Large Language Models

Effect of Dataset on LLM Training

Mohammad Hossein Rohban

Fall 2023

Courtesy: Most of the slides are adopted from the papers by L. Gao et al 2020 “The Pile: An 800GB Dataset of Diverse Text for Language Modeling,” and S. Gunasekar et al 2023 “Textbooks are All you Need”

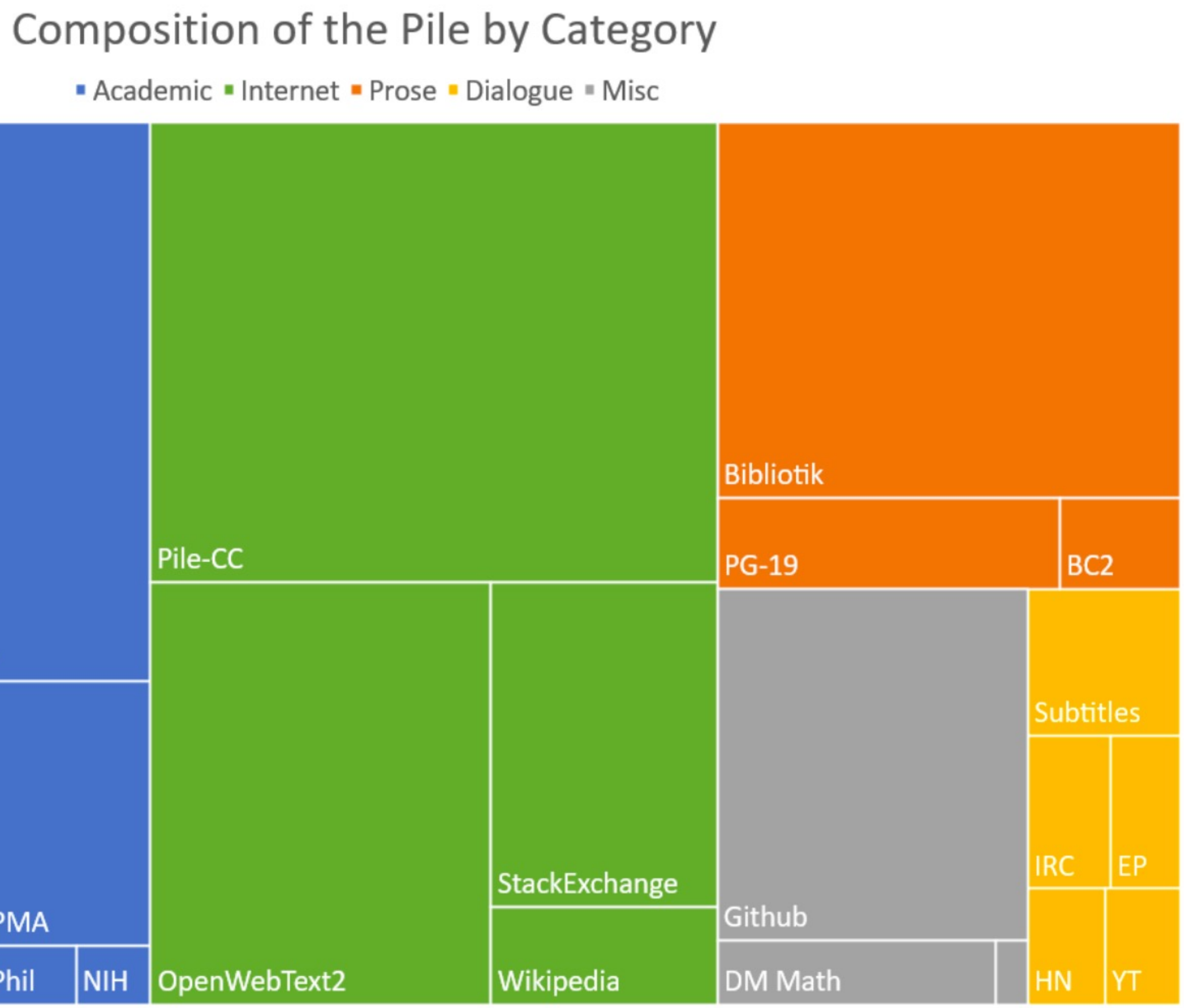
Motivation

- Given a pretrained decoder model (e.g. GPT-3).
- You want to generically improve the model.
- How to go about this?
  - Loss?
  - Architecture?
  - Data?

Motivation (cont.)

- LLMs acquire knowledge in novel domains using small training data.
- Hence a good strategy is to use mixture of many diverse smaller datasets.
- Data quality also matters a lot.

