Leaderboards

For benchmarking the performance of various LLMs and SLMs on a specific task (code generation in our case).

- **Big Code Models Leaderboard** evaluates on HumanEval (Python) + irrelevant for me MultiPL-E (C++, Java, and JavaScript): https://huggingface.co/spaces/bigcode/bigcode-models-leaderboard
- **EvalPlus** evaluates using HumanEval+ version 0.1.10; MBPP+ version 0.2.0. Models are ranked according to pass@1 using greedy decoding: https://evalplus.github.io/leaderboard.html
- **LiveCodeBench** - holistic and contamination-free evaluation of coding capabilities of LLMs: https://livecodebench.github.io/leaderboard.html. Optional – to estimate the usefulness for the Praxis.
- **CanAICode results**: https://huggingface.co/spaces/mike-ravkine/can-ai-code-results .
- **Awesome Code LLM:** https://github.com/huybery/Awesome-Code-LLM
- **SEAL Leadervoard** – evaluates multiple coding languages: https://scale.com/leaderboard/coding. Optional – to estimate the usefulness for the Praxis
- **Vellum LLM Leaderboard** evaluates on HumanEval among other metrics: https://www.vellum.ai/llm-leaderboard. Optional – to estimate the usefulness for the Praxis

Evaluation Datasets

Used to evaluate the quality of the generated code

1. **HumanEval:** popular benchmark for evaluating code generation models - contains programming problems with corresponding unit tests that can be used to verify the correctness of generated solutions.
   Source: https://github.com/openai/human-eval/tree/master/data
2. **MBPP (Mostly Basic Python Problems):** designed to evaluate code generation models on Python programming tasks. It consists of a large number of Python problems with a set of unit tests that assess the correctness of the generated Python code.
   Source: https://github.com/google-research/google-research/tree/master/mbpp
3. **LiveCodeBench Dataset:** holistic and contamination-free evaluation of coding capabilities of LLMs
   Source: https://huggingface.co/livecodebench

Evaluation Code

- **My code** - Testing LLMs on the Code Generation Task: https://github.com/agnedil/Praxis
- **EvalPlus** evaluation code: https://github.com/evalplus/evalplus/

- **LiveCodeBech** evaluation code: https://github.com/LiveCodeBench/LiveCodeBench

Training Datasets

Used to fine-tune SLMs to improve their code generation capabilities

1. **Tested-143k-Python-Alpaca**

Description: Python dataset with 143,327 examples of code that passed automatic tests to ensure high quality.
Link: https://huggingface.co/datasets/Vezora/Tested-143k-Python-Alpaca

2. **CodeFeedback-Filtered-Instruction**

Description: a curated collection of code instruction queries extracted from open-source code instruction tuning datasets. It significantly advances code generation capabilities by integrating execution and iterative refinement functionalities.
Link: https://huggingface.co/datasets/m-a-p/CodeFeedback-