



Open and responsible development of
Large Language Models for code

Raymond Li | Research Engineer @ ServiceNow

Today's talk

1 The BigCode Community

2 The Stack

3 StarCoder

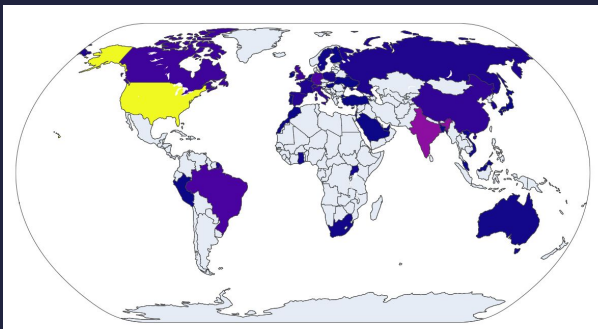
1. The BigCode community



BigCode: open-scientific collaboration

We are building LLMs for code in a collaborative way:

- 500+ participants
- 30+ countries



🌟 STARCODER:

MAY THE SOURCE BE WITH YOU!

Raymond Li² Loubna Ben Allal¹ Yangtian Zi⁴ Niklas Muennighoff¹ Denis Kocetkov²
Chenghao Mou⁵ Marc Marone⁸ Christopher Akiki^{9,10} Jia Li⁵ Jenny Chim¹¹
Qian Liu¹³ Evgenii Zheltonozhskii¹⁴ Terry Yue Zhuo^{15,16} Thomas Wang¹
Olivier Dehaene¹ Mishig Davaadorj¹ Joel Lamy-Poirier² João Monteiro²
Oleh Shliazhko² Nicolas Gontier² Nicholas Meade^{6,17} Armel Randy¹ Ming-Ho Yee⁴
Logesh Kumar Umapathi¹⁸ Jian Zhu¹⁹ Benjamin Lipkin²⁰ Muhtasham Oblokulov²¹
Zhiruo Wang⁷ Rudra Murthy²² Jason Stillerman²³ Siva Sankalp Patel²²
Dmitry Abulkhanov⁵ Marco Zocca³⁴ Manan Dey²⁵ Zhihan Zhang²⁶ Nour Fahmy²⁷
Urvashi Bhattacharyya²⁸ Suriya Gunasekar²⁹ Wenhao Yu²⁶ Swayam Singh³⁰
Sasha Luccioni¹ Paulo Villegas³¹ Maxim Kunakov³² Fedor Zhdanov³²
Manuel Romero⁵ Tony Lee³³ Nadav Timor³⁴ Jen Ding³⁵ Claire Schlesinger⁴
Hailey Schoelkopf³⁷ Jan Ebert³⁸ Tri Dao³³ Mayank Mishra²² Alex Gu²⁰
Jennifer Robinson³ Carolyn Jane Anderson³⁶ Brendan Dolan-Gavitt³⁹
Danish Contractor⁵ Siva Reddy^{2,6} Daniel Fried⁷ Dzmitry Bahdanau² Yacine Jernite¹
Carlos Muñoz Ferrandis¹ Sean Hughes³ Thomas Wolf¹ Arjun Guha^{4,12}
Leandro von Werra^{1,*} Harm de Vries^{2,*}

¹Hugging Face ²ServiceNow Research ³ServiceNow ⁴Northeastern University ⁵Independent
⁶Mila ⁷Carnegie Mellon University ⁸Johns Hopkins University ⁹Leipzig University
¹⁰ScaDS.AI ¹¹Queen Mary University of London ¹²Roblox ¹³Sea AI Lab ¹⁴Technion –
Israel Institute of Technology ¹⁵Monash University ¹⁶CSIRO's Data61 ¹⁷McGill University
¹⁸Saama AI Research Lab ¹⁹University of British Columbia ²⁰MIT ²¹Technical University of
Munich ²²IBM Research ²³University of Vermont ²⁴UnfoldML ²⁵SAP ²⁶University of
Notre Dame ²⁷Columbia University ²⁸Discover Dollar Pvt Ltd ²⁹Microsoft Research
³⁰University of Allahabad ³¹Telefonica I+D ³²Toloka ³³Stanford University ³⁴Weizmann
Institute of Science ³⁵The Alan Turing Institute ³⁶Wellesley College ³⁷Eleuther AI
³⁸Forschungszentrum Jülich ³⁹NYU

Corresponding authors (*) can be contacted at contact@bigcode-project.org

Developing Code LLMs is not only a technical problem!

