

# FALL 2024 COS597R: DEEP DIVE INTO LARGE LANGUAGE MODELS

Danqi Chen, Sanjeev Arora



PRINCETON  
UNIVERSITY

Lecture 6: Data Curation

<https://princeton-cos597r.github.io/>

# Required reading

## **dolm: an Open Corpus of Three Trillion Tokens for Language Model Pretraining Research**

Luca Soldaini   Rodney Kinney   Akshita Bhagia   Dustin Schwenk    
David Atkinson  Russell Authur  Ben Begin   Khyathi Chandu   
Jennifer Dumas  Yanai Elazar   Valentin Hofmann  Ananya Harsh Jha   
Sachin Kumar  Li Lucy  Xinxi Lyu  Nathan Lambert  Ian Magnusson   
Jacob Morrison  Niklas Muennighoff Aakanksha Naik  Crystal Nam   
Matthew E. Peters  Abhilasha Ravichander  Kyle Richardson  Zejiang Shen   
Emma Strubell  Nishant Subramani  Oyvind Tafjord  Pete Walsh   
Luke Zettlemoyer  Noah A. Smith   Hannaneh Hajishirzi    
Iz Beltagy  Dirk Groeneveld  Jesse Dodge   
Kyle Lo  

Also highly recommended: <https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1>



**FineWeb: decanting the web for the finest  
text data at scale**

# This lecture

- An overview of pre-training data
- What is good data?
- What are important steps of data filtering? Where do we see open research going?

## What is open data?

- The data and **toolkits** are released to the public
- The process of curation is well documented

---

# An overview of pre-training data

# The status of pre-training data

- Pre-training data has been **highly opaque** and arguably decides the quality of pre-trained models
  - e.g., OpenAI, Anthropic, Google, ..
- SOTA open-weight models only describe their pre-training data composition vaguely too

## 2.1 Pretraining Data

LLAMA 2

Our training corpus includes a new mix of data from publicly available sources, which does not include data from Meta's products or services. We made an effort to remove data from certain sites known to contain a high volume of personal information about private individuals. We trained on 2 trillion tokens of data as this provides a good performance–cost trade-off, up-sampling the most factual sources in an effort to increase knowledge and dampen hallucinations.

LLAMA 3

**Data.** Compared to prior versions of Llama (Touvron et al., 2023a,b), we improved both the quantity and quality of the data we use for pre-training and post-training. These improvements include the development of more careful pre-processing and curation pipelines for pre-training data and the development of more rigorous quality assurance and filtering approaches for post-training data. We pre-train Llama 3 on a corpus of about 15T multilingual tokens, compared to 1.8T tokens for Llama 2.

# Where do they even got all data?

- **Option #1:** build Web crawler themselves e.g., OpenAI, Anthropic
- **Option #2:** clean and extract from a public repository of crawled webpages

## ClaudeBot

Last updated 17 hours ago.

### What is ClaudeBot?

#### About

ClaudeBot is a web crawler operated by Anthropic to download training data for its LLMs (Large Language Models) that power AI products like Claude.

#### Track ClaudeBot on Your Website

You can see when ClaudeBot visits your website using the API or WordPress plugin.

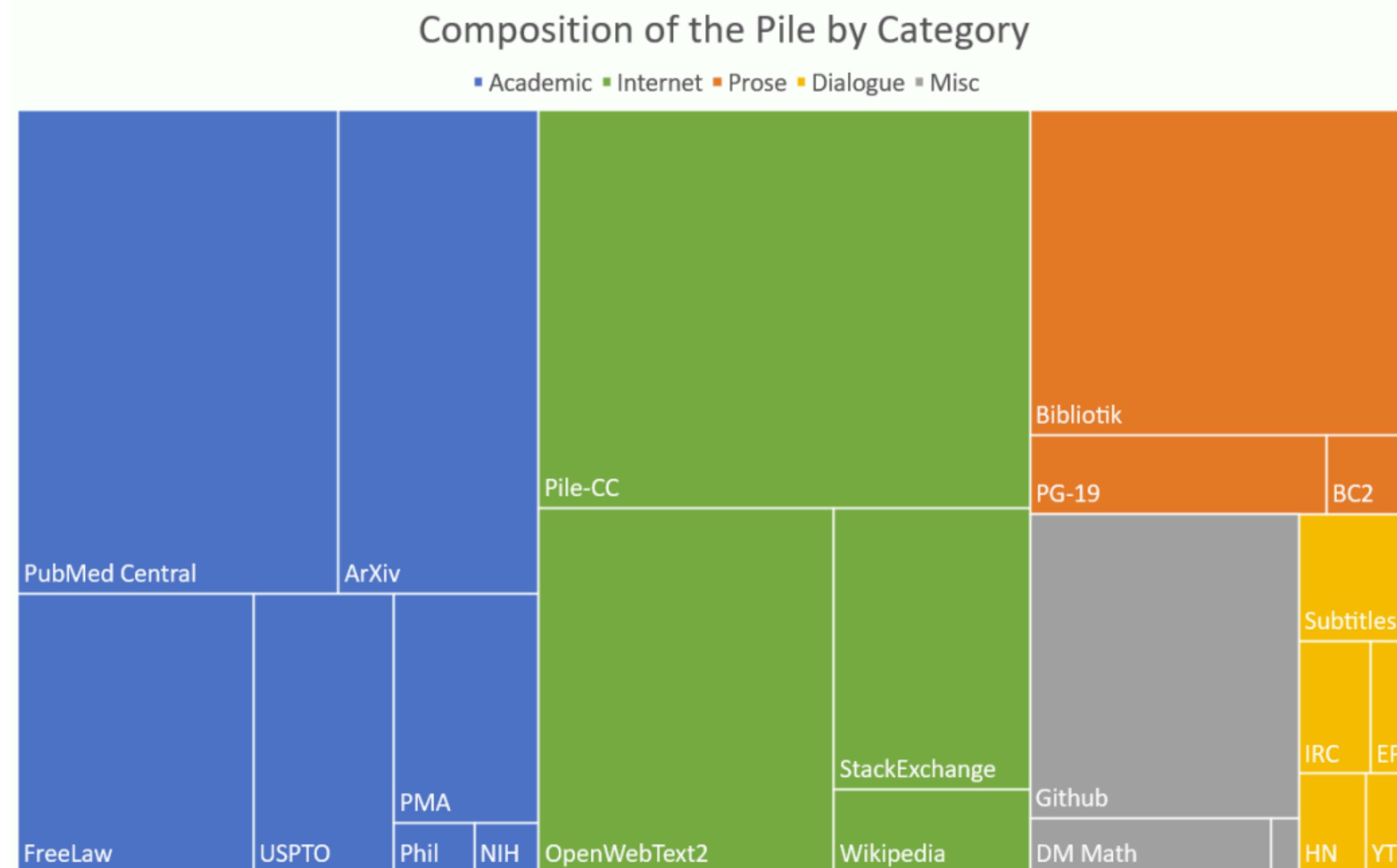
[Set Up Agent Analytics](#)



- Since 2007
- April 2024: the latest CC crawl contains 2.7 billion web pages, 38B TB of uncompressed HTML text content

# Pre-training data in the public

- **C4 (Raffel et al., 2020)**: 175B tokens, cleaned from Common Crawl
- **Pile (Gao et al., 2020)**: 387B tokens, from diverse sources



# Pre-training data in the public

- **RedPajama v1:** 1.2T tokens, an open source effort to reproduce Llama v1 training data

Dataset	Token Count
Commoncrawl	878 Billion
C4	175 Billion
GitHub	59 Billion
ArXiv	28 Billion
Wikipedia	24 Billion
StackExchange	20 Billion
Total	1.2 Trillion

