# TEXTBOOKS ARE ALL YOU NEED

**Phi-1**, **Phi-1.5**, Phi-2 (MIT license) from Microsoft

Yan Xu

# Key Idea

**Breaking existing scaling laws with high-quality data**

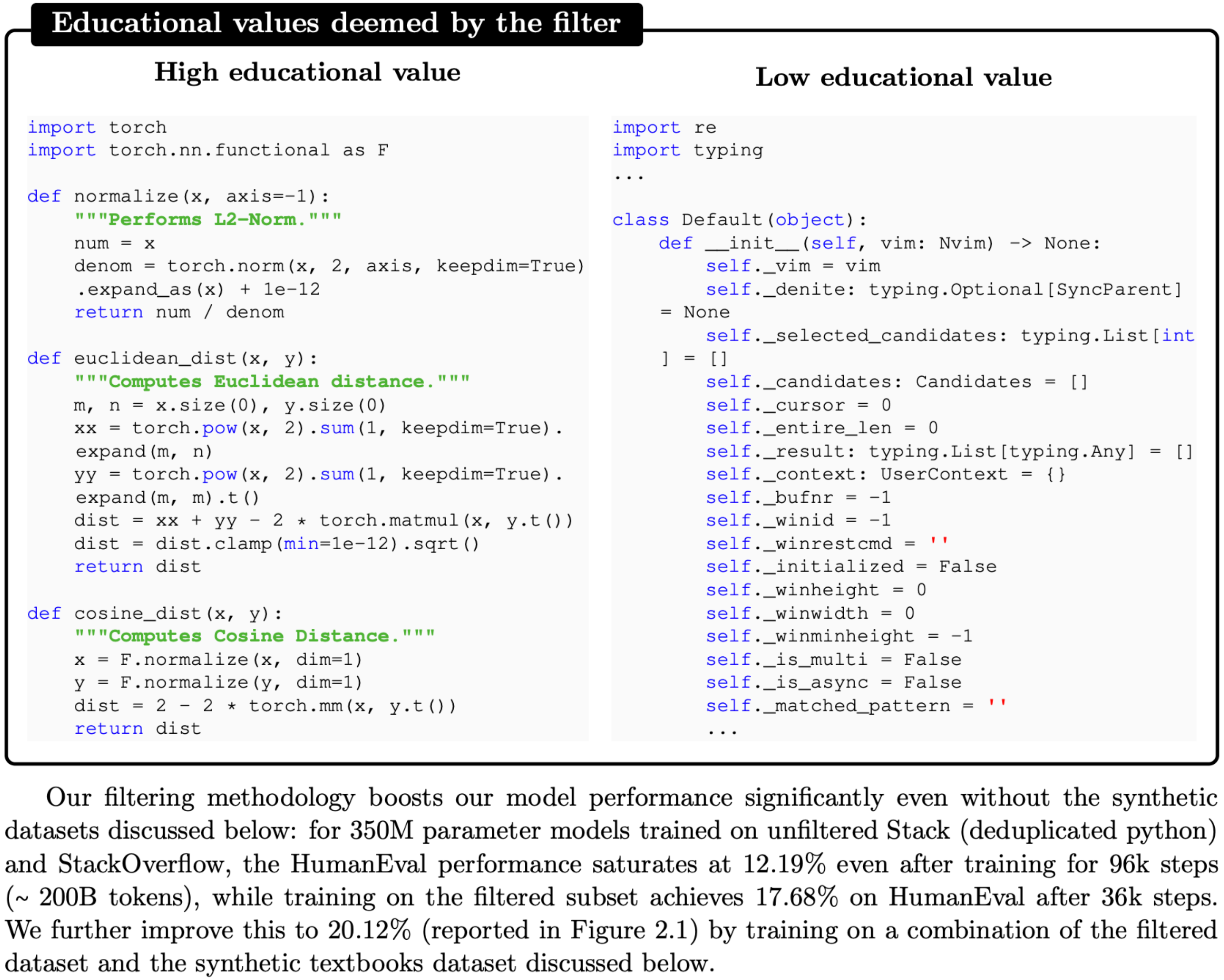
- Use high-quality textbook-style data that have higher education values
- Use synthetic textbook data with an existing LLM (e.g. GPT-3.5)
- Train a smaller model
- Achieve comparable or better performance

**Small Language Model with big ambition:**
Can we achieve ChatGPT's level of capability at the one billion parameters scale?

# Phi-1

- Phi-1 is a decoder-only transformer-based model with 1.3B parameters, trained for 4 days on 8 A100s
- Trained with a selection of “textbook quality” data.
    - A **filtered code-language dataset**, which is a subset of The Stack and StackOverflow, the filter classifier is built by:
        - GPT-4 as the annotator (prompted to “determine its educational value for a student whose goal is to learn basic coding concepts”.)
        - train a random forest classifier based on the codegen embeddings
    - A synthetic textbook dataset consisting of <1B tokens of GPT-3.5 generated Python textbooks.
