How to Use Open-source LLMs with Python

Presenters: Robert Tang

ISMB 2024 · JULY 12-16 · MONTREAL CANADA

Yale

NIH National Library of Medicine
National Center for Biotechnology Information

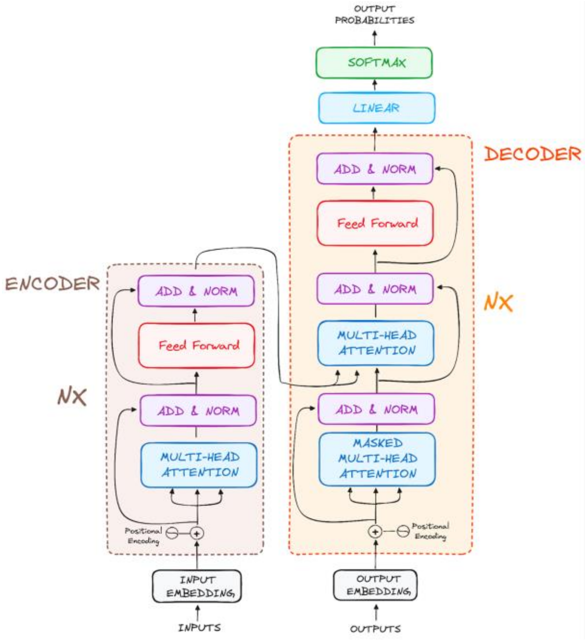
Open-source LLMs

Presenters: Robert Tang, Tom Qiu

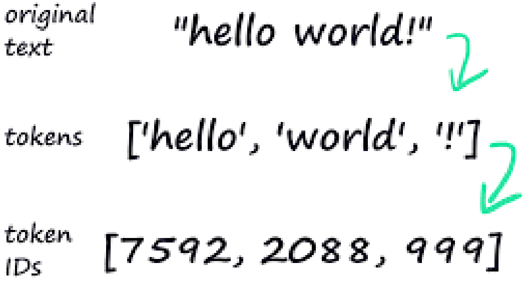
Learning Objectives of the session

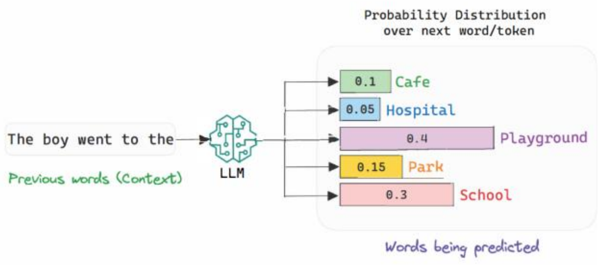
- Why open source models
- More specifics on 3 such models
  - Llama3
  - Mistral
  - Deepseek
- Performance comparison
- Methods to improve accuracy of such LLMs for certain tasks (e.g. prompting techniques)
- Walkthrough of Llama 3 in Python with Google Colab

An Overview of Transformers



1. Encoding
2. Attention
3. Multilayer perceptron
4. Repeat
5. Decoding
6. Probability distribution



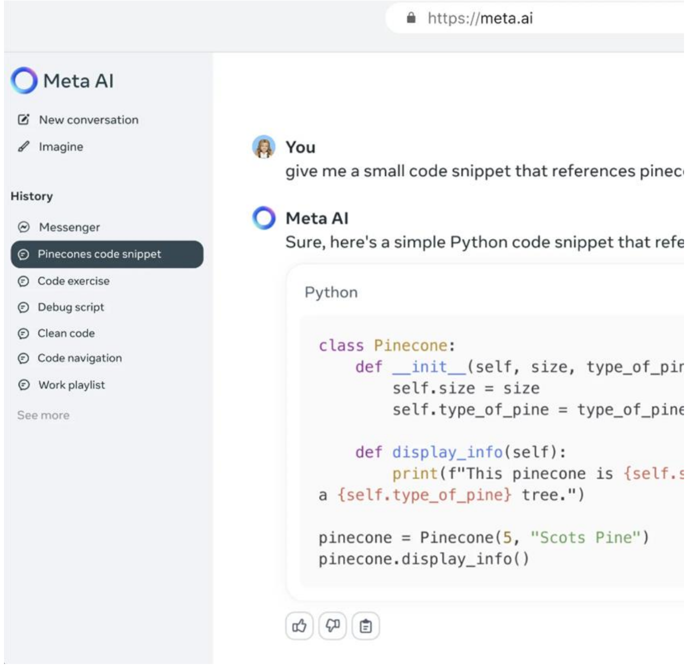


[Why Are There So Many Tokenization Methods For Transformers?]()
[How Transformers Work: A Detailed Exploration of Transformer Architecture]()
[How do Language models(LLM) work ?]()

Open-Source LLMs

1. Transparency
   a. Make powerful AI tools available to a broader audience
   b. Ethical development process
2. Collaboration
   a. AI community can collectively contribute
3. Customization
   a. Allow developers to customize and fine-tune models to suit specific needs and applications
4. Cost
   a. Proprietary models like GPT-4 often come with significant costs

