Multilingual Model-Based Quality Filtering for LLM Pretraining

Maximilian Idahl

ellamind

elluminate

ellamind

frontier-science

- Project Home
- Experiments
- Prompt Templates
- Collections
- Criteria Library
- Models
- Compliance
- Optimization
- Eval Agent
- API Keys
- Settings

- Quick Start
- Documentation
- Support
- What's New 3

elluminate version: v1.0.5post0

maximilian@ellamind.c...

Privacy Policy Imprint

Home

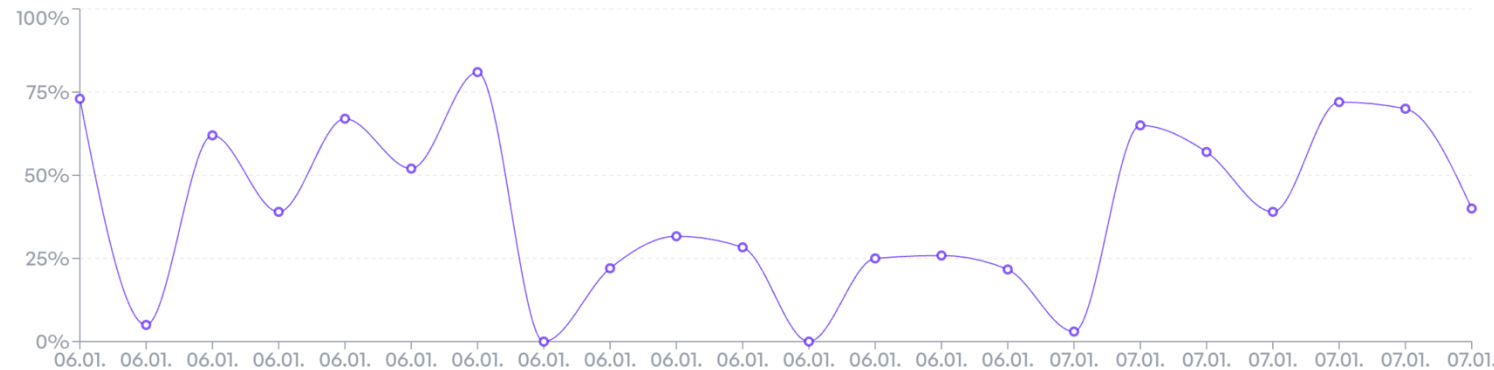
Evening maximilian@ellamind.com! Still going strong?

Monday, February 2, 2026

Settings | Visit Docs

Performance Score Timeline

Filters | View Full Timeline



Experiments

View Experiments

frontier-science-olympiad-deu v1 + frontier-science-olympiad-deu...	O4 Mini # 100 samples 15.7s avg	40.0% score
frontier-science-olympiad-deu v1 + frontier-science-olympiad-deu...	Gemini 3 Flash ... # 100 samples 94.7s avg	70.0% score
frontier-science-olympiad-deu v1 + frontier-science-olympiad-deu ...	Gemini 3 Pro pr... # 100 samples 82.7s avg	72.0% score
frontier-science-olympiad-deu v1 + frontier-science-olympiad-deu ...	Claude Sonnet ... # 100 samples 17.5s avg	39.0% score
frontier-science-olympiad-deu v1 + frontier-science-olympiad-deu ...	Claude Opus 4.5 # 100 samples 19.3s avg	57.0% score

Response Stats

View Models

LLM Config: Select model...

Timeframe: 7 Days | 30 Days | 90 Days

Total Responses
1.8K
All project responses

Average Score
45.0%
Overall performance

Avg Duration
47.3s
Response time

Average Tokens
↑ **385t** **324t** ↓ **1.6Kt**
per response

0: Intro + Motivation

Spend 80% of Your Compute Budget on Data, Not the Final Training Run

Data Quality as a High Leverage Investment

Data quality is the single most impactful factor for LLM performance

- more than model architecture
- more than training tricks

Data Quality as a High Leverage Investment

Some evidence:

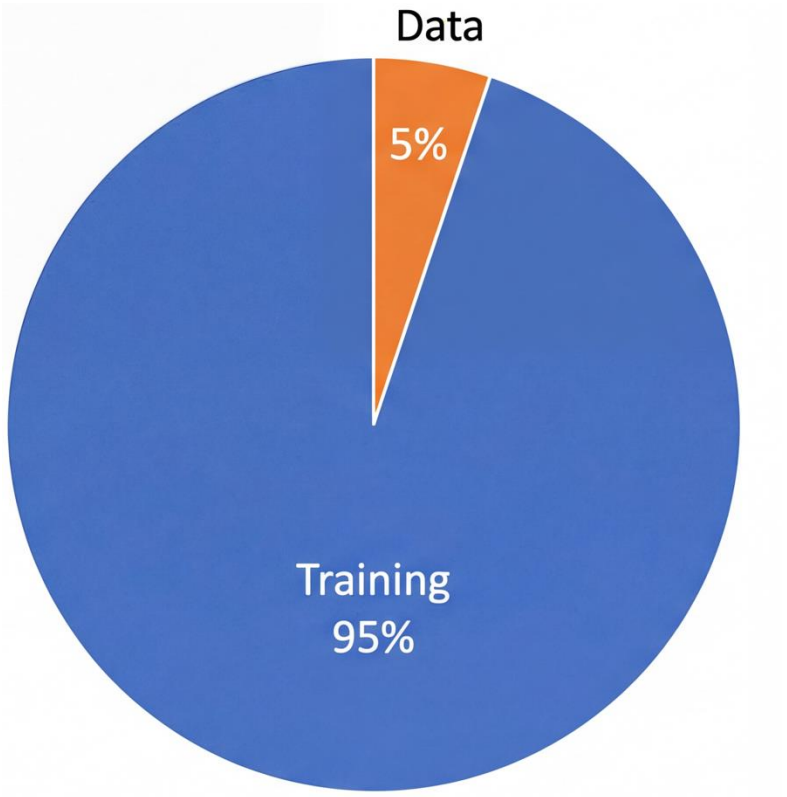
1. ”we match a 350B token baseline with only 38B tokens” [1]
2. “quality classifier made us achieve competitive results at a fraction of cost” [2]
3. “we match Qwen3-32B with 6x fewer tokens through data curation” [3]
4. “our dataset enables 7.7x faster training through document rephrasing” [4]

Implication: If you have limited compute -> invest in data

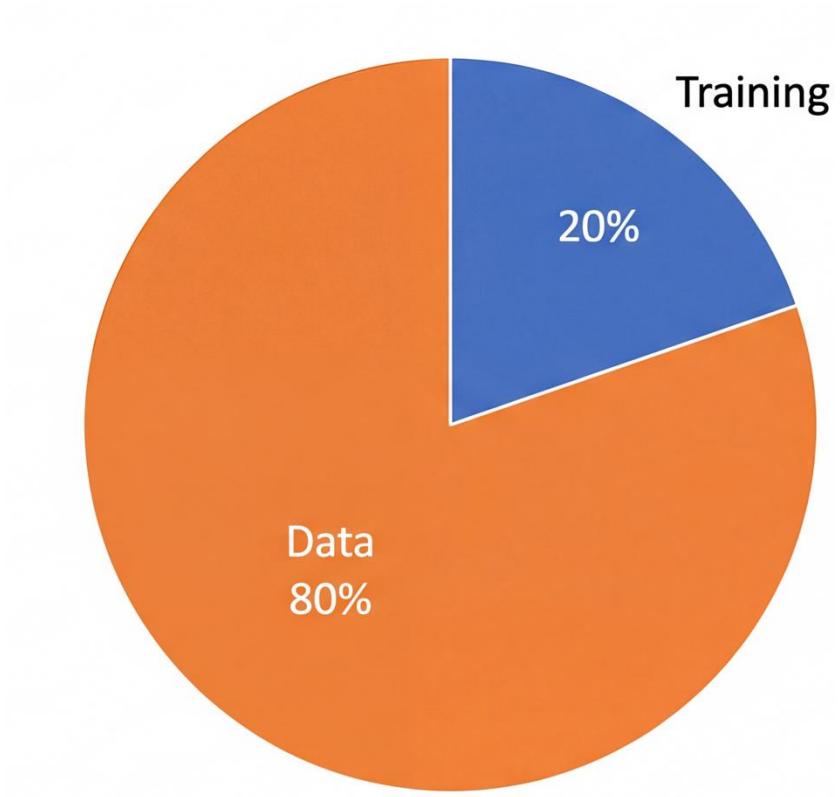
[1] 2406.17557 [2] 2406.11794 [3] 2512.13961 [4] 2508.10975

The Compute Allocation Problem

Current practice



My position



Efficiency multipliers of 6-9× from data work justify allocating 80%+ to data

Three modes of data compute

1. Selection (annotation, quality filtering, …)
2. Transformation (rephrasing, restructuring, extraction, translation, …)
3. Generation (synthetic data at scale)

Motivation for our recent work

Scaling selection compute is among the highest-return uses of GPU-hours

Data investments compound across model generations;
training runs depreciate within months

The Multilingual Gap

Many advances in model-based filtering are English-only:

- FineWeb-Edu: English only
- DCLM: English only
- Nemotron-CC: English only

Significant performance gap between English and other languages.

The Quality Definition Problem

What does "quality" even mean?

- Wikipedia like?
- Educational value?
- Downstream-task-like?
- Benchmark-like?

Can a single scalar score capture the complexity of data quality?