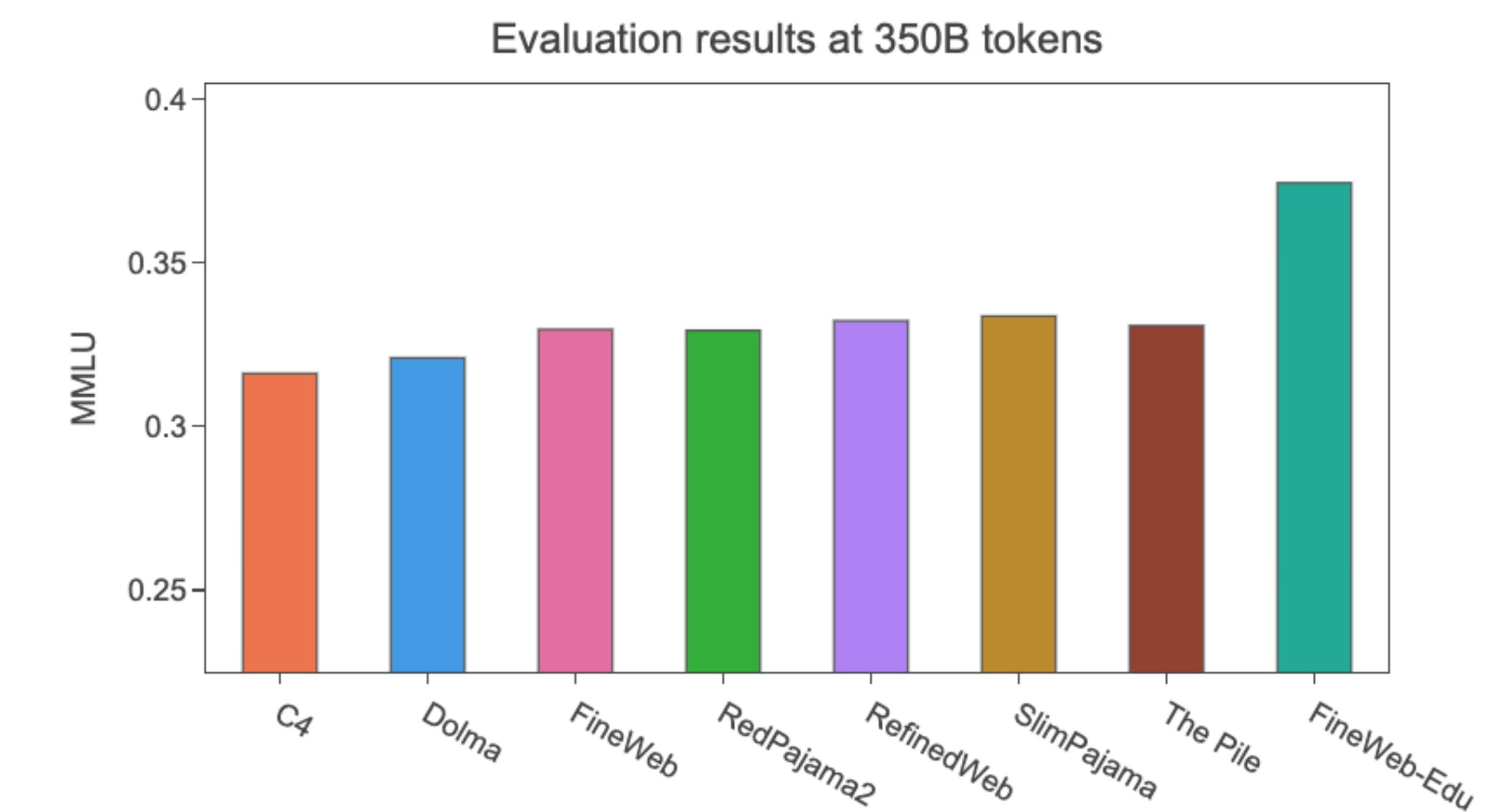
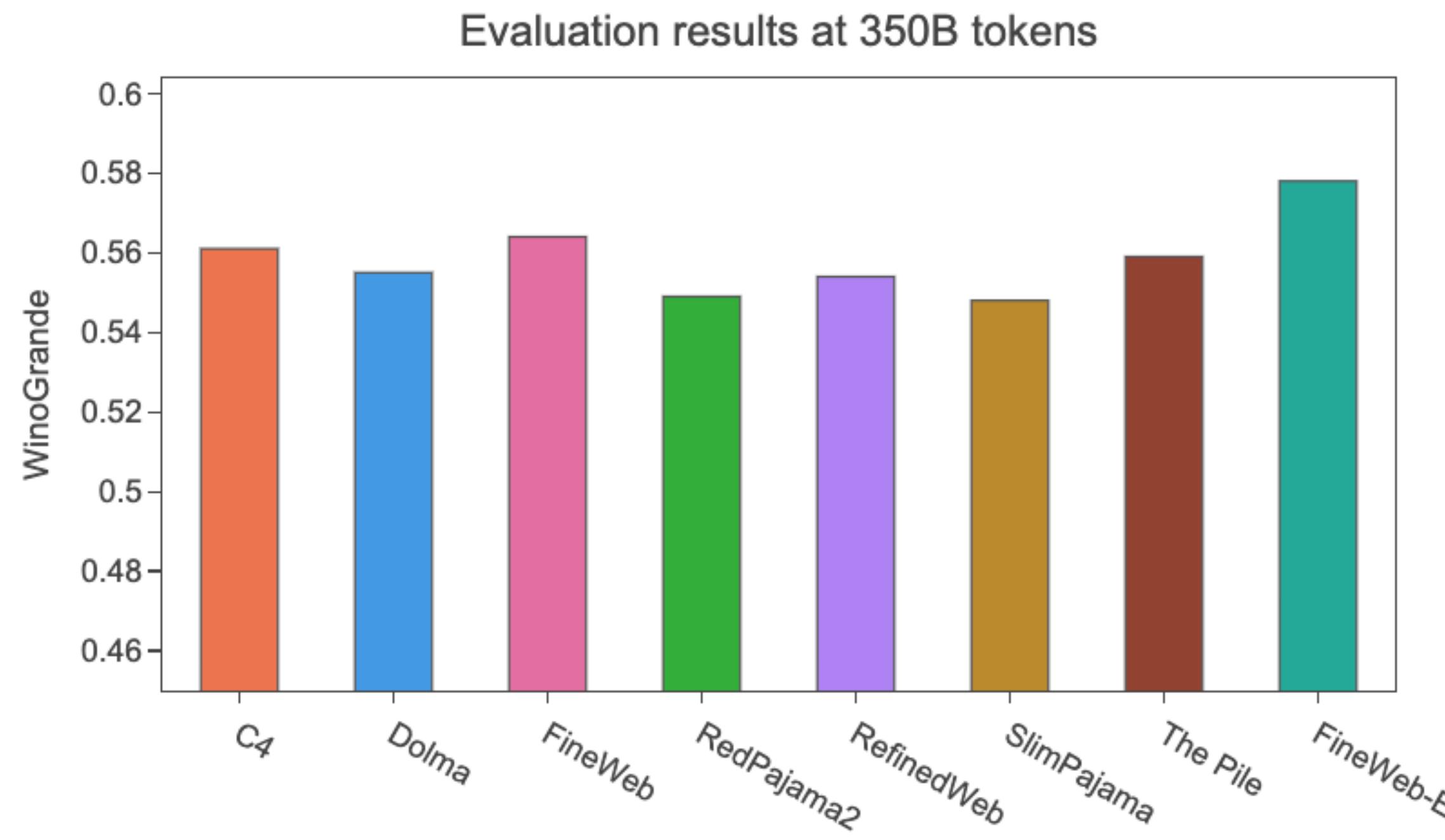


- SlimPajama: 627B cleaned and deduplicated version of RedPajama

- RedPajama-v2: 30 trillion tokens from CC, only lightly curated..

# Pre-training data in the public

- **RefinedWeb** (Penedo et al., 2023): 600B tokens only from Common Crawl
- **FineWeb** (Penedo et al., 2024): 15T tokens only from Common Crawl



- FineWeb-Edu is a subset of FineWeb focusing on educational content

# Pre-training data in the public

- **Dolma** (Soldaini et al., 2024)



Source	Doc Type	Llama tokens (billions)
Common Crawl	🌐 web pages	2,281
The Stack	💻 code	411
C4	🌐 web pages	198
Reddit	💬 social media	89
PeS2o	🎓 STEM papers	70
Project Gutenberg	📖 books	6.0
Wikipedia, Wikibooks	🔖 encyclopedic	4.3

Total = 3T tokens



---

## What is good data?

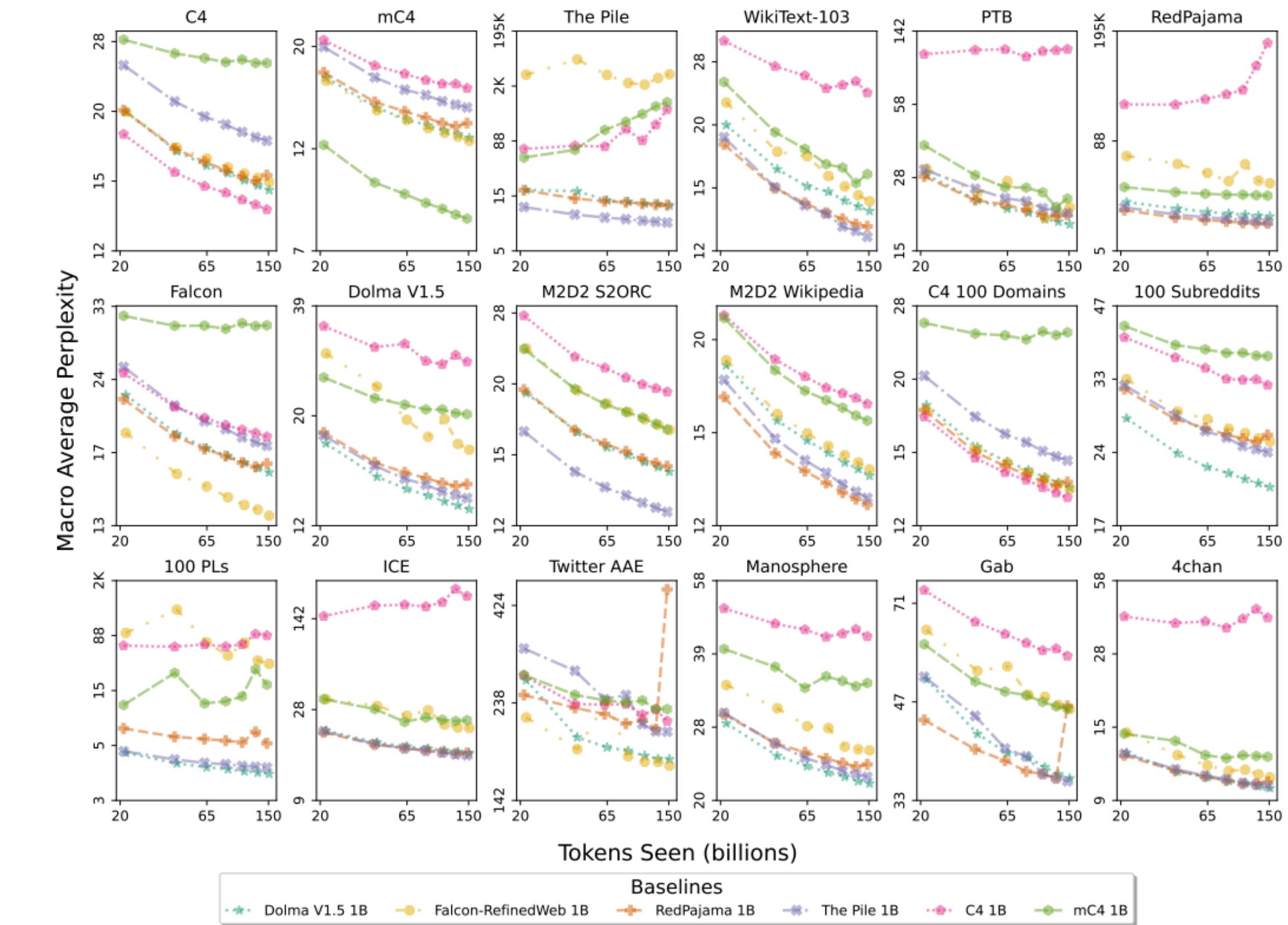
“High-quality” is not well-defined and not even a property of documents that can be clearly perceived through direct human observations..