- Consent of data subjects
 - Do machine learning models comply with copyright laws?
- Privacy risks
 - Scraped data likely contains personal identifiable information
- Software safety and security
 - Code LLMs may be used to generate malware or may provide code suggestions that are less secure

More info on the Governance Card: <https://hf.co/datasets/bigcode/governance-card>

Closed development of LLMs

- Model only available through API, which limits research on:
 - Safety and alignment
 - The model's inner workings (i.e. representations)
 - Adaptation methods like LoRA, and continuous prompt-tuning
- Training data and filtering is not disclosed:
 - Limits scientific reproducibility
 - Potential benchmark contamination
 - People don't know if their data is used, and there's no way to remove it

Open & Responsible Research on LLMs

- Distributing the training data
 - Data licensing needs to permit this
- Allow people to inspect the collected data + opt-out
 - See The Stack
- Transparency builds trust in AI systems:
 - Be open about shortcomings in PII detection models
 - Be open about Code LLM limitations: Model cards

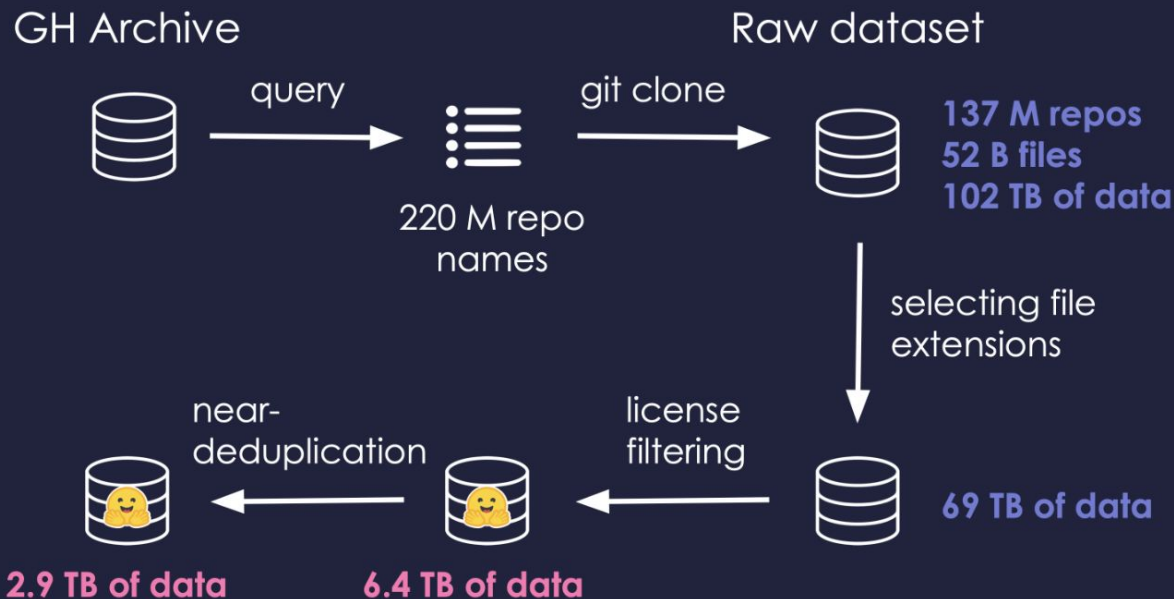
3. The Stack



The Stack

A dataset with **6.4TB** of **permissively licensed** source code in **358 programming languages** with a **data inspection** tool and **opt-out** mechanism

Data Collection



Find the filtered and deduplicated datasets at: www.hf.co/bigcode

Am I In The Stack?



BigCode

The Stack is an open governance interface between the AI community and the open source community.

Am I in The Stack?

As part of the BigCode project, we released and maintain [The Stack](#), a 3.1 TB dataset of permissively licensed source code in 30 programming languages. One of our goals in this project is to give people agency over their source code by letting them decide whether or not it should be used to develop and evaluate machine learning models, as we acknowledge that not all developers may wish to have their data used for that purpose.

This tool lets you check if a repository under a given username is part of The Stack dataset. Would you like to have your data removed from future versions of The Stack? You can opt-out following the instructions [here](#).

The Stack version:

v1.1

Your GitHub username:

Check!

Opt-out

Yes, there is code from 5 repositories in The Stack:

lvwerra/datasets

lvwerra/jupyterplot

lvwerra/pandas-profiling

lvwerra/transformers

lvwerra/trl

Opt-out

If you want your data to be removed from the stack and model training open an issue with [this link](#) (if the link doesn't work try right a right click and open it in a new tab) or visit <https://github.com/bigcode-project/opt-out-v2/issues/new?&template=opt-out-request.md>.

<https://huggingface.co/spaces/bigcode/in-the-stack>



StarCoder

Training Data - The Stack

- **Strong near-deduplication**

- language agnostic and shows consistent performance gains

- **Language selection**

- 86 languages out of 358
- Includes GitHub issues, git commits, structured Jupyter notebooks

- **Data quality inspection**

- inspect 100 samples per source and derive filtering heuristics

- **Removing PII**

- StarPII: Model to detect/remove emails, password and keys

Architecture choices

What do people want from a code model?

