Introducing GneissWeb - a state-of-the-art LLM pre-training dataset

Shahrokh Daijavad
Research Scientist
IBM Almaden Research Center

Shahrokh Daijavad

March 6, 2025

Agenda

- At a Glance
- What is GneissWeb?
- An Overview of GneissWeb Recipe
- Summary of Results
- Data Prep Kit role in GneissWeb
- Recipe Notebook
- Summary

At a Glance

- At IBM, responsible AI implies transparency in training data: Introducing GneissWeb (pronounced “niceWeb”), a state-of-the-art LLM pre-training dataset with ~10 trillion tokens derived from FineWeb, with open recipes, results, and tools for reproduction!

- For the announcement on Feb. 21st and details, please refer to: https://research.ibm.com/blog/gneissweb-for-granite-training

- In this session, we will go over how GneissWeb was created and discuss the tools and techniques used. We will provide a Jupyter notebook recipe that you can try at your leisure.

What is GneissWeb?

- A dataset built on top of FineWeb
  - Hugging Face had introduced FineWeb V1.1.0, a large-scale dataset for LLM pre-training, consisting of 15 trillion tokens (which takes up 44TB of disk space).
  - FineWeb is derived from 96 Common Crawl snapshots, focusing on English text by applying a series of processing steps, including language classification, deduplication, and heuristic rule-based quality filters.
  - Models trained on FineWeb are shown to outperform those trained on other publicly available datasets, such as C4, RefinedWeb, Dolma, RedPajamav2, SlimPajama, and The Pile.
  - While we focused on FineWeb V1.1.0 to prepare GneissWeb, our recipe can also be applied to FineWeb V1.2, which was recently released.

What is GneissWeb? (continued)

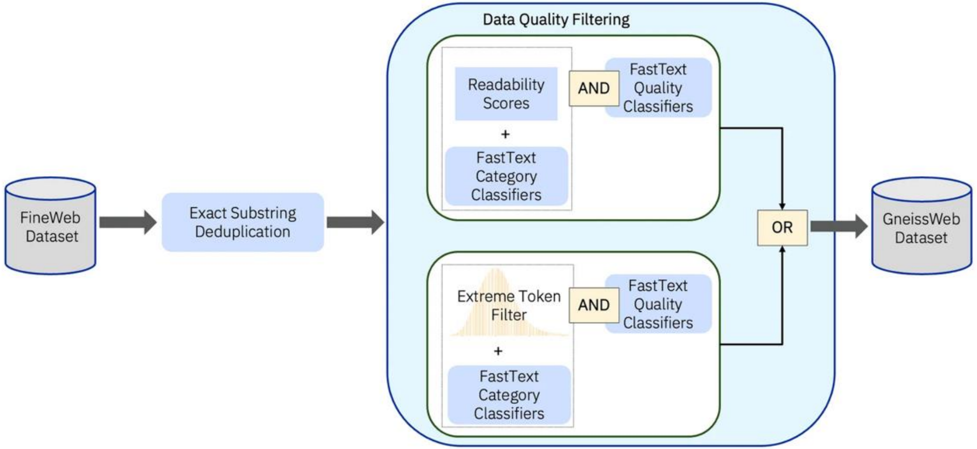
- We started with the goal of distilling roughly 10 trillion high-quality tokens from FineWeb V1.1.0, so that we get a sufficiently large number of quality tokens suitable for Stage-1 pre-training

- Unlike the FineWeb.Edu families, which rely on a single quality annotator and perform aggressive filtering, we developed a multi-faceted ensemble of quality annotators to enable fine-grained quality filtering

- This allowed us to achieve a finer trade-off between the quality and quantity of the tokens retained

