Base-1.

Qwen3-Max-Base

The Qwen3-Max model has over 1 trillion parameters and was pretrained on 36 trillion tokens. Its architecture follows the design paradigm of the Qwen3 series, incorporating our proposed global-batch load balancing loss.

MiniMax M3

Coding & Agentic Frontier. 1M-context MSA. Native Multimodality

The first open-weight model with three frontier capabilities.

A natively multimodal model. The entire data pipeline was rebuilt to scale pretraining data to 100T+, with multimodal training from step zero achieving deep alignment between textual and visual semantic spaces. Multimodal is a native core capability, not a superficial add-on.

Can we generate new data?

Some approaches to synthetic data creation:

- Generate from scratch
- Translation
- Transformation of existing data

Generate from scratch - Cosmopedia

Open synthetic dataset designed for LLM pretraining

- Inspired by Microsoft's Phi "Textbooks Are All You Need " idea

Given a web excerpt, outline, topic, wikiHow title → Generate new educational content

Total:

- 30 million files
- 25B tokens

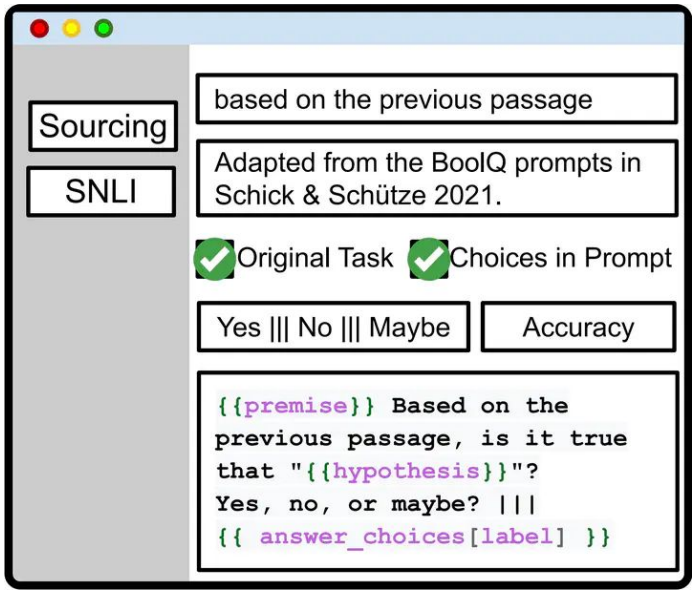
Limitations??

Translation

This one is trivial, but comes with it’s own issue, can you guess?

Transformation of Existing Data

- Transform labelled training datasets or even knowledge graphs



Bach, Stephen, et al. "Promptsource: An integrated development environment and repository for natural language prompts." Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. 2022.

Transformation of Existing Data

Extract Instructions from Web documents

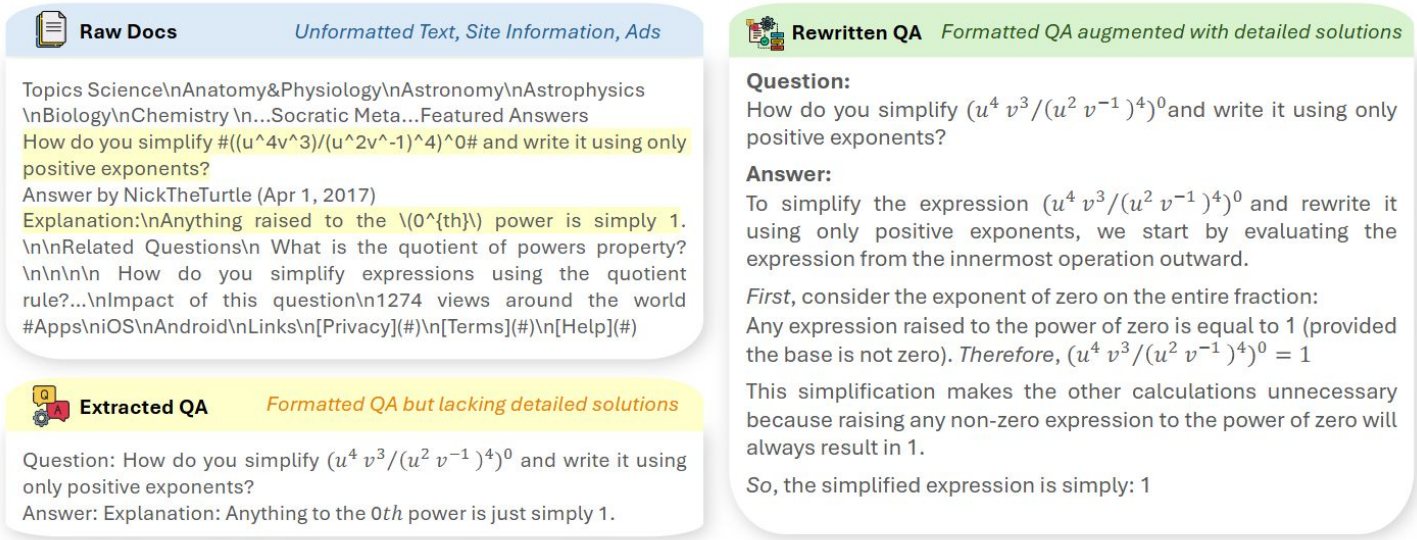
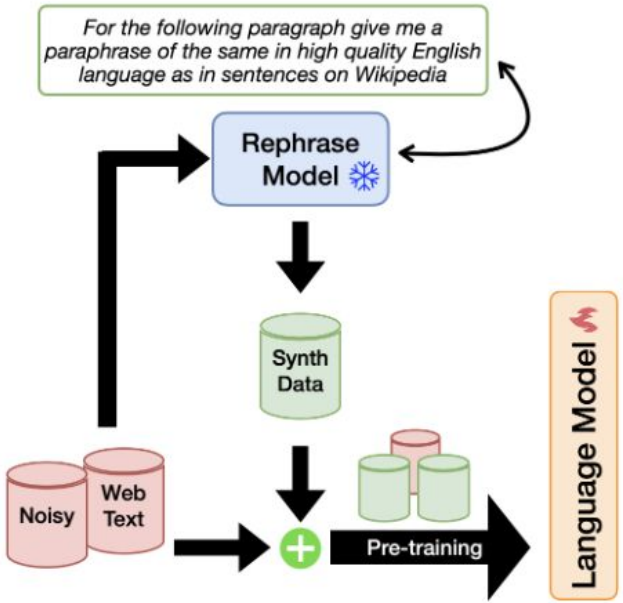


Figure 4: An illustrating example from WEBINSTRUCT for the extraction and refinement step.