Roadmap for this talk

1. ~~An excerpt of the landscape of filtered pretraining datasets~~
2. Heuristic Filtering: Rules and patterns
3. Early Model-Based Filtering: Perplexity / KenLM
4. Modern Model-Based Filtering: FastText, Encoders, LLM-as-Judge
5. propella: Multi-property annotation at scale
6. Conclusion & Future Directions

2. Heuristic filtering

The Heuristic Toolkit

Common heuristic categories

Category	Examples
Length	Min/max document length, min/max line length
Character ratios	Alphabetic %, numeric %, symbol %
Repetition	N-gram repetition, line repetition
Punctuation	Terminal punctuation required, excessive punctuation
Blocklists	Bad words, spam phrases, adult content
URL-based	Domain blocklists, TLD filtering

Which Heuristics Actually Matter?

FineWeb [1] ablations revealed:

High impact:

- MinHash deduplication (≥75% 5-gram overlap)
- Language ID confidence threshold
- Terminal punctuation requirement

Moderate impact:

- Line length filters
- Repetition removal

Low/negative impact:

- Overly aggressive bad-word filters
- Some C4 rules hurt performance

Not all heuristics are created equal.
Ablate everything!

[1] 2406.17557

The Limits of Heuristics

1. Language-specific assumptions
    - "Lines must end with punctuation" fails for languages without sentence-final punctuation
    - Word length heuristics don't work for languages without spaces

2. Can't capture semantic quality
    - A grammatically correct, well-formatted spam page passes all heuristics
    - A valuable but messy forum post might get filtered

3. No nuance
    - Often binary keep/discard, no quality gradation
    - Can't do curriculum learning or targeted filtering

Heuristics Are Still Useful

Despite limitations, heuristics remain valuable for:

- First-pass noise removal (boilerplate, HTML artifacts)
- Computational efficiency (fast, no GPU needed)
- Transparency (easy to understand and audit)
- Baseline filtering before model-based refinement

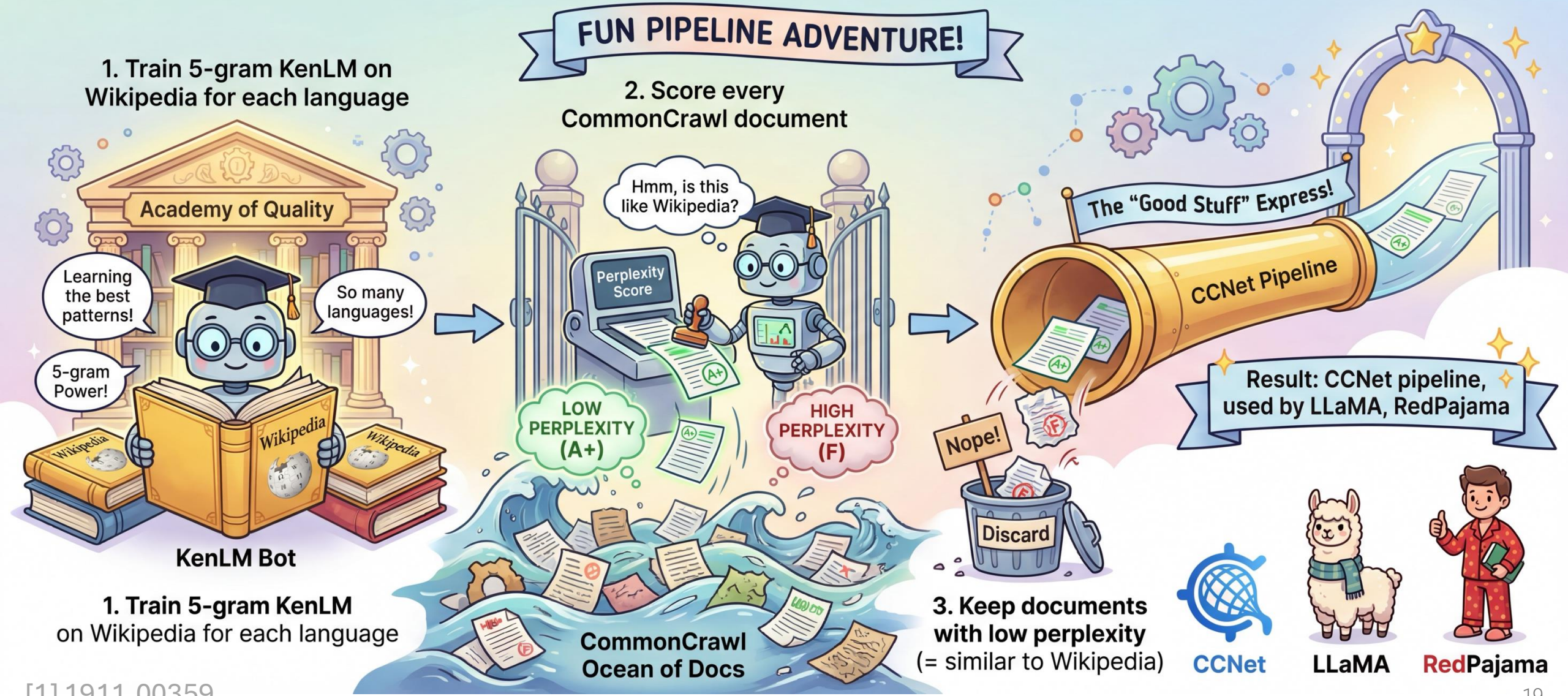
-> Use heuristics for gross noise removal, then apply model-based filtering for quality scoring.

“Don't apply heuristic filters to high-quality documents. They remove valuable content!” [1]

[1] 2412.02595

3: Early Model-based Filtering

Idea: Train a language model on “high-quality” text (Wikipedia), then score documents by perplexity.



[1] 1911.00359

Perplexity Filtering: Pros and Cons

+ Language-specific (one model per language)
+ Fast inference (n-gram models are efficient)
+ Captures fluency and coherence
+ Works for any language with Wikipedia

− Biased toward Wikipedia style/topics
− Misses valuable non-encyclopedic content (code, conversational, technical)
− Doesn't capture semantic quality (fluent nonsense scores well)
− Wikipedia size varies dramatically by language

Beyond Perplexity: The Need for Semantic Quality

Perplexity measures: How surprised is the model by this text?

But we want to know: Is this text useful for training an LLM?

These are different questions!

-> Move from fluency-based to content-based quality scoring.

4: Modern Model-based filtering

The Classifier Paradigm Shift

What is "quality"?

How do we get labels for it?

Two strategies emerged:

1. Curated positive examples: Use high-quality sources (Wikipedia, textbooks)
2. LLM-as-judge: Use large LLMs to annotate samples

FastText

A shallow neural network with bag-of-n-grams

- Very fast: CPU inference, easily parallelized
- Subword features help with rare words
- No context modeling. Pure lexical signal
- Training data matters more than architecture [2]:
  - Best: Diverse high-quality sources (not just Wikipedia)

[1] 1607.04606 [2] 2406.11794

Transformer Embedding + Classifier Approach

Upgrade: Use pretrained transformer embeddings.

1. Encode document with encoder-only transformer, such as
   - XLM-RoBERTa
   - Snowflake Arctic Embed
   - ModernBERT
2. Train lightweight classifier (MLP or linear regression) on top
3. Score documents

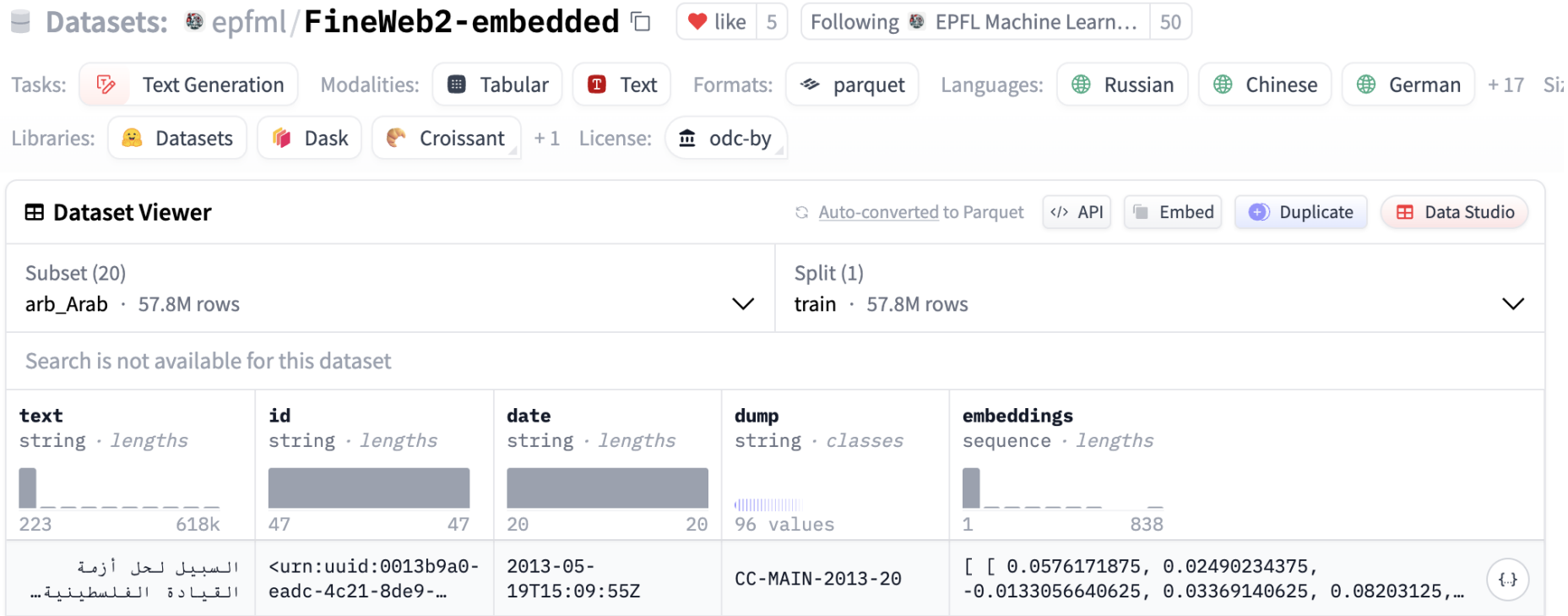
\+ Captures semantic similarity, not just lexical

\+ Cross-lingual transfer: Classifier trained on German can work for Dutch

\+ Better quality signal for nuanced content

− Requires GPUs

+ Embed once, score for cheap:



FineWeb2-embedded

Dataset summary

FineWeb2-embedded is an extension of the **FineWeb2** dataset, annotated with **document-level XLM-RoBERTa embeddings** for **20 languages**, making the dataset **useful for a variety of tasks**, including document clustering, filtering, and other multilingual research.

