StarCoder

Reaching for the Stars

Alan Zhang and Michael Thede

Introduction

Most well-performing models are closed-access, leading to concerns of privacy, copyright, and openness for LLMs and their training datasets.

Enter StarCoder!

- Many different language and sources represented.
- Special architectural features.

Goals:

- Copyright-free, PII-free, public dataset with opt-out process
- Open source model with great performance
- Tools to detect generations that copied from training set

Data Curation

01
Languages
Selection from all available languages found

02
Quality Control
Visual filtering from random selection

03
Automatic Filters
Avoid data files, generated files, XML/HTML boilerplate

04
Jupyter
Converted notebooks to scripts and structured Python/Markdown

05
Github
Issues, Pull Requests, and Commits cleaned and anonymized

06
Deduplication
Removed similar code files via MinHashes/LSH

Personal Identifiable Information (PII) Detection

Goal: Detect PII automatically by building PII-filtering model

- Utilized crowd-workers to annotate PII.
- Avoided PII types with high annotation error rate
- Transformer-based encoder finetuned on this dataset
- Pseudo-labeling to improve password and key detection

Github only required names to redact, and this setup worked for all other styles of PII.

PII type	Count	Recall	Precision
IP_ADDRESS	2526	85%	97%
KEY	308	91%	78%
PASSWORD	598	91%	86%
ID	1702	53%	51%
EMAIL	5470	99%	97%
EMAIL_EXAMPLE	1407		
EMAIL_LICENSE	3141		
NAME	2477	89%	94%
NAME_EXAMPLE	318		
NAME_LICENSE	3105		
USERNAME	780	74%	86%
USERNAME_EXAMPLE	328		
USERNAME_LICENSE	503		
AMBIGUOUS	287		

Model Creation - Models

StarCoderBase

Trained on base dataset
1 Trillion tokens

StarCoder

Fine-tuned on additional
35 Billion Python tokens

Model Creation - Data formatting

Code metadata formatted, randomly prepended

Fill-in-the-middle transformations (FIM) to allow filling in context in the middle of code

Formatted Github Issues, Jupyter structured notebooks, and Git Commits structurally

Answers and docstrings to AI benchmarks filtered out

Model Creation - Model

15.5B parameters, Decoder-only Transformer with Multi-Query Attention

Used FlashAttention to reduce memory footprint and speed up computation

250k iterations with batch size of 4M = 1 trillion tokens trained on, using Adam optimizer.

StarCoder — finetuned on 2 epochs on Python subset = 35B tokens extra

Hyperparameter	SantaCoder	StarCoder
Hidden size	2048	6144
Intermediate size	8192	24576
Max. position embeddings	2048	8192
Num. of attention heads	16	48
Num. of hidden layers	24	40
Attention	Multi-query	Multi-query
Num. of parameters	≈ 1.1B	≈15.5B

Model Creation - Training Data

512 A100 GPUs over 64 nodes, combined in a GPU cluster from AWS

Used both tensor and pipeline parallelism, with 16 GPUs per replica

TL;DR Lots of parallelism and optimization across GPUs

Carbon footprint estimated to be 16.68 tonnes of CO2eq emitted from 320,256 GPU hours at 280W per GPU.

Model	Size	HumanEval	MBPP
Open-access			
LLaMA	7B	10.5	17.7
LLaMA	13B	15.8	22.0
SantaCoder	1.1B	18.0	35.0
CodeGen-Multi	16B	18.3	20.9
LLaMA	33B	21.7	30.2
CodeGeeX	13B	22.9	24.4
LLaMA-65B	65B	23.7	37.7
CodeGen-Mono	16B	29.3	35.3
StarCoderBase	15.5B	30.4	49.0
StarCoder	15.5B	33.6	52.7
Closed-access			
LaMDA	137B	14.0	14.8
PaLM	540B	26.2	36.8
code-cushman-001	12B	33.5	45.9
code-davinci-002	175B	45.9	60.3

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

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**	—	—

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**

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**

Big Picture

Why Care?

Positives

Openness — open community focused on responsible development and use, allowing for the model creation process to be learned from

Performance — StarCoder's performance can be used as a benchmark for future AI models, and it could be run locally on a (somewhat powerful) PC

Limitations — Shortcomings of the model can be (and has been) researched and improved upon

Dataset — The Stack, and their process of cleaning up the dataset, are available and explained in detail

Considerations

Openness — higher risk of misuse without appropriate guardrails

Limitations — StarCoder's limitations, like any LLM, include the potential to generate harmful content, hallucinations, and generally incorrect/useless data

Dataset — The Stack's automatic filtering may not be perfect for code licensing, PII, and malicious code. Additionally, interest for an opt-in process instead of opt-out has been shown.

Recent Updates

StarCoder 2 and **The Stack v2** have been released last month, with model changes and performance improvements.

As of now, there are several open-source Code LLMs with similar, if not better, performance than StarCoder 2, including **DeepSeek Coder** and **Dolphin Mistral**.

GPT-4 powered multi-model code generation models rank highest in code generation benchmarks, and there is still a gap with

Many models use **The Stack** to train and research new improvements in the LLM field, and build off of **StarCoder**, as the original paper has been cited >250 times.

Questions?

References

- https://doi.org/10.48550/arXiv.2305.06161
- https://doi.org/10.48550/arXiv.2402.19173
- https://huggingface.co/spaces/bigcode/bigcode-models-leaderboard
- https://huggingface.co/blog/starcoder