An Overview of GneissWeb Recipe

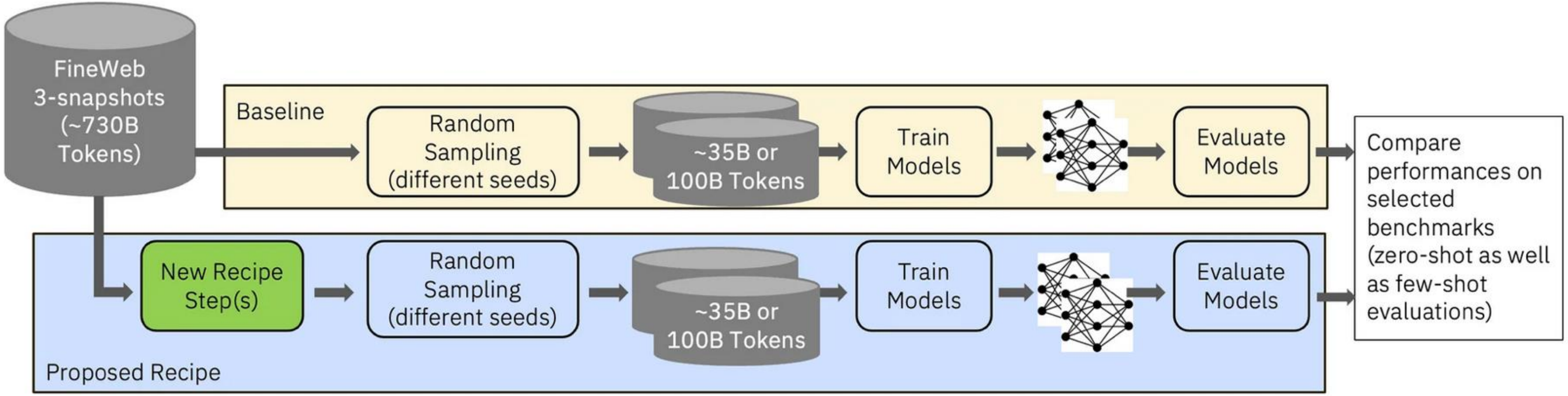
- The GneissWeb dataset was obtained by applying the following processing steps to FineWeb:
    - Exact substring deduplication at line level
    - Custom built fastText quality filter
    - Custom built fastText category classifier
    - Custom built Category-aware readability score quality filter
    - Custom built Category-aware extreme_tokenized quality filter



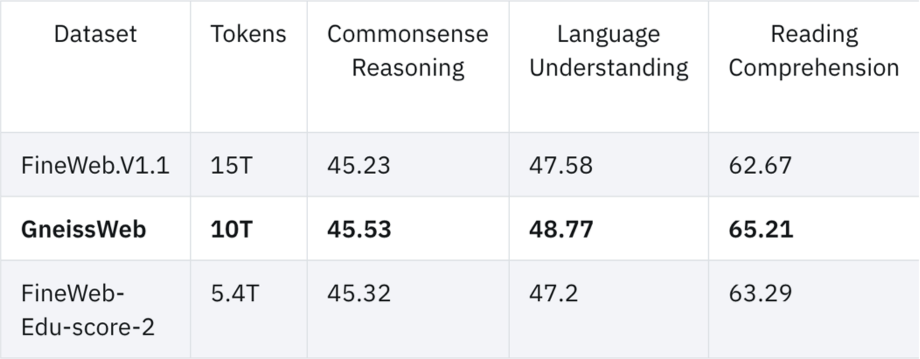
FineWeb Dataset
Exact Substring Deduplication
Data Quality Filtering
Readability Scores
+
FastText Category Classifiers
AND
FastText Quality Classifiers
Extreme Token Filter
+
FastText Category Classifiers
AND
FastText Quality Classifiers
OR
GneissWeb Dataset

Summary of Results

- To compare GneissWeb against the baselines, we trained decoder models with 1.4B, 3B, and 7B parameters on a Llama architecture
- We trained and evaluated our models on an LSF (Load Sharing Facility) cluster with each node equipped with eight H100 GPUs
- We evaluated our ablation models using lm-evaluation-harness on two categories of tasks: 11 High-Signal tasks (0-shot and few-shot) and 20 Extended tasks (0-shot and few-shot)