The FineWeb-Edu Breakthrough

The recipe:

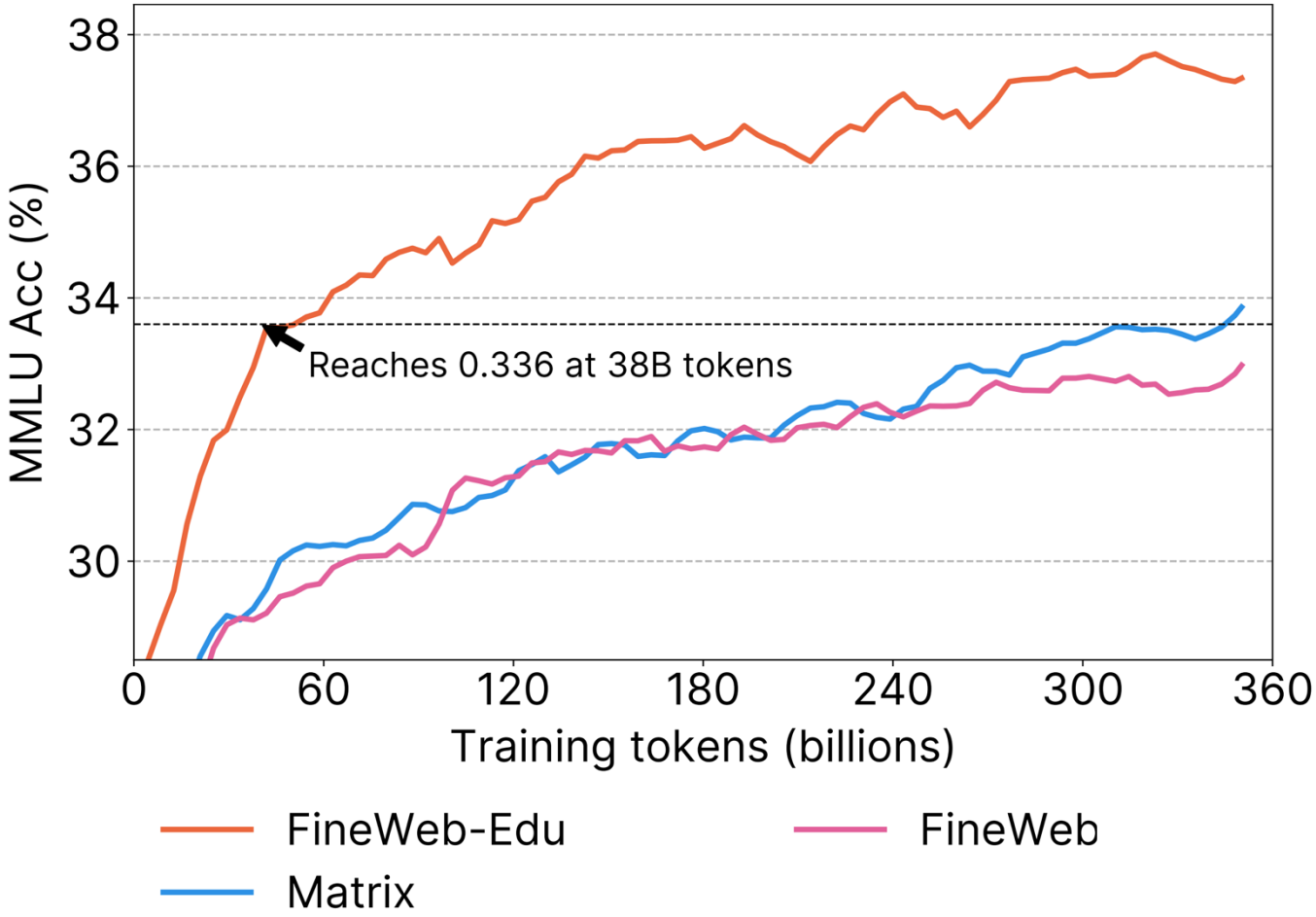
1. Collect LLM annotations: Use Llama-3-70B to score 460K documents for "educational value" (0-5 scale)
2. Distillation: Train linear regressor on transformer embeddings to predict LLM scores
3. Scale: Score all documents in FineWeb (15T)
4. Filter: score ≥ 3

-> FineWeb-edu: 1.3T tokens of "educational" content

-> Massive gains on various benchmarks

[1] 2406.17557

FineWeb-Edu Results



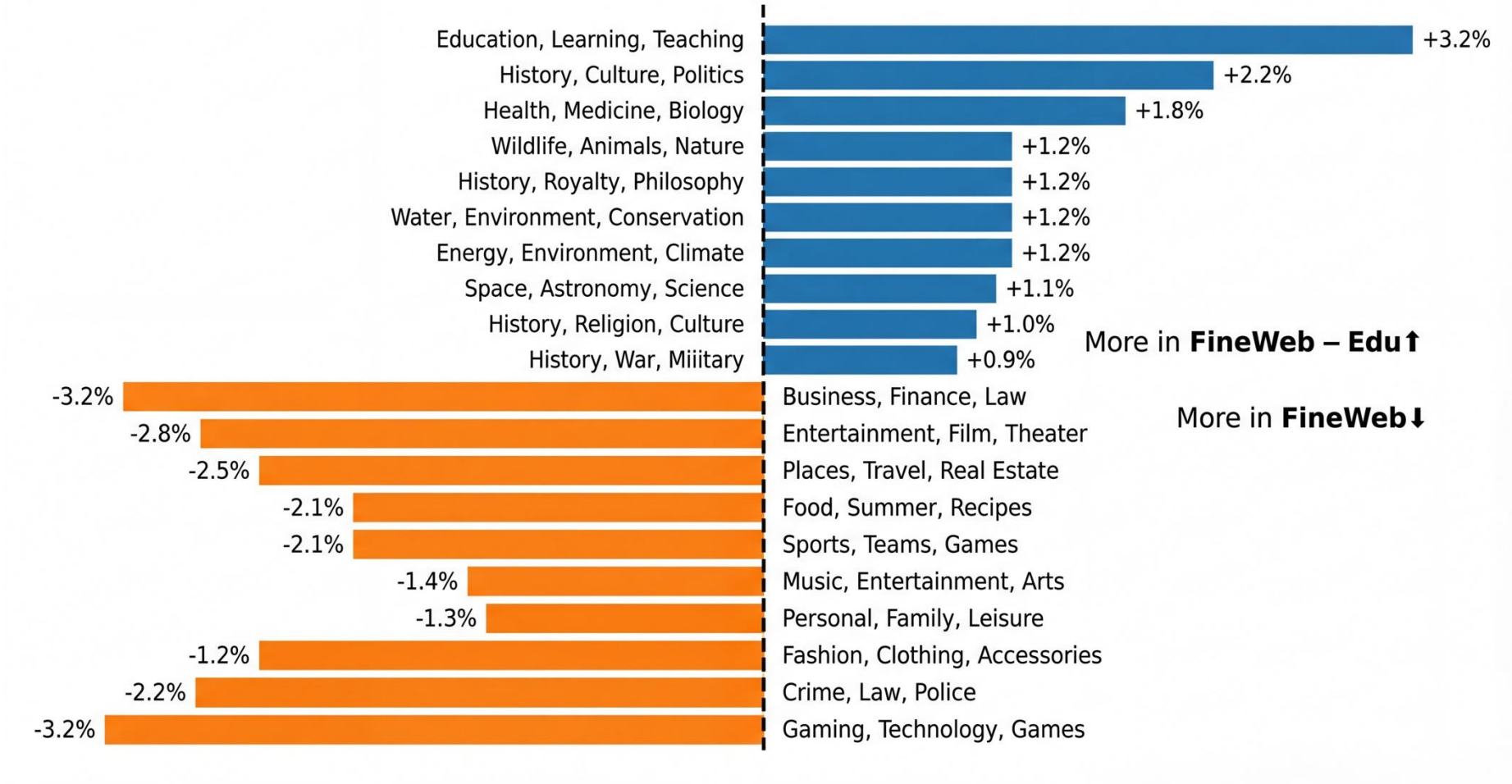
~10× token efficiency!