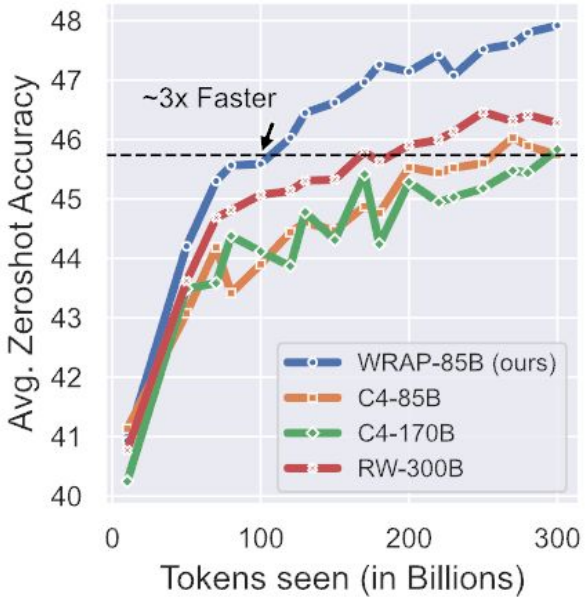
Yue, Xiang, et al. "Mammoth2: Scaling instructions from the web." Advances in Neural Information Processing Systems 37 (2024): 90629-90660.

Transformation of Existing Data

Rephrase good and bad documents



Prompt diversity is key

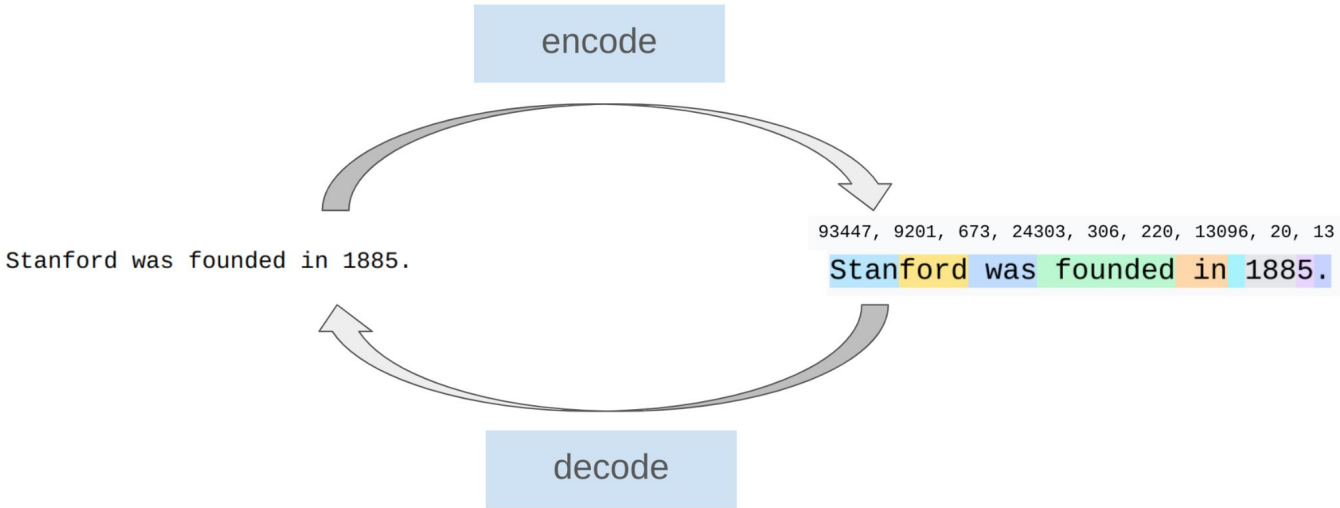


Yue, Xiang, et al. "Mammoth2: Scaling instructions from the web." Advances in Neural Information Processing Systems 37 (2024): 90629-90660.

Let’s Train

Tokenization

- Words explode the vocabulary and fail on rare or new forms.
- Characters are too long for efficient context use.
- Subwords/Tokens balance vocabulary size and sequence length.



Tokenization - BPE - Learn Frequent Merges

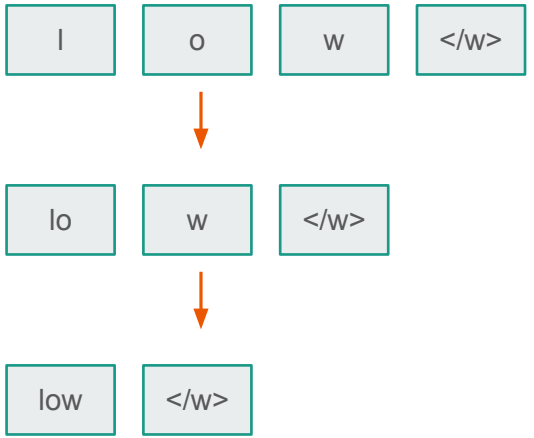
Byte Pair Encoding starts from small units (bytes) and repeatedly merges frequent adjacent pairs

- Similar to a compression algorithm

Process:

1. Start with byte/character vocabulary
2. Count adjacent token pairs
3. Merge the most frequent pair
4. Repeat until target vocab size
5. Encode new text using the learned merges

Typical Vocab Size: 32K-256K



Tokenization - BPE - Examples

English and CAPITALIZATION

🐦

show_tokens False None elif == >= else: two tabs:" " Three tabs: " "

12.0*50=600

GPT-2	English and CAP ITAL IZ ATION □□□□□□ show _ t ok ens False None el if == >= else : two tabs :" " Three tabs : " " 12 . 0 * 50 = 600
FLAN-T5	English and CA PI TAL IZ ATION <unk> <unk> show _ to ken s Fal s e None e l if = = > = else : two tab s : " " Three tab s : " " 12. 0 * 50 = 600 </s>
GPT-4	English and CAPITAL IZATION □□□□□□ show _tokens False None elif == >= else : two tabs :" " Three tabs : " " 12 . 0 * 50 = 600
StarCoder	English and CAPITAL IZATION □□□□□ show _ tokens False None elif == >= else : two tabs :" " Three tabs : " " 1 2 . 0 * 5 0 = 6 0 0