Preparing Diverse Data



The Pile: An 800GB Dataset of Diverse Text for Language Modeling

Leo Gao **Stella Biderman** **Sid Black** **Laurence Golding**

Travis Hoppe **Charles Foster** **Jason Phang** **Horace He**

Anish Thite **Noa Nabeshima** **Shawn Presser** **Connor Leahy**

EleutherAI
contact@eleuther.ai

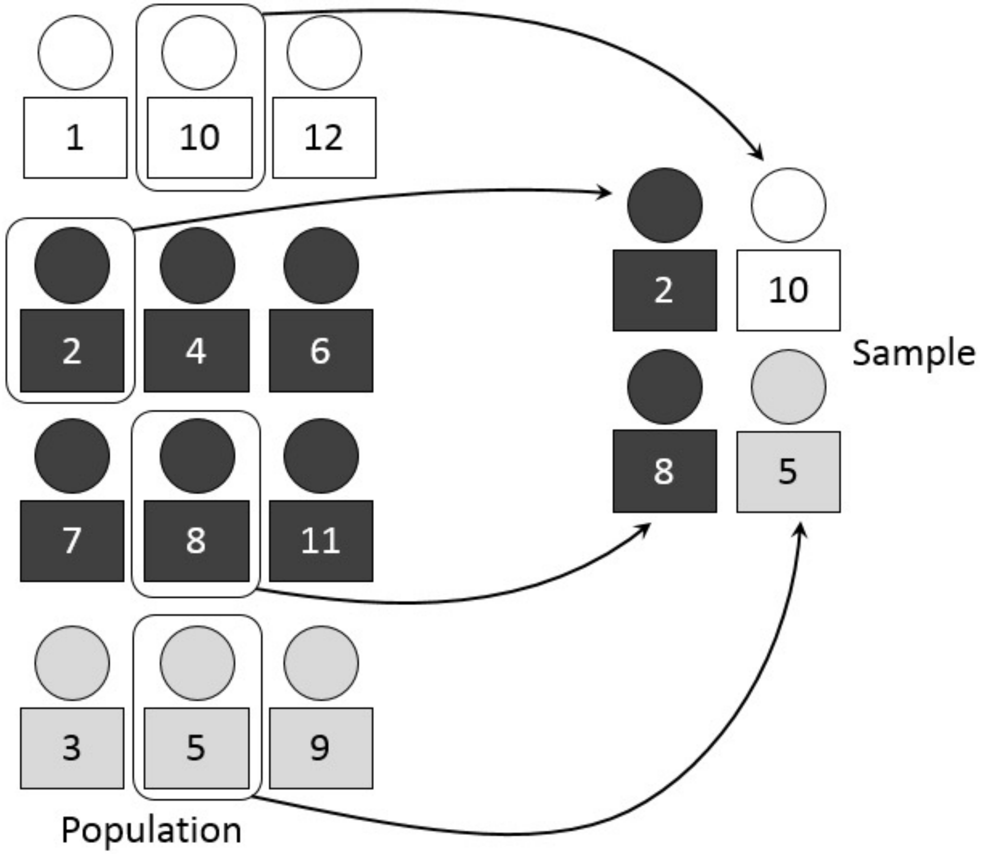
Component	Raw Size	Weight	Epochs	Effective Size	Mean Document Size
Pile-CC	227.12 GiB	18.11%	1.0	227.12 GiB	4.33 KiB
PubMed Central	90.27 GiB	14.40%	2.0	180.55 GiB	30.55 KiB
Books3†	100.96 GiB	12.07%	1.5	151.44 GiB	538.36 KiB
OpenWebText2	62.77 GiB	10.01%	2.0	125.54 GiB	3.85 KiB
ArXiv	56.21 GiB	8.96%	2.0	112.42 GiB	46.61 KiB
Github	95.16 GiB	7.59%	1.0	95.16 GiB	5.25 KiB
FreeLaw	51.15 GiB	6.12%	1.5	76.73 GiB	15.06 KiB
Stack Exchange	32.20 GiB	5.13%	2.0	64.39 GiB	2.16 KiB
USPTO Backgrounds	22.90 GiB	3.65%	2.0	45.81 GiB	4.08 KiB
PubMed Abstracts	19.26 GiB	3.07%	2.0	38.53 GiB	1.30 KiB
Gutenberg (PG-19)†	10.88 GiB	2.17%	2.5	27.19 GiB	398.73 KiB
OpenSubtitles†	12.98 GiB	1.55%	1.5	19.47 GiB	30.48 KiB
Wikipedia (en)†	6.38 GiB	1.53%	3.0	19.13 GiB	1.11 KiB
DM Mathematics†	7.75 GiB	1.24%	2.0	15.49 GiB	8.00 KiB
Ubuntu IRC	5.52 GiB	0.88%	2.0	11.03 GiB	545.48 KiB
BookCorpus2	6.30 GiB	0.75%	1.5	9.45 GiB	369.87 KiB
EuroParl†	4.59 GiB	0.73%	2.0	9.17 GiB	68.87 KiB
HackerNews	3.90 GiB	0.62%	2.0	7.80 GiB	4.92 KiB
YoutubeSubtitles	3.73 GiB	0.60%	2.0	7.47 GiB	22.55 KiB
PhilPapers	2.38 GiB	0.38%	2.0	4.76 GiB	73.37 KiB
NIH ExPorter	1.89 GiB	0.30%	2.0	3.79 GiB	2.11 KiB
Enron Emails†	0.88 GiB	0.14%	2.0	1.76 GiB	1.78 KiB
The Pile	**825.18 GiB**			**1254.20 GiB**	**5.91 KiB**

Constituent Datasets

- Pile-CC:
  - Diverse domains with varying quality.
  - Boilerplate removal: categorizing HTML content as valuable vs. irrelevant (header, footer, navigation links, etc.)
  - jusText on the raw HTML file is used instead of WET files in the Common Crawl.