But: Edu filter biases towards certain topics (education, history, science) and away from others (entertainment, business, travel)

[1] 2406.17557

FineWeb-Edu Results

But: Edu filter biases towards certain topics (education, history, science) and away from others (entertainment, business, travel)



[1] 2406.17557

DataComp-LM

A simple FastText classifier, trained on carefully selected data, achieves competitive results.

The Recipe:

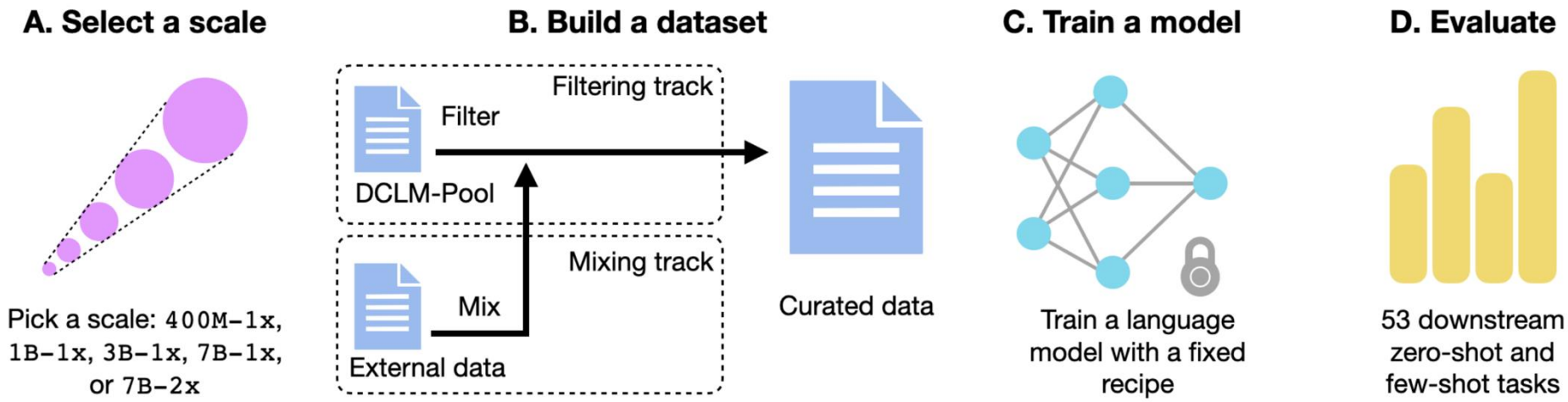
1. Labels:
    - Positive: High-quality instruction data, Wikipedia, curated sources
    - Negative: Random CommonCrawl sample
2. Train binary FastText classifier (2-gram features)
3. Score all documents in Pool, threshold to filter best N

Out of many positive/negative combinations, some beat the FineWeb-Edu-scorer performance at a fraction of compute cost.

[1] 2406.11794

DataComp-LM

Make it a competition!



[1] 2406.11794

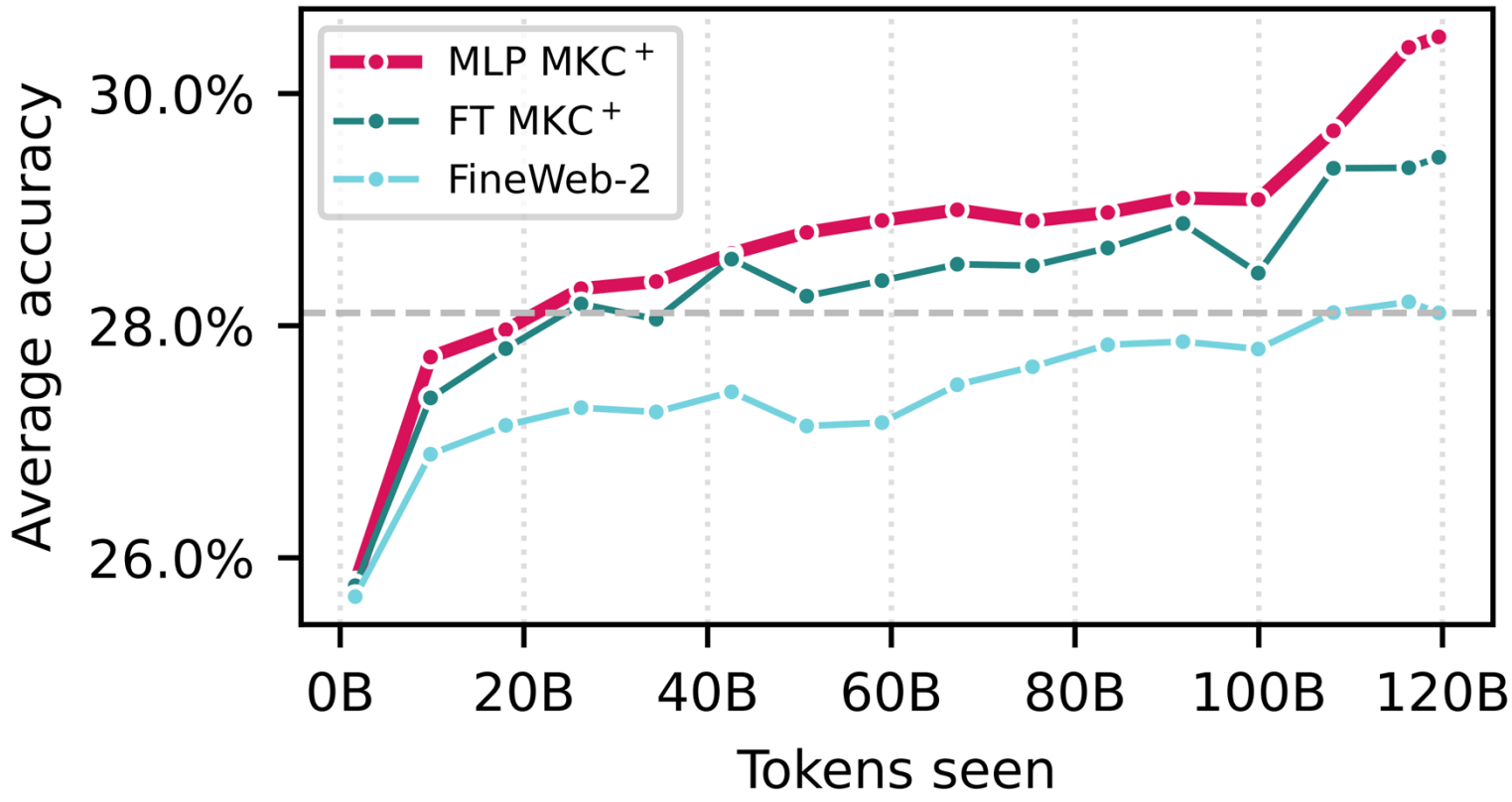
FineWeb-2-HQ

Extending DCLM to multilingual:

1. Labels:
    - Positive: Aya Collection + Dataset, MMMLU, OpenAssistant-2, Include-Base-44)
    - Negative: Random FineWeb-2 sample
2. Train separate FastText/MLP scorers per language

[1] 2502.10361

FineWeb-2-HQ



The training data mixture matters: diverse sources outperform single-source.

[1] 2502.10361

Multilingual LLM-as-Judge

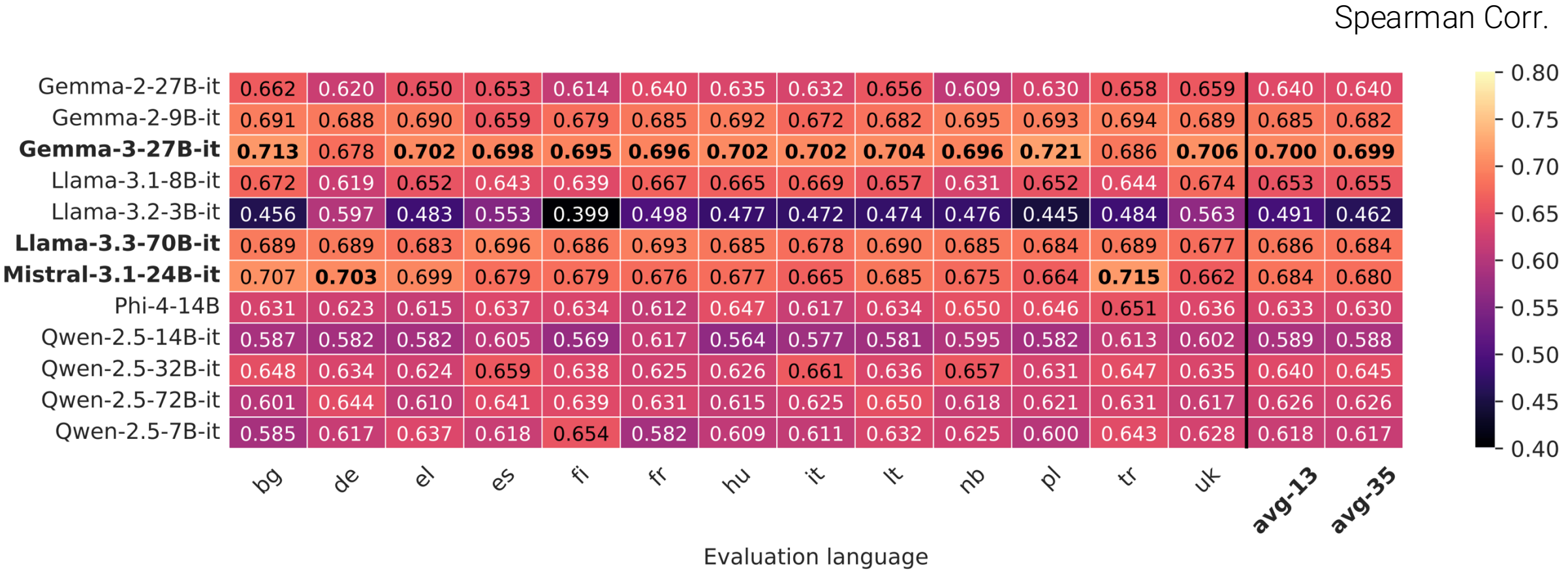
JQL: Judging Quality across languages

- Human edu-score labels: 511 English documents annotated by 15 humans

  -> Machine-translate to 35 languages
- Evaluate various LLM-judges
- Distillation: Train lightweight regressors on Snowflake Arctic Embed