Training Objective - Next Word/Token Prediction

Self-Supervised Objective: The label is within the example

Negative Log-Likelihood

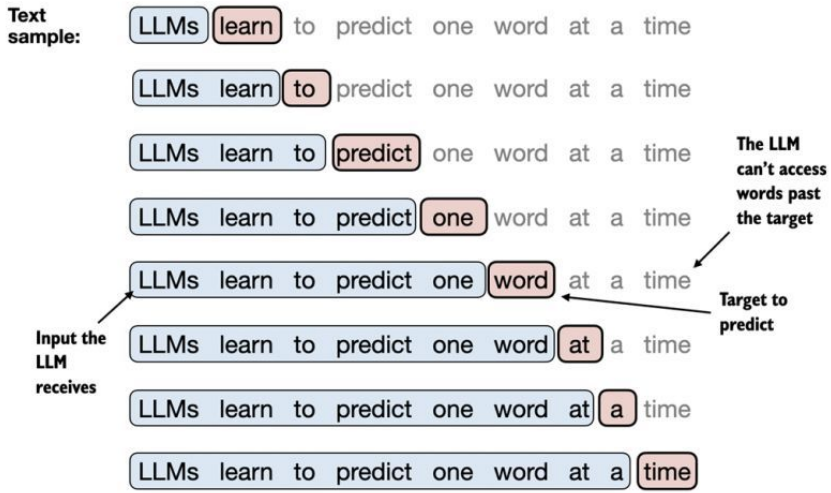
$$\mathcal{L} = -\sum_{t=1}^{T-1} \log p_\theta(x_{t+1} \mid x_1, \ldots, x_t)$$

Or

Cross-Entropy

Choose which token out the vocabulary to predict

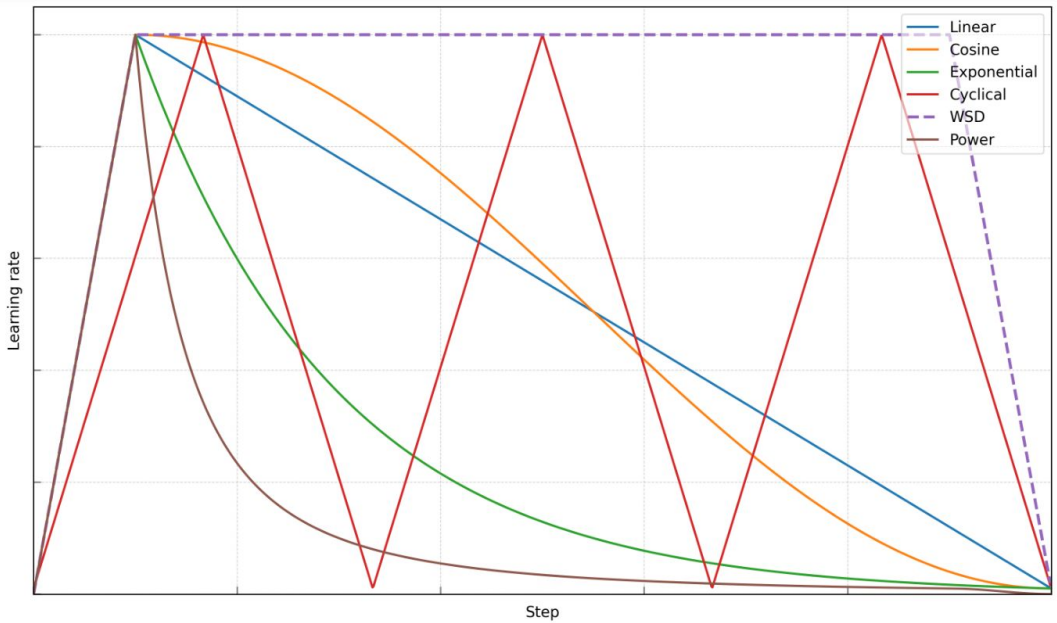
Model the probability of all tokens



Learning Rate

Warmup-Stable-Decay (WSD)

Allows Flexibility for experimentations



Learning Rate

Choosing the right LR matters:

Maximal Update Parameterization, usually written **µP** or **MuP**

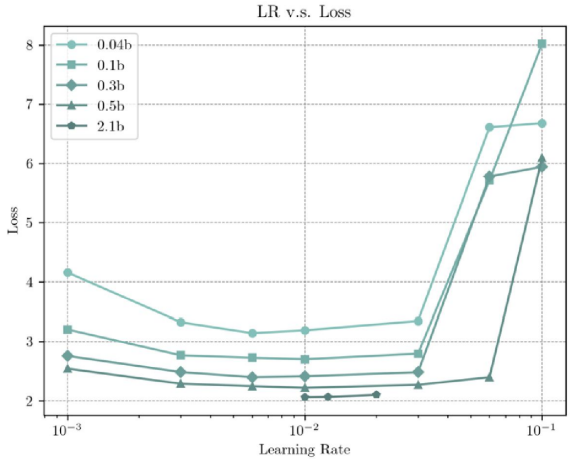


Figure 3: Loss vs Learning Rate. After applying for the Tensor Program, the learning rate shift becomes minimal.

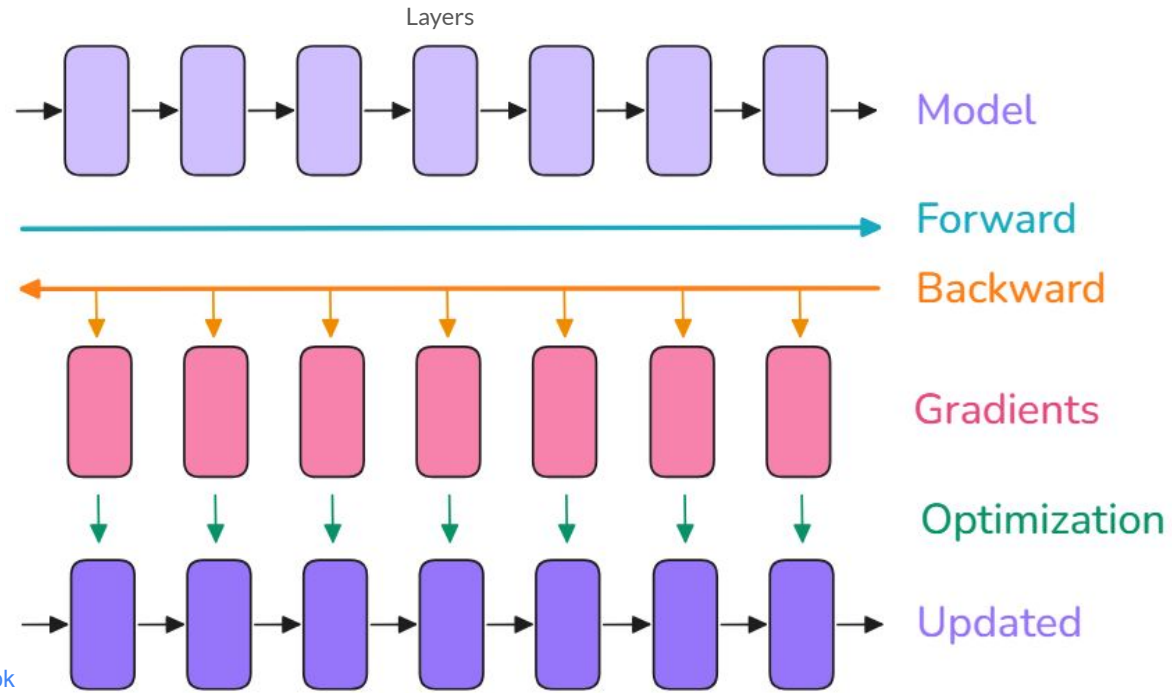
Yang, G., et al. "Tensor programs V: Tuning large neural networks via zero-shot hyperparameter transfer. arXiv." arXiv preprint arXiv:2203.03466 (2022).