    - A small synthetic exercises dataset consisting of ∼180M tokens of Python exercises and solutions.

# Filtered code-language dataset



**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
```

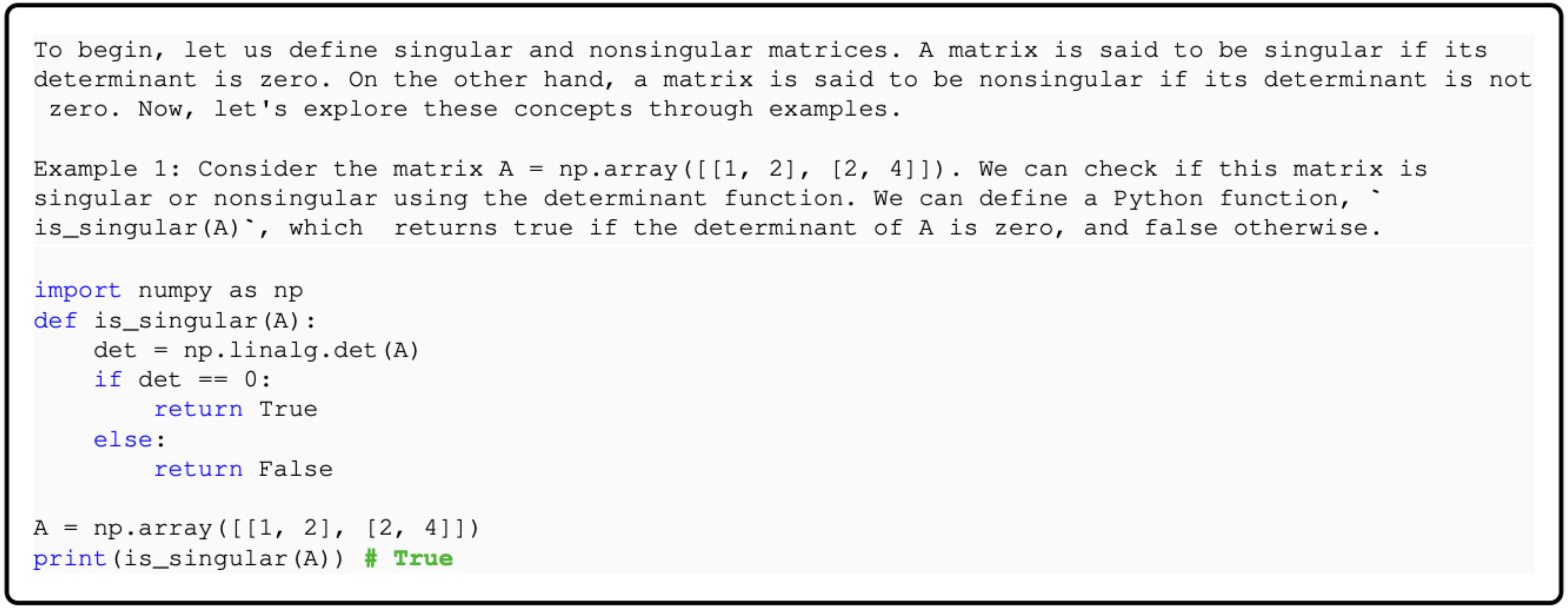
**Low educational value**

```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 = ''
        ...
```

Our filtering methodology boosts our model performance significantly even without the synthetic datasets discussed below: for 350M parameter models trained on unfiltered Stack (deduplicated python) and StackOverflow, the HumanEval performance saturates at 12.19% even after training for 96k steps (~ 200B tokens), while training on the filtered subset achieves 17.68% on HumanEval after 36k steps. We further improve this to 20.12% (reported in Figure 2.1) by training on a combination of the filtered dataset and the synthetic textbooks dataset discussed below.

# Synthetic textbook dataset



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.