# How to compare datasets/perform data ablations?

- You have to run a lot of ablation experiments with reasonably-sized models/data
  - **Dolma**: 1.2B models, 150B tokens
  - **FineWeb**: 1.8B models, 28B tokens (according to Chinchilla law)

- **Evaluation #1: Perplexity**

- Evaluate on a held-out validation set (see Dolma paper D.2), or fitting to many diverse domains, e.g., Paloma (Magnusson et al., 2024)

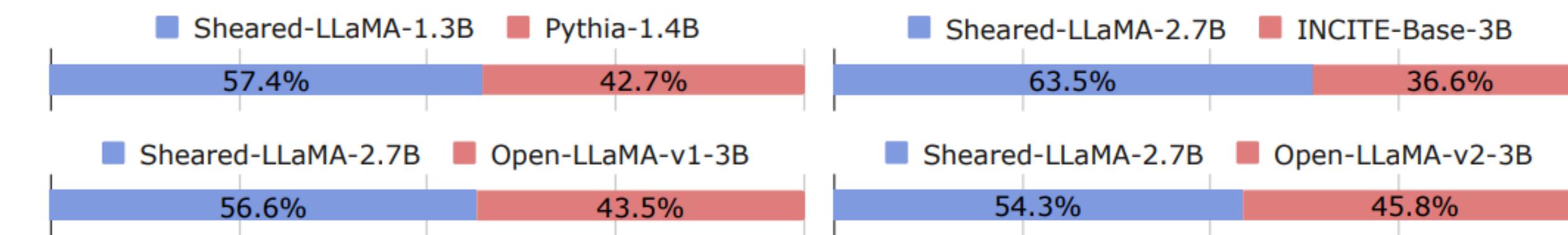


# How to compare datasets/perform data ablations?

- **Evaluation #2:** Compare generations of different models and let humans rate them (e.g., ChatBot Arena), or use LLM-as-judge
  - Caveat: usually needs to go through instruction tuning stage (next lecture)



*Figure 3.* Instruction following win rates of models trained with QuRating ( $\tau = 2.0$ ) vs. uniform data selection after instruction fine-tuning on 10K ShareGPT examples.



*Figure 3:* Sheared-LLaMAs outperform Pythia-1.4B, INCITE-Base-3B, OpenLLaMA-3B-v1 and OpenLLaMA-3B-v2 in instruction tuning.

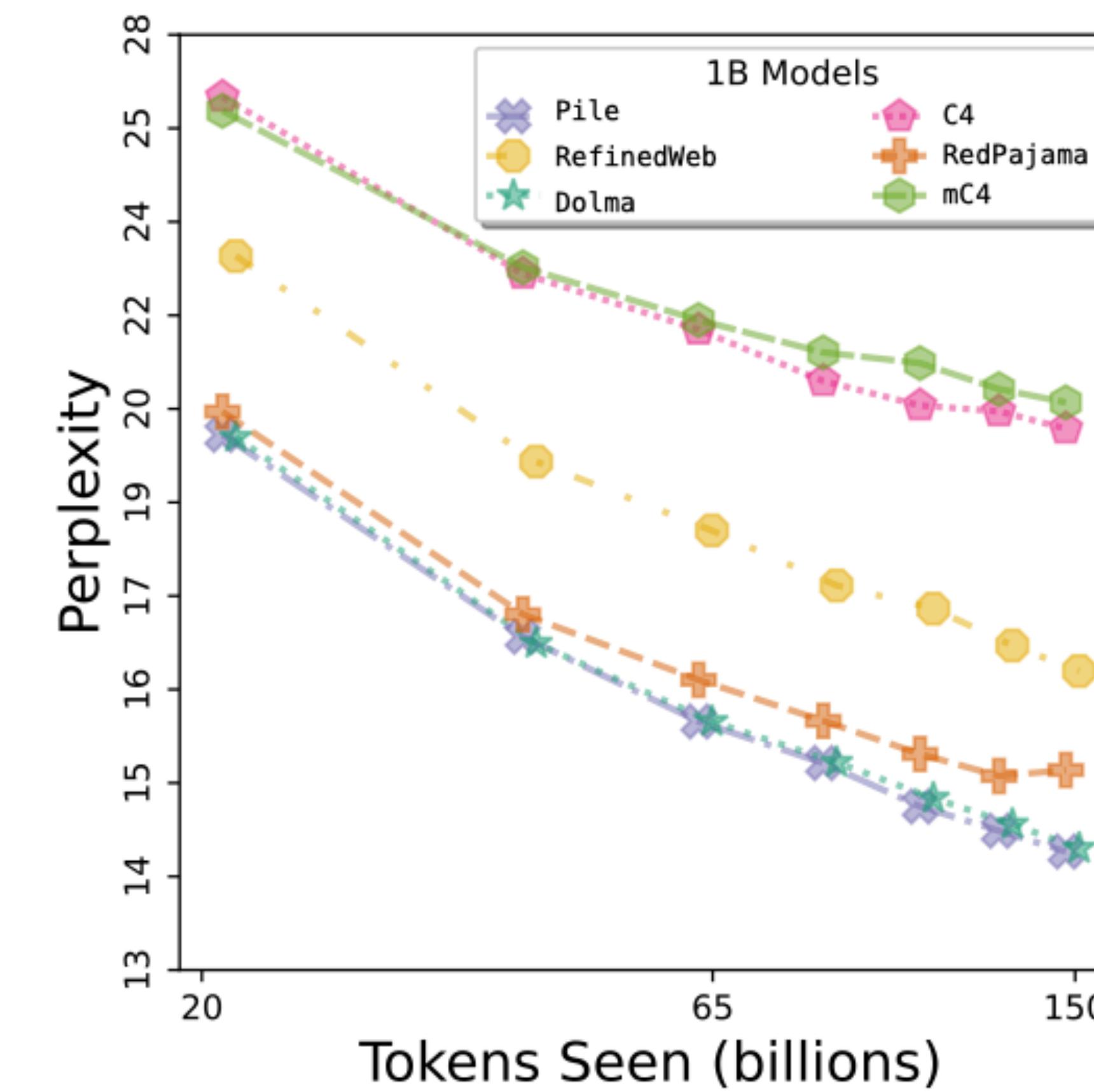
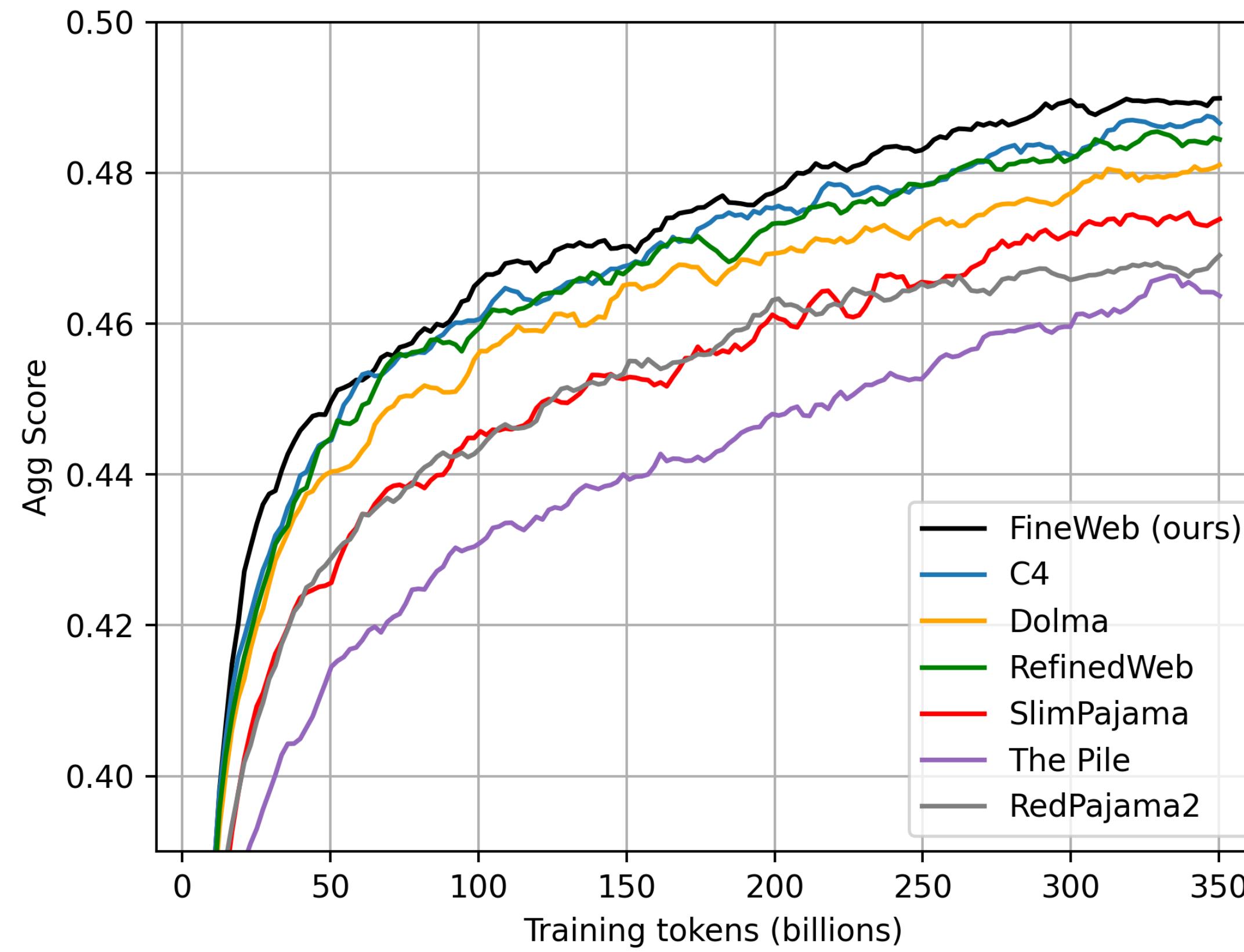
- From **QuRating** (Wettig et al., 2024)

- From **Sheared LLMA** (Xia et al., 2024)

# How to compare datasets/perform data ablations?

- **Evaluation #3** (current default): in-context learning (zero-shot or few-shot) on a diverse set of downstream tasks
  - Caveat: the trends might vary a lot on different tasks; overall score doesn't tell the full picture
  - You need to select a set of “early-signal” and reliable tasks too
  - Dolma's selection: ARC, BoolQ, HellaSwag, OpenBookQA, PIQA, SciQ, WinoGrande
  - FineWeb's selection: **CommonsenseQA**, HellaSwag, OpenBookQA, PIQA, **SIQA**, WinoGrande, ARC, **MMLU**