Why so many GPUs?

Model Training

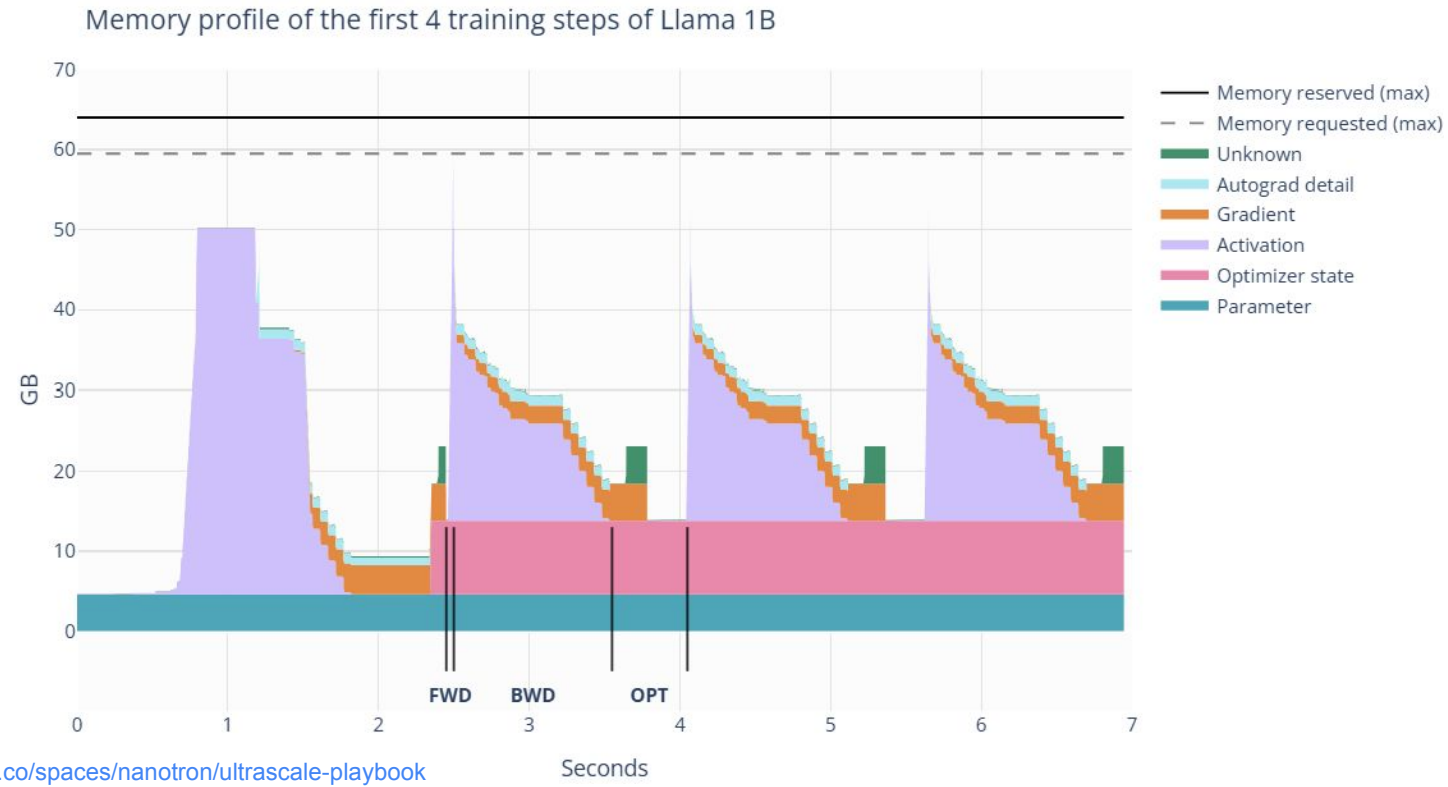
Data flow through a single training step on a single GPU

What if the model doesn't fit!



https://huggingface.co/spaces/nanotron/ultrascale-playbook

Model Training - Memory Usage During training



https://huggingface.co/spaces/nanotron/ultrascale-playbook

Model Training - Memory Usage During training

We will need multi GPU for >7B models

Model parameters	FP32 or BF16 w/o FP32 grad acc	BF16 w/ FP32 grad acc
1B	16 GB	20 GB
7B	112 GB	140 GB
70B	1120 GB	1400 GB
405B	6480 GB	8100 GB

Using FP8 training instead of BF16 would further decrease the memory usage, but it is less stable. This is a very active research topic (see this tweet), and we'll cover it in more detail later.

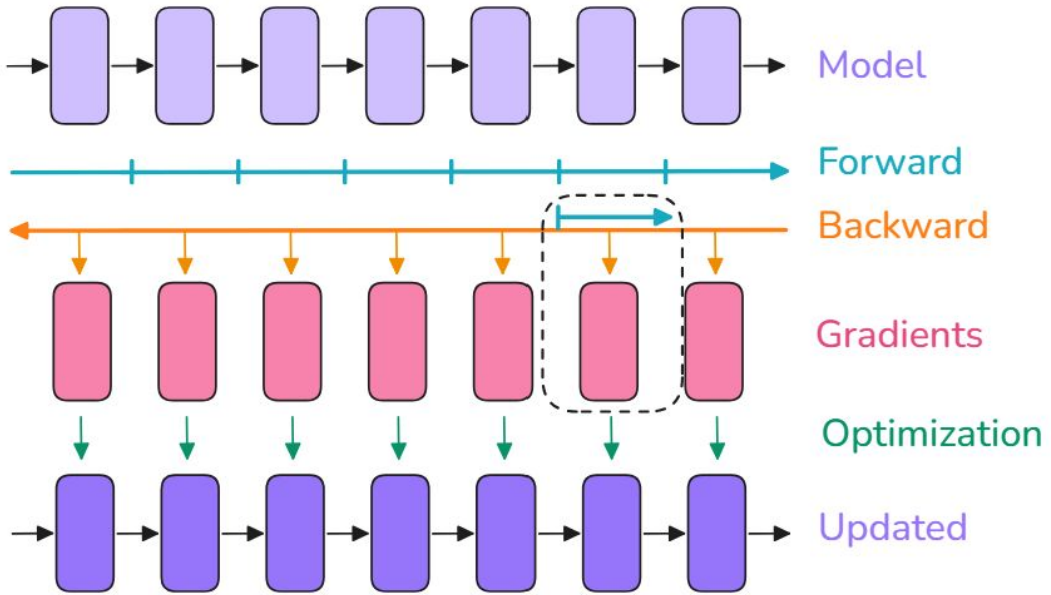
https://huggingface.co/spaces/nanotron/ultrascale-playbook