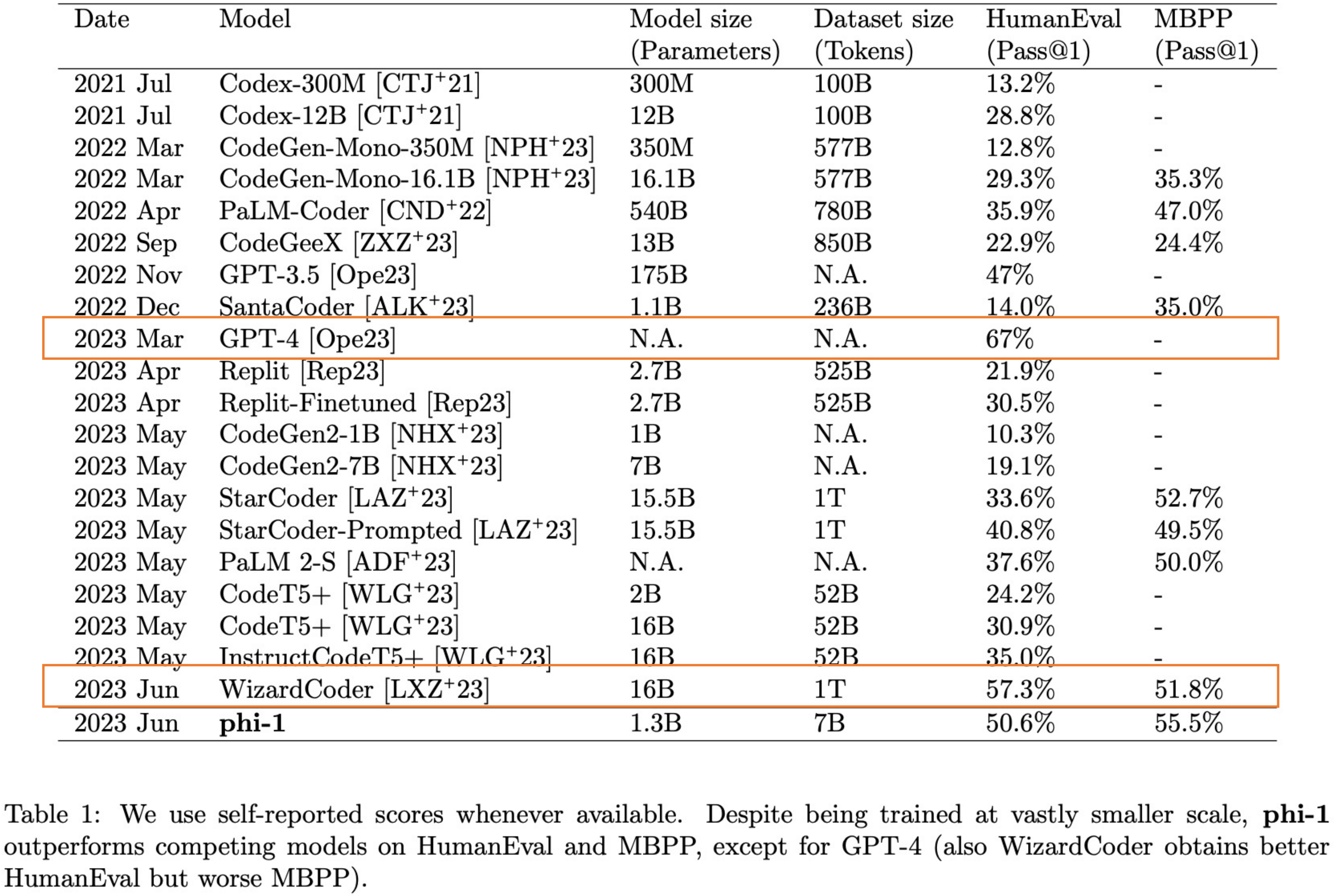
Example 1: Consider the matrix A = np.array([[1, 2], [2, 4]]). We can check if this matrix is singular or nonsingular using the determinant function. We can define a Python function, ` is_singular(A)`, which returns true if the determinant of A is zero, and false otherwise.

```
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
```

1B tokens of GPT-3.5 generated Python textbooks, synthesized to provide a high-quality source of natural language heavy text interleaved with relevant code snippets. We further targeted the content of these textbooks to cover topics that promote reasoning and basic algorithmic skills.

# Phi-1 on Code

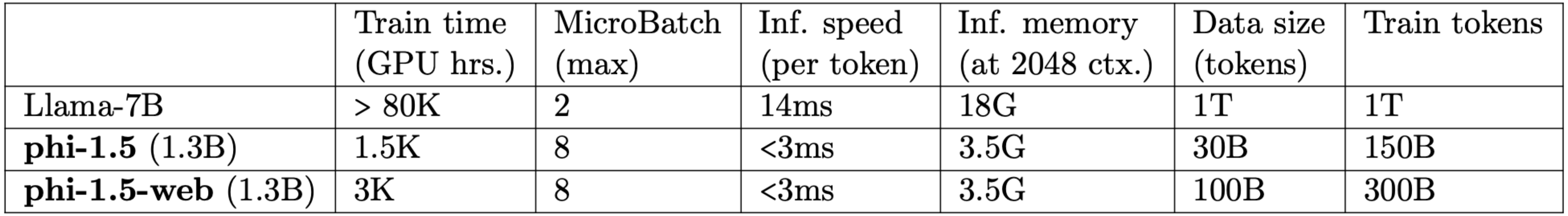


| Date | Model | Model size<br>(Parameters) | Dataset size<br>(Tokens) | HumanEval<br>(Pass@1) | MBPP<br>(Pass@1) |
|---|---|---|---|---|---|
| 2021 Jul | Codex-300M [CTJ+21] | 300M | 100B | 13.2% | - |
| 2021 Jul | Codex-12B [CTJ+21] | 12B | 100B | 28.8% | - |
| 2022 Mar | CodeGen-Mono-350M [NPH+23] | 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).

# Phi-1.5

It is a exactly the same as our previous model phi-1. We use rotary embedding with rotary dimension 32, and context length 2048. We also use flash-attention [DFE+ 22, Dao23] for training speed up, and we use the tokenizer of codegen-mono [NPH+ 22].