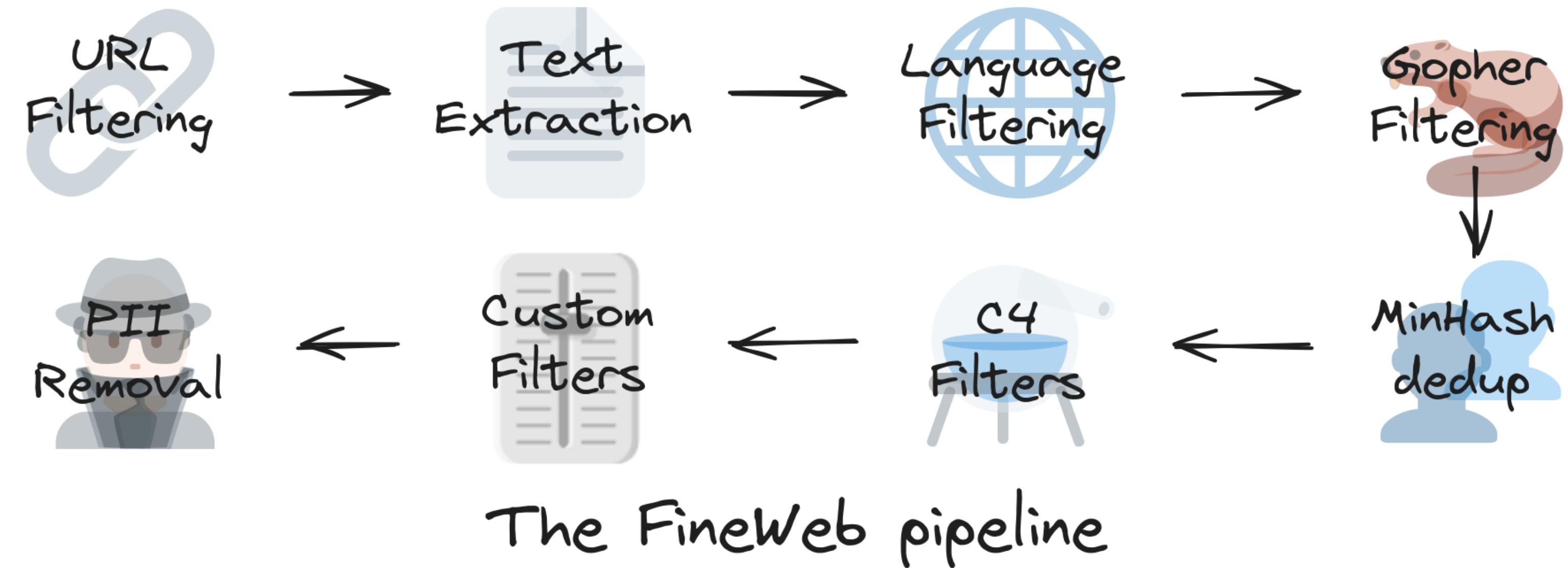
# Perplexity vs downstream evaluation



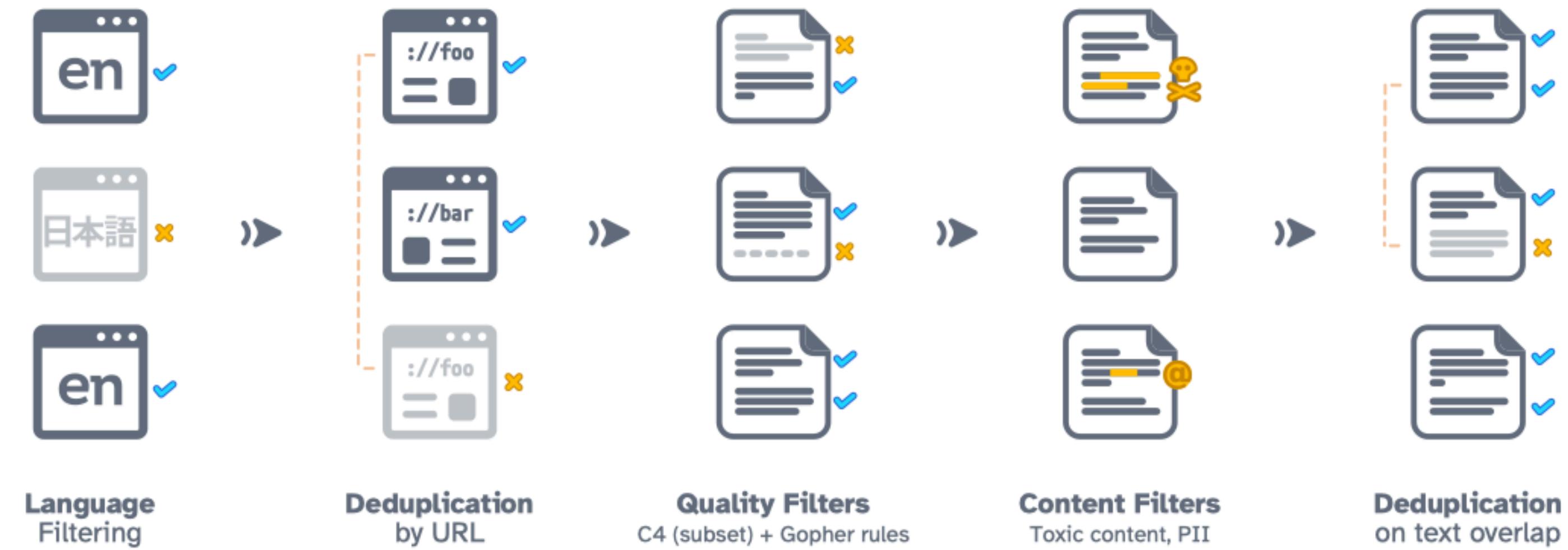
---

## Important steps for data filtering

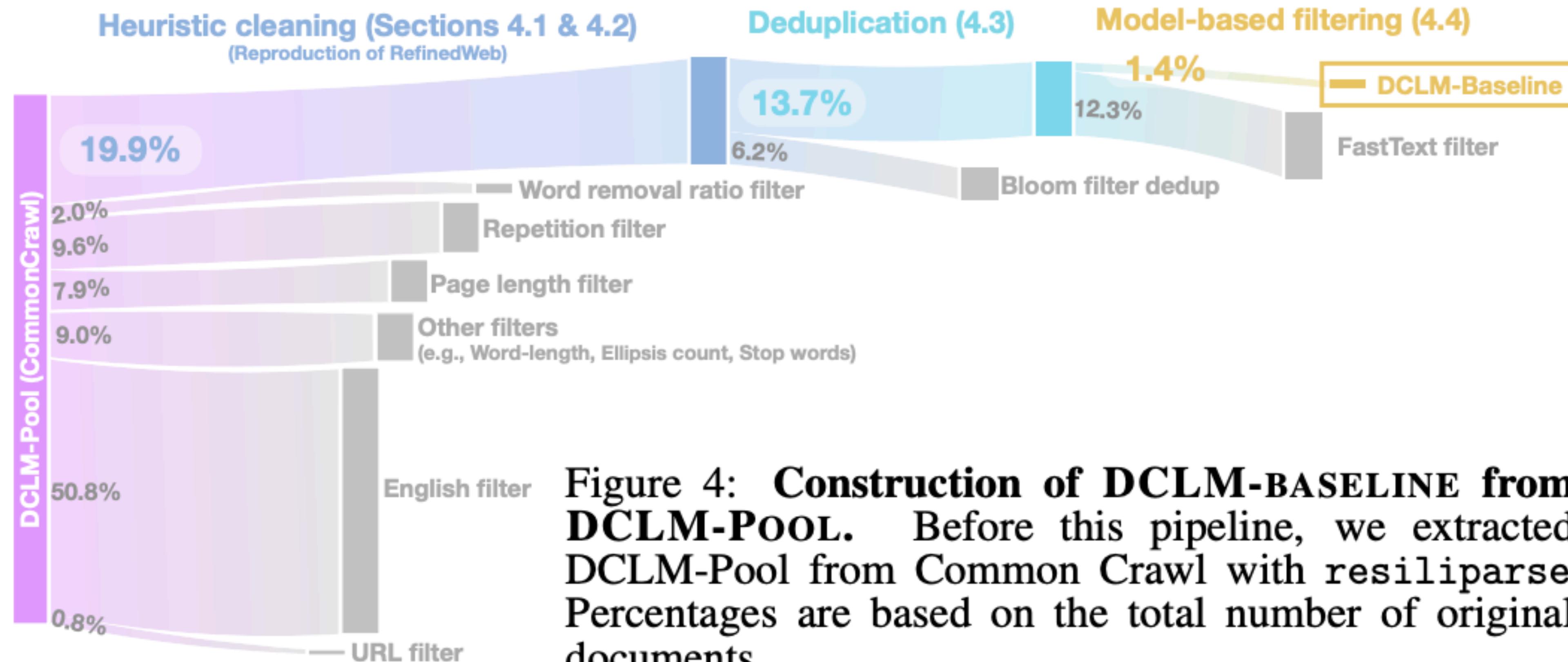
- **FineWeb pipeline:**



- **Dolma pipeline:**



- **DCLM pipeline:**



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

# Important steps (roughly)

- **Text extraction:** pulls content from raw HTML (example: trafilatura)
- **Language filtering:** keep only English text (with a document score  $\geq$  a threshold)
- **Deduplication:** improves training efficiency, reduces memorization and improves generalization
  - Examples: MinHash, suffix array, near-duplicate Bloom filtering
- **Quality filtering:** removes low-quality and repetitive information
  - Can be either heuristic or model-based
- **Content filtering**
  - Remove toxic and harmful content - usually by trained classifier
  - Remove personal identifiable information (PII) - usually by regular expressions