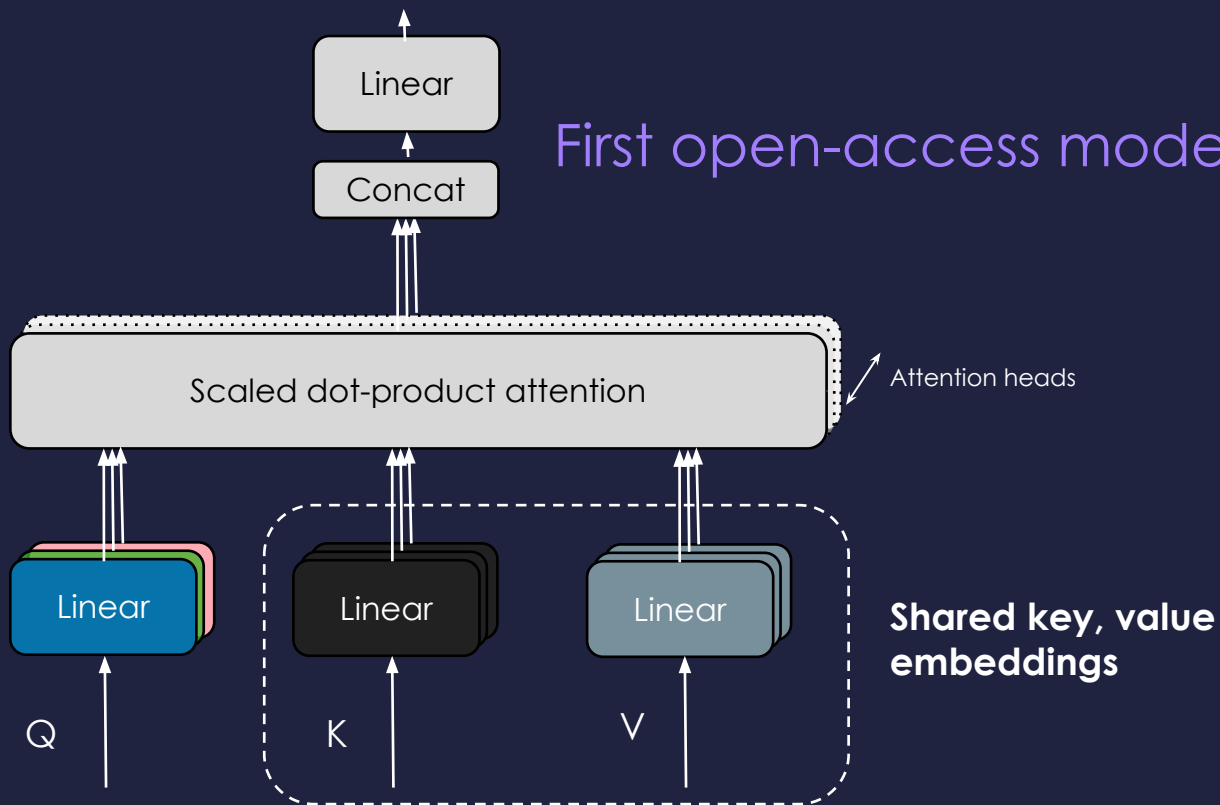
- Fast inference
 - Integrated into an IDE - very low latency
- Cheap generations
 - Generating batches or long sequences requires lot of memory
- Long context
 - Including code from repo can improve performance
- Bi-directional context
 - Code before and after cursor can be important for completion

Architecture choices

What do people want from a code model?

- Fast inference
 - **15B parameters with code optimizations**
- Cheap generations
 - **Multi-Query Attention for reduced memory footprint**
- Long context
 - **Flash Attention to scale to 8,192 tokens context**
- Bi-directional context
 - **Fill-in-the-middle training objective**

Architecture choices: MQA



First open-access model with MQA!

Training setup

Infrastructure: 512 GPUs

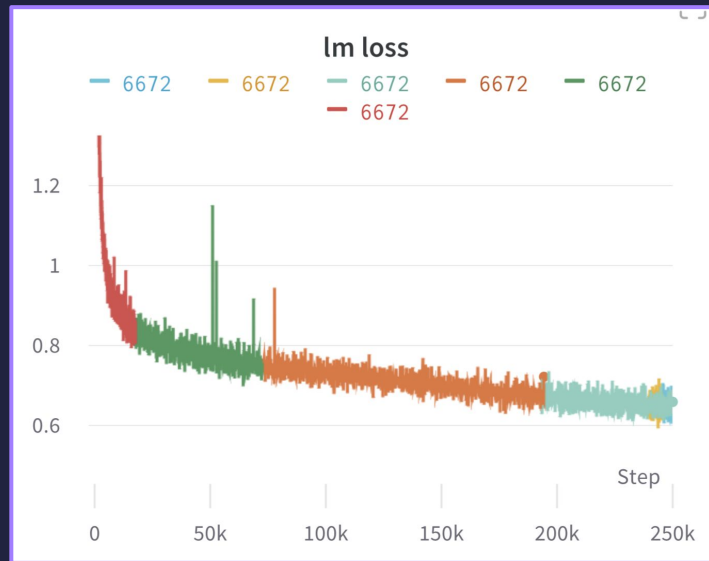
Model Distribution: TP=4, PP=4, DP=32

Batch size: 4M tokens
(or 512 at 8,192 sequence length)