Llama3



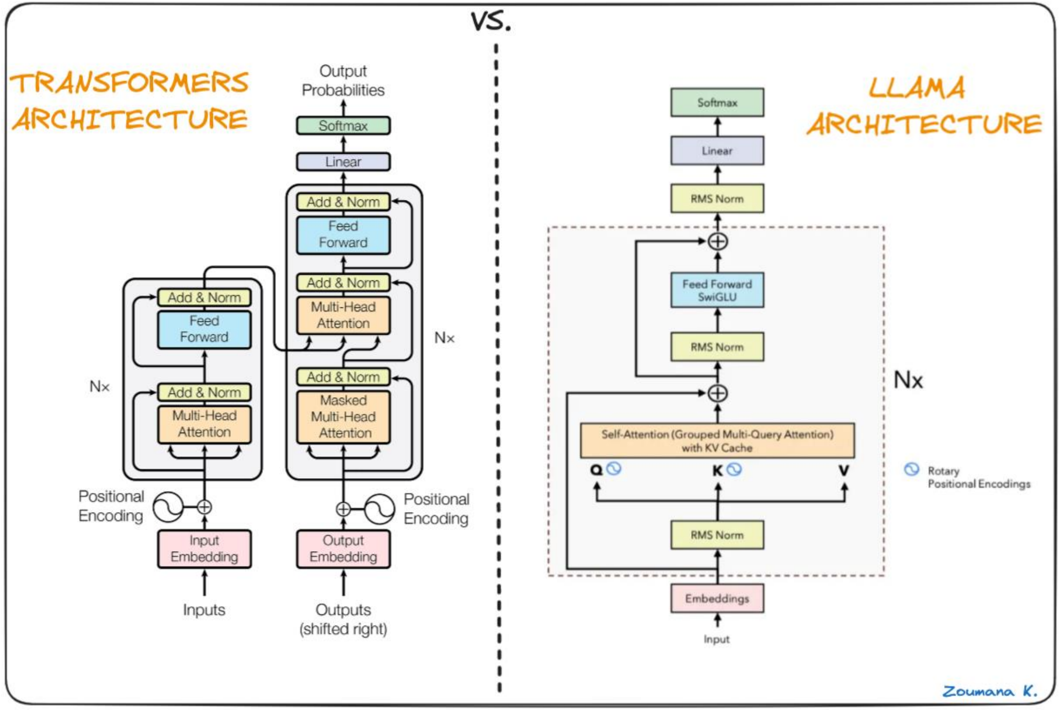
https://llama.meta.com/llama3/

“With Llama 3, we set out to build the best open models that are on par with the best proprietary models available today. We wanted to address developer feedback to increase the overall helpfulness of Llama 3 and are doing so while continuing to play a leading role on responsible use and deployment of LLMs.”

Introduced April 2024.

“Our new 8B and 70B parameter Llama 3 models … establish a new state-of-the-art for LLM models at those scales … [and] are the best models existing today at the 8B and 70B parameter scale.”

Llama3

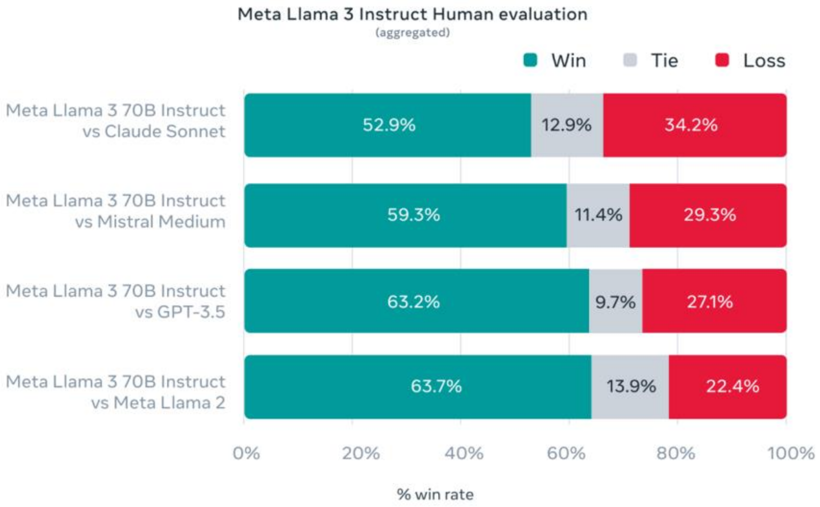


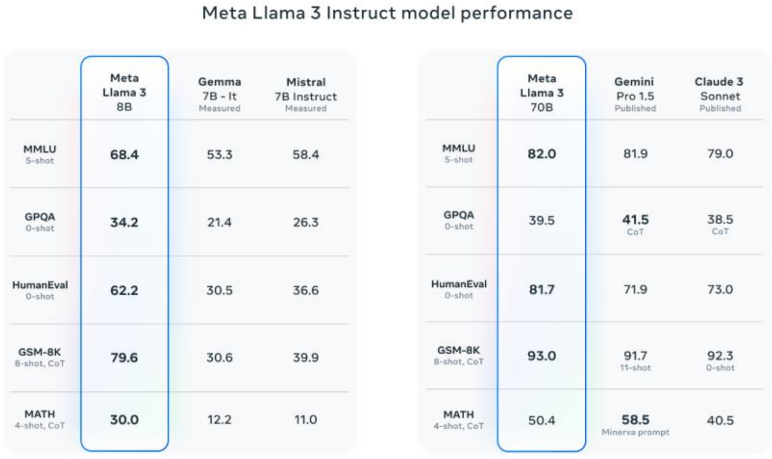
Significant improvement over Llama 2

- Improvement on benchmarks
- Larger token vocabulary
- Increased context length (8k tokens)

LLaMA explained!

Llama3





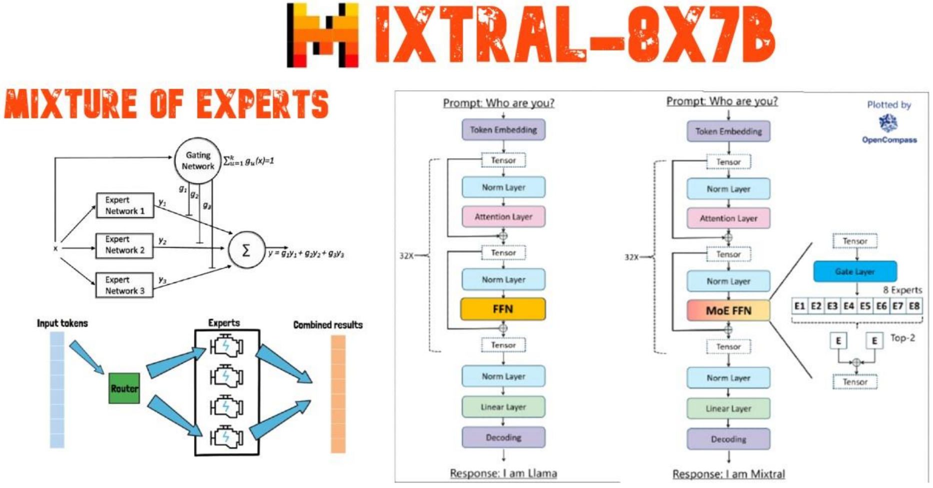
Meta Llama 3 Instruct model performance

	Meta Llama 3 8B	Gemma 7B - It (Measured)	Mistral 7B Instruct (Measured)
MMLU 5-shot	**68.4**	53.3	58.4
GPQA 0-shot	**34.2**	21.4	26.3
HumanEval 0-shot	**62.2**	30.5	36.6
GSM-8K 8-shot, CoT	**79.6**	30.6	39.9
MATH 4-shot, CoT	**30.0**	12.2	11.0