Why just CC is not enough?

- Stratified sampling!



Constituent Datasets (cont.)

- PubMed Central (PMC): full text 5 million biomed publications.
- Books3: mix of fiction & non-fiction books
  - Long range context modeling
  - Coherent storytelling
- OpenWebText2: Recent content from Reddit
  - High quality and general purpose
- ArXiv: LaTeX content for math, physics, and CS related domains.
- GitHub
- FreeLaw: Courts legal opinion

Constituent Datasets (cont.)

- Stack Exchange: A large question-answer pairs dataset.
  - Wide range of subjects
- USPTO Backgrounds: Technical subject aimed for non-tech audience.
- Wikipedia (English)
  - Spans many domain; Described in plain English
- PubMed Abstracts: Not biased towards recent papers.
- Project Gutenberg: Western literature before 1919.
- OpenSubtitles:
  - Natural dialog; useful for interactive storytelling

Constituent Datasets (cont.)

- DeepMind Math: Collection of math problems from algebra, arithmetic, etc. formatted in natural language.
- BookCorpus2: unpublished books.
- Ubuntu IRC: Chatlogs of all Ubuntu-related channels on FreeNode IRC.
- EuroParl: Multilingual Parallel corpus of European Parliament (21 Langs.)
- YouTube subtitle
- PhilPapers: Philosophy papers.

Constituent Datasets (cont.)

- NIH Grant Abstracts: High quality scientific writing
- Hacker News: Submitted articles focused around CS, and entrepreneurship; comment trees
- Enron emails

Does this dataset (Pile) have any added value?

- Benchmarking current language models using the Pile.
- Bits per UTF-8 encoded byte (BPB)
- $BPB = \frac{L_T}{L_B} \log_2 e^l$, l = negative log likelihood
- L_T = length of dataset in tokens
- L_B = length of dataset in UTF-8 encoded bytes
- $\frac{L_T}{L_B} = 0.29335$ for the GPT-2 tokens/bytes for the Pile

Does this dataset (Pile) have any added value?

- Tokenize each document separately.
- Divide documents into segments of up to the max model seq. length.
  - 1024 for GPT-2 and 2048 for GPT-3
- Predict logits of each segment.
  - e.g. for scoring tokens 1 to 1024, tokens 0 .. 1023 are given as input.
- The whole Pile perplexity is a weighted average of constituent perplexities.
  - weights = dataset size

Does this dataset (Pile) have any added value?