Training length: 1T tokens / 250k steps

Training time: 24 days

Tool: Megatron-LM (w/ MQA + FlashAttn)
(<https://github.com/bigcode-project/Megatron-LM>)



“smooth sailing”



StarCoderBase

Trained on 1T tokens in
80+ languages

StarCoder

Additionally trained on
35B Python tokens

Evaluation - HumanEval/MBPP

Common observed failure mode:

```
# solution here
```

Prompt fix:

```
<filename>solutions/solutions_1.py
```

```
# Here is the correct implementation of the code exercise
```

Didn't observe similar gains for

CodeGen-Mono or StarCoderBase

Model	HumanEval	MBPP
LLaMA-7B	10.5	17.7
LaMDA-137B	14.0	14.8
LLaMA-13B	15.8	22.0
CodeGen-16B-Multi	18.3	20.9
LLaMA-33B	21.7	30.2
CodeGeeX	22.9	24.4
LLaMA-65B	23.7	37.7
PaLM-540B	26.2	36.8
CodeGen-16B-Mono	29.3	35.3
StarCoderBase	30.4	49.0
code-cushman-001	33.5	45.9
StarCoder	33.6	52.7
StarCoder-Prompted	40.8	49.5

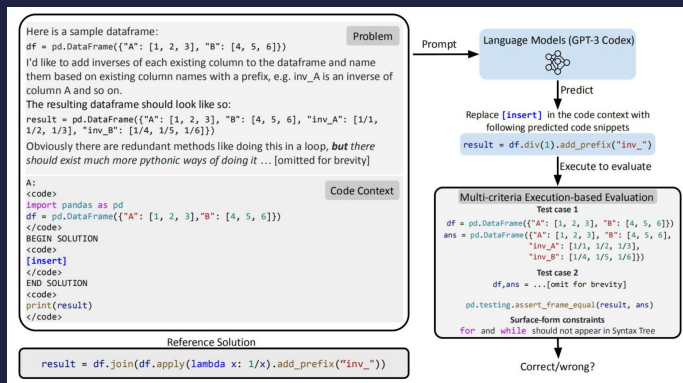
Evaluation - MultiPL-E

MultiPL-E contains translated versions of HumanEval

Language	CodeGen-16B-Multi	CodeGeeX	code-cushman-001	StarCoder	StarCoderBase
cpp	21.00	16.87	30.59	31.55	30.56
c-sharp	8.24	8.49	22.06	21.01	20.56
d	7.68	9.15	6.73	13.57	10.01
go	13.54	11.04	19.68	17.61	21.47
java	22.20	19.14	31.90	30.22	28.53
julia	0.00	0.29	1.54	23.02	21.09
javascript	19.15	16.92	31.27	30.79	31.70
lua	8.50	10.96	26.24	23.89	26.61
php	8.37	13.51	28.94	26.08	26.75
perl	3.42	8.09	19.29	17.34	16.32
python	19.26	21.62	30.71	33.57	30.35
r	6.45	3.92	10.99	15.50	10.18
ruby	0.00	3.34	28.63	1.24	17.25
racket	0.66	3.31	7.05	0.07	11.77
rust	4.21	7.88	25.22	21.84	24.46
scala	2.37	8.95	27.62	27.61	28.79
bash	0.61	2.75	11.74	10.46	11.02
swift	1.25	7.26	22.12	22.74	16.74
typescript	20.07	10.11	31.26	32.29	32.15

- outperforming open-access models
- competitive with code-cushman-001

Evaluation - DS-1000




- benchmark with data science problems
- **StarCoder / StarCoderBase** competitive edge
- fine-tuning improves completion mode

Format	Model	Matplotlib	NumPy	Pandas	PyTorch	SciPy	Scikit-Learn	TensorFlow	Overall
	Number of problems:	155	220	291	68	106	115	45	1,000
Completion	InCoder-6B	28.3	4.4	3.1	4.4	2.8	2.8	3.8	7.4
Completion	CodeGen-16B-Mono	31.7	10.9	3.4	7.0	9.0	10.8	15.2	11.7
Completion	code-cushman-001	40.7	21.8	7.9	12.4	11.3	18.0	12.2	18.1
Completion	StarCoderBase	47.0	27.1	10.1	19.5	21.7	27.0	20.5	23.8
Completion	StarCoder	51.7	29.7	11.4	21.4	20.2	29.5	24.5	26.0
Insertion	InCoder-6B	28.3*	4.6	2.9	4.4	2.8	3.1	7.8	7.5
Insertion	StarCoderBase	47.0*	26.3	10.9	16.6	20.2	30.2	22.3	24.0
Insertion	StarCoder	51.7*	30.8	10.3	21.0	20.2	27.4	20.0	25.4

VSCode extension

Auto-complete



Users > swayam > Desktop >  main.py > ...

```
1 def is_prime(num):  
2     return False
```




```
def is_prime(num):  
    if num == 2:  
        return True  
    if num % 2 == 0:  
        return False  
    for i in range(3, num, 2):  
        if num % i == 0:  
            return False
```

<https://marketplace.visualstudio.com/items?itemName=HuggingFace.huggingface-vscode>

Membership test

Users > swayam > Desktop >  main.py >  is_prime

```
1 def is_prime(num):  
2     return False  
3  
4 def is_prime(num):  
5     if num == 2:  
6         return True  
7     if num % 2 == 0:  
8         return False  
9     for i in range(3, num, 2):  
10        if num % i == 0:  
11        return False
```

 Highlighted code was found in the stack.  