	Meta Llama 3 70B	Gemini Pro 1.5 (Published)	Claude 3 Sonnet (Published)
MMLU 5-shot	**82.0**	81.9	79.0
GPQA 0-shot	39.5	**41.5** CoT	38.5 CoT
HumanEval 0-shot	**81.7**	71.9	73.0
GSM-8K 8-shot, CoT	**93.0**	91.7 11-shot	92.3 0-shot
MATH 4-shot, CoT	50.4	**58.5** Minerva prompt	40.5

https://ai.meta.com/blog/meta-llama-3/

Mistral AI



Tuning.AI

Mistral AI is a French company selling artificial intelligence (AI) products. It was founded in April 2023 by previous employees of Meta Platforms and Google DeepMind.

“Our mission is to make frontier AI ubiquitous, and to provide tailor-made AI to all the builders. This requires fierce independence, strong commitment to open, portable and customisable solutions, and an extreme focus on shipping the most advanced technology in limited time.”

[Understanding Mixtral-8x7B: A Dive into Mixture of Experts (MoE) Architecture]

Mistral AI

Open source models

		Input	Output
open-mistral-7b	A 7B transformer model, fast-deployed and easily customisable.	$0.25 /1M tokens	$0.25 /1M tokens
open-mixtral-8x7b	A 7B sparse Mixture-of-Experts (SMoE). Uses 12.9B active parameters out of 45B total.	$0.7 /1M tokens	$0.7 /1M tokens
open-mixtral-8x22b	Mixtral 8x22B is currently the most performant open model. A 22B sparse Mixture-of-Experts (SMoE). Uses only 39B active parameters out of 141B.	$2 /1M tokens	$6 /1M tokens

Mixture-of-Experts architecture, extending context length up to 65K tokens for its open-mixtral-8x22b model (larger than many models, including GPT-4's 32K tokens)

https://mistral.ai/technology/

Mistral AI

Automate large scale text generation & processing

Use our Mistral models to process, summarize, classify or translate any kind of text. Summarize a long report, classify customer reviews, translate emails into other languages, generate marketing campaigns - the possibilities are endless!

Build an internal assistant with RAG and function calling

Give your employees easy access to internal company knowledge by building retrieval-augmented (RAG) applications. You can use our state-of-the-art embedding model for that, as well as the function calling capacities of our models.

Empower your developers with a coding assistant

Leverage your developers with a custom-built coding co-pilot and accelerate your coding speed! Mistral models have proven to be particularly strong in coding, making them a great asset to accelerate application development or even IT legacy modernization.

Tailor your applications to your customer base

Our multilingual models can be tuned to follow a certain editorial line and used to generate content as you see fir for your customers.

https://mistral.ai/business/#use-cases

DeepSeek

Introduced May 2024.

Emphasizes efficiency and cost, while achieving similar, if not better performance than many other state-of-the-art models, especially in Chinese, math, coding, and reasoning

https://www.deepseek.com/

DeepSeek

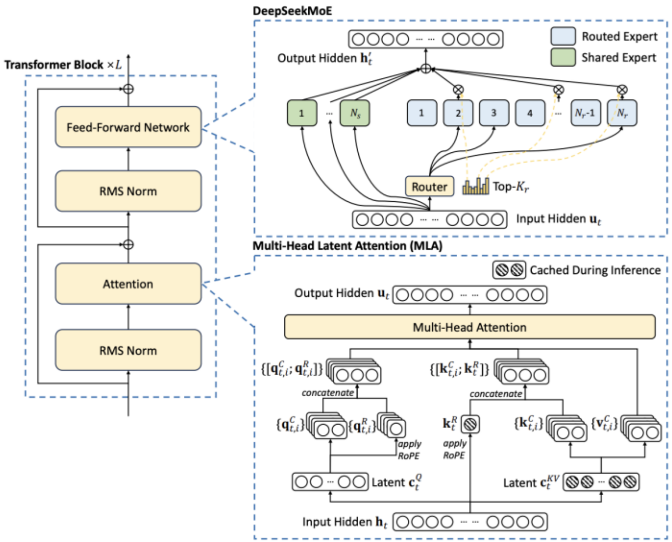


Figure 2 | Illustration of the architecture of DeepSeek-V2. MLA ensures efficient inference by significantly reducing the KV cache for generation, and DeepSeekMoE enables training strong models at an economical cost through the sparse architecture.

SOTA-model is DeepSeek-V2 with 236B parameters, which features innovative architectures like Multi-head Latent Attention and DeepSeekMoE for optimized inference. Also features a 128K context window size

Liu et al., DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model (2024)

DeepSeek

Chinese Performance vs. API Price

Elites in AlignBench, DeepSeek-V2's performance is in the top tier globally with unbeatable API pricing.

	Open source	Chinese General AlignBench	English General MT-Bench	Knowledge MMLU	Arithmetic GSM8K	Math MATH	Reasoning BBH	Coding HumanEval
DeepSeek-V2	Yes	7.91	8.97	77.8	92.2	53.9	79.7	81.1
GPT-4-Turbo-1106	-	8.01	9.32	84.6	93.0	64.1	-	82.2
GPT-4-0613	-	7.53	8.96	86.4	92.0	52.9	83.1	84.1
GPT-3.5	-	6.08	8.21	70.0	57.1	34.1	66.6	48.1
Gemini1.5 Pro	-	7.33	8.93	81.9	91.7	58.5	84.0	71.9
Claude3 Opus	-	7.62	9.00	86.8	95.0	61.0	86.8	84.9
Claude3 Sonnet	-	6.70	8.47	79.0	92.3	40.5	82.9	73.0
Claude3 Haiku	-	6.42	8.39	75.2	88.9	40.9	73.7	75.9
abab-6.5	-	7.97	8.82	79.5	91.7	51.4	82.0	78.0
abab-6.5s	-	7.34	8.69	74.6	87.3	42.0	76.8	68.3
ERNIE-4.0	-	7.89	7.69	-	91.3	52.2	-	72.0
GLM-4	-	7.88	8.60	81.5	87.6	47.9	82.3	72.0
Moonshot-v1	-	7.22	8.59	-	89.5	44.2	-	82.9
Baichuan 3	-	-	8.70	81.7	88.2	49.2	84.5	70.1
Qwen1.5 72B	Yes	7.19	8.61	76.2	81.9	40.6	65.9	68.9
LLaMA 3 70B	Yes	7.42	8.95	80.3	93.2	48.5	80.1	76.2
Mixtral 8×22B	Yes	6.49	8.66	77.8	87.9	49.8	78.4	75.0