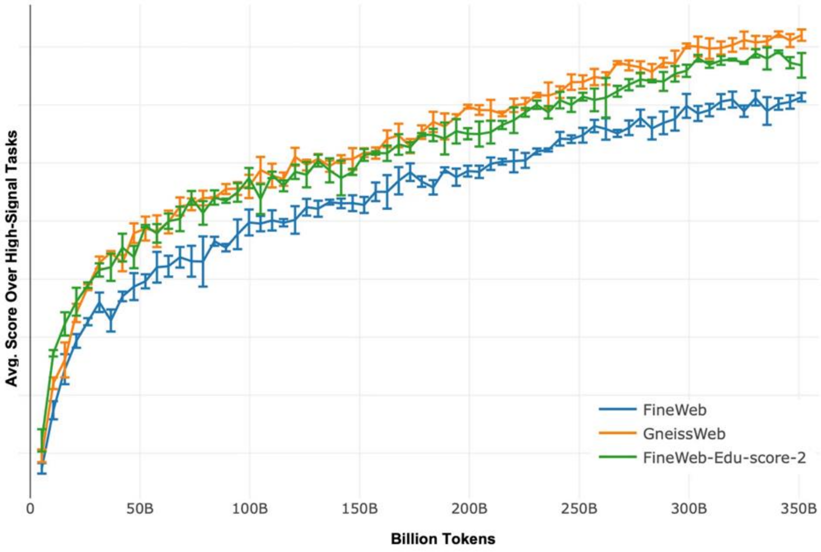

FineWeb 3-snapshots (~730B Tokens)
Baseline
Random Sampling (different seeds)
~35B or 100B Tokens
Train Models
Evaluate Models
New Recipe Step(s)
Random Sampling (different seeds)
~35B or 100B Tokens
Train Models
Evaluate Models
Proposed Recipe
Compare performances on selected benchmarks (zero-shot as well as few-shot evaluations)

Summary of Results (continued)



Dataset	Tokens	Commonsense Reasoning	Language Understanding	Reading Comprehension
FineWeb.V1.1	15T	45.23	47.58	62.67
GneissWeb	**10T**	**45.53**	**48.77**	**65.21**
FineWeb-Edu-score-2	5.4T	45.32	47.2	63.29

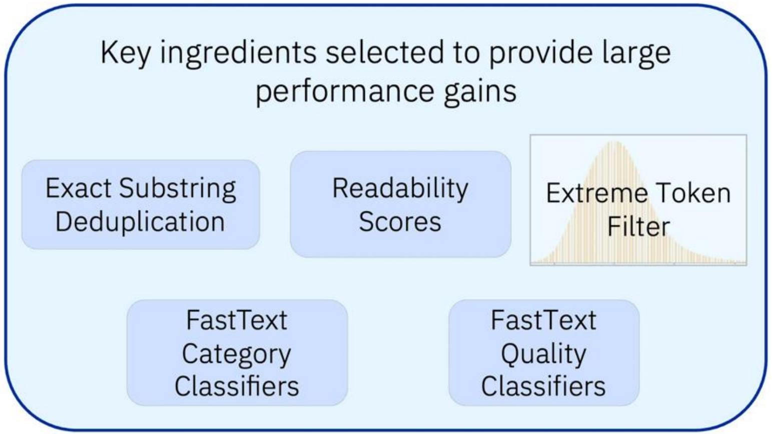
Comparison of average evaluation scores grouped by categories for 1.4 billion models trained on 350 billion tokens



Average evaluation score on High-Signal tasks versus the number of tokens for 1.4 Billion parameter models. The model trained on GneissWeb consistently outperforms the ones trained on FineWeb.V1.1.0 and FineWeb-Edu-score-2

Data Prep Kit role in GneissWeb

Five new transforms were developed and are available in the **Data Prep Kit** repo: https://github.com/IBM/data-prep-kit