Source: HF Code Autocomplete (Extension)

[Go to stack search](#)

Jupyter extension

The estimated image \tilde{f} using hard thresholding suffers from many artifacts. It is possible to improve the result by using soft thresholding, defined as:

$$\tilde{f} = S_T^1(f) = \sum_m s_T^1(\langle f, \psi_m \rangle) \psi_m$$

where $s_T^1(\alpha) = \max\left(0, 1 - \frac{T}{|\alpha|}\right) \alpha$.

Display the soft thresholding function $s_T^1(\alpha)$.

```
In [1]: import numpy as np
import matplotlib.pyplot as plt

def soft_thresholding(alpha, T):
    return np.maximum(0, 1 - T / np.abs(alpha)) * alpha

def plot_soft_thresholding(T):
    x = np.linspace(-3, 3, 100)
    y = soft_thresholding(x, T)
    plt.plot(x, y)
    plt.title('Soft thresholding function')
    plt.xlabel('alpha')
    plt.ylabel('s_T^1(alpha)')
    plt.show()

plot_soft_thresholding(1)
plot_soft_thresholding(2)
plot_soft_thresholding(3)
```

```
-----
NameError                                 Traceback (most recent call last)
Cell In [1], line 13
    10 plt.ylabel('s_T^1(alpha)')
    11 plt.show()
----> 13 plot_soft_thresholding(1)
    14 plot_soft_thresholding(2)
    15 plot_soft_thresholding(3)

Cell In [1], line 5, in plot_soft_thresholding(T)
----> 4 def plot_soft_thresholding(T):
      5     x = np.linspace(-3, 3, 100)
      6     y = soft_thresholding(x, T)
      7     plt.plot(x, y)

NameError: name 'np' is not defined
```

- The markdown context is passed to the model as well as cell outputs
- Leverages the commit format of starcoder to fix bugs and errors

<https://github.com/bigcode-project/jupytercoder>

Thank you!



Questions?

www.bigcode-project.org

hf.co/bigcode

Feedback from the opt-out form

- *“It should be opt-in instead of opt-out”*
- *“It is unfair to use my code without compensation”*
- *“There’s PII in my code and I don’t want it to be publicly exposed”*
- *“My code is of poor quality and unsuitable for training your AI model”*
- *“I am not confident about the current state of AI code generation. I am concerned that the generated code could be traced back to me and I’m held liable for issues in that code.”*

Jennifer Ding’s community research: it’s both **better to know** AND **better to have a choice**.

Training Data - Formatting

Code

```
<reponame>REPONAME<filename>FILENAME<gh_stars>STARS\nCode<eos>
```

Git commits

```
<commit_before>code<commit_msg>text<commit_after>code<eos>
```

Jupyter Notebooks

```
<jupyter_start><jupyter_text>TEXT<jupyter_code>CODE<jupyter_output>  
OUTPUT<jupyter_text> ...
```

GitHub Issues

```
<issue_start>title + USERID: comment<issue_comment>USERID: Comment  
... <issue_closed (optional)> <eos>
```

Evaluating Infilling

- StarCoder supports infilling or fill-in-the-middle: condition generation on code before *and after* the insertion point
- One of a handful of recent open models that support infilling
- Applications
 - Docstring generation
 - Type prediction
 - Many more
- StarCoder outperforms other infilling models significantly on three infilling tasks

Model	Java	JavaScript	Python
InCoder-6B	0.49	0.51	0.31
SantaCoder	0.62	0.60	0.44
StarCoder	0.73	0.74	0.62

Single-line code completion for three languages
(SantaCoder/InCoder benchmarks)

Model	Non-None F1	All F1
InCoder-6B	59.1	46.8
SantaCoder	66.9	78.5
StarCoderBase	77.4	86.6
StarCoder	77.1	86.4

Python return-type prediction
(InCoder/TypeWriter benchmarks)

	Packages type check		
	✓	Total	%
InCoder	30	128	23.4
StarCoderBase	49	128	38.3

TypeScript type inference
(TypeWeaver benchmarks)

Perplexity with long contexts

Window Size	Language									
	cpp	c-sharp	c	go	java	javascript	php	r	ruby	rust
2K tokens	2.01	1.90	1.71	1.35	1.65	1.98	1.73	1.72	2.16	1.84
8K tokens	1.79	1.66	1.61	1.21	1.54	1.68	1.43	1.48	2.02	1.65

- Derived test data from GPL repositories on GitHub. GPL was excluded from training data.
- Demonstrates StarCoder can benefit from information within long files or repositories.
- Longer contexts provides noticeable decreases in perplexity.