https://www.deepseek.com/

Performance

	provider	1M input tokens	1M output tokens	Image
GPT 4 Turbo	openai	$10	$30	150px = $0,00255
GPT 4 Omni	openai	$5	$15	150px = $0,001275
Gemini 1.5 Pro	google	$3.50/-128K $7/+128K	$10.5/-128K $21/+128K	
Gemini 1.5 Flash	google	$0.35/-128K $0.70/+128K	$0.53/-128K $1.05/+128K	
Claude 3 Opus	anthropic	$15	$75	
Claude 3 Soonet	anthropic	$3	$15	
Claude 3 Haiku	bedrock	$0.25	$1.25	
Cohere R+	bedrock	$3	$15	
Llama 3 70B	bedrock	$2.5	$3.05	
Mistral Large	mistral	$4	$12	

model	HellaSwag	MMLU	ARC-C	DROP	WinoGrande	MATH	BIG-Bench	HumanEval	Natural2Code	WMT23	GSM8K	GPQA
GPT4	95.3	86.4	96.3	80.9	87.5	52.9	83.1	67.0	73.9	73.8	**92.0**	35.7
GPT-4 Turbo	**96**	88.4	x	86	x	72.6	83.9	73.1	75	x	x	48
GPT 4 Omni	x	**88.7**	x	83.4	x	**76.6**	x	**90.2**	x	x	x	**53.6**
Claude 3 Opus	95.4	86.8	**96.4**	83.1	**88.5**	60.1	**86.8**	84.9	x	x	95	50.4
Gemini Pro 1.5	92.5	81.9	x	78.9	x	58.5	84	71.9	**77.7**	**75.2**	91.7	41.5
Gemini 1.5 Flash	x	78.9	x	x	x	54.9	85.5	x	77.2	x	x	39.5
Cohere R+	91.2	88.2	x	x	84.3	x	x	71.4	x	x	x	x
Mistral Large	89.2	81.2	94.2	x	87.5	45	x	45.1	x	x	81	x
Mistral 8x22B	88.5	77.7	91.3	x	84.7	41.8	x	45.1	x	x	78.6	x
Llama 70B	85.6	82	93	79.7	83.1	30	81.3	81.7	x	x	93	39.5
Llama 400B	x	86.1	x	**83.5**	x	57.8	x	84.1	x	x	x	48

https://medium.com/@daniellefranca96/battle-of-the-top-llama-3-claude-3-gpt4-omni-gemini-1-5-pro-light-and-more-3ff560cf6b58

Prompting to improve quality of LLMs generated responses

1. Clear and specific instructions
   i. Vague: "Tell me about Python."
   ii. Specific: "Explain the main differences between Python 2 and Python 3."
2. Contextual information
   i. “How does backpropagation work?” vs. “In the context of neural networks, how does backpropagation work?"
3. Role play
   i. "You are a data scientist explaining neural networks to a beginner. How would you describe it?"
4. Iterative refinement
   i. "What are the advantages of renewable energy?" -> "What are the economic advantages of renewable energy sources like solar and wind?"

Break!

We will reconvene in 15 mins. Meanwhile, we are available for Q/As.

Next in line: Hands On Tutorial

Hands On Tutorial with Open Source LLMs

Presenters: Robert Tang

Link to a python Jupyter Notebook

https://colab.research.google.com/drive/1xxsIbGLf7BCOgXgjyTuVEDeire1zzNsB#scrollTo=5KONMiTvPzRY

Code Generation in Bioinformatics

Presenters: Robert Tang

Yale

NIH National Library of Medicine
National Center for Biotechnology Information

Code LLMs

Presenters: Robert Tang

Yale

Learning Objectives of the session

- Brief overview of how code LLMs differ from non-code LLMs
- Introduction to code LLM development
    - Pretraining, fine-tuning, evaluation of code LLMs
    - Prompting and instruction tuning
- Brief overview of how to overcome limitations of code LLMs

Code LLMs

Code Completion

```
#write a quick sort algorithm in Python.

def quick_sort(arr):
    if len(arr) <= 1:
        return arr
    pivot = arr[0]
    less_than_pivot = []
    greater_than_pivot = []
    for i in range(1, len(arr)):
        if arr[i] < pivot:

less_than_pivot.append(arr[i])
        else:

greater_than_pivot.append(arr[i])
    return quick_sort(less_than_pivot) +
[pivot] + quick_sort(greater_than_pivot)

# Test the function
print(quick_sort([7, 2, 3, 1, 4]))
```

Fill-In-The-Blank

```
def quick_sort(arr):
    if len(arr) <= 1:
        return arr
    pivot = arr[0]
    left = []
    right = []
<|fim_hole|>
            left.append(arr[i])
        else:

right.append(arr[i])
    return quick_sort(left) +
[pivot] + quick_sort(right)
```

```
def quick_sort(arr):
    if len(arr) <= 1:
        return arr
    pivot = arr[0]
    left = []
    right = []
    for i in range(1,
len(arr)):
        if arr[i] < pivot:
            left.append(arr[i])
        else:

right.append(arr[i])
    return quick_sort(left) +
[pivot] + quick_sort(right)
```