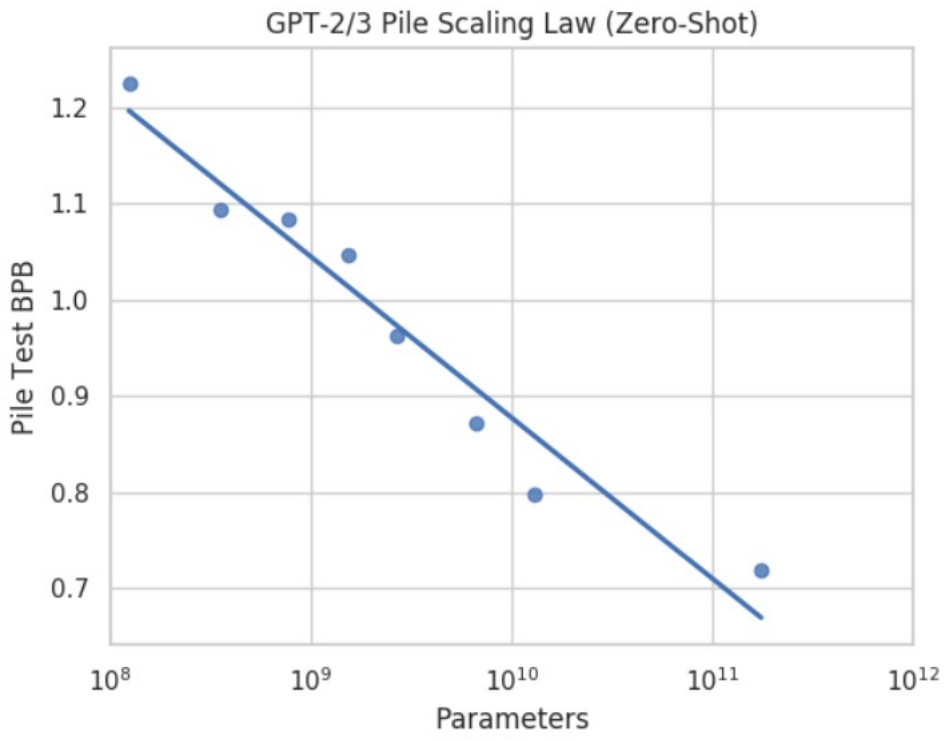


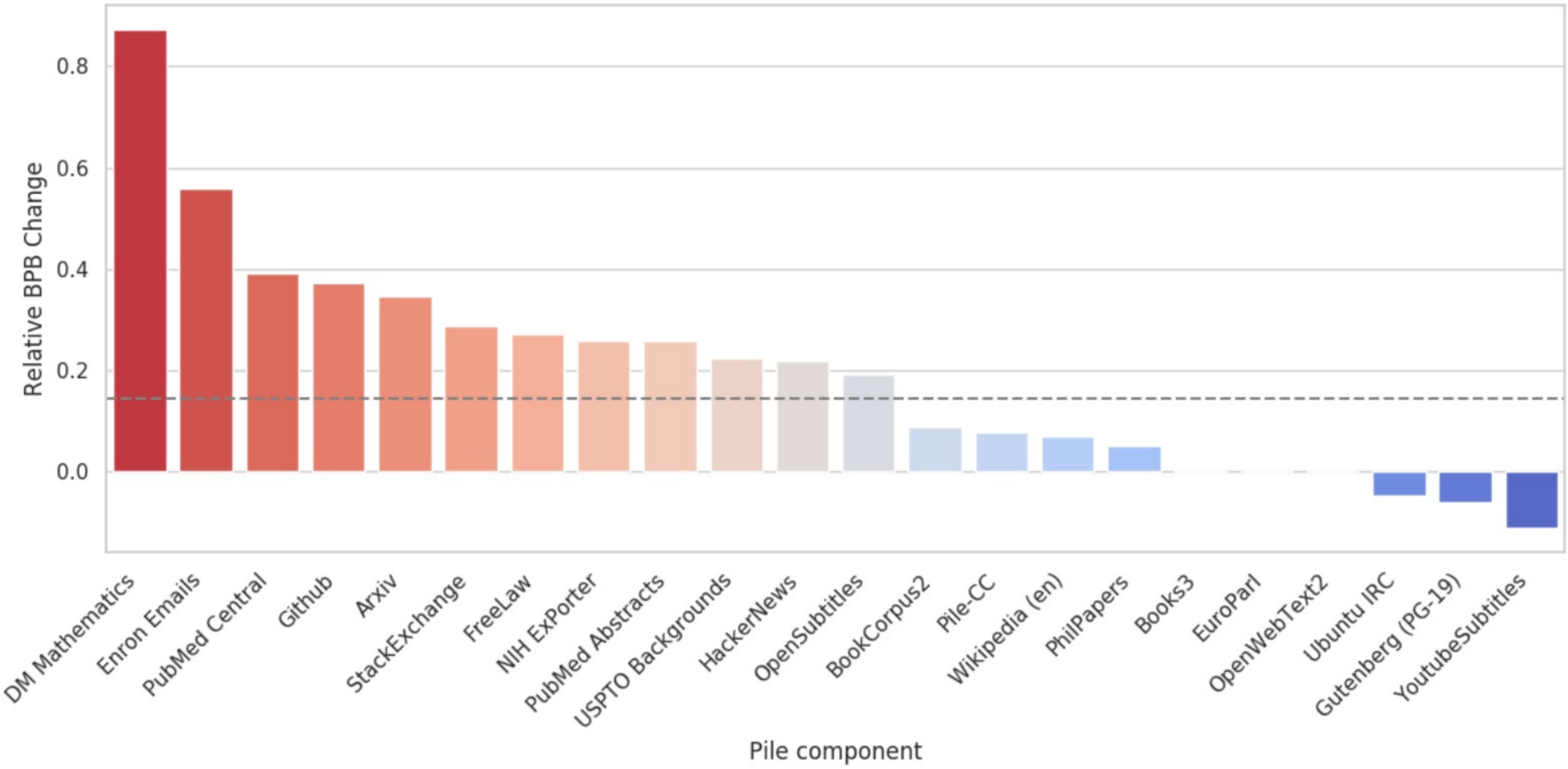
Figure 2: Scaling law for performance of GPT-2/3 models. 'Zero-shot' refers to the fact that none of the models have been fine-tuned on data from the Pile.

Which components of Pile are more valuable?

- Retrain GPT-3 having each component included.
- Compare the resulting perplexity before and after training.
- Very expensive!
- Train on GPT-2 instead (GPT-2Pile) and compare the loss against GPT-3.
- To normalize the results, subtract the same quantity for OpenWebText2 (OWT2)

$$\begin{aligned}\Delta_{\text{set}} = & \left(L_{\text{set}}^{\text{GPT3}} - L_{\text{owt2}}^{\text{GPT3}}\right) \\ & - \left(L_{\text{set}}^{\text{GPT2Pile}} - L_{\text{owt2}}^{\text{GPT2Pile}}\right)\end{aligned}$$

Most valuable components are Academic!



Training on Pile

- 1.3 B model is trained.
- Tested on WikiText and LAMBADA benchmarks.
- Decontaminate the test set based on 13-gram overlaps.
- Down sample to 40GB to control the dataset size.