Key ingredients selected to provide large performance gains

Exact Substring Deduplication

Readability Scores

Extreme Token Filter

FastText Category Classifiers

FastText Quality Classifiers

1. **Exact Substring Deduplication:** We apply exact substring deduplication to remove any substring of predetermined length that repeats verbatim more than once by adapting the implementation from Lee et al. (2022) based on suffix arrays

   ○ We shard each snapshot of FineWeb-V1.1.0 into sets of roughly equal size and apply exact substring deduplication on each shard independently. Also, rather than removing all copies of a duplicate substring, we retain the first occurrence of each duplicate substring and remove any subsequent matches exceeding 50 consecutive tokens.

Data Prep Kit role in GneissWeb (continued)

2. **Readability Score:** Readability scores are formulas based on text statistics (such as sentence length, average number of words, and the number of syllables) designed to assess how easily the text can be read and understood. We apply readability scores as a novel quality metric to facilitate identifying and filtering hard-to-read, low-quality documents.

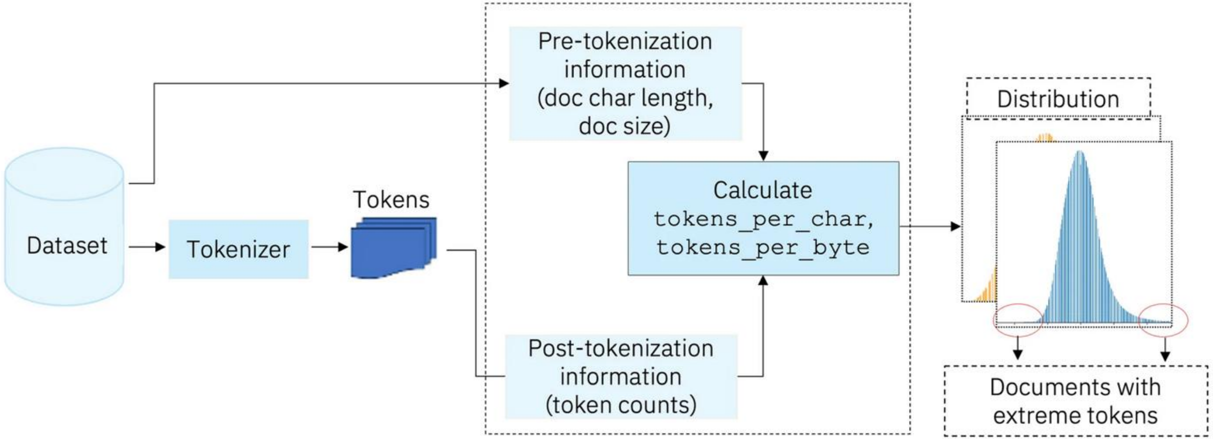
3. **Fasttext Quality Classifier:** The fastText family of binary classifiers has been shown to perform well in identifying high-quality pre-training documents. Specifically, DCLM trained a fastText classifier on a mix of instruction-formatted data. In addition to DCLM-fastText, we trained a custom fastText classifier on a mix of high-quality synthetic data and data annotated by LLM for high educational value.

Data Prep Kit role in GneissWeb (continued)

4. **Fasttext Category Classifier:** The quality score distributions in certain categories that potentially contain higher education level documents differ from the overall distribution across all categories in our dataset. In particular, we observe that the following categories have significantly different distributions than the overall distribution across all categories: **science, education, technology & computing, and medical health.** Thus, for each of these key categories, we annotate whether each document falls into the category.

Data Prep Kit Application (continued)

5. **Extreme tokenized documents removal:** We propose novel annotations that effectively leverage information from the "pre-tokenization" stage (document char length, document size) and the "post-tokenization" stage (token counts) to identify potential low-quality documents. We refer to the documents with extremely high or low number of tokens per character (or tokens per byte) as *extreme-tokenized documents,* and we remove them.