Notable Code LLM models

CodeGeeX

A Multilingual Code Generation Model

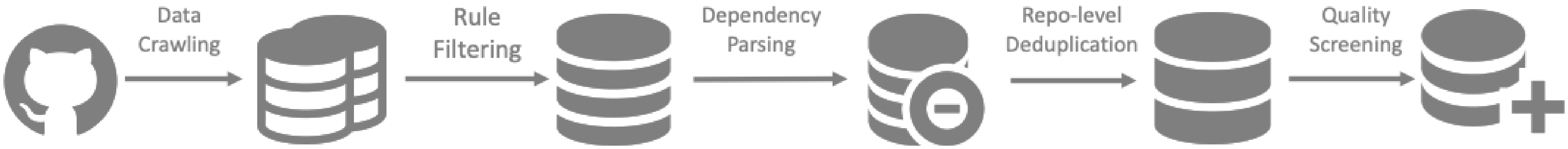
Meta CodeLLaMA

GPT - 4

Starcoder

Deepseek Coder (V1, V2)

Development Process



Pretraining
- Natural language queries
- Prioritize language understanding
- Usually 500B-2T tokens

GitHub Code Scraping
- Scrape GitHub for millions of repositories
- Use novel code filtering, highlighting, and analysis techniques
- Perform repository-level dependency analysis

Fine-tuning
- Use a prompt to provide more structured outputs (next slide)
- Model learns to generate more structured code based on repositories

Prompting/Instruction Tuning

```
### Complete the following code
sample, following the user instruction
and the given context.

### Context
{context}

### User Input
{user_input}

### Response
<model output here>
```

Prompt + Dataset

Instruction Tuning

- Model learns to generate more structured outputs
- Used by some models, extremely useful for agent-related tasks

Evaluation

Model	Size	HumanEval Python	HumanEval Multilingual	MBPP	DS-1000
Pre-Trained Models					
Codex-001	-	33.5%	26.1%	45.9%	20.2%
Codex-002	-	-	-	-	39.2%
StarCoder	16B	36.0%	28.7%	46.8%	27.2%
CodeGeeX2	6B	36.0%	24.5%	42.4%	22.9%
CodeLlama	7B	31.7%	29.2%	41.6%	22.1%
CodeLlama	13B	36.0%	35.4%	48.4%	26.8%
CodeLlama	34B	48.2%	41.0%	55.2%	34.3%
DeepSeek-Coder-Base	1.3B	34.8%	28.3%	46.2%	16.2%
DeepSeek-Coder-MQA-Base	5.7B	48.7%	41.3%	57.2%	27.7%
DeepSeek-Coder-Base	6.7B	49.4%	44.7%	60.6%	30.5%
DeepSeek-Coder-Base	33B	**56.1%**	**50.3%**	**66.0%**	**40.2%**
Instruction-Tuned Models					
GPT-3.5-Turbo	-	76.2%	64.9 %	70.8%	-
GPT-4	-	**84.1%**	**76.5%**	**80.0%**	-
DeepSeek-Coder-Instruct	6.7B	78.6%	66.1%	65.4%	-
DeepSeek-Coder-Instruct	33B	**79.3%**	**69.2%**	**70.0%**	-

Pass@K Metric

$$:= \mathop{\mathbb{E}}_{\text{Problems}}\left[1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}\right]$$

n = # generated
c = # passed
k = # considered

SWE-bench

SWE-bench: repository-level benchmark

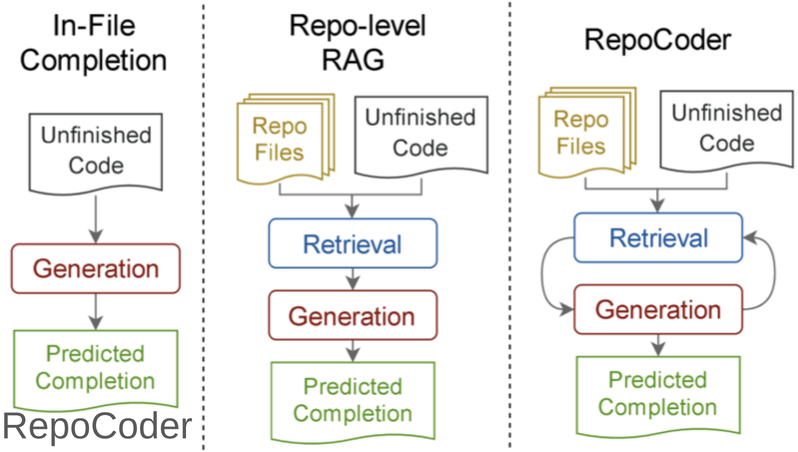
Source: https://github.com/deepseek-ai/DeepSeek-Coder