Model Training - Reduce Memory Usage

Activation recomputation - Gradient Checkpointing

Discard some activations during the forward pass to save memory and spend some extra compute to recompute these on the fly during the backward pass

FlashAttention already does this



https://huggingface.co/spaces/nanotron/ultrascale-playbook

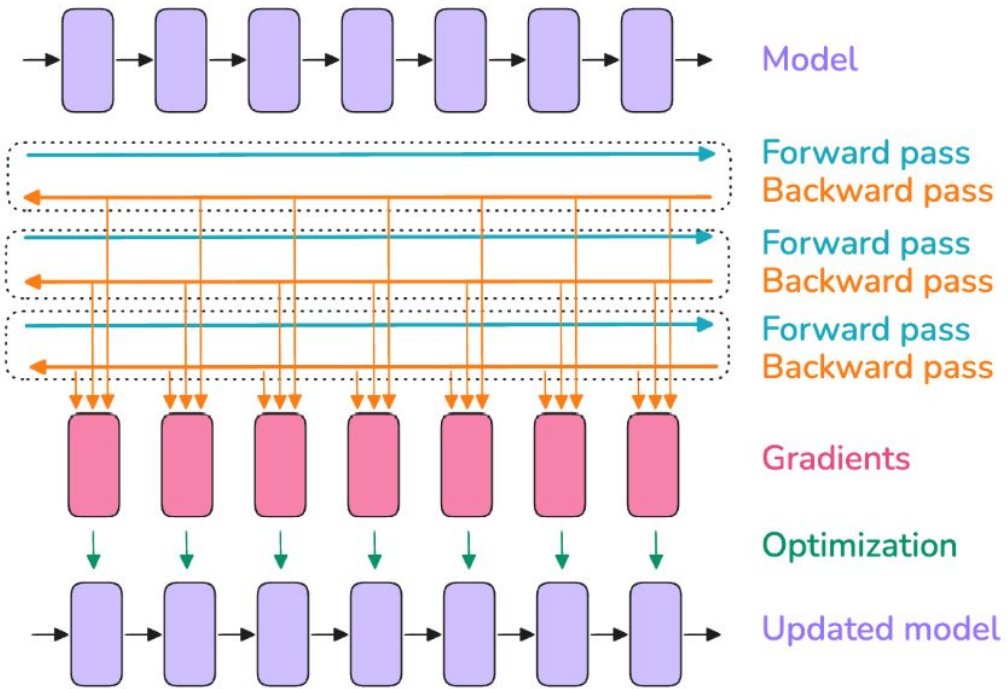
Model Training - Reduce Memory Usage

Gradient Accumulation

1- Split the Batch into mini batches

2- Compute the gradients for each mini-batch

3- Just save the gradient and accumulate it (sum the gradients)