# Heuristic vs model-based filtering

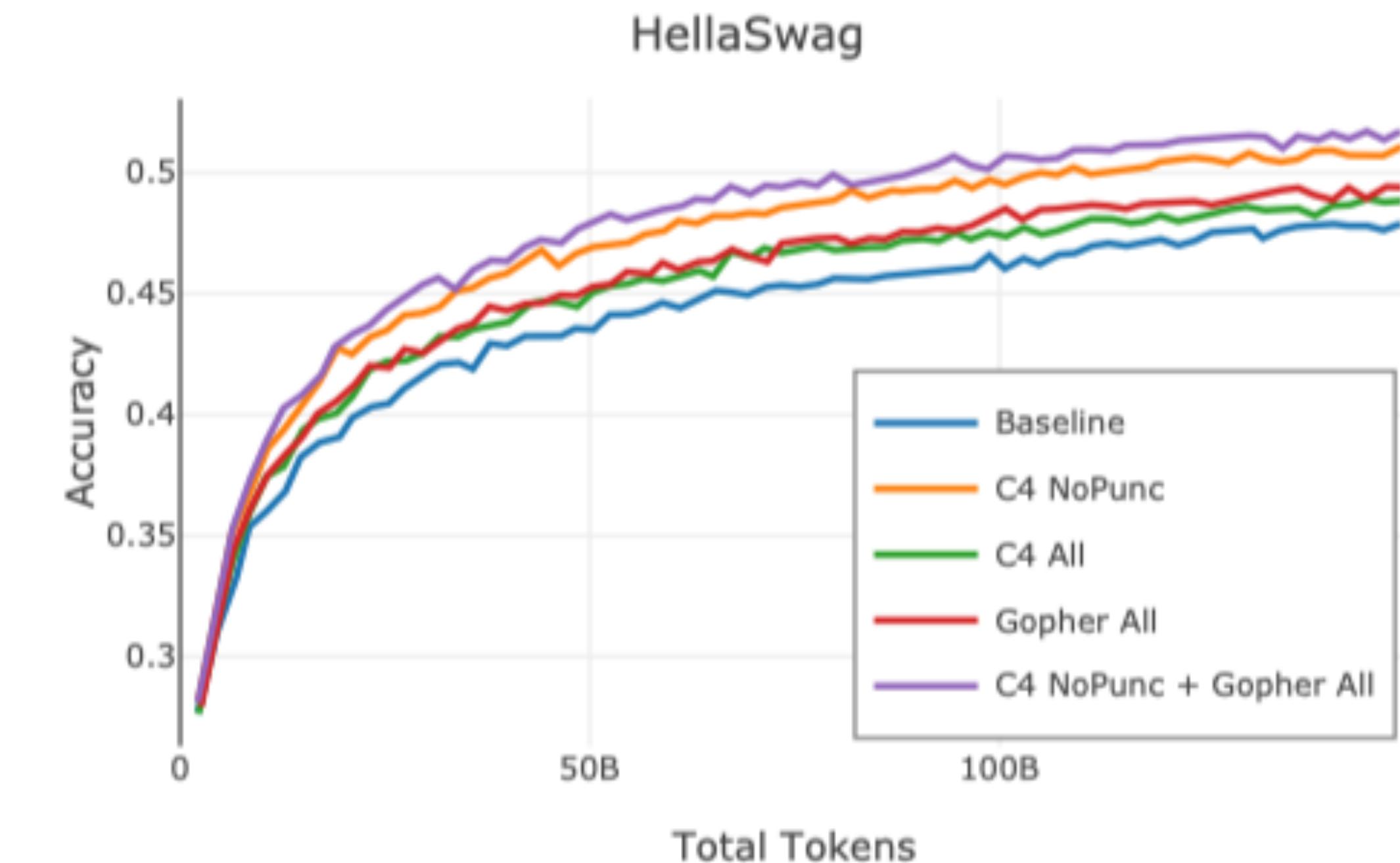
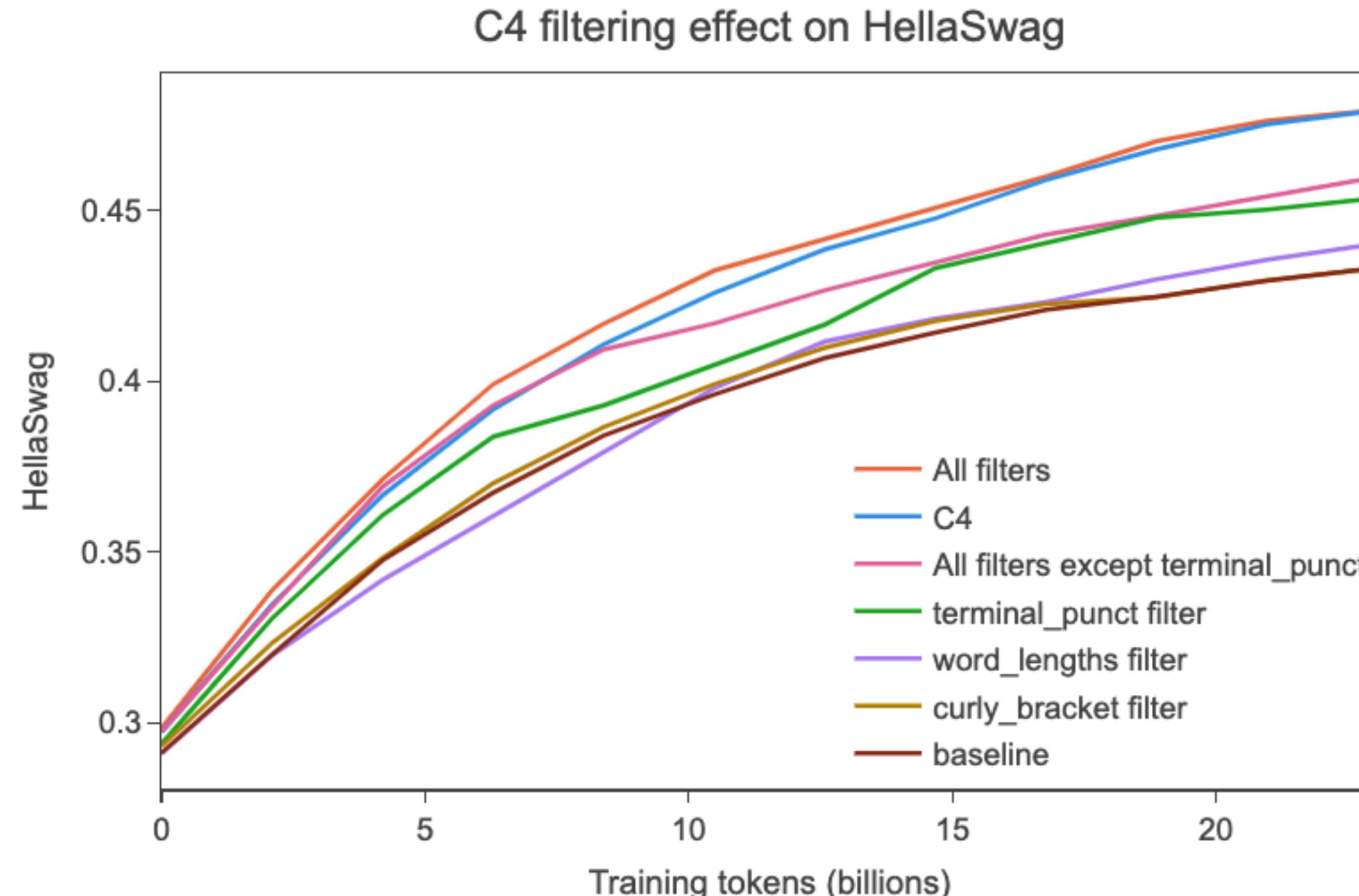
## C4 rules (Raffel et al., 2020)

- We only retained lines that ended in a terminal punctuation mark (i.e. a period, exclamation mark, question mark, or end quotation mark).
- We discarded any page with fewer than 5 sentences and only retained lines that contained at least 3 words.
- We removed any page that contained any word on the “List of Dirty, Naughty, Obscene or Otherwise Bad Words”.<sup>6</sup>
- Many of the scraped pages contained warnings stating that Javascript should be enabled so we removed any line with the word Javascript.
- Some pages had placeholder “lorem ipsum” text; we removed any page where the phrase “lorem ipsum” appeared.
- Some pages inadvertently contained code. Since the curly bracket “{” appears in many programming languages (such as Javascript, widely used on the web) but not in natural text, we removed any pages that contained a curly bracket.

## Gopher Rules (Rae et al., 2021)

```
def gopher_rules_pass(sample) -> bool:  
    """ function returns True if the sample complies with Gopher rules """  
    signals = json.loads(sample["quality_signals"])  
  
    # rule 1: number of words between 50 and 10'000  
    word_count = signals["rps_doc_word_count"][0][2]  
    if word_count < 50 or word_count > 10_000:  
        return False  
  
    # rule 2: mean word length between 3 and 10  
    mean_word_length = signals["rps_doc_mean_word_length"][0][2]  
    if mean_word_length < 3 or mean_word_length > 10:  
        return False  
  
    # rule 2: symbol to word ratio below 0.1  
    symbol_word_ratio = signals["rps_doc_symbol_to_word_ratio"][0][2]  
    if symbol_word_ratio > 0.1:  
        return False  
  
    # rule 3: 90% of lines need to start without a bullet point  
    n_lines = signals["ccnet_nlines"][0][2]  
    n_lines_bulletpoint_start = sum(map(lambda ln: ln[2], signals["rps_lines_start_w":]))  
    if n_lines_bulletpoint_start / n_lines > 0.9:  
        return False  
  
    # rule 4: the ratio between characters in the most frequent 2-gram and the total  
    # of characters must be below 0.2  
    top_2_gram_frac = signals["rps_doc_frac_chars_top_2gram"][0][2]  
    if top_2_gram_frac > 0.2:  
        return False  
  
    # rule 5: ...
```