GSM8K

Problem: Beth bakes 4, 2 dozen batches of cookies in a week. If these cookies are shared amongst 16 people equally, how many cookies does each person consume?

Solution: Beth bakes 4 2 dozen batches of cookies for a total of $4*2 = \ll 4*2=8 \gg 8$ dozen cookies

There are 12 cookies in a dozen and she makes 8 dozen cookies for a total of $12*8 = \ll 12*8=96 \gg 96$ cookies

She splits the 96 cookies equally amongst 16 people so they each eat $96/16 = \ll 96/16=6 \gg 6$ cookies

Final Answer: 6

Model	Size	GSM8K CoT	+maj1@100	GSM8K PAL	+maj1@40
StarCoderBase	15.5B	8.4	—	21.5	31.2
CodeGen-Multi	16B	3.18	—	8.6	15.2
CodeGen-Mono	16B	2.6	—	13.1	22.4
LLaMA	7B	11.0	18.1	10.5	16.8
	13B	17.8	29.3	16.9	28.5
	33B	35.6	53.1	38.7	50.3
	65B	50.9	69.7	—	—

- StarCoderBase performs better with PAL than with CoT
- Outperforms CodeGen-16B and LLaMA-13B

MMLU

Multiple-choice questions in
57 knowledge domains

Model	Size	MMLU 5-shot acc, %
CodeGen-Multi	16B	27.8
GPT-NeoX	20B	32.9
StarCoder	15.5B	33.9
StarCoderBase	15.5B	34.2
LLaMA	7B	35.1
LLaMA	13B	46.9

CoQA

Conversational question answering
on diverse text passages

Model	Size	CoQA zero-shot F1 score
CodeGen-Multi	16B	0.59
StarCoderBase	15.5B	0.67
StarCoder	15.5B	0.67
LLaMA	7B	0.71
LLaMA	13B	0.73
GPT-NeoX	20B	0.73

- StarCoder models significantly outperform other open code LLMs but still fall short of text-only LLMs like LLaMa

Harmful generations

StereoSet:

3 sentence completions (stereotypical, anti-stereotypical, unrelated), which one did the model prefer?

Model	Stereotype Score	Language Model Score	ICAT Score
LLaMA-13B	63.40	87.62	64.14
CodeGen-Multi-16B	61.29	87.25	67.55
StarCoderBase	55.53	86.18	76.65

- StarCoder is less Stereotypical (overall) than LLaMa and CodeGen
- With slightly lower Language Model Score
- On average (ICAT) StarCoder is better than both LLaMa and CodeGen

RealToxicityPrompts:

Given toxic prompts (10k), how much toxicity the model generates?

Model	Classifier	Word List
LLaMA-13B	0.74	1.43
CodeGen-Multi-16B	0.21	0.82
StarCoderBase	0.42	1.12

- **Classifier:** percentage of responses with toxic score > 0.5
- **WordList:** percentage of responses with 1+ offensive word
- StarCoder generates less toxic content than LLaMa, but little more than CodeGen

Reasoning tasks in HELM

Model	Size	Open Access	Synth. Reason. (AS)	Synth. Reason. (NL)	bAbI	Dyck	GSM8K	MATH	MATH (CoT)	LSAT	Legal Support
code-davinci-002	175B		54.0	68.4	68.6	80.5	56.8	41.0	43.3	—	—
text-davinci-003	175B		50.2	73.4	65.3	75.1	50.6	39.0	44.9	23.3	62.2
Luminous Supreme	70B		31.2	—	50.4	72.9	11.2	14.9	5.7	21.2	53.0
StarCoderBase	15.5B	✓	44.0	21.0	50.4	85.4	8.4	15.1	7.0	19.0	53.2
Cohere Command Beta	52.4B		24.3	24.5	47.3	42.1	13.8	13.3	7.5	22.9	60.6
J1-Jumbo v1	178B		26.3	17.4	54.3	44.5	5.4	8.9	3.3	23.2	48.4
J1-Grande v2 beta	17B		28.6	13.9	47.0	61.7	9.6	12.7	6.8	19.1	56.2
code-cushman-001	12B		34.1	16.4	48.1	45.1	4.9	9.9	7.2	—	—
OPT	175B	✓	22.5	24.8	50.7	49.4	4.0	6.5	2.6	22.0	53.2
GPT-NeoX	20B	✓	20.4	16.7	46.8	74.7	5.3	14.1	7.1	19.1	51.5
BLOOM	176B	✓	30.4	19.7	44.7	54.5	9.5	4.3	5.5	20.9	54.3
GLM	130B	✓	25.2	25.4	44.3	54.9	6.1	0	5.9	19.3	45.1
UL2	20B	✓	20.5	21.7	50.1	14.0	2.4	0	0	20.7	50.6
OPT	66B	✓	19.3	21.3	40.8	47.1	1.8	4.8	2.9	17.5	52.7
YaLM	100B	✓	5.6	6.1	34.6	63.3	0	0	0	2.3	48.4
T5	11B	✓	19.6	10.1	41.2	34.7	2.3	0	0	15.9	55.8