| | Train time (GPU hrs.) | MicroBatch (max) | Inf. speed (per token) | Inf. memory (at 2048 ctx.) | Data size (tokens) | Train tokens |
|---|---|---|---|---|---|---|
| Llama-7B | > 80K | 2 | 14ms | 18G | 1T | 1T |
| **phi-1.5** (1.3B) | 1.5K | 8 | <3ms | 3.5G | 30B | 150B |
| **phi-1.5-web** (1.3B) | 3K | 8 | <3ms | 3.5G | 100B | 300B |

Table 1: Comparison of compute of different models using a single A100-80G with context length 2048 and fp16.

# Phi-1.5: Training Datasets

- (7B tokens) Phi-1’s training data (code)
- (20B tokens) Newly created synthetic, “textbook-like” data for the purpose of teaching common sense reasoning and general knowledge of the world (science, daily activities, theory of mind, etc.).
    - carefully selected 20K topics to seed the generation of this new synthetic data.
    - In our generation prompts, we use samples from web datasets for diversity.
    - We point out that the only non-synthetic part in our training data for phi-1.5 consists of the 6B tokens of filtered code dataset used in phi-1’s training (see [GZA+ 23]).
- (95B tokens) Filtered Web dataset:
    - (88B tokens) Filtered from the Falcon refined web dataset [PMH+ 23]
    - (7B tokens) of code data filtered from The Stack [KLA+ 22] and StackOverflow

# Prompts for Synthetic Datasets

Possible prompts targeting different audiences, the topic and context can be seeded by web data

Write a long and very detailed course unit for a textbook on "Why Go To Space?" intended for **young children.**

We are currently writing the first chapter: "1. Introduction".
You will be writing the first sub-unit for this chapter.
Write the new sub-unit titled "1.1. Why do we spend billions of dollars exploring space?" while trying to be:

…(truncated)

**Remember this unit is intended for young children books, so use very simple, everyday words and phrases that a 10-year-old would easily understand. Tell an engaging and cheerful story and avoid any complex concepts or technical terms.**

Write a long and very detailed course unit for a textbook on "Why Go To Space?" intended for **professionals and researchers in the field.**

We are currently writing the first chapter: "1. Introduction".
You will be writing the first sub-unit for this chapter.
Write the new sub-unit titled "1.1. Why do we spend billions of dollars exploring space?" while trying to be:

…(truncated)

**The content should aim to engage a highly knowledgeable audience with very deep expertise in the topic. Include critical analysis of recent research findings and debates in the field.**

Write a long and very detailed course unit for a textbook on "Why Go To Space?" intended for **high school students.**

We are currently writing the first chapter: "1. Introduction".
You will be writing the first sub-unit for this chapter.
Write the new sub-unit titled "1.1. Why do we spend billions of dollars exploring space?" while trying to be:

…(truncated)

**Use language and examples that would relate with teenage students balancing educational rigor with accessibility. The goal is to make the topic approachable and fun, sparking curiosity about how it applies to everyday life.**

# Phi-1.5: Training Datasets

**Phi-1.5** is trained on

- Phi-1’s training data (20%)
- Newly created synthetic textbook data (80%)

**phi-1.5-web-only (no synthetic)**

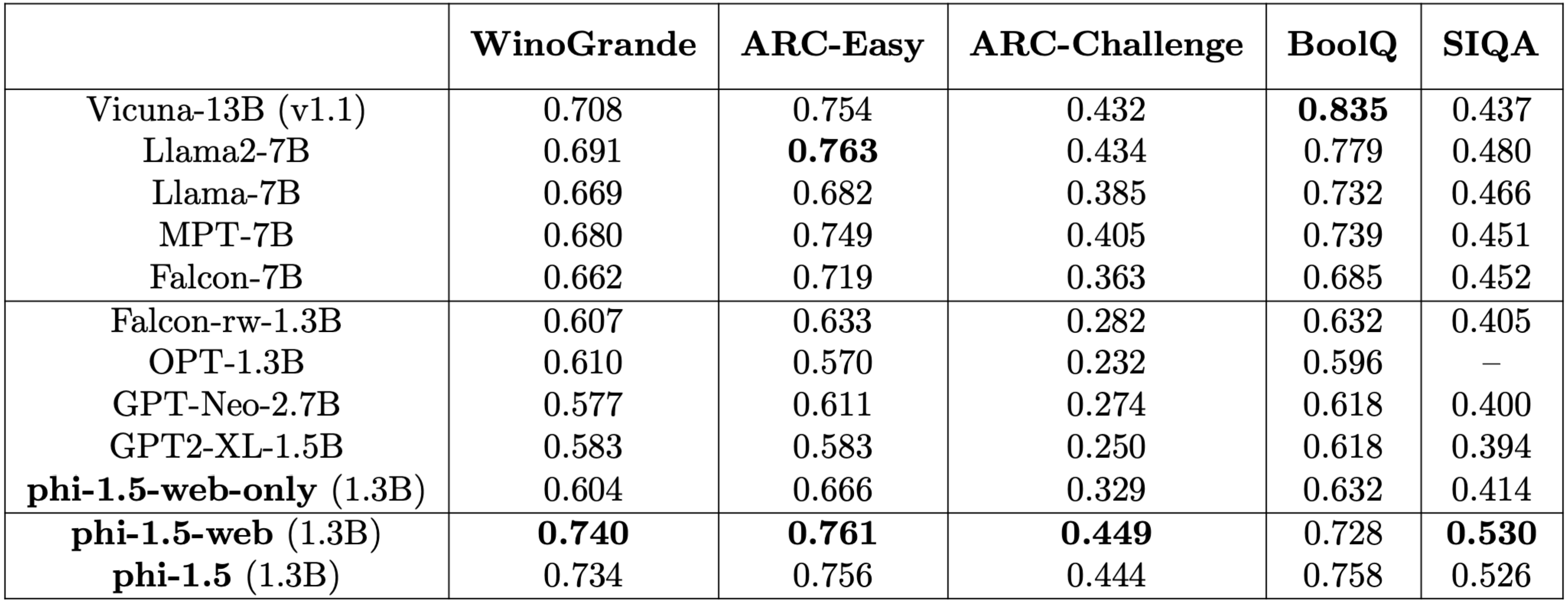
- Filtered web data with about 80% training tokens from NLP data sources and 20% from code datasets (no synthetic data).

**Phi-1.5-web** is trained on

- Phi-1’s training data (20%)
- Filtered web data (40%)
- Newly created synthetic textbook data (40%)

None of these models have undergone alignment finetuning (instruction finetuning or RLHF).

# Common Sense Reasoning

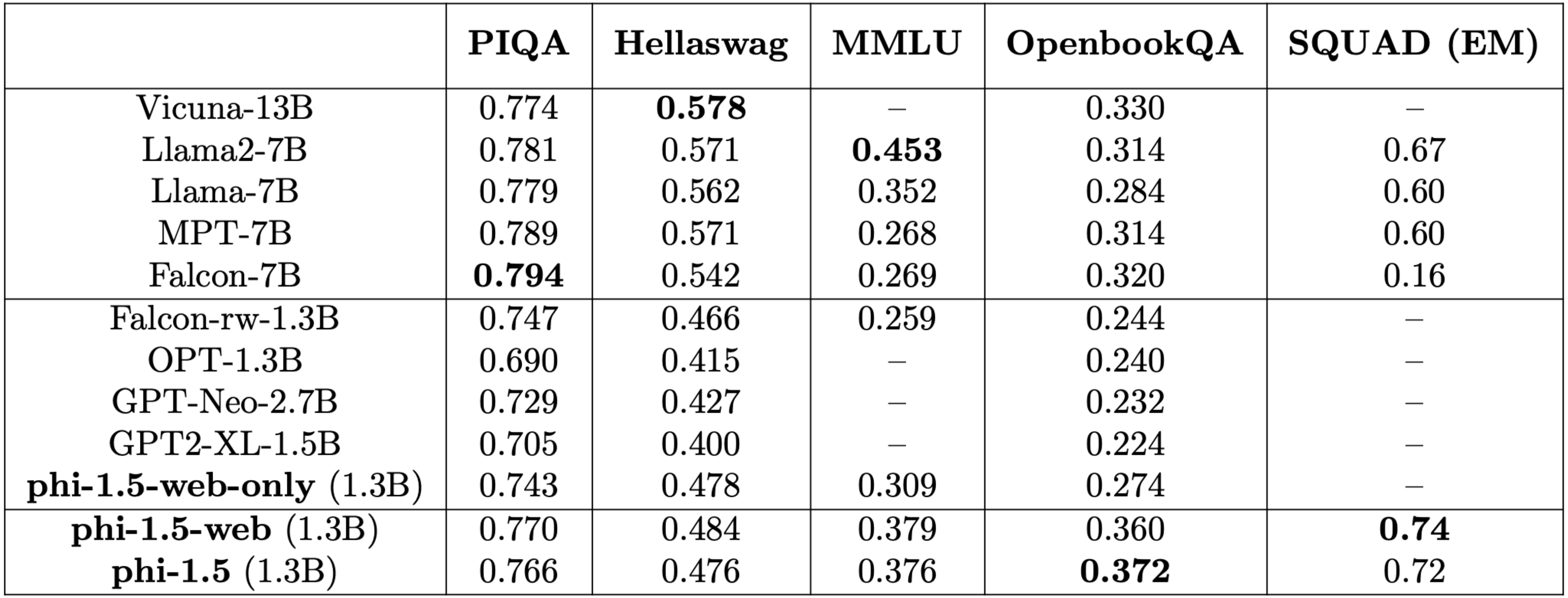


| | **WinoGrande** | **ARC-Easy** | **ARC-Challenge** | **BoolQ** | **SIQA** |
|---|---|---|---|---|---|
| Vicuna-13B (v1.1) | 0.708 | 0.754 | 0.432 | **0.835** | 0.437 |
| Llama2-7B | 0.691 | **0.763** | 0.434 | 0.779 | 0.480 |
| Llama-7B | 0.669 | 0.682 | 0.385 | 0.732 | 0.466 |
| MPT-7B | 0.680 | 0.749 | 0.405 | 0.739 | 0.451 |
| Falcon-7B | 0.662 | 0.719 | 0.363 | 0.685 | 0.452 |
| Falcon-rw-1.3B | 0.607 | 0.633 | 0.282 | 0.632 | 0.405 |
| OPT-1.3B | 0.610 | 0.570 | 0.232 | 0.596 | – |
| GPT-Neo-2.7B | 0.577 | 0.611 | 0.274 | 0.618 | 0.400 |
| GPT2-XL-1.5B | 0.583 | 0.583 | 0.250 | 0.618 | 0.394 |
| **phi-1.5-web-only** (1.3B) | 0.604 | 0.666 | 0.329 | 0.632 | 0.414 |
| **phi-1.5-web** (1.3B) | **0.740** | **0.761** | **0.449** | 0.728 | **0.530** |
| **phi-1.5** (1.3B) | 0.734 | 0.756 | 0.444 | 0.758 | 0.526 |

Table 2: Common Sense Reasoning Benchmarks.