Dataset
Tokenizer
Tokens
Pre-tokenization information (doc char length, doc size)
Calculate tokens_per_char, tokens_per_byte
Post-tokenization information (token counts)
Distribution
Documents with extreme tokens

Combining GneissWeb components into a winning recipe

Exact Substring Deduplication

- Removes sequence-level duplicates within and across documents

Custom Data Quality Classifiers

- FastText classifier trained on Cosmopedia synthetic data, coupled with DCLM-fastText

Filtering based on Readability Scores

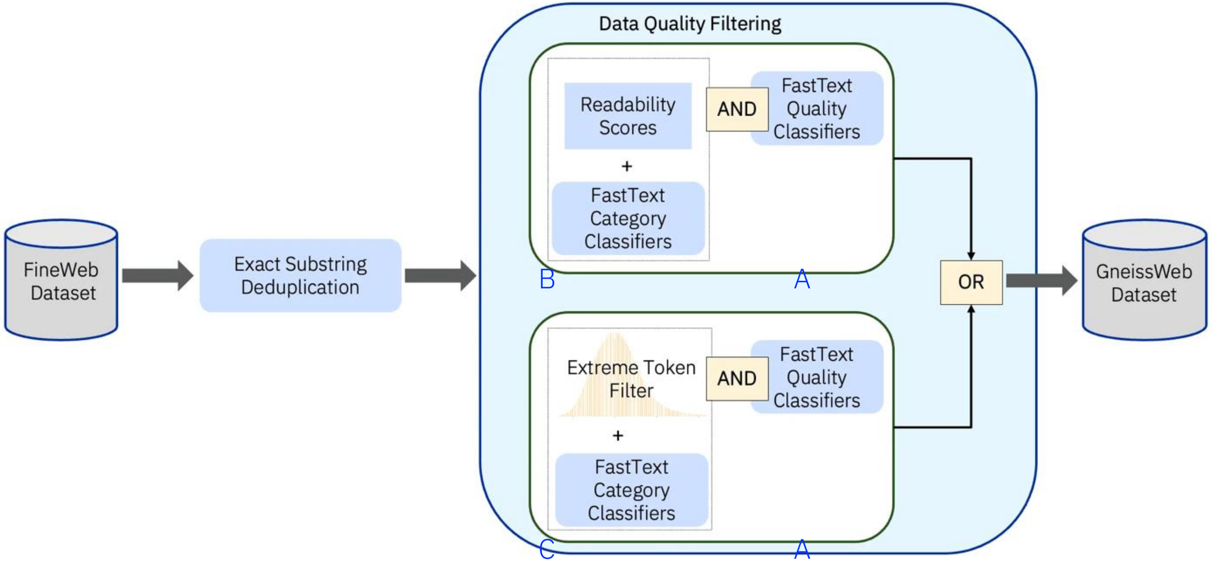
- Filter *hard-to-read* low-quality documents using linguistic metrics for accessing text difficulty

Filtering Extreme- Tokenized Documents

- Remove documents with extremely high or low number of tokens per character

Document Category Classifiers

- Leverage category annotations to strengthen quality filters



FineWeb Dataset
Exact Substring Deduplication
Data Quality Filtering
Readability Scores
+
FastText Category Classifiers
AND
FastText Quality Classifiers
B
A
Extreme Token Filter
+
FastText Category Classifiers
AND
FastText Quality Classifiers
C
A
OR
GneissWeb Dataset

GneissWeb Recipe:

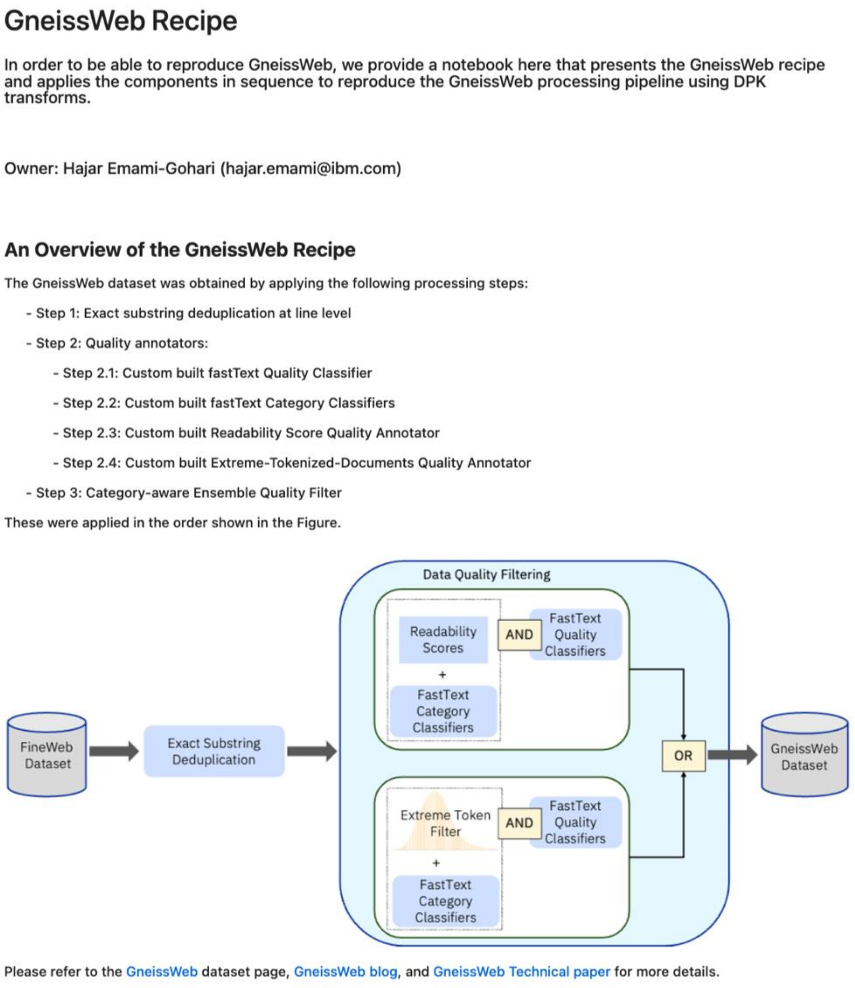
Exact substring deduplication → ((A AND B) OR (A AND C))

GneissWeb ensemble filtering rule: A document is retained if either the fastText combination and category-aware readability score filter agree to retain, or the fastText combination and category-aware extreme-tokenized filter agree to retain

Novel Components highlighted in color 🟥

GneissWeb Recipe

A notebook is available here:



GneissWeb Recipe

In order to be able to reproduce GneissWeb, we provide a notebook here that presents the GneissWeb recipe and applies the components in sequence to reproduce the GneissWeb processing pipeline using DPK transforms.

Owner: Hajar Emami-Gohari (hajar.emami@ibm.com)

An Overview of the GneissWeb Recipe

The GneissWeb dataset was obtained by applying the following processing steps:

- Step 1: Exact substring deduplication at line level
- Step 2: Quality annotators:
  - Step 2.1: Custom built fastText Quality Classifier
  - Step 2.2: Custom built fastText Category Classifiers
  - Step 2.3: Custom built Readability Score Quality Annotator
  - Step 2.4: Custom built Extreme-Tokenized-Documents Quality Annotator
- Step 3: Category-aware Ensemble Quality Filter

These were applied in the order shown in the Figure.

FineWeb Dataset
Exact Substring Deduplication
Data Quality Filtering
Readability Scores
+
FastText Category Classifiers
AND
FastText Quality Classifiers
Extreme Token Filter
+
FastText Category Classifiers
AND
FastText Quality Classifiers
OR
GneissWeb Dataset

Please refer to the GneissWeb dataset page, GneissWeb blog, and GneissWeb Technical paper for more details.

Summary

👉 > 2% avg improvement in benchmark performance over FineWeb

👉 Huggingface page

👉 Data prep kit detailed recipe

👉 Recipe models for reproduction

👉 announcement

👉 Paper