[1] 2505.22232

Multilingual LLM-Judge Performance



	bg	de	el	es	fi	fr	hu	it	lt	nb	pl	tr	uk	**avg-13**	**avg-35**
Gemma-2-27B-it	0.662	0.620	0.650	0.653	0.614	0.640	0.635	0.632	0.656	0.609	0.630	0.658	0.659	0.640	0.640
Gemma-2-9B-it	0.691	0.688	0.690	0.659	0.679	0.685	0.692	0.672	0.682	0.695	0.693	0.694	0.689	0.685	0.682
Gemma-3-27B-it	**0.713**	0.678	**0.702**	**0.698**	**0.695**	**0.696**	**0.702**	**0.702**	**0.704**	**0.696**	**0.721**	0.686	**0.706**	**0.700**	**0.699**
Llama-3.1-8B-it	0.672	0.619	0.652	0.643	0.639	0.667	0.665	0.669	0.657	0.631	0.652	0.644	0.674	0.653	0.655
Llama-3.2-3B-it	0.456	0.597	0.483	0.553	0.399	0.498	0.477	0.472	0.474	0.476	0.445	0.484	0.563	0.491	0.462
Llama-3.3-70B-it	0.689	0.689	0.683	0.696	0.686	0.693	0.685	0.678	0.690	0.685	0.684	0.689	0.677	0.686	0.684
Mistral-3.1-24B-it	0.707	**0.703**	0.699	0.679	0.679	0.676	0.677	0.665	0.685	0.675	0.664	**0.715**	0.662	0.684	0.680
Phi-4-14B	0.631	0.623	0.615	0.637	0.634	0.612	0.647	0.617	0.634	0.650	0.646	0.651	0.636	0.633	0.630
Qwen-2.5-14B-it	0.587	0.582	0.582	0.605	0.569	0.617	0.564	0.577	0.581	0.595	0.582	0.613	0.602	0.589	0.588
Qwen-2.5-32B-it	0.648	0.634	0.624	0.659	0.638	0.625	0.626	0.661	0.636	0.657	0.631	0.647	0.635	0.640	0.645
Qwen-2.5-72B-it	0.601	0.644	0.610	0.641	0.639	0.631	0.615	0.625	0.650	0.618	0.621	0.631	0.617	0.626	0.626
Qwen-2.5-7B-it	0.585	0.617	0.637	0.618	0.654	0.582	0.609	0.611	0.632	0.625	0.600	0.643	0.628	0.618	0.617

Strong open LLMs can judge educational value for many EU languages

[1] 2505.22232

FinePDFs-Edu - Domain-Specific Filtering

PDF challenges:

- OCR noise
- Layout complexity
- Mixed content (text, tables, figures)

- Teacher selection: Qwen3-235B for labeling (best MSE vs. Sonnet-4)
- Student: mmBERT-based classifier
  -> One model per language

HuggingFaceFW/finepdfs_edu_classifier_nno_Latn
0.3B • Updated Oct 6, 2025 • 5

HuggingFaceFW/finepdfs_edu_classifier_tam_Taml
0.3B • Updated Oct 6, 2025 • 4

HuggingFaceFW/finepdfs_edu_classifier_afr_Latn
0.3B • Updated Oct 6, 2025 • 6

HuggingFaceFW/finepdfs_edu_classifier_bcc_Arab
0.3B • Updated Oct 6, 2025 • 5

HuggingFaceFW/finepdfs_edu_classifier_wuu_Hani
0.3B • Updated Oct 6, 2025 • 4

HuggingFaceFW/finepdfs_edu_classifier_hye_Armn
0.3B • Updated Oct 6, 2025 • 4

HuggingFaceFW/finepdfs_edu_classifier_cym_Latn
0.3B • Updated Oct 6, 2025 • 5

The Limitation of Single-Score Filtering

Since FineWeb-Edu in 2024, the community has largely relied on single scores to filter training data.

One scalar from tiny encoder or FastText models.

Is that a problem?

The Limitation of Single-Score Filtering

Problems with single-score:

- Educational value ≠ all quality dimensions
- Not flexible (e.g., ”Now I want reasoning-heavy content")
- Want to compose various filters

A single scalar score cannot capture the complexity of data quality.

Register Matters

Register annotation method:

- XLM-RoBERTa-Large fine-tuned on multilingual, register annotated data [2]
- Multi-label classification
- Hierarchical scheme: 9 main registers → 25 subregisters

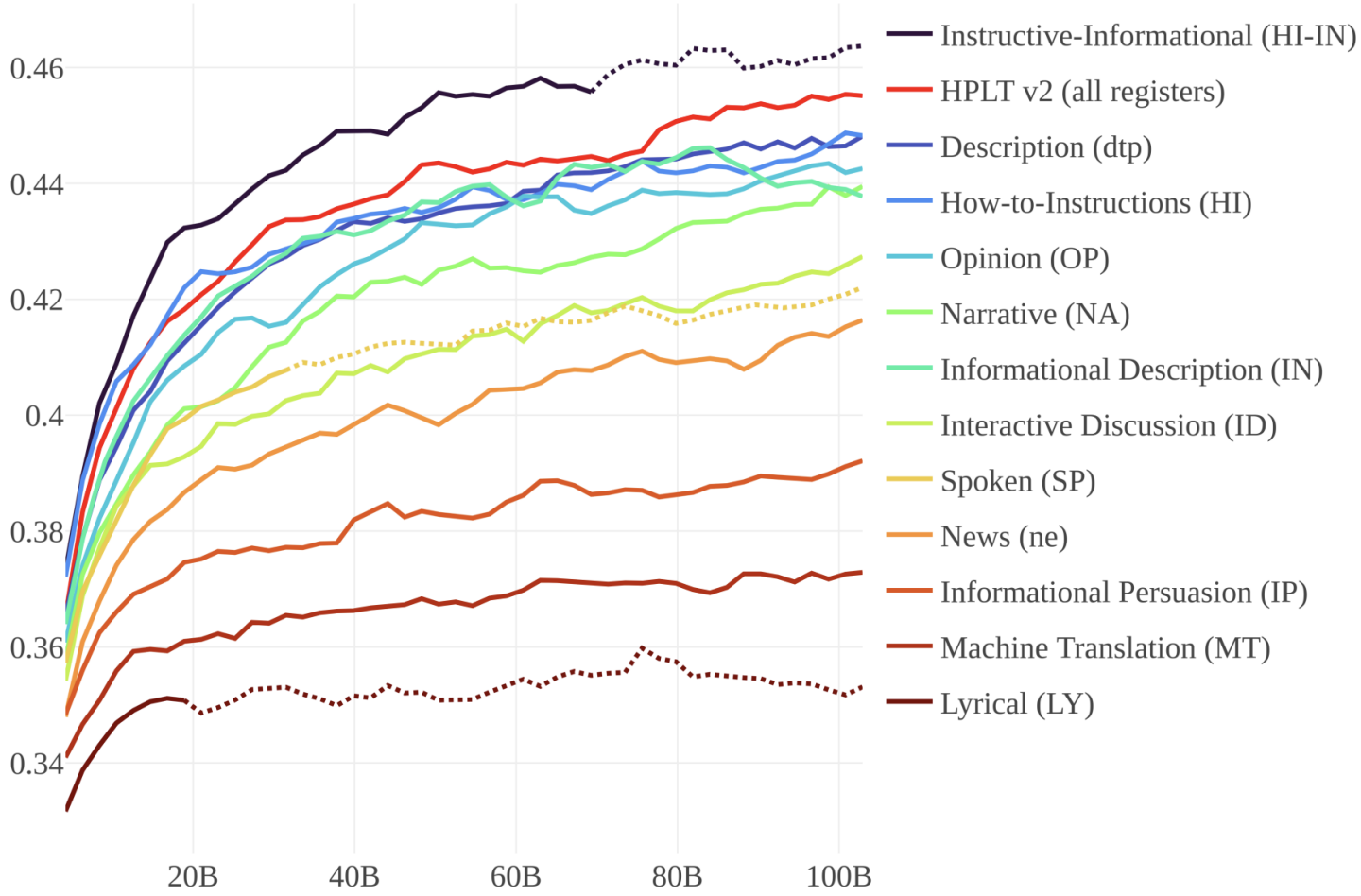
  Examples:

  Narrative (main) → News (subregister)

  Informational Description (main) → Description (subregister)