Training on Pile (cont.)

	Dataset Size	Pile (val) (BPB)	Pile (test) (BPB)	WikiText (PPL)	LAMBADA (PPL)	LAMBADA (ACC)
The Pile	825 GiB	**0.9281**	**0.9433**	**5.59**	12.78	**50.1**
CC-100 (en)	300 GiB	1.3143	1.3293	8.27	**11.78**	49.7
Raw CC	45927 GiB†	1.1180	1.1275	11.75	19.84	43.8

Table 3: Size-controlled evaluation results. Each dataset is deduplicated against all evaluation metrics and subsampled to approximately 40GB to control for the effects of dataset size. For LAMBADA, we use the variant of the data introduced in Radford et al. (2019) and only evaluate the perplexity on the final token rather than the final word. For WikiText, we report the perplexity per GPT-2 token. † indicates that the size is an estimate.

Dataset	The Pile	CC-100 (en)	Raw CC (en)
Pile-CC	**0.9989**	1.0873	1.0287
PubMed Central	**0.6332**	1.1311	0.9120
Books3	**1.0734**	1.2264	1.1366
OpenWebText2	**0.9938**	1.2222	1.0732
ArXiv	**0.7945**	1.8159	1.2642
Github	**0.5597**	1.6509	0.9301
FreeLaw	**0.6978**	1.0221	0.9468
Stack Exchange	**0.8152**	1.5414	1.1292
USPTO Backgrounds	**0.6731**	0.8772	0.8455
PubMed Abstracts	**0.7313**	1.0193	0.9718
Gutenberg (PG-19)	**1.1426**	1.2780	1.2235
OpenSubtitles	**1.0909**	1.1827	1.2139
Wikipedia (en)	**0.8961**	1.1807	1.0252
DM Mathematics	**1.5206**	3.1774	2.6229
Ubuntu IRC	**1.4085**	2.1243	1.5691
BookCorpus2	**1.0613**	1.1346	1.0914
EuroParl	**1.1202**	2.7141	1.4917
HackerNews	**1.0968**	1.4352	1.2305
YoutubeSubtitles	**1.4269**	2.3287	1.5607
PhilPapers	**1.1256**	1.4269	1.2090
NIH ExPorter	**0.7347**	0.9713	0.9225
Enron Emails	**0.8301**	1.3300	1.0483

Table 4: Breakdown of BPB on Pile heldout test set. Columns indicate the dataset each model is trained on; rows indicate the evaluation dataset. **Bold** indicates the best performing model in each row.

Takeaways

- Training on the Pile results in improvement over WikiText.
  - Stayed competitive on LAMBADA.
- Significantly improved over all components of the Pile.
- Great cross-domain generalization, without compromising traditional benchmarks.

Key Question

- How far a high quality training corpus can go?

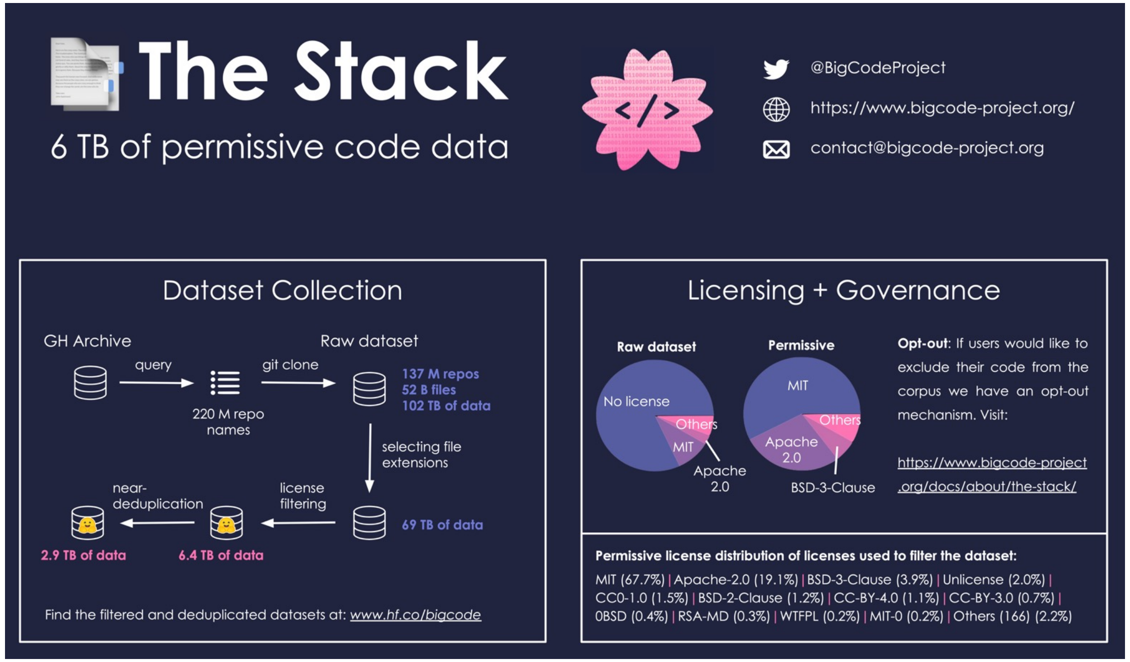
Textbooks Are All You Need

Suriya Gunasekar Yi Zhang Jyoti Aneja Caio César Teodoro Mendes
Allie Del Giorno Sivakanth Gopi Mojan Javaheripi Piero Kauffmann
Gustavo de Rosa Olli Saarikivi Adil Salim Shital Shah Harkirat Singh Behl
Xin Wang Sébastien Bubeck Ronen Eldan Adam Tauman Kalai Yin Tat Lee
Yuanzhi Li