https://huggingface.co/spaces/nanotron/ultrascale-playbook

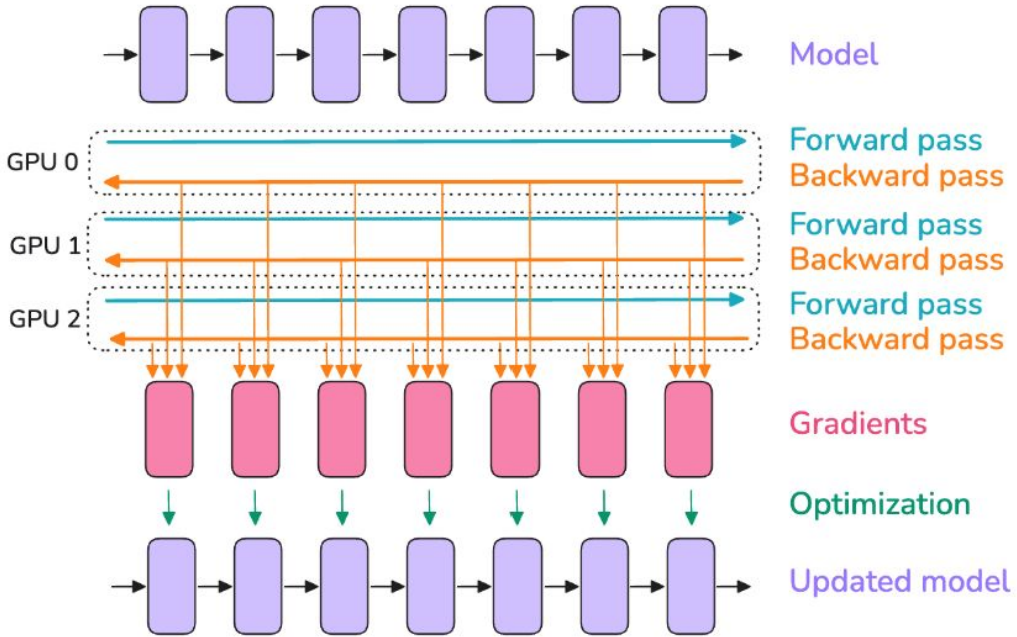
Model Training - Data Parallelism

Run each mini batch on a different gpu

Distribute all gradients to all gpus and do the backward pass and updates

This is called ALL-Reduce

Wasted time for the GPU communication



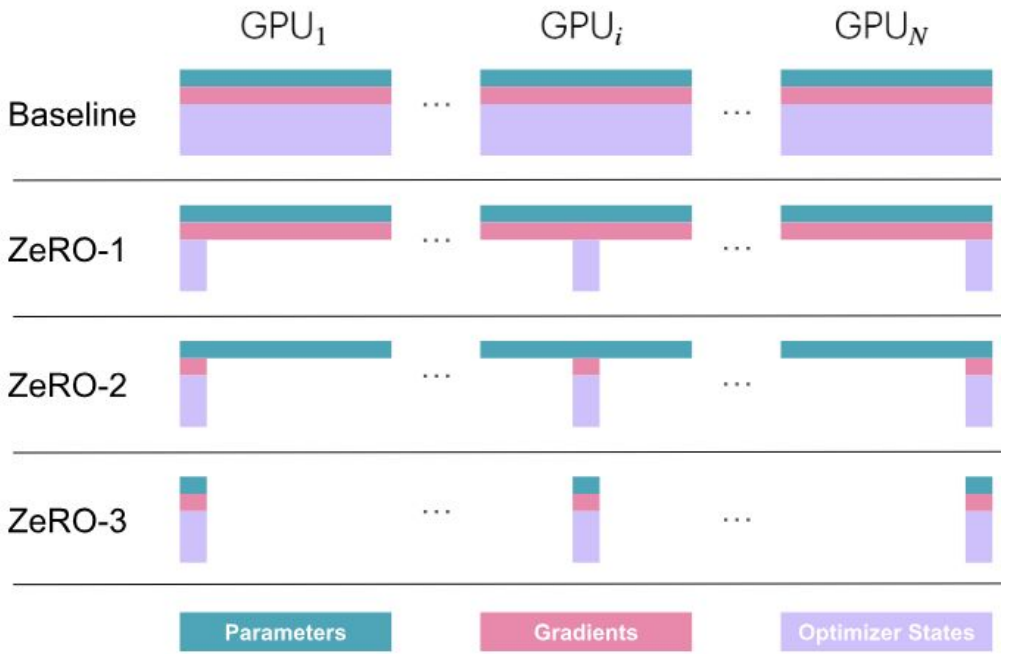
https://huggingface.co/spaces/nanotron/ultrascale-playbook

Model Training - Data Parallelism - ZeRO

DeepSpeed Zero Redundancy Optimizer (ZeRO)

Shard the parameters, gradients, and optimizer states over all gpus

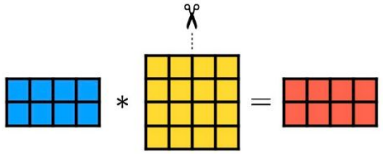
Slices are then reconstructed when and if needed



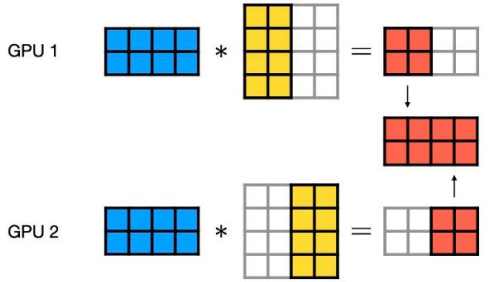
https://huggingface.co/spaces/nanotron/ultrascale-playbook

Model Training - Tensor Parallelism

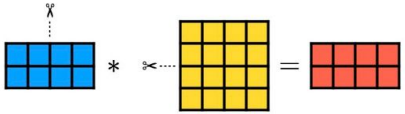
Split tensors across GPUs

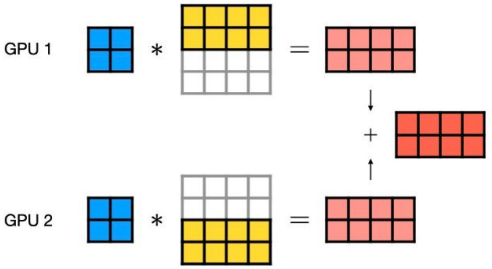


Column-wise



Row - wise





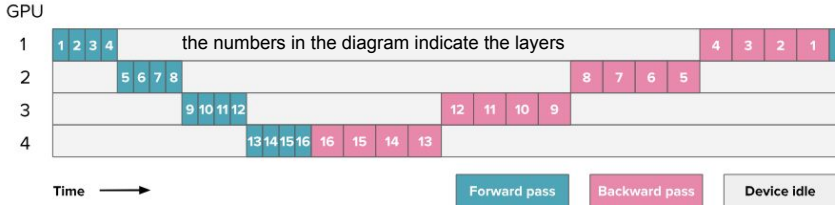
Need to overlap communication with computation

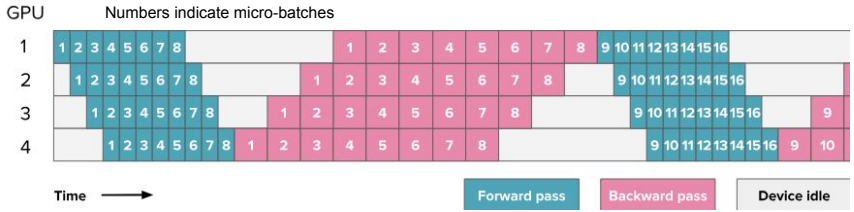
https://huggingface.co/spaces/nanotron/ultrascale-playbook