# Remarks on heuristic filters



- From **FineWeb**

- From **Dolma**

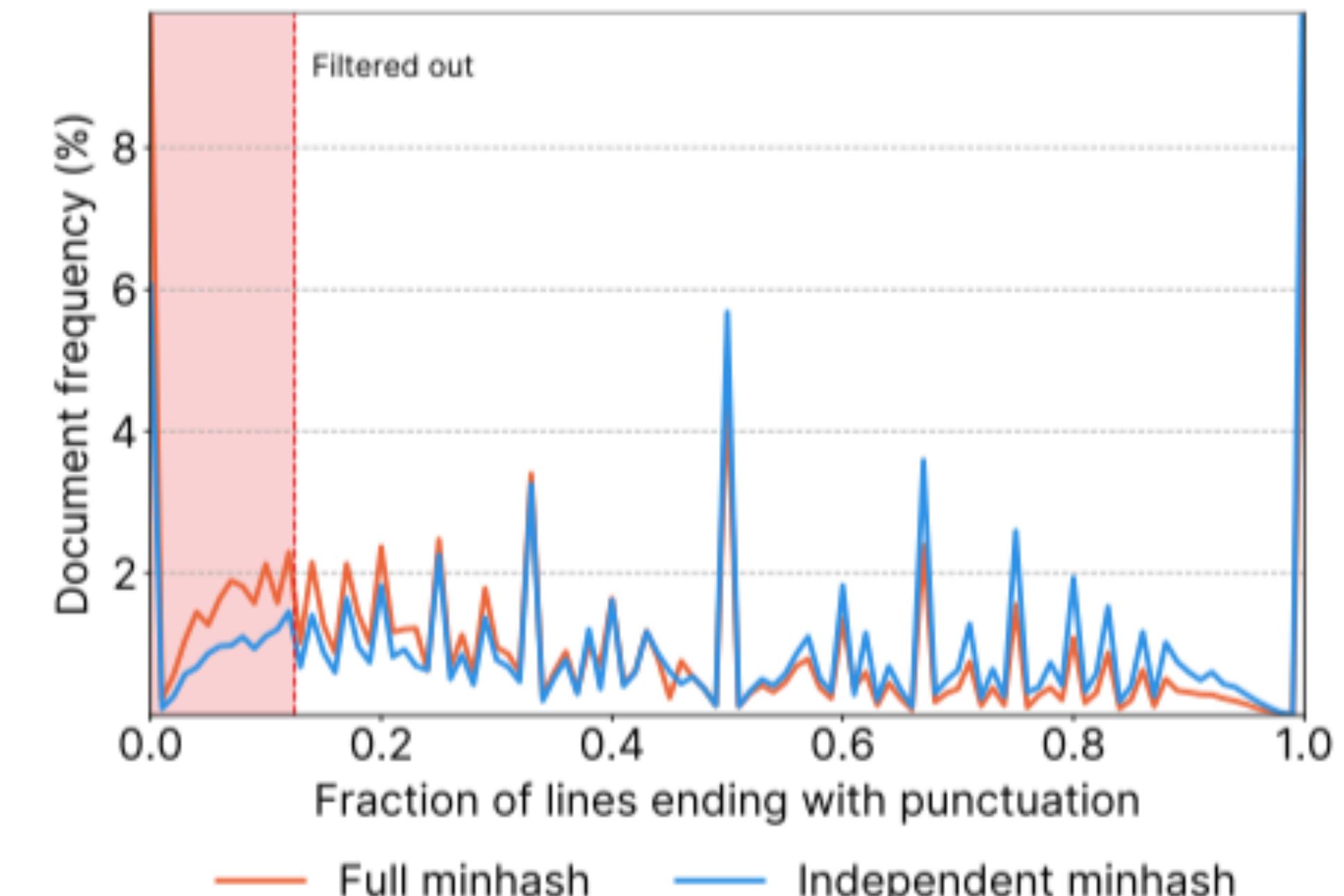
# How to develop heuristic filters?

## A STATISTICAL APPROACH TO DEVELOP HEURISTIC FILTERS

To develop new heuristic filters and select their thresholds we devised a systematic process:

1. we started by collecting a very large list of high level statistics of our datasets (over **fifty** different metrics) ranging from common document-level metrics (e.g. number of lines, avg. line/word length, etc) to inter-document repetition metrics (inspired by MassiveText), on both a high quality and a lower quality web dataset;
2. we selected the metrics for which the Wasserstein distance between the two distributions (of the metric computed on each dataset) was larger;
3. we inspected the histograms of the two distributions and empirically chose a threshold that would make the lower quality dataset more closely resemble the higher quality one on this metric;
4. we validated the resulting filter (metric-threshold pair) by using it on a reference dataset and running small ablations.

• From **FineWeb**



Blue = high-quality  
Red = low-quality

# Heuristic vs model-based filtering

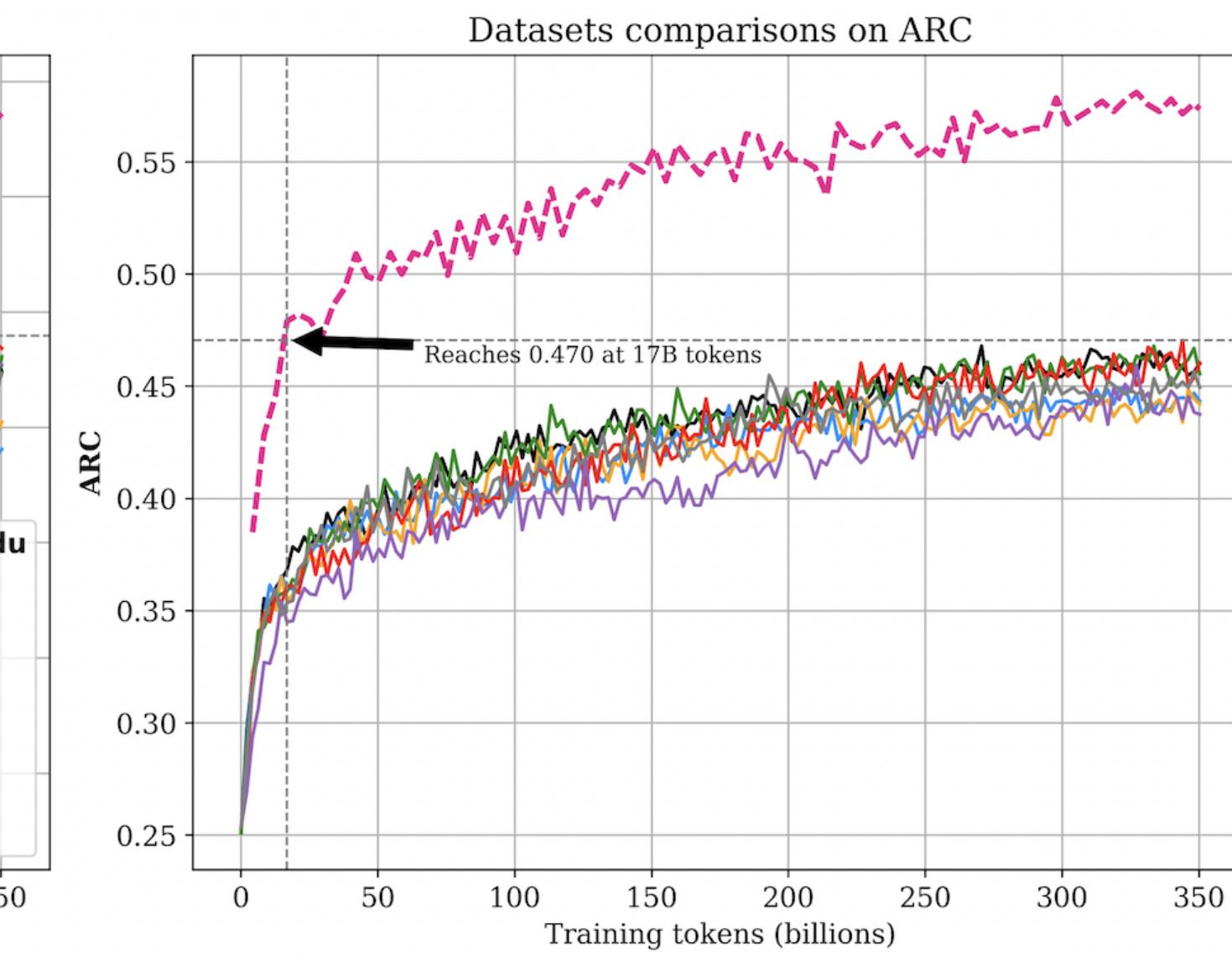
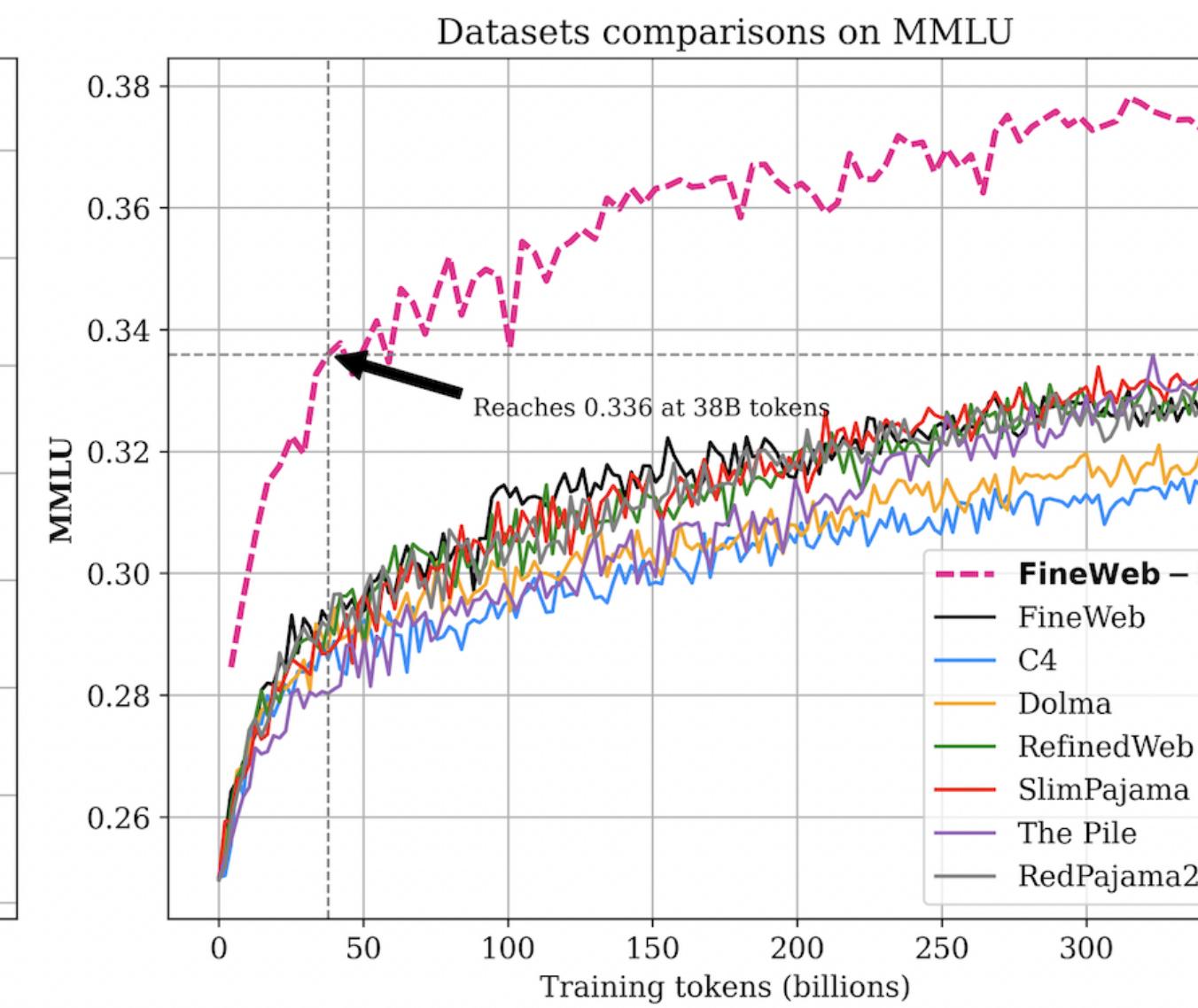
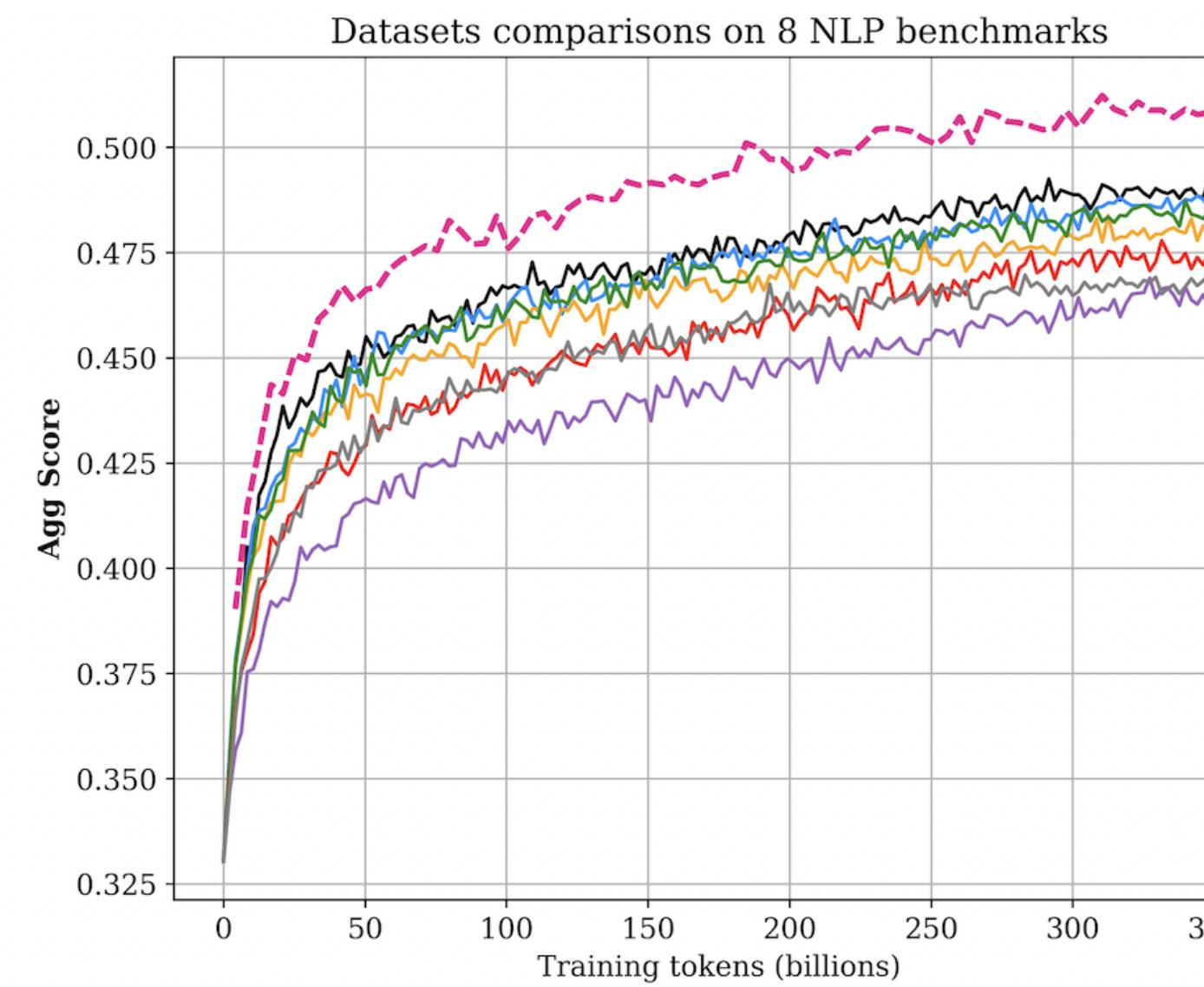
- **Perplexity filtering:** using a language model trained on “high-quality” text to measure the quality of any pertained text
  - **Dolma:** “KenLM perplexity that groups documents based on Wikipedia-likeness”
  - **QuRating:** “we use a pretrained ShearedLlama-2.7B model to select documents with highest/lowest perplexity scores”
  - **DCLM:** “We utilize a 154M parameter Transformer trained on a mix of English Wikipedia, the books subset of RedPajama v1, and peS2o”