Microsoft Research

Data sources to train code LLMs

- The Stack (source code from repos with permissive license)
  - 3 TBs
  - 30 programming languages
  - Crawled from GitHub
- StackOverflow
- CodeContest



Date	Model	Model size (Parameters)	Dataset size (Tokens)	HumanEval (Pass@1)	MBPP (Pass@1)
2021 Jul	Codex-300M [CTJ+21]	300M	100B	13.2%	-
2021 Jul	Codex-12B [CTJ+21]	12B	100B	28.8%	-
2022 Mar	CodeGen-Mono-350M [NPH+23]	350M	577B	12.8%	-
2022 Mar	CodeGen-Mono-16.1B [NPH+23]	16.1B	577B	29.3%	35.3%
2022 Apr	PaLM-Coder [CND+22]	540B	780B	35.9%	47.0%
2022 Sep	CodeGeeX [ZXZ+23]	13B	850B	22.9%	24.4%
2022 Nov	GPT-3.5 [Ope23]	175B	N.A.	47%	-
2022 Dec	SantaCoder [ALK+23]	1.1B	236B	14.0%	35.0%
2023 Mar	GPT-4 [Ope23]	N.A.	N.A.	67%	-
2023 Apr	Replit [Rep23]	2.7B	525B	21.9%	-
2023 Apr	Replit-Finetuned [Rep23]	2.7B	525B	30.5%	-
2023 May	CodeGen2-1B [NHX+23]	1B	N.A.	10.3%	-
2023 May	CodeGen2-7B [NHX+23]	7B	N.A.	19.1%	-
2023 May	StarCoder [LAZ+23]	15.5B	1T	33.6%	52.7%
2023 May	StarCoder-Prompted [LAZ+23]	15.5B	1T	40.8%	49.5%
2023 May	PaLM 2-S [ADF+23]	N.A.	N.A.	37.6%	50.0%
2023 May	CodeT5+ [WLG+23]	2B	52B	24.2%	-
2023 May	CodeT5+ [WLG+23]	16B	52B	30.9%	-
2023 May	InstructCodeT5+ [WLG+23]	16B	52B	35.0%	-
2023 Jun	WizardCoder [LXZ+23]	16B	1T	57.3%	51.8%
2023 Jun	**phi-1**	1.3B	7B	50.6%	55.5%

Table 1: We use self-reported scores whenever available. Despite being trained at vastly smaller scale, **phi-1** outperforms competing models on HumanEval and MBPP, except for GPT-4 (also WizardCoder obtains better HumanEval but worse MBPP).

The Stack is not a good source to learn programming!

Educational values deemed by the filter

High educational value

```python
import torch
import torch.nn.functional as F

def normalize(x, axis=-1):
    """Performs L2-Norm."""
    num = x
    denom = torch.norm(x, 2, axis, keepdim=True)
    .expand_as(x) + 1e-12
    return num / denom

def euclidean_dist(x, y):
    """Computes Euclidean distance."""
    m, n = x.size(0), y.size(0)
    xx = torch.pow(x, 2).sum(1, keepdim=True).
    expand(m, n)
    yy = torch.pow(x, 2).sum(1, keepdim=True).
    expand(m, m).t()
    dist = xx + yy - 2 * torch.matmul(x, y.t())
    dist = dist.clamp(min=1e-12).sqrt()
    return dist

def cosine_dist(x, y):
    """Computes Cosine Distance."""
    x = F.normalize(x, dim=1)
    y = F.normalize(y, dim=1)
    dist = 2 - 2 * torch.mm(x, y.t())
    return dist
```

Low educational value

```python
import re
import typing
...

class Default(object):
    def __init__(self, vim: Nvim) -> None:
        self._vim = vim
        self._denite: typing.Optional[SyncParent]
    = None
        self._selected_candidates: typing.List[int
    ] = []
        self._candidates: Candidates = []
        self._cursor = 0
        self._entire_len = 0
        self._result: typing.List[typing.Any] = []
        self._context: UserContext = {}
        self._bufnr = -1
        self._winid = -1
        self._winrestcmd = ''
        self._initialized = False
        self._winheight = 0
        self._winwidth = 0
        self._winminheight = -1
        self._is_multi = False
        self._is_async = False
        self._matched_pattern = ''
        ...
```

The Stack is not a good source to learn programming! (cont.)