[1] 2504.01542 [2] 2406.19892

Register Matters a Lot

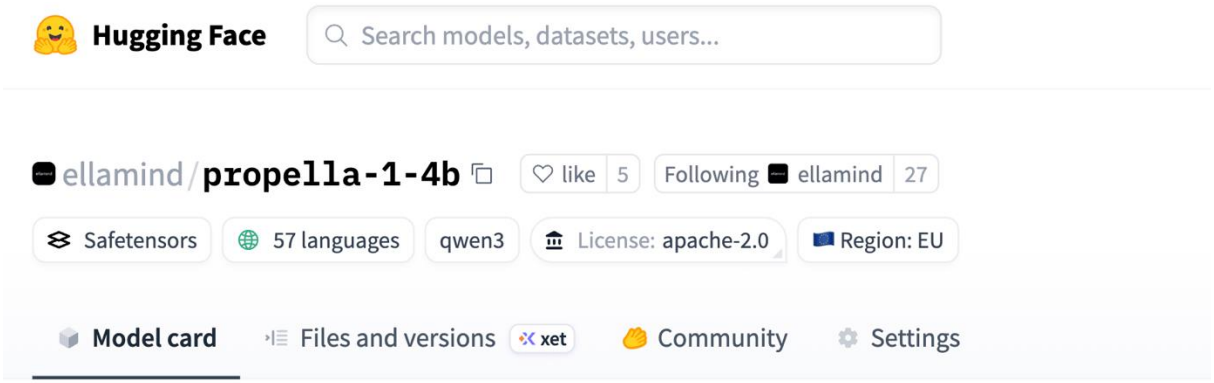


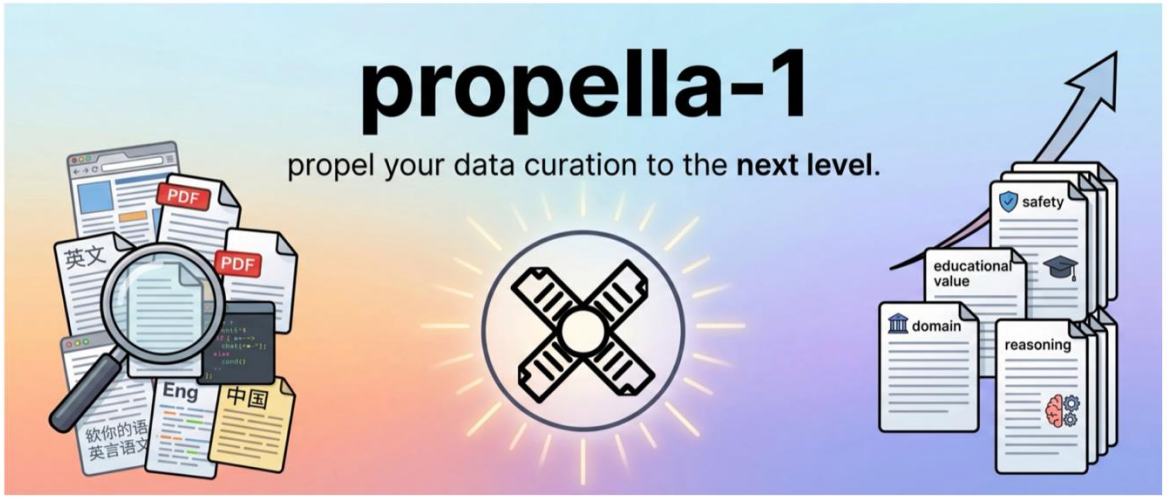
The type of text (register) has a substantial effect on model performance

[1] 2504.01542 [2] 2406.19892

5: Multi-property Annotation at Scale

How we built propella-1





Motivation

MultiSynt: an open multilingual synthetic dataset for LLM pre-training

x

Urgent need:

- Select good seed documents for synth data generation

  -> garbage in, garbage out

Goals

- Score documents on various dimensions (beyond edu)
- Support all kinds of text
- Support many languages

-> Use small decoder models: strong performance, long context

German-edu-scorer [1]

Teacher: Command R plus

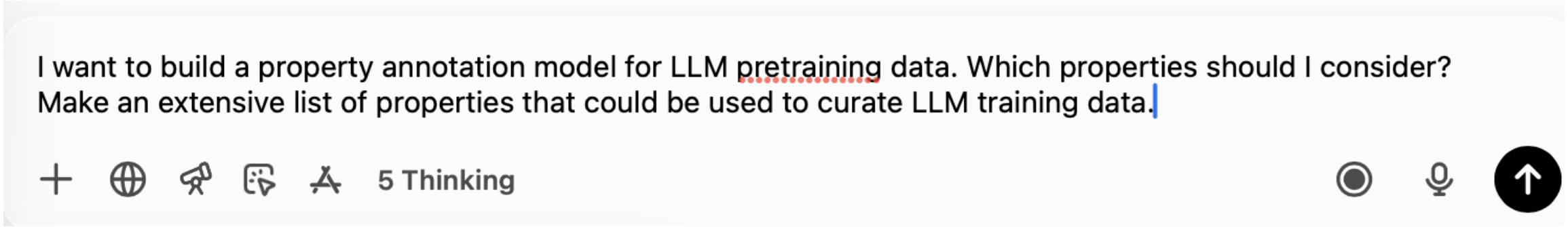
Students

- Bert (512): 85%
- T5 (512): 88%
- Qwen2-1.5b (32k): 95%

[1] hf.co/pL-Community/GermanEduScorer-Qwen2-1.5b

Which Properties to Annotate?

How it started:



Which Properties to Annotate?

Two weeks of iterating:

- 17 initial properties
- Later 18, added
  “one sentence description”
- ~14k tokens long rubric

Detailed Property Descriptions & Annotation Guidelines

Core Content Properties

1. Content Integrity

What we're measuring: Completeness and technical quality of the content itself, regardless of navigation ratio.

Values & Criteria:

`complete` - Full, intact content as intended

- Content appears complete with proper beginning, middle, and end
- All essential elements present (introduction, body, conclusion where appropriate)
- No obvious truncation or missing sections
- Example: Complete articles, full tutorials, intact documents

`mostly_complete` - Minor elements missing but core content intact

...

[1] hf.co/ellamind/propella-1-4b/blob/main/property_descriptions.md

propella-1 Properties

Category	Properties
Core Content	Content Integrity, Content Ratio, Content Length
Classification	One-Sentence Description, Content Type, Business Sector, Technical Content
Quality & Value	Content Quality, Information Density, Educational Value, Reasoning Indicators
Audience & Purpose	Audience Level, Commercial Bias, Time-Sensitivity
Safety & Compliance	Content Safety, PII Presence
Geographic	Regional Relevance, Country Relevance

Multi-select, ordinal, and free-text properties

A Diverse Data Sample

lang	percent
eng_Latn	35.08
spa_Latn	3.98
ita_Latn	3.97
fra_Latn	3.95
deu_Latn	3.86
pol_Latn	3.81
code	2.82
math	2.77
sft	2.41
ukr_Cyrl	0.95
…	…

source	percent
hplt3_unfiltered	39.59
fineweb	16.01
fineweb2	13.23
finepdfs	8.09
fineweb2_removed	6.28
fineweb_edu_dedup	3.92
thestack	2.04
finemath	2.00
openhermes	2.00
finewiki	1.35
…	…

Training Data

- Obtained from various frontier models (Dec. 2025)
- Problem: Strict content-filters

  -> labeled some very bad documents by hand

Training Setup

- Base: Qwen-3 architecture (0.6B, 1.7B, 4B variants)
- Target: Annotations as a JSON-object ← no whitespace, saves output tokens later
- Training:
  - 64K context length (recommend truncating at 50K chars)
  - fp8 precision
  - 4x H100 (couple of hours)

[1] https://axolotl.ai

Inference

Sending request via OpenAI SDK

```python
from openai import OpenAI
from propella import (
    create_messages,
    AnnotationResponse,
    get_annotation_response_schema,
)

document = "Hi, its me Max."

client = OpenAI(base_url="http://localhost:8000/v1", api_key="EMPTY")

response = client.chat.completions.create(
    model="ellamind/propella-1-4b",
    messages=create_messages(document),
    response_format={
        "type": "json_schema",
        "json_schema": {
            "name": "AnnotationResponse",
            "schema": get_annotation_response_schema(flatten=True, compact_whites
            "strict": True,
        }
    },
)
response_content = response.choices[0].message.content
```

Output

▼ Result


```
{
    "content_integrity": "complete",
    "content_ratio": "complete_content",
    "content_length": "minimal",
    "one_sentence_description": "A short personal greeting introducing someone nai
    "content_type": [
        "conversational"
    ],
    "business_sector": [
        "general_interest"
    ],
    "technical_content": [
        "non_technical"
    ],
    "information_density": "dense",
    "content_quality": "good",
    "audience_level": "general",
    "commercial_bias": "none",
    "time_sensitivity": "evergreen",
    "content_safety": "safe",
    "educational_value": "none",
    "reasoning_indicators": "none",
```