- StarCoderBase substantially outperforms other open LLMs
- State-of-the-art performance on Dyck benchmark



Prompting StarCoder

```
1 Below are a series of dialogues between various people and
  an AI technical assistant. The assistant tries to be
  helpful, polite, honest, sophisticated, emotionally aware,
  and humble-but-knowledgeable. The assistant is happy to help
  with code questions, and will do its best to understand
  exactly what is needed. It also tries to avoid giving false
  or misleading information, and it caveats when it isn't
  entirely sure about the right answer. That said, the
  assistant is practical and really does its best, and doesn't
  let caution get too much in the way of being useful.
2
3 -----
4
5 Human: Write a function that takes two lists and returns a
  list that has alternating elements from each input list.
6
7 Assistant: Sure. Here is a function that does that.
8
9 ....
10 def alternating(list1, list2):
11     results = []
12     for i in range(len(list1)):
13         results.append(list1[i])
14         results.append(list2[i])
15     return results
16 ....
17
18 Human: Can you write some test cases for this function?
19
20 Assistant: Sure, here are some tests.
21
22 ....
23 assert alternating([10, 20, 30], [1, 2, 3]) == [10, 1, 20,
24 2, 30, 3]
25 assert alternating([True, False], [4, 5]) == [True, 4,
26 False, 5]
27 assert alternating([], []) == []
28 ....
29
30 Human: Modify the function so that it returns all input
  elements when the lists have uneven length. The elements
  from the longer list should be at the end.
31
32 Assistant: Here is the modified function.
```

excerpt from TA prompt

Inspired by Anthropic's HHH prompt: the TA prompt can make the model act as a tech assistant!

Can you give me Python code that tells me the weather in the city Bern?

Sure. Here is a simple script that uses the OpenWeatherMap API to retrieve data about the weather forecast in Bern.

```
import requests
import json
api_key = "<KEY>"
url = "http://api.openweathermap.org/data/2.5/forecast"
params = {"q": "Bern", "units": "metric", "APPID": api_key }
response = requests.get(url, params=params)
json_data = response.json()
print("Temperature:", round((json_data["main"]["temp"] - 273.15)))
print("Wind speed:", round(json_data["wind"]["speed"]))
print("Humidity:", round(json_data["main"]["humidity"], 1))
```

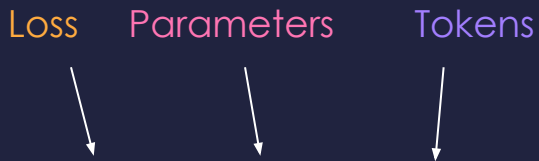
CodeML OpenRAIL-M v1

- Royalty free access and use of the model
- Enables downstream distribution of derivatives
- Includes ethical use restrictions based on technical capabilities of the Model (e.g malware generation)
- Improved OpenRAIL license for commercialization

Chinchilla Scaling Laws

[[Hoffman et al](#)]

Loss Parameters Tokens


$$L(N, D) = E + \frac{A}{N^\alpha} + \frac{B}{D^\beta}$$

$$E = 1.69, A = 406.4, B = 410.7, \alpha = 0.32, \beta = 0.28.$$

Compute-optimal Parameters and Tokens

$$C = 6ND$$

+ set the partial
derivative of L with
respect to $N/D = 0$

$$N_{opt}(C) = G \left(\frac{C}{6} \right)^{\frac{\beta}{\alpha+\beta}}, D_{opt}(C) = G^{-1} \left(\frac{C}{6} \right)^{\frac{\alpha}{\alpha+\beta}}, G = \left(\frac{\alpha A}{\beta B} \right)^{\frac{1}{\alpha+\beta}}$$