Training along with our synthetic data to get phi-1-web, one can see a large boost in performance, achieving similar performance to models that are 5x larger. Without any web data at all, phi-1.5 is also comparable to all of the other models.

# Language Understanding



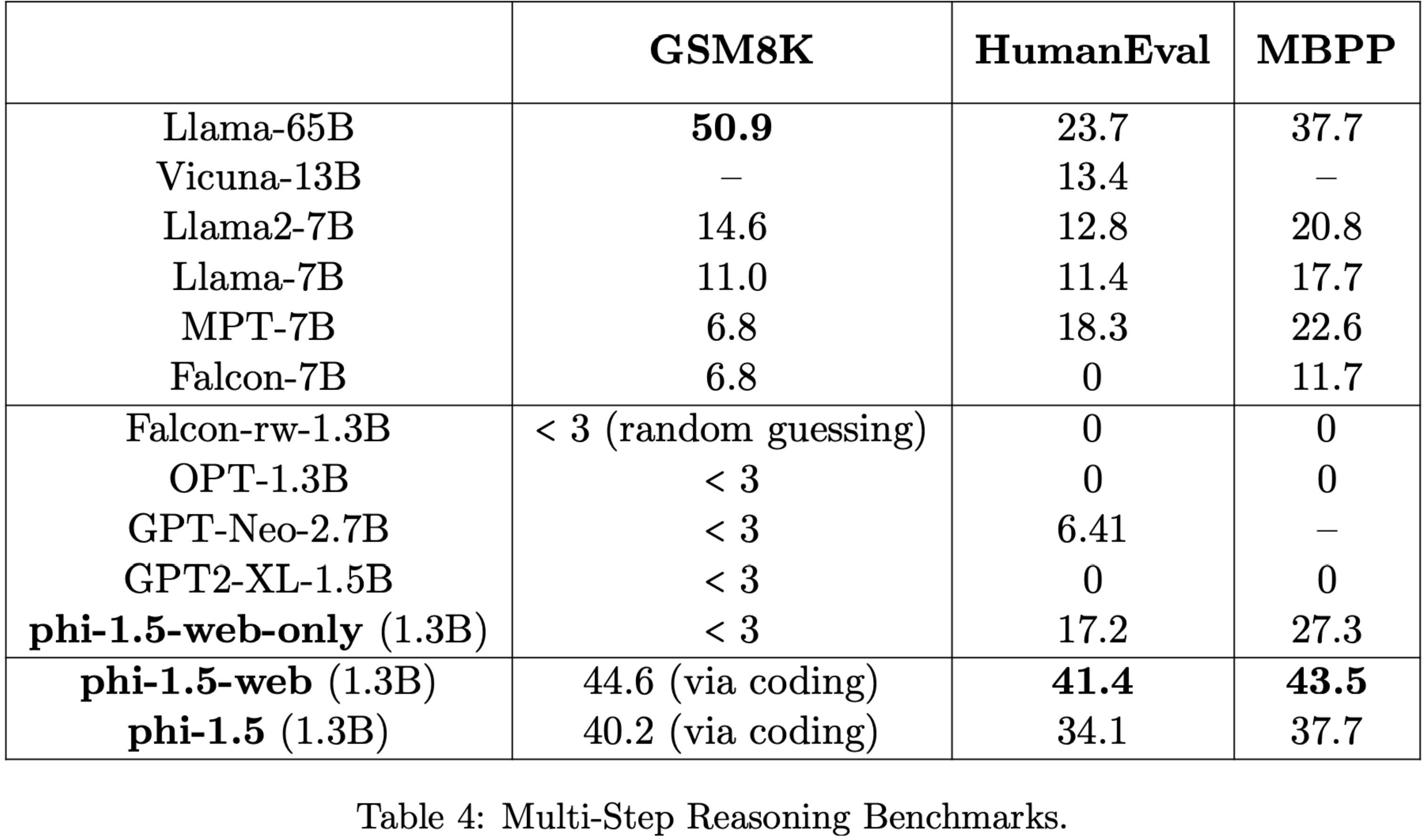
| | **PIQA** | **Hellaswag** | **MMLU** | **OpenbookQA** | **SQUAD (EM)** |
|---|---|---|---|---|---|
| Vicuna-13B | 0.774 | **0.578** | – | 0.330 | – |
| Llama2-7B | 0.781 | 0.571 | **0.453** | 0.314 | 0.67 |
| Llama-7B | 0.779 | 0.562 | 0.352 | 0.284 | 0.60 |
| MPT-7B | 0.789 | 0.571 | 0.268 | 0.314 | 0.60 |
| Falcon-7B | **0.794** | 0.542 | 0.269 | 0.320 | 0.16 |
| Falcon-rw-1.3B | 0.747 | 0.466 | 0.259 | 0.244 | – |
| OPT-1.3B | 0.690 | 0.415 | – | 0.240 | – |
| GPT-Neo-2.7B | 0.729 | 0.427 | – | 0.232 | – |
| GPT2-XL-1.5B | 0.705 | 0.400 | – | 0.224 | – |
| **phi-1.5-web-only** (1.3B) | 0.743 | 0.478 | 0.309 | 0.274 | – |
| **phi-1.5-web** (1.3B) | 0.770 | 0.484 | 0.379 | 0.360 | **0.74** |
| **phi-1.5** (1.3B) | 0.766 | 0.476 | 0.376 | **0.372** | 0.72 |

Table 3: Language Understanding and Knowledge Benchmarks.

# Multi-step Reasoning

elementary school math

coding



| | **GSM8K** | **HumanEval** | **MBPP** |
|---|---|---|---|
| Llama-65B | **50.9** | 23.7 | 37.7 |
| Vicuna-13B | – | 13.4 | – |
| Llama2-7B | 14.6 | 12.8 | 20.8 |