**“Model and heuristic filters are orthogonal”**

# Heuristic vs model-based filtering

FINEWEB-EDU

They generate **450k annotations** by **llama-3-instruct** for identifying educational content



# Heuristic vs model-based filtering

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:

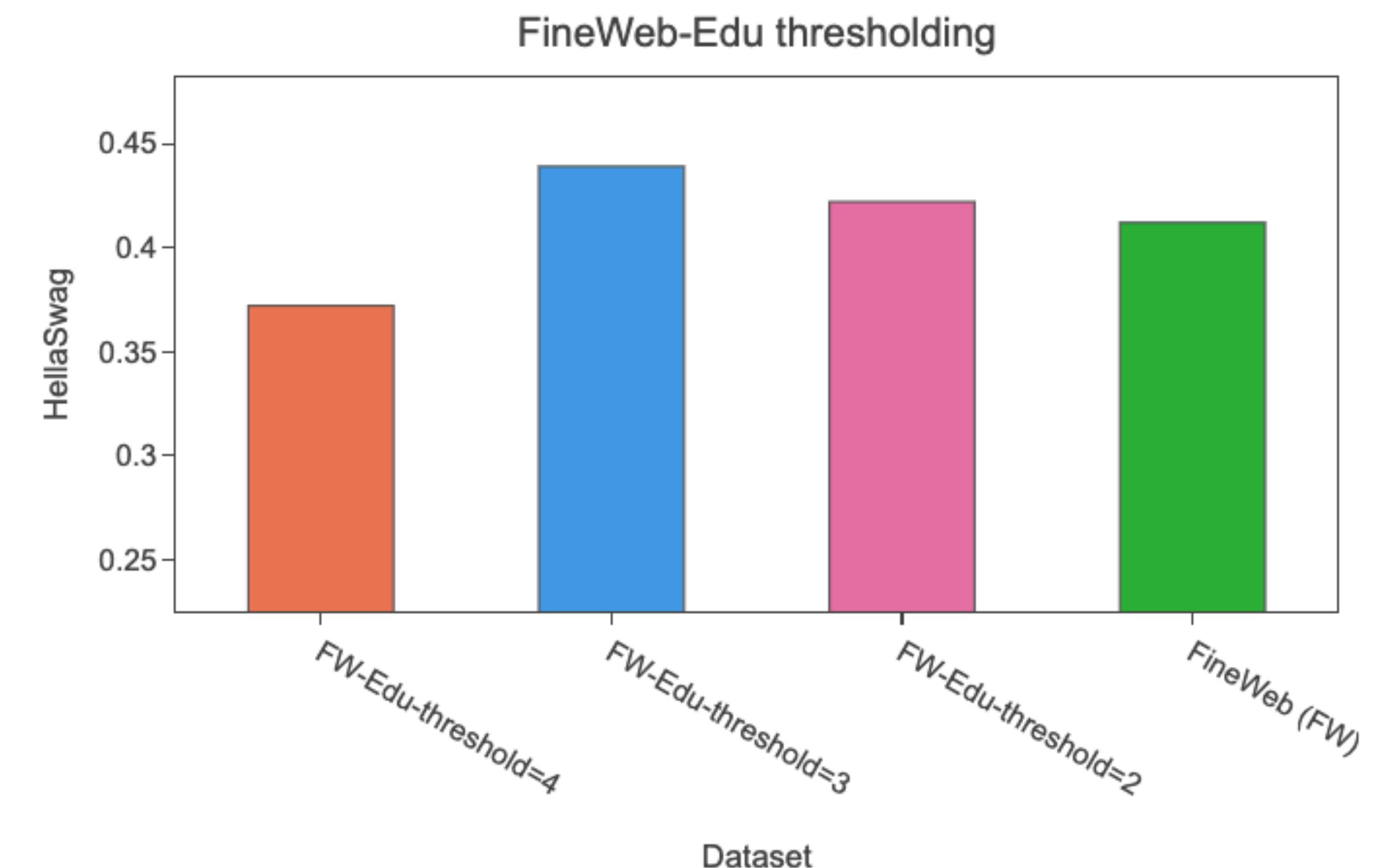
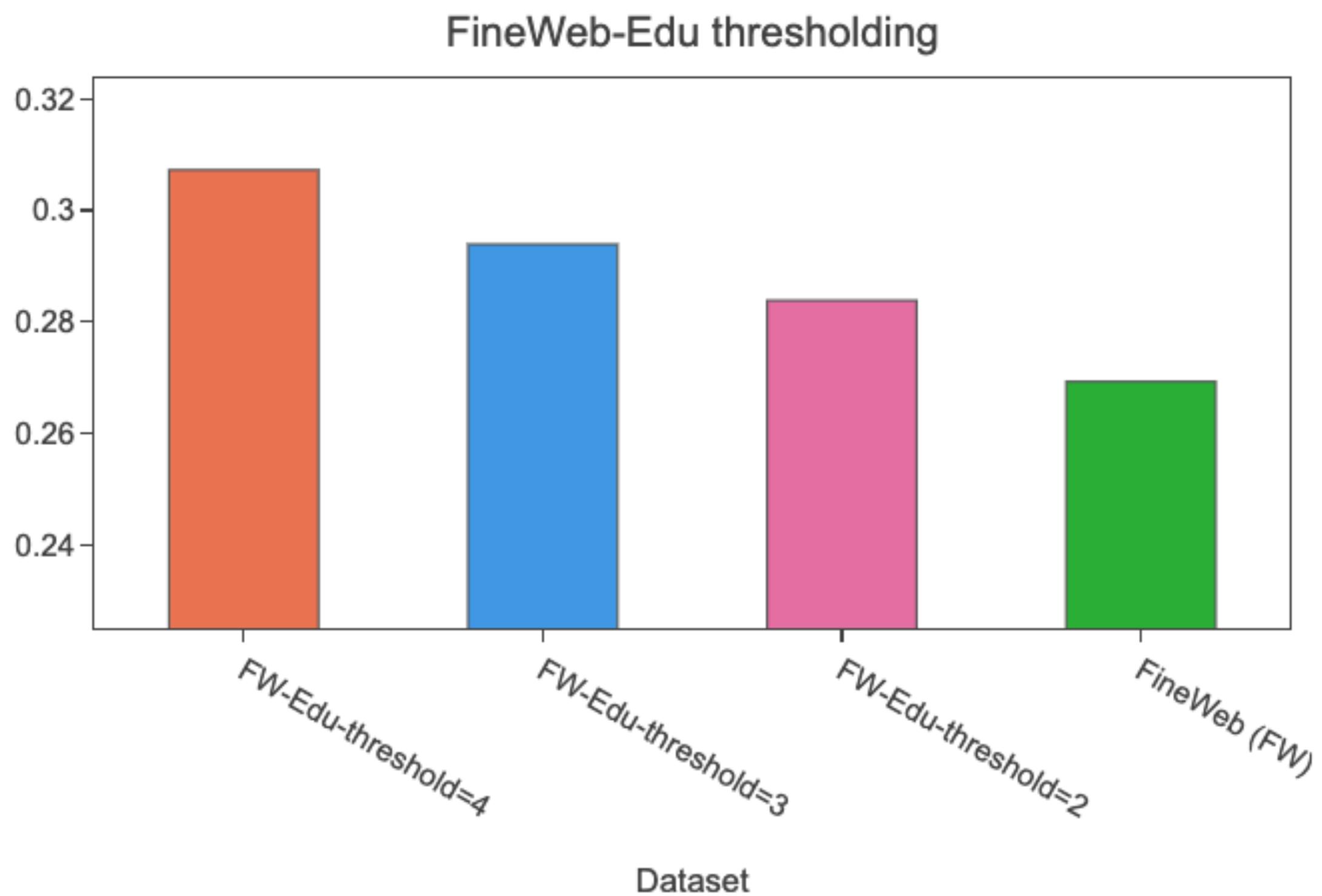
- Add 1 point if the extract provides some basic information relevant to educational topics, even if it includes some irrelevant or non-academic content like advertisements and promotional material.
- Add another point if the extract addresses certain elements pertinent to education but does not align 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>"