Current Research

Agents

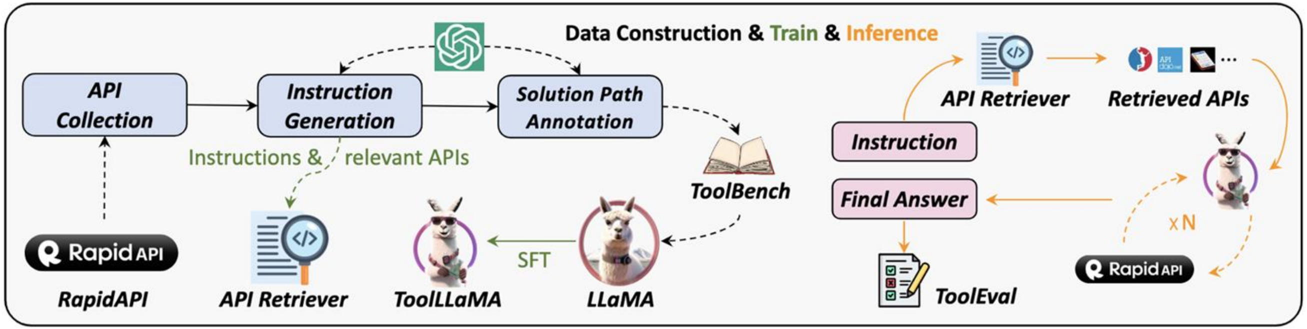
OpenDevin

Repository Level Code Completion



RepoCoder
https://arxiv.org/pdf/2303.12570

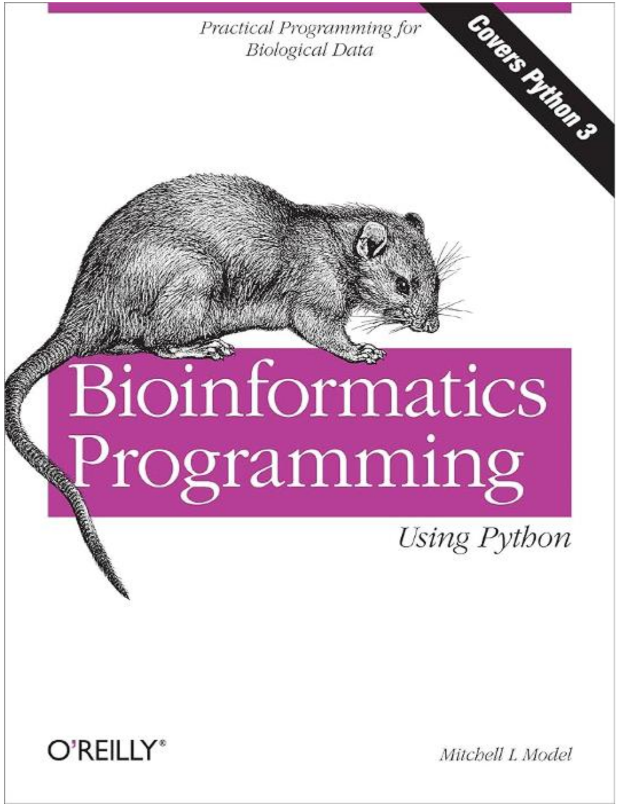
ToolLLM
https://github.com/OpenBMB/ToolBench

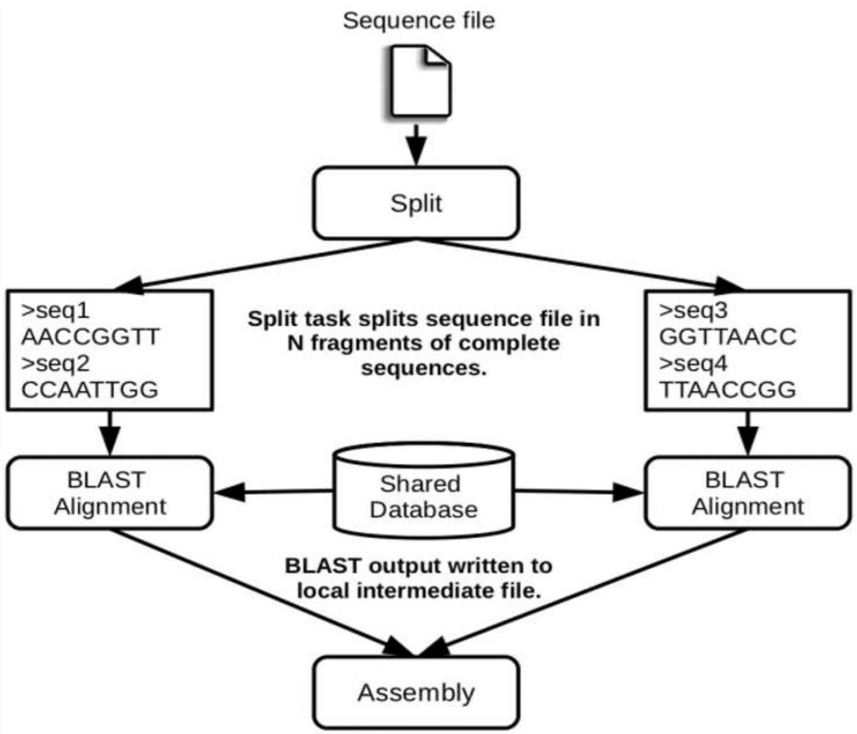


BioCoder: A Benchmark for Bioinformatics Code Generation with Large Language Models

Presenters: Robert Tang

How about bioinformatics code generation?