| Llama-7B | 11.0 | 11.4 | 17.7 |
| MPT-7B | 6.8 | 18.3 | 22.6 |
| Falcon-7B | 6.8 | 0 | 11.7 |
| Falcon-rw-1.3B | < 3 (random guessing) | 0 | 0 |
| OPT-1.3B | < 3 | 0 | 0 |
| GPT-Neo-2.7B | < 3 | 6.41 | – |
| GPT2-XL-1.5B | < 3 | 0 | 0 |
| **phi-1.5-web-only** (1.3B) | < 3 | 17.2 | 27.3 |
| **phi-1.5-web** (1.3B) | 44.6 (via coding) | **41.4** | **43.5** |
| **phi-1.5** (1.3B) | 40.2 (via coding) | 34.1 | 37.7 |

Table 4: Multi-Step Reasoning Benchmarks.

zero-shot pass@1 accuracy

We can see that phi-1.5 outperforms all existing models, including Llama 65B on coding tasks. One can also see that the web data does help more here, as phi-1.5-web outperforms phi-1.5 somewhat significantly on those reasoning tasks.

# Phi-2: The surprising power of small language models

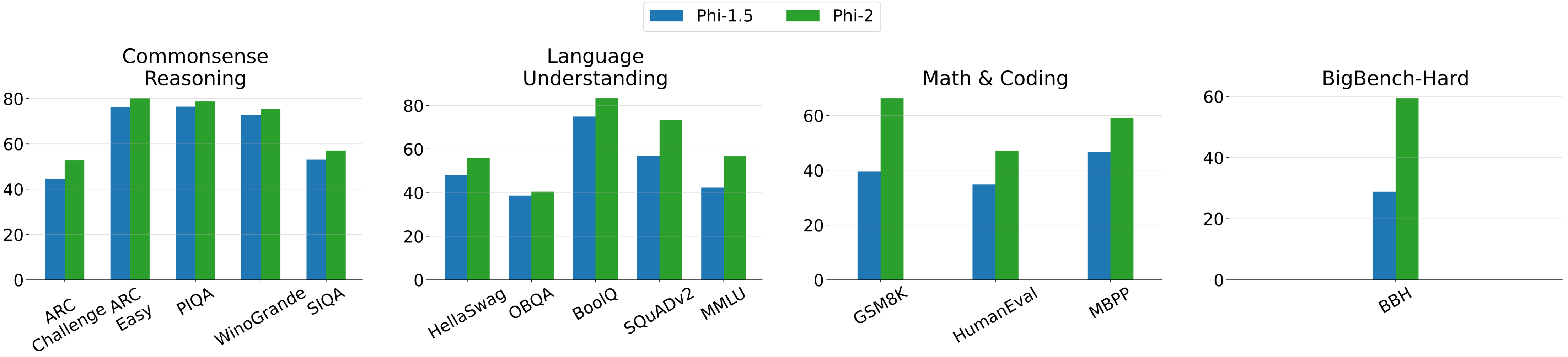
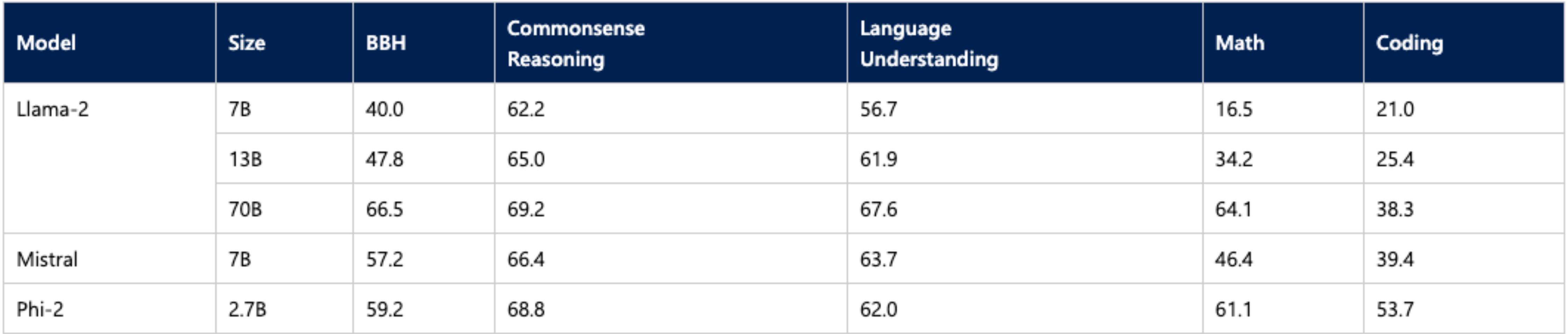


**Figure 2.** Comparison between Phi-2 (2.7B) and Phi-1.5 (1.3B) models. All tasks are evaluated in 0-shot except for BBH and MMLU which use 3-shot CoT and 5-shot, respectively.

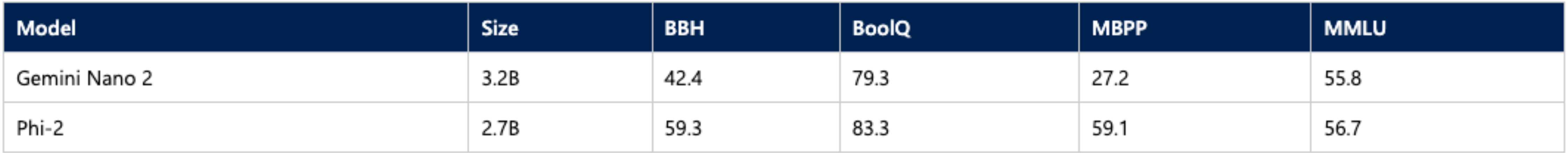
Phi-2 is trained on 1.4T tokens (increased from 30B tokens in Phi-1.5) from multiple passes on a mixture of Synthetic and Web datasets for NLP and coding.

# Phi-2 outperforms Mistral-7B, and the latter outperforms the Llama-2 models (7B, 13B, and 70B)



| Model | Size | BBH | Commonsense Reasoning | Language Understanding | Math | Coding |