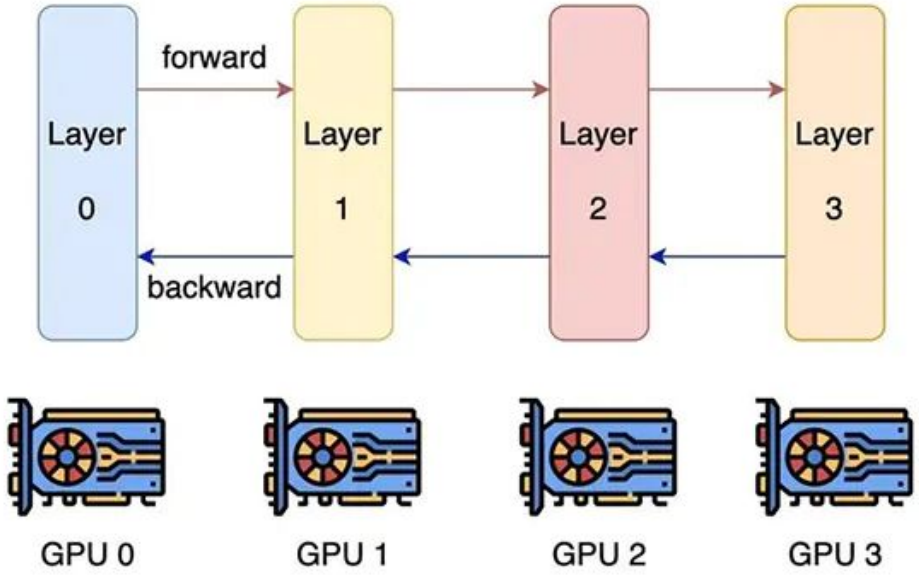
Model Training - Pipeline Parallelism

Split layers across GPUs

Need to overlap communication with computation







https://huggingface.co/spaces/nanotron/ultrascale-playbook

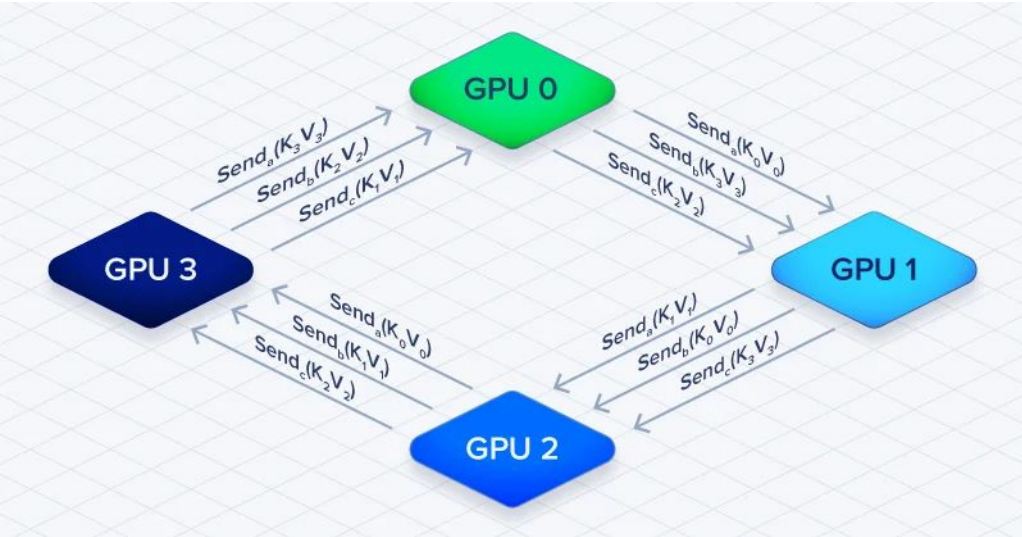
Model Training - Context Parallelism

When scaling to 128k+ tokens, even 1 example can't fit multiple GPUs within a node

Split the input along the sequence dimension

Attention layer needs special attention because it's context dependent

Ring Attention



https://www.exxactcorp.com/blog/deep-learning/how-llms-reach-large-token-context-windows

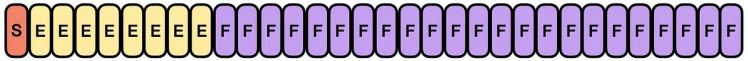
https://huggingface.co/spaces/nanotron/ultrascale-playbook

Model Training - Reduced Precision Training

Instead of BF16 let's go to FP8 or even FP4

Mantissa

float 32



Sign (1 bit) Exponent (8 bits) Fraction (23 bits)

$$(-1)^{b_{31}} \times 2^{(b_{30}b_{29}\ldots b_{23})_2 - 127} \times (1.b_{22}b_{21}\ldots b_0)_2 ,$$

which yields

$$\text{value} = (-1)^{\text{sign}} \times 2^{(E-127)} \times \left(1 + \sum_{i=1}^{23} b_{23-i} 2^{-i}\right) .$$

Source: https://en.wikipedia.org/wiki/Single-precision_floating-point_format

Model Training - Reduced Precision Training

FP8: DeepSeek v3 showed it works

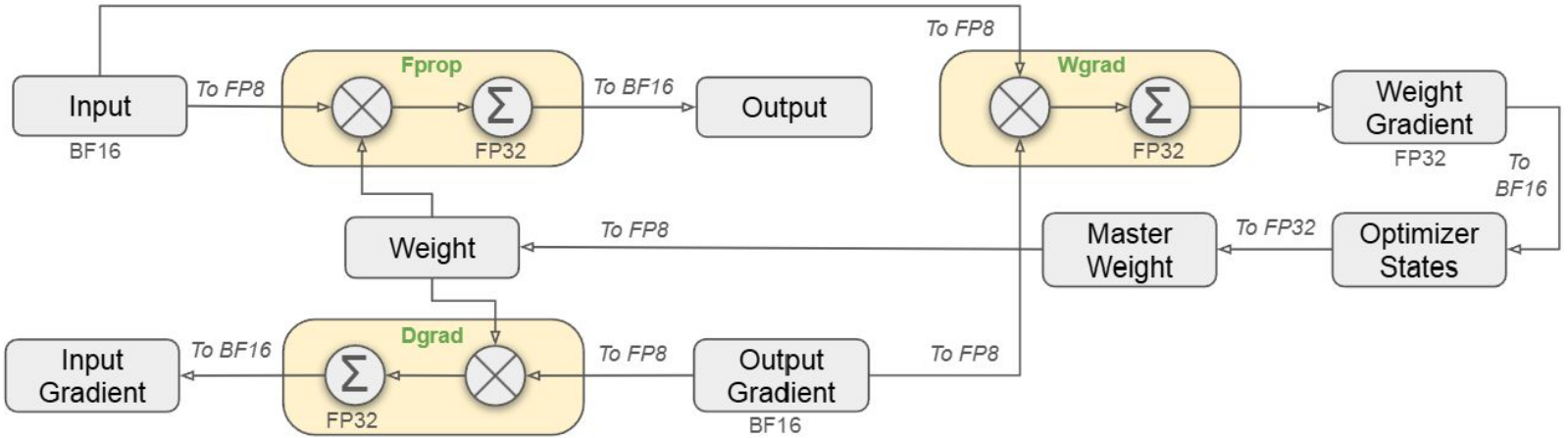
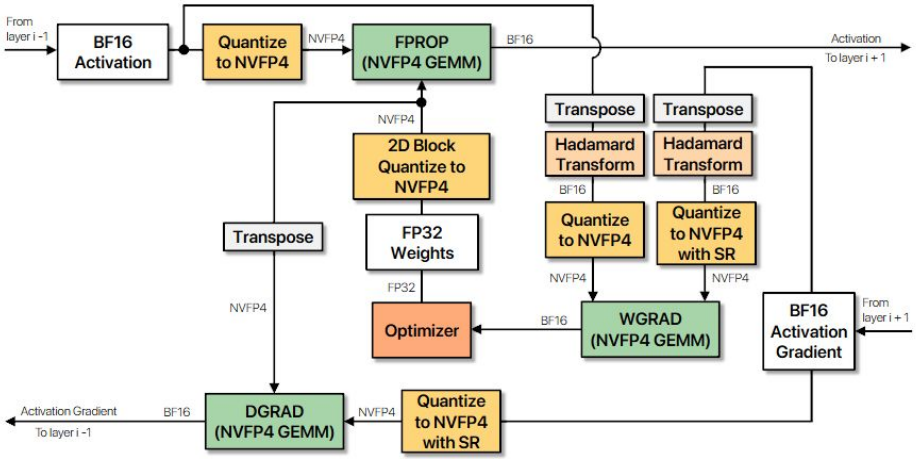