Building a reliable benchmark

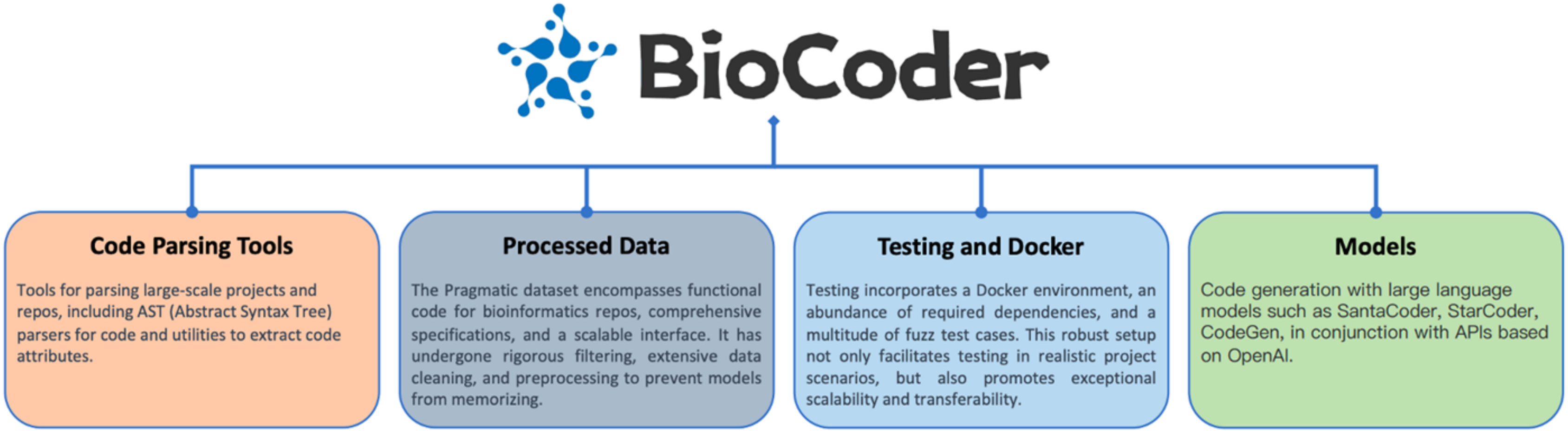
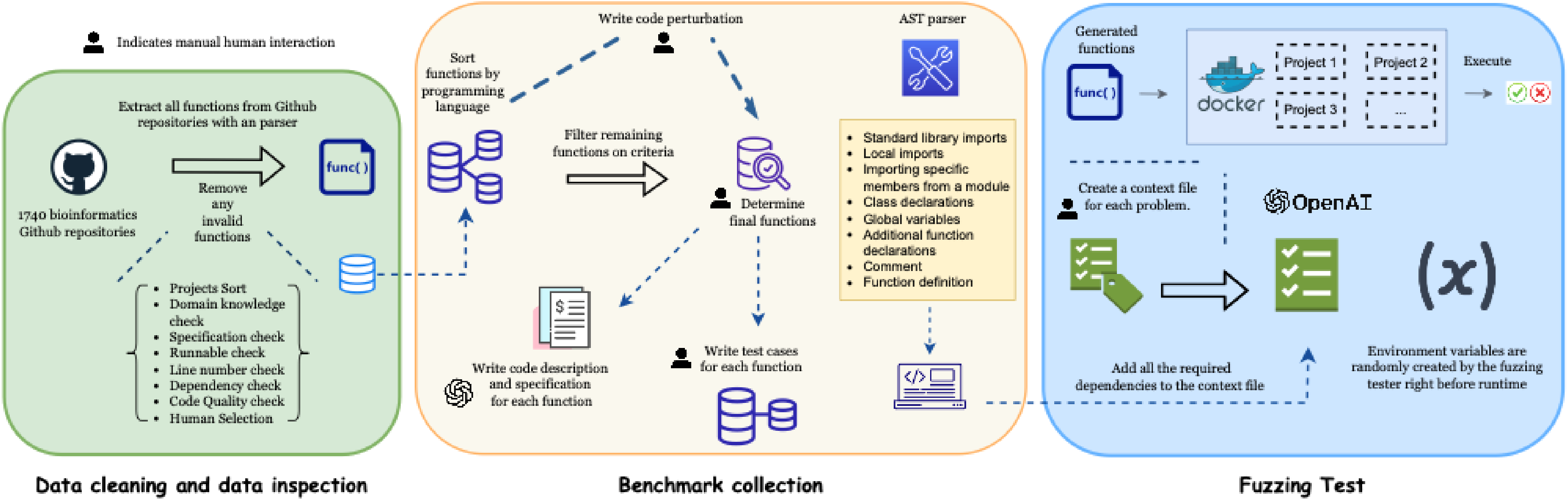
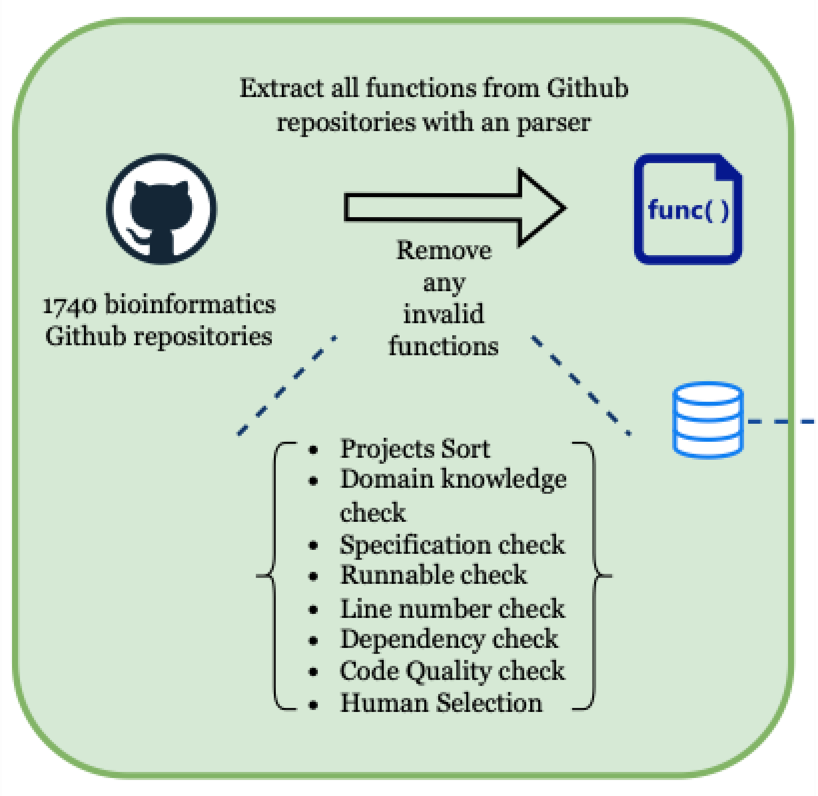


Figure 1: Overview of our contribution in BIOCODER.