Compute vs model-size trade-off

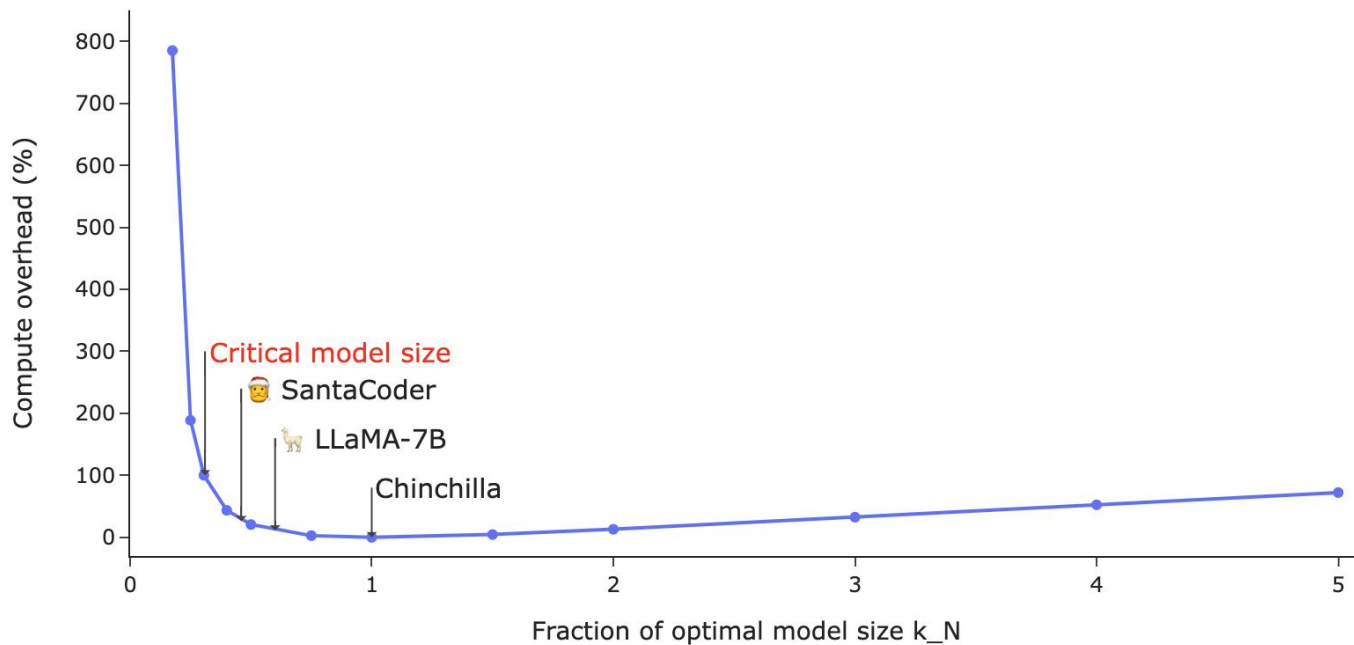
$$L(N_{opt}, D_{opt}) = L(k_N N_{opt}, k_D D_{opt})$$

with some math:

$$k_D = \left(1 - (k_N^{-\alpha} - 1) \frac{AN_{opt}^{-\alpha}}{BD_{opt}^{-\beta}} \right)^{\frac{1}{-\beta}}$$

$$C_{new} = 6(k_N N_{opt})(k_D D_{opt})$$

The compute vs model-size curve



Updated Chinchilla table

[Blogpost](#)

	$k_N = 1$		$k_N = 0.5$		$k_N = 0.3$	
C	N	D	N	D	N	D
2.21e+19	0.40 B	9.22 B	0.20 B	22.28 B	0.12 B	63.20 B
1.62e+20	0.99 B	27.20 B	0.50 B	65.70 B	0.30 B	186.35 B
2.46e+22	9.87 B	415.53 B	4.93 B	1003.77 B	2.96 B	2847.27 B
1e+23	18.73 B	889.63 B	9.37 B	2149.02 B	5.62 B	6095.86 B
1.71e+24	68.60 B	4154.24 B	34.30 B	10035.16 B	20.58 B	28465.50 B

Next steps



StarCoder+

- ★ StarCoder+: StarCoder fine-tuned on **600B** tokens from the Falcon English Web dataset & the Stack
- Strong natural language performance and coding capabilities
- **StarChat-Beta**: StarCoder+ instruction-tuned

Model	MMLU	ARC	HellaSwag	TruthfulQA	HumanEval
Falcon-7B	35	47.9	78.1	34.3	-
LLaMA-7B	34.2	46.6	75.6	34.1	10.5
LLaMA-13B	37.7	50.8	78.9	39.9	15.8
Stable-Vicuna	38.8	48.1	76.4	46.5	-
Vicuna 13B 1.1	39.6	47.4	78	49.8	-
StarCoderPlus	45.1	48.9	77.3	37.9	26.9
StarChat-Beta	44.7	52	80.6	47.2	26.7

<https://hf.co/bigcode/starcoderplus>

StarChat Beta

Chat

Create a simple HTML website for a fictional company called "TechCoAI". Please make an HTML file that adheres to the following specifications:

- The webpage should have a title that says "TechCoAI - Welcome".
- The background color of the webpage should be light blue (#E6F1F6).
- The website should have a navigation menu with three links: "Home" "About" and "Contact ", these links should be displayed in dark blue.
- Include a paragraph of text that describes TechCoAI's mission and vision: " The mission of TechCoAI is to build the next generation of AI models with responsible practices."
- Finally, create a footer with the copyright information.
- The footer should display the copyright symbol (©), the current year 2023, and "TechCoAI. All rights reserved."
- Remember to follow these instructions precisely and ensure that the HTML code is properly structured and formatted.

```
<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
```



Home About Us Contact us

Welcome To Tech Co Ai

We are building the Next Generation Of AI Models With Responsible Practices.

© 2023 TechCoAi | All Rights Reserved

<https://huggingface.co/spaces/HuggingFaceH4/starchat-playground>