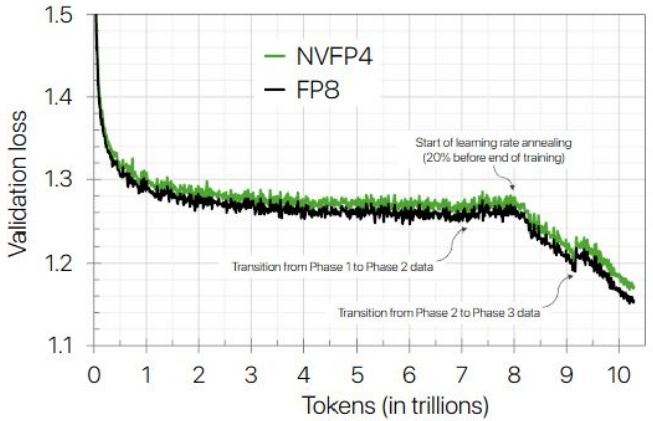
Figure 6 | The overall mixed precision framework with FP8 data format. For clarification, only the `Linear` operator is illustrated.

Liu, Aixin, et al. "Deepseek-v3 technical report." *arXiv preprint arXiv:2412.19437* (2024).

Model Training - Reduced Precision Training

FP4 might work





ıre 2 | Validation loss of NVFP4 and FP8 pretraining for the 12B model using 10T tokens.

Abecassis, Felix, et al. "Pretraining large language models with nvfp4." *arXiv preprint arXiv:2509.25149* (2025).

Other Challenges

GPU stability: In a 54-day snapshot of Llama 3 405B pretraining, they saw **419 job interruptions on 16K-GPU**

- You need elastic training and fast recovery (loading a model might take an hour of cluster time)

Loss spikes and Rollbacks:

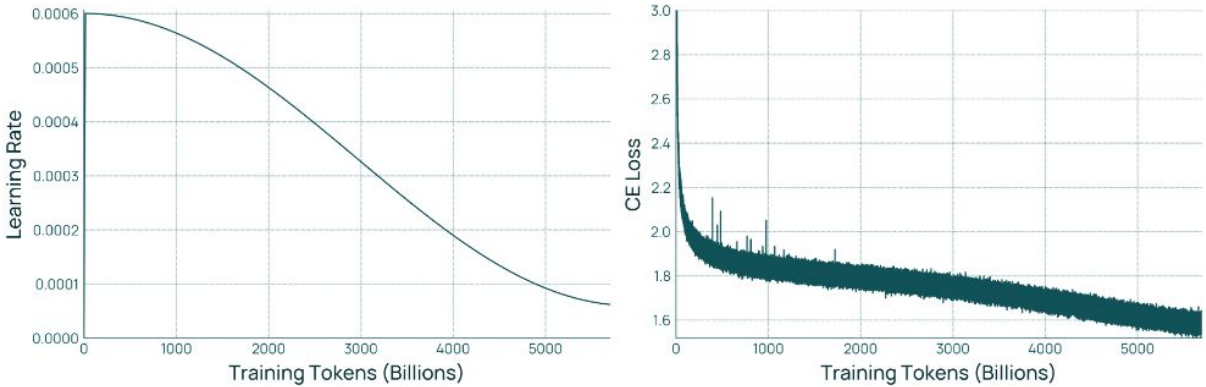
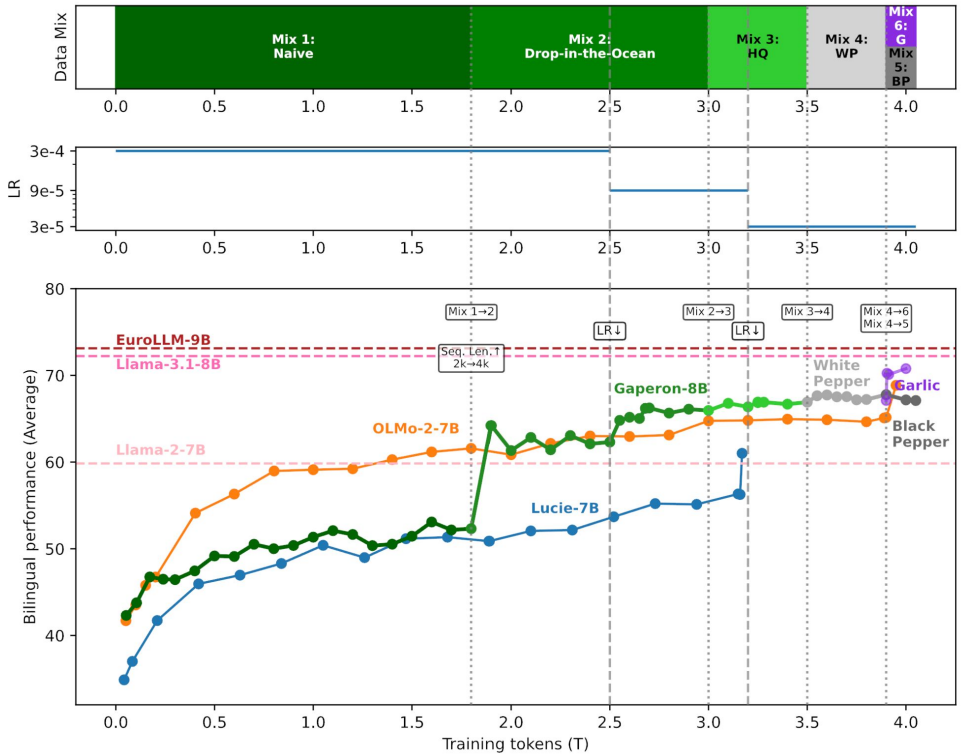


Figure 4 Learning rate schedule and loss for Olmo 3 Base 32B. The learning rate schedule is a cosine schedule over

Story of Gaperon



Godey, N., Antoun, W., Touchent, R., Bawden, R., de la Clergerie, É., Sagot, B., & Seddah, D. (2025). Gaperon: A Peppered English-French Generative Language Model Suite. *arXiv preprint arXiv:2510.25771*.

Resources

Check out all blogs at HF:

- https://huggingface.co/spaces/nanotron/ultrascale-playbook
- https://huggingface.co/spaces/HuggingFaceFW/FinePDFsBlog
- https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1

Karpathy LLM training series on Youtube: https://www.youtube.com/@AndrejKarpathy

Karpathy LLM training code: https://github.com/karpathy/nanogpt

Stas's ML Engineering Book: https://github.com/stas00/ml-engineering

CS336 Language Modeling from Scratch from Stanford: https://cs336.stanford.edu/

Thank you

Get the slides from: wiss.dev