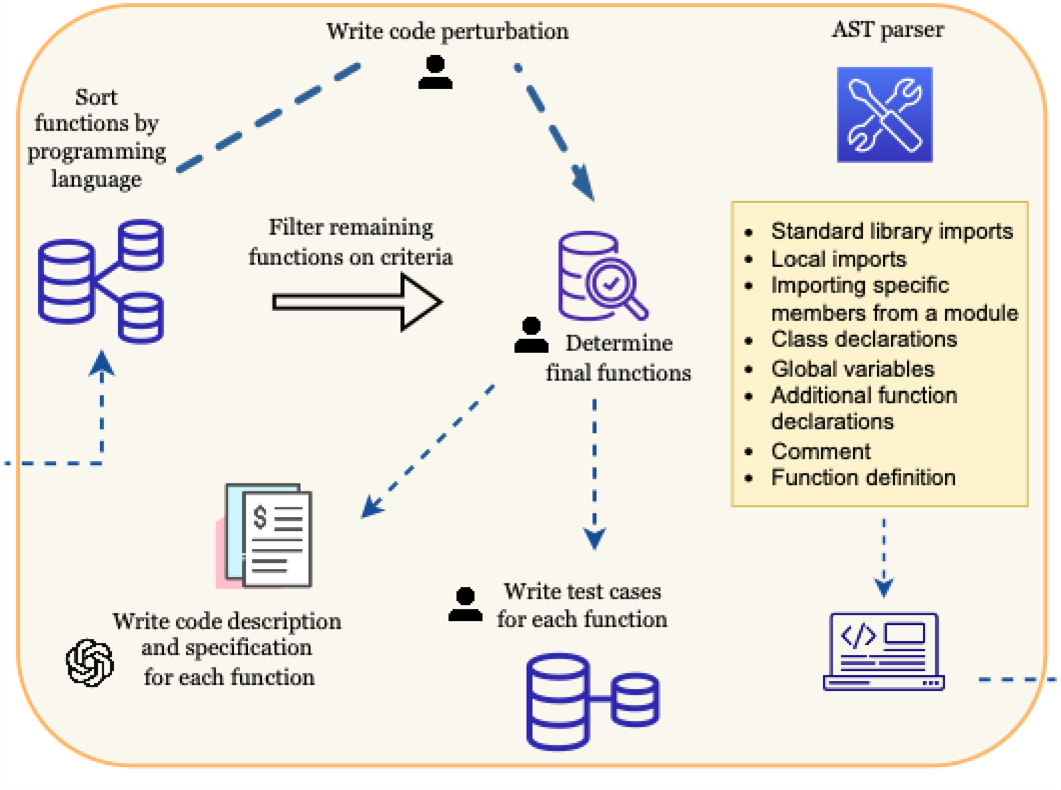
Benchmark construction process



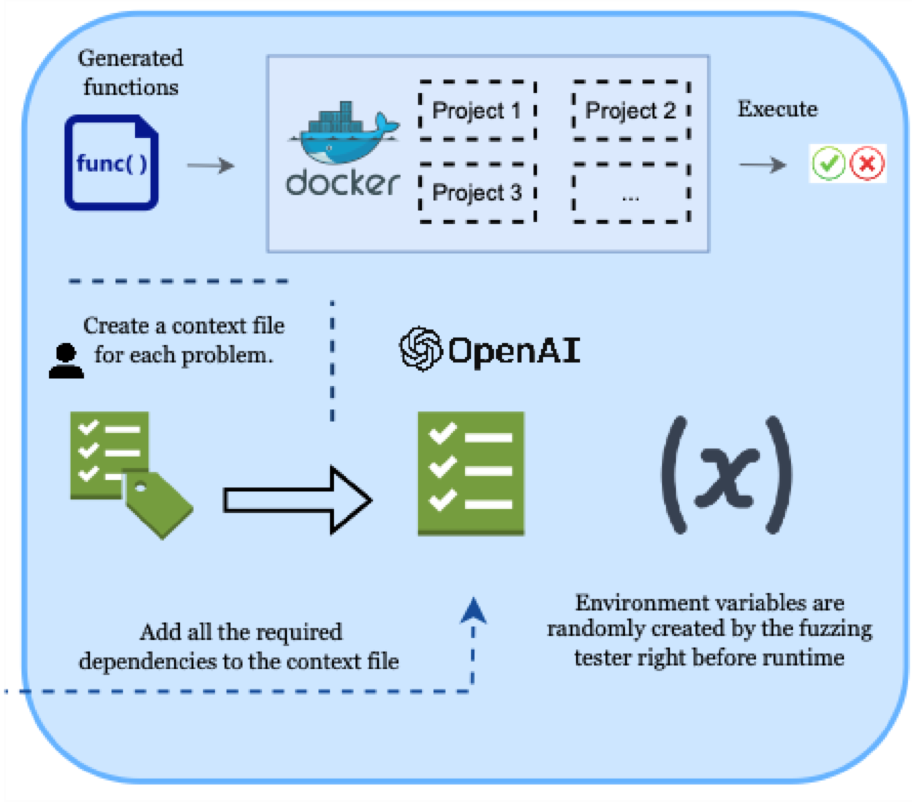
Benchmark construction process



Benchmark construction process



Benchmark construction process



```
Here are the imports:
from collections import defaultdict
import re
import numpy as np
```

```
Here are the global variables:
trans_dict = defaultdict(lambda : 5, A=0, C=1, G=2, T=3)
trans_dict['-'] = 4
```

```
Here are the class declarations:
class Sequence(object):
      attributes:
self.label,self.sequence,self.length,self.unaligned_length,self.frequency,self.np_seque
nce
      methods:
      def __init__(self, label, sequence):
            summary: Initializes a class instance with the specified label and sequence
information.
            param: label (str) - the label of the sequence.
            param: sequence (str) - the nucleotide sequence.
            return: None - the function does not return any value.
      def __eq__(self, other):
            Parameters:
            - self (object) - the first object to be compared
            - other (object) - the second object to be compared
            Return:
            - (bool) - returns True if the objects are equal and False if they are not
equal.
```

```
summary: Returns a string with the sequence in fasta format param: None return: str -
The FASTA representation of the sequence The function is located in the class Sequence
```

```
def to_fasta(self):
```

Area where imports for function is defined

Area where global variabls for function is defined (Python only)

Area where external classes for function is defined

Area where summary for function is defined

Area where function signature is defined

Rank	Model	Details	Pass@1	Pass@5	Pass@10	Pass@20
1 Mar 14, 2023	gpt-4 *Azure OpenAI*	Completion t=0.7, top-p=0.95 len = 8192	38.439	48.491	50.619	52.229
2 Mar 01, 2023	gpt-3.5-turbo *Azure OpenAI*	Completion t=0.7, top-p=0.95 len = 8192	24.682	33.997	37.132	40.127
3 May 09, 2023	Starcoder *Bigcode* Li et al., '23	Completion t=0.7, top-p=0.95 len = 8192	4.682	15.225	21.200	27.166
4 Dec 22, 2022	SantaCoder *Bigcode* Allal et al., '22	Completion t=0.7, top-p=0.95 len = 2048	2.965	9.848	14.227	18.181
5 Nov 08, 2022	InCoder-6B *Facebook AI* Fried et al., '22	Completion t=0.7, top-p=0.95 len = 2048	1.688	5.320	8.332	12.006
6 May 03, 2023	CodeGen2-7B *Salesforce Research* Nijkamp et al., '23	Completion t=0.7, top-p=0.95 len = 2048	0.860	2.494	3.962	6.242
7 Nov 08, 2022	CodeGen-6B *Salesforce Research* Nijkamp et al., '22	Completion t=0.7, top-p=0.95 len = 2048	0.637	0.637	0.637	0.637
8 May 15, 2023	InstructCodeT5+ 16B *Salesforce Research* Wang et al., '23	Completion t=0.7, top-p=0.95 len = 2048	0	0	0	0

Hands On Tutorial with BioCoder

Presenters: Robert Tang, Tom Qiu

Hands on Tutorial

https://colab.research.google.com/drive/18EiOJFG7zNmkSoDj7--wiCXMpVq9W1PX?usp=sharing