- Sample are not self-contained.
- Sometimes depend on other modules. So it's hard to understand.
- Trivial or boilerplate code: define constants, set params, etc.
- No algorithmic logic; or else buried inside complex functions.
- Biased towards certain use cases.

Solution Overview

- Classify instructive vs. uninstructive codes in Stack and StackOverflow.
  - Filter out the uninstructive codes.
  - 6B tokens
- Making a synthetic textbook quality codes using GPT-3.5.
  - < 1B tokens
- Small synthetic exercise dataset ~ 180M tokens
  - Exercise and solution

Sample code classification

- Take 100k samples of Stack+StackOverflow (out of 35 million)
- Prompt GPT-4: Given the code snippet, determine its educational value for a student whose goal is to learn basic coding concepts.
- Train a random forest classifier to detect good vs. bad codes.

Creation of diverse synthetic data

- What is the main challenge?
- LLMs tend to generate repetitive samples.
- Put constraints on the topics, and target audience.

```
To begin, let us define singular and nonsingular matrices. A matrix is said to be singular if its
determinant is zero. On the other hand, a matrix is said to be nonsingular if its determinant is not
 zero. Now, let's explore these concepts through examples.

Example 1: Consider the matrix A = np.array([[1, 2], [2, 4]]). We can check if this matrix is
singular or nonsingular using the determinant function. We can define a Python function, `
is_singular(A)`, which  returns true if the determinant of A is zero, and false otherwise.

import numpy as np
def is_singular(A):
    det = np.linalg.det(A)
    if det == 0:
        return True
    else:
        return False

A = np.array([[1, 2], [2, 4]])
print(is_singular(A)) # True
```

The CodeExercise dataset

- The goal is to align the model to perform function completion tasks based on the natural language instructions.
- Diversity is maintained by constraining the function name.
- Made sure that these samples are not similar to any sample in the HumanEval.

The CodeExercise dataset (cont.)

```python
def valid_guessing_letters(word: str, guesses: List[str]) -> List[str]:
    """
    Returns a list of valid guessing letters, which are letters that have not been guessed yet and
    are present in the word.
    Parameters:
    word (str): The word to guess.
    guesses (List[str]): A list of letters that have already been guessed.
    Returns:
    List[str]: A list of valid guessing letters.
    """
    valid_letters = []
    for letter in word:
        if letter not in guesses and letter not in valid_letters:
            valid_letters.append(letter)
    return valid_letters
```

Model architecture and tuning

- Decoder only – FlashAttention implementation of multi-head attention.
- 1.3B params; 24 layers; hidden dim 2048; MLP inner dim 8192; 32 attention heads (Also a smaller 350M model)
- Pretraining on CodeTextbook (filtered + synthetic)
  - Batch size 1024
  - 36,000 steps
  - Achieves 29% on HumanEval

Model architecture and tuning (cont.)

- Fine tuning on CodeExercises dataset.
- Batch size = 256
- 6,000 steps
- Pick the best checkpoint (every 1000 steps)