|---|---|---|---|---|---|---|
| Llama-2 | 7B | 40.0 | 62.2 | 56.7 | 16.5 | 21.0 |
| | 13B | 47.8 | 65.0 | 61.9 | 34.2 | 25.4 |
| | 70B | 66.5 | 69.2 | 67.6 | 64.1 | 38.3 |
| Mistral | 7B | 57.2 | 66.4 | 63.7 | 46.4 | 39.4 |
| Phi-2 | 2.7B | 59.2 | 68.8 | 62.0 | 61.1 | 53.7 |

**Table 1.** Averaged performance on grouped benchmarks compared to popular open-source SLMs.



| Model | Size | BBH | BoolQ | MBPP | MMLU |
|---|---|---|---|---|---|
| Gemini Nano 2 | 3.2B | 42.4 | 79.3 | 27.2 | 55.8 |
| Phi-2 | 2.7B | 59.3 | 83.3 | 59.1 | 56.7 |

**Table 2.** Comparison between Phi-2 and Gemini Nano 2 Model on Gemini's reported benchmarks.

# Key take-aways

- **Creation of synthetic datasets** will become, in the near future, an important technical skill and a central topic of research in AI.
- This result challenges the prevailing notion that the capabilities of LLMs are solely determined by their scale, suggesting that **data quality plays an even more important role** than previously thought.
- Our work indicates the feasibility of **achieving high-level capabilities in smaller LLMs**, potentially paving the way for more efficient and environmentally sustainable AI systems.
- Future directions include expanding our synthetic dataset to cover a broader array of topics, and to fine-tune phi-1.5/2 for more specific tasks.