Evaluation Setup

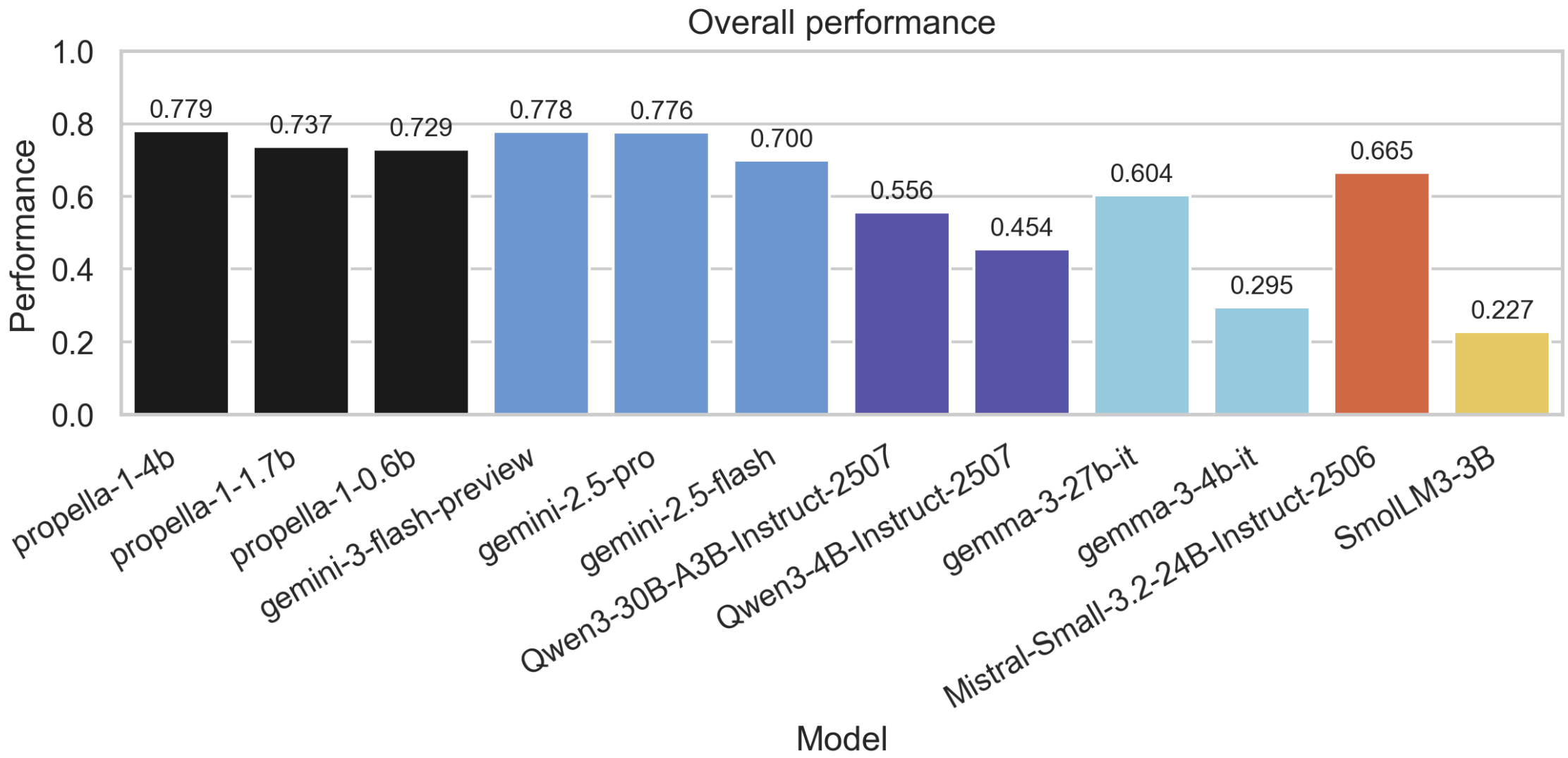
Evaluation Against Frontier Models

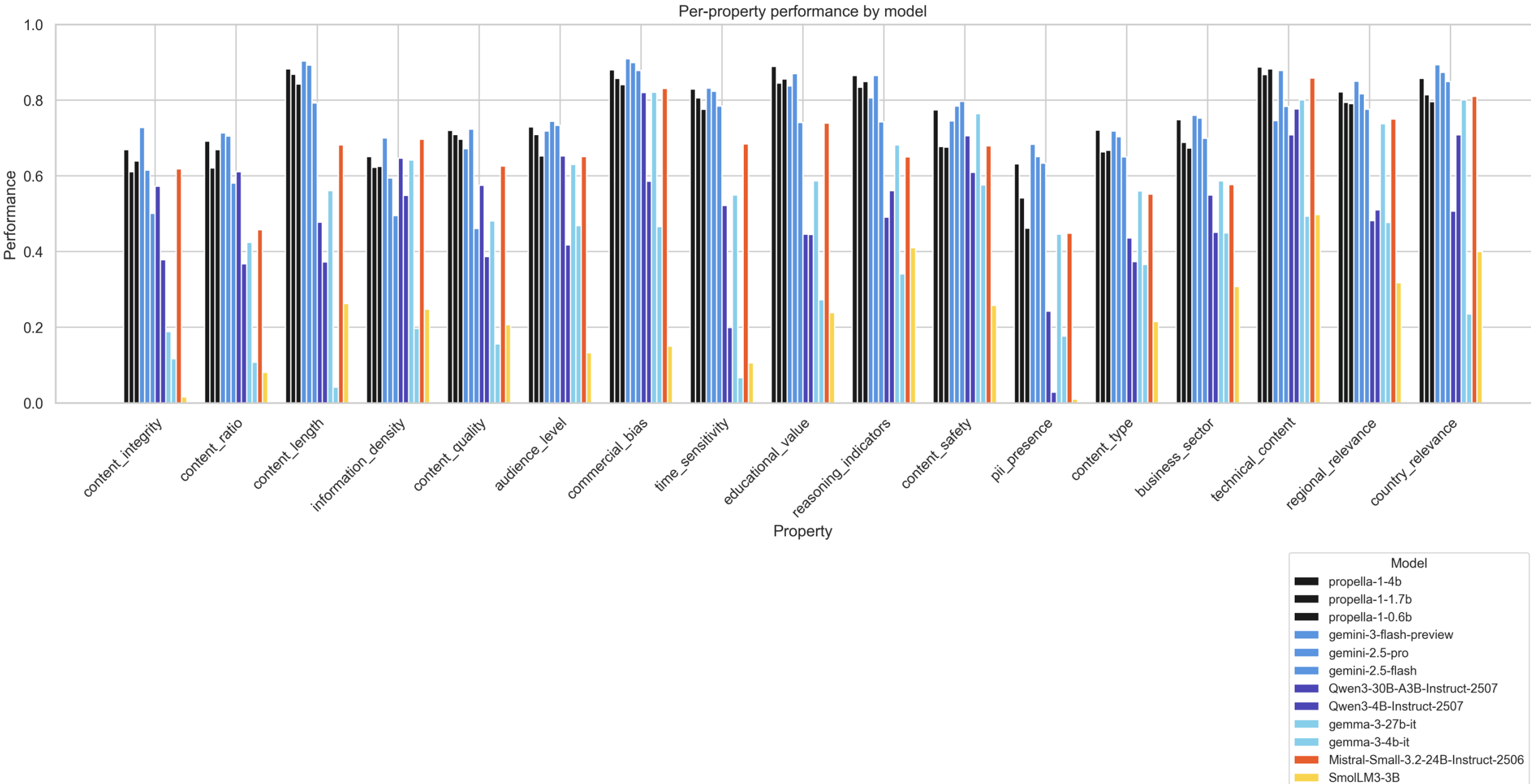
- Ground truth: Gemini-3-Pro annotations (reasoning_effort: high) for 3K documents.

- Metrics by property type:
  - Ordinal (11 properties): Quadratic Weighted Kappa
  - Binary (1 property): F1
  - Multi-select (5 properties): IoU/Jaccard

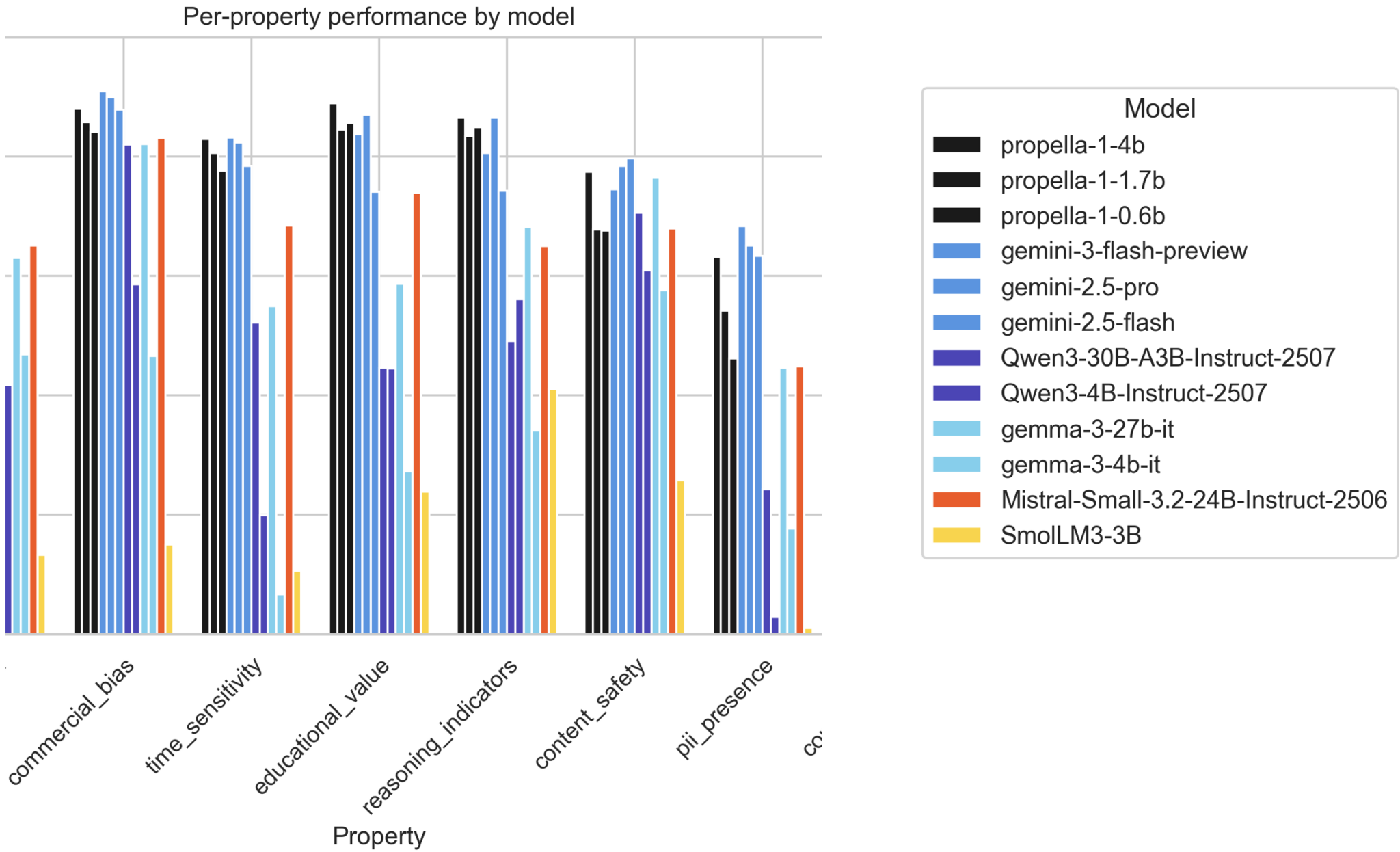
- Overall Score
  - A weighted average of the primary metric for each property type:

    overall = (11/17 × avg_QWK) + (1/17 × avg_F1) + (5/17 × avg_IoU)


Overall performance
Performance
1.0
0.8
0.6
0.4
0.2
0.0
0.779
0.737
0.729
0.778
0.776
0.700
0.556
0.454
0.604
0.295
0.665
0.227
propella-1-4b
propella-1-1.7b
propella-1-0.6b
gemini-3-flash-preview
gemini-2.5-pro
gemini-2.5-flash
Qwen3-30B-A3B-Instruct-2507
Qwen3-4B-Instruct-2507
gemma-3-27b-it
gemma-3-4b-it
Mistral-Small-3.2-24B-Instruct-2506
SmolLM3-3B
Model



[1] hf.co/ellamind/propella-1-4b/blob/main/res/per_property_scores_by_model.png



[1] hf.co/ellamind/propella-1-4b/blob/main/res/per_property_scores_by_model.png

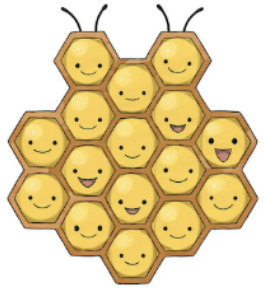
Going Fast

Key enablers: SGL + llguidance

Model	GPU	Docs/s	hours-per-1M docs	Prompt TPS	Output TPS	Total TPS
propella-1-4b	A100 80GB	10.3	27.0	19.1k	1.5k	20.5k
propella-1-4b	H100 96GB	22.4	12.4	41.6k	3.2k	44.8k
propella-1-4b (fp8)	H100 96GB	27.0	10.3	50.1k	3.9k	54.0k
propella-1-1.7b	A100 80GB	17.8	15.6	33.0k	2.6k	35.6k
propella-1-1.7b	H100 96GB	35.8	7.8	66.5k	5.2k	71.8k
propella-1-1.7b (fp8)	H100 96GB	39.1	7.1	72.7k	5.7k	78.4k
propella-1-0.6b	H100 96GB	39.9	7.0	74.2k	5.7k	79.9k
propella-1-0.6b	A100 80GB	21.5	12.9	40.0k	3.1k	43.1k

[1] github.com/sgl-project/sglang [2] guidance-ai.github.io/llguidance/llg-go-brrr

Scaling up



inference-hive

Run offline LLM inference at scale using SLURM

inference-hive is a toolkit to run distributed LLM inference on SLURM clusters. Configure a few cluster, inference server and data settings, and scale your inference workload across thousands of GPUs.

[1] github.com/ellamind/inference-hive

Scaling up

Max Idahl
@maxidahl

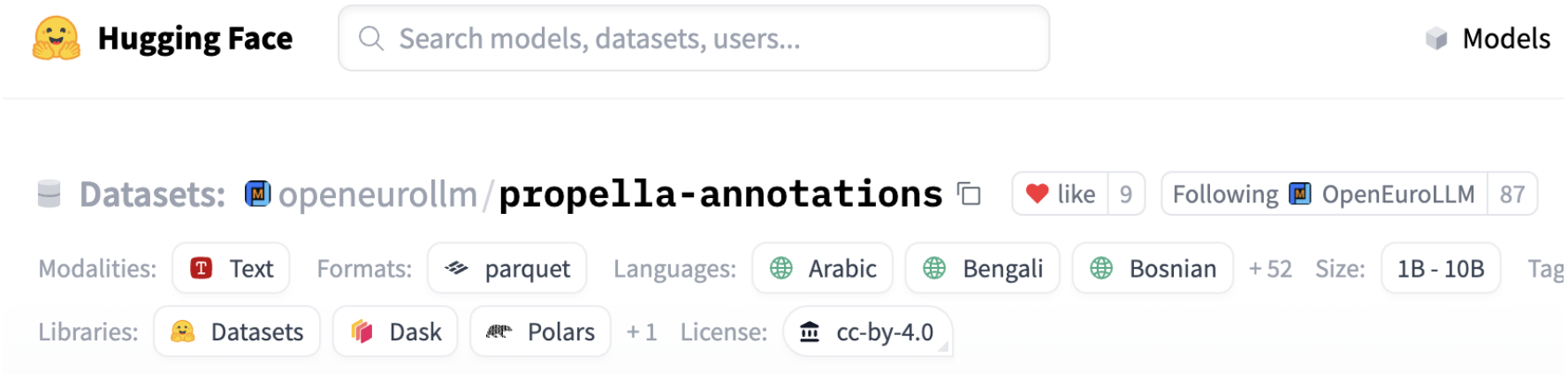
Last weekend we ran **propella-1-4b** on **3936 GPUs**, and annotated the **entire German FineWeb-2 in ~3.5h**.

In total, we annotated over 1.7B documents. Lots more incoming.

So far we have:
- 1.4B for FineWeb-2 (DE, ES, FR, IT, SV, FI)
- 145M for FinePDFs (10+ EU languages)
- 156M for Nemotron-CC HQ.

What dataset should we annotate next?

propella-annotations



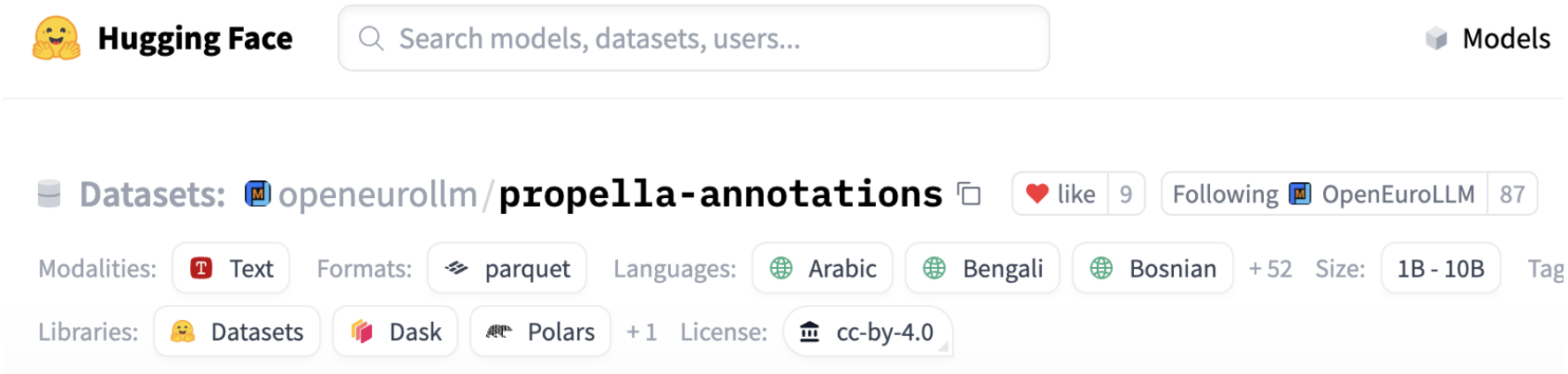
Lots of annotations available, including:

FineWeb-2 FinePDFs FineWiki HPLT3

Nemotron-CC German-commons SYNTH

[1] hf.co/datasets/openeurollm/propella-annotations

propella-annotations



Lots of annotations available, including:

FineWeb-2 FinePDFs FineWiki HPLT3

Nemotron-CC German-commons SYNTH

[1] hf.co/datasets/openeurollm/propella-annotations

The Synthetic Pretraining Future

Filter-then-augment

- Filtering improves training token efficiency but reduces dataset size

  -> Synthetic data generation is the solution

1. **Filter aggressively**: Use quality classifiers to identify high-utility tokens, accepting that this drastically reduces dataset size
2. **Augment systematically**: Use transformation and generation (rephrasing, Q&A pair generation, knowledge extraction) to expand the filtered corpus back to frontier scale
3. **Validate continuously**: Use ablation experiments to verify that the resulting tokens maintain the efficiency multiplier

Recommended Reading: Synthetic Pretraining – Blog from Alexander Doria [1]

[1] vintagedata.org/blog/posts/synthetic-pretraining