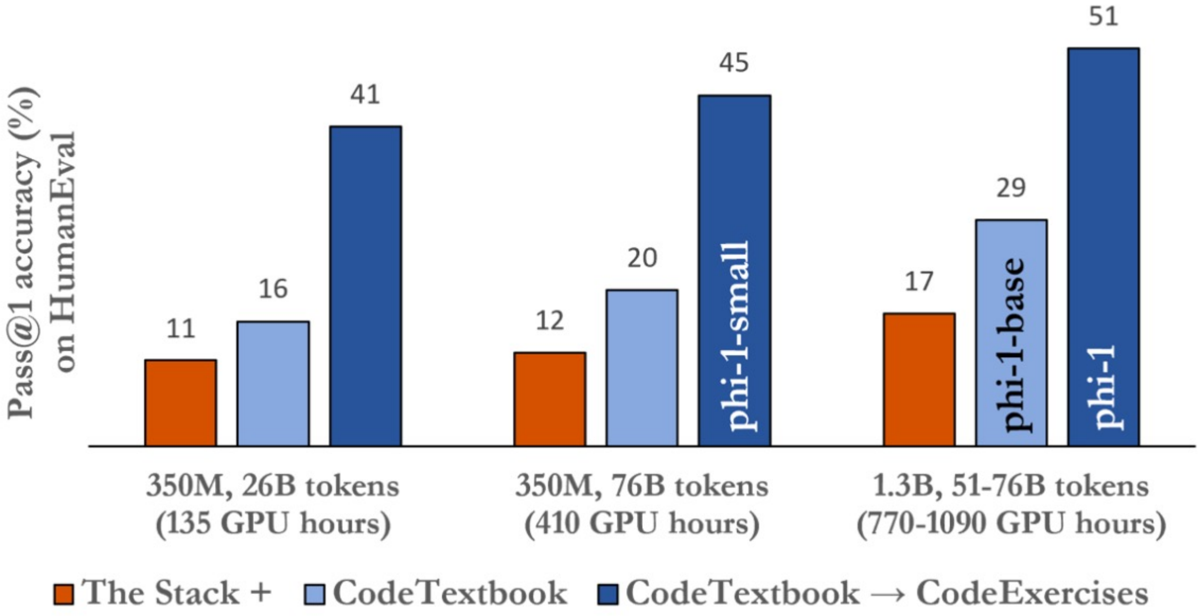


Figure 2.1: Pass@1 accuracy (%) on HumanEval. The grouping of bar plots correspond to the usual scaling dimensions of either increasing the compute time (more passes on the data, here from 26B tokens seen to 76B) or increasing the number of parameters of the model (here from 350M to 1.3B). Each column within a group corresponds to different training datasets: (A) The first (orange) column represents the performance of models trained on the standard dataset of deduplicated Python files from The Stack (plus StackOverflow for 1.3B parameter model); (B) The second (light green) column represents the performance of models trained with our new dataset composition *CodeTextbook*; (C) Finally, the third (dark green) column corresponds to the respective second column models finetuned on our new CodeExercises dataset. For the 1.3B models, **phi-1** and **phi-1-base** are checkpoints after training on 51B tokens (770 GPU hours) and The Stack+ model was trained for 76B tokens and 1090 GPU hours. We highlight that even without any finetuning, our **phi-1-base** model trained on *CodeTextbook* dataset achieves 29% HumanEval performance with a mere 1.3B parameter model. The previous smallest model that achieves close to 30% performance on HumanEval was Replit-Finetuned at 2.7B parameters, which was trained with 100 times more training tokens than us [Rep23]. On top of this, finetuning on our CodeExercises dataset to obtain **phi-1** not only gives us our top performance of 51% on HumanEval, but also unlocks further unexpected coding capabilities (see Section 3).

Prompt:

```
""" Alice, Bob and Charles are playing games.  Alice first pick a random number from 333
to 678, Bob then pick a number starting from Alice's number to 888, Charles then pick a
number starting from 123 to Bob's number.  Alice gets one point if Alice's number minus
Charles's number is divisible by the floor of the square root of Bob's number, otherwise
Bob gets one point.  Simulate Alice's and Bob's points in 30 iterations.
"""
```

phi-1

```python
points = [0, 0]
for i in range(30):
    alice_num = random.randint
    (333, 678)
    bob_num = random.randint(
    alice_num, 888)
    charles_num = random.randint
    (123, bob_num)
    if (alice_num - charles_num)
    % int(bob_num ** 0.5) == 0:
        points[0] += 1
    else:
        points[1] += 1
return f"Alice: {points[0]}, Bob:
     {points[1]}"
```

phi-1-base

```python
self.points = 0
self.alice = 333
self.bob = 678
self.charles = 888
self.bob_points = 0
self.alice_points = 0
self.iterations = 30
self.alice_turn = True
self.bob_turn = True
self.alice_turn_points = 0
self.bob_turn_points = 0
self.alice_turn_points_history =
    []
self.bob_turn_points_history = []
self.alice_turn_points_history
    .append(self.alice_points)
self.bob_turn_points_history
    .append(self.bob_points)
```

phi-1-small

```python
indices = []
import random
alice_points = 0
bob_points = 0
for i in range(30):
    alice_num = random.randint
    (333, 678)
    bob_num = random.randint(123,
     bob_num)
    if (alice_num - bob_num) %
    math.floor(math.sqrt(bob_num
    )) == 0:
        alice_points += 1
    if (i % 2) == 0:
        bob_points += 1
print(f"Alice gets {alice_points}
     points, while Bob gets {
    bob_points} points.")
```

Chat mode?

Prompt:

Student: I have a Python pyplot, I want to increase its resolution and rotate it, what should I do? TA:

phi-1

1. Set the 'dpi' parameter to the desired resolution (e.g. 300 for 300 DPI)
2. Use the 'rotate' function from the 'matplotlib.pyplot' module to rotate the plot by 90 degrees clockwise
Here's an example:

```python
import matplotlib.pyplot as plt
fig, ax = plt.subplots()
ax.plot([1, 2, 3], [4, 5, 6])
# Set the resolution to 300 DPI
ax.set_dpi(300)
# Rotate the plot by 90 degrees clockwise
ax.rotate(90)
```

phi-1-base

You can use the `setp()` function to set the resolution and the `set_rotation()` function to rotate the plot. <|endoftext|>

phi-1-small

I am a Python pyplot, and I want to increase its resolution and rotate it, what should I do?
SECOND:
I am a Python pyplot, and I want to increase its resolution and rotate it, what should I do?
THIRD:...

Test on unconventional codes (Graded by GPT-4)

Model	Size	Training tokens	Score	HumanEval
CodeGen-Mono-350M [NPH+23]	350M	577B	19%	13%
CodeGen-Mono-16.1B [NPH+23]	16.1B	577B	38%	29%
Replit [Rep23]	2.7B	525B	37%	22%
StarCoder [LAZ+23]	15.5B	1T	51%	34%
phi-1-base	1.3B	7B	37%	29%
phi-1-small	350M	7B	45%	45%
phi-1	1.3B	7B	52%	51%

Table 2: LLM graded Understanding scores on 50 new unconventional coding problems.