# Heuristic vs model-based filtering

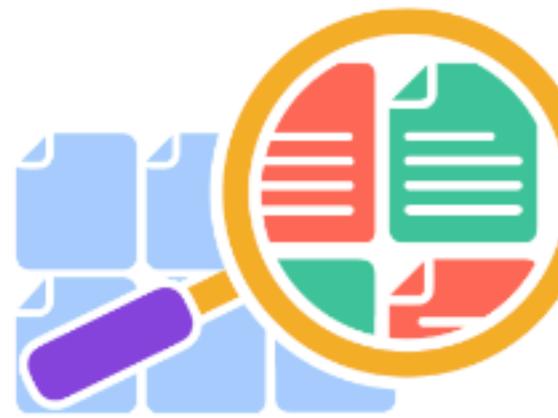


# Heuristic vs model-based filtering

When collecting *MassiveText*, we decide to use only simple heuristics for filtering out low quality text. In particular, we do not attempt to filter out low quality documents by training a classifier based on a “gold” set of text, such as English Wikipedia or pages linked from Reddit ([Radford et al., 2019](#)), as this could inadvertently bias towards a certain demographic or erase certain dialects or sociolects from representation. Filtering text for quality, while preserving coverage of dialects and avoiding biases, is an important direction for future research.

From the **Gopher** paper

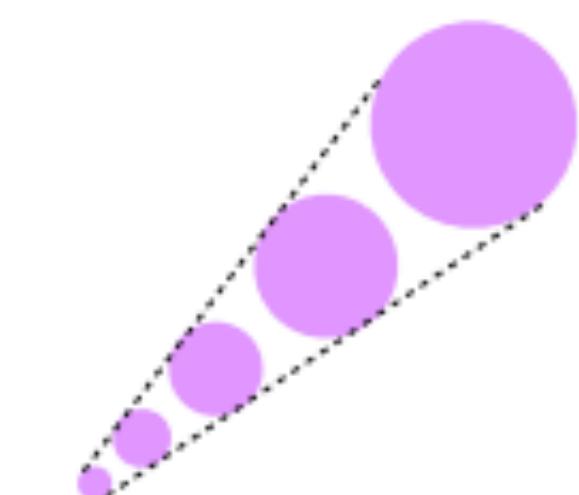
# DataComp for language models (DCLM)



## DataComp - LM

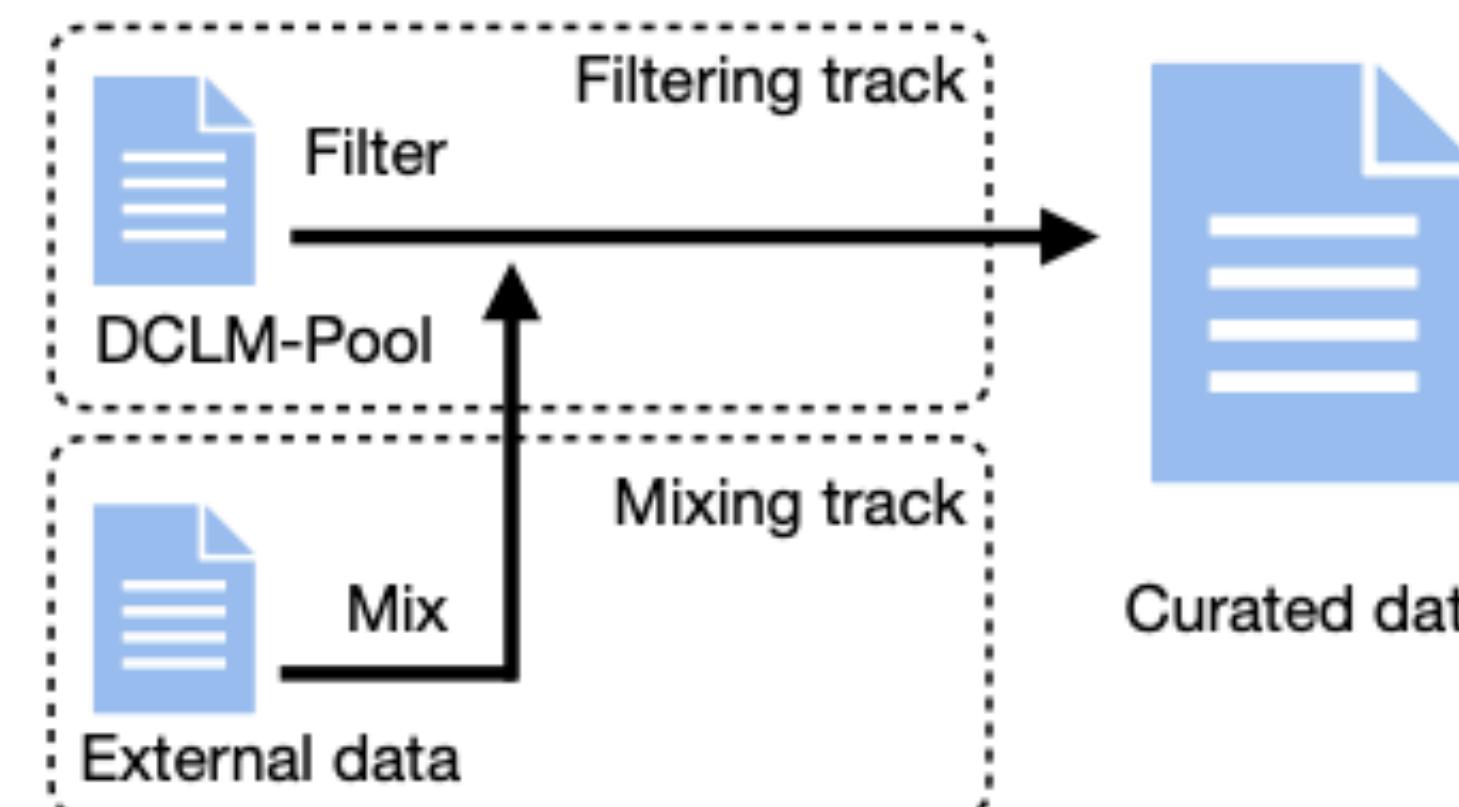
Welcome to DataComp, the machine learning benchmark where the models are fixed and the challenge is to find the best possible data!

### A. Select a scale



Pick a scale: 400M-1x,  
1B-1x, 1B-5x, 7B-1x,  
or 7B-2x

### B. Build a dataset



### C. Train a model



Train a language  
model with a fixed  
recipe

### D. Evaluate



53 downstream  
zero-shot and  
few-shot tasks

# DCLM-baseline

“We used instruction-formatted data, drawing examples from OpenHermes 2.5 [157] (OH-2.5) and high-scoring posts from the r/ExplainLikeIMFive (ELI5) subreddit”

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 [176]	27.2	14.0
AskLLM [139]	28.6	14.3
Perplexity filtering	29.0	15.0
Top-k average logits	29.2	14.7
fastText [81] OH-2.5 + ELI5	<b>30.2</b>	<b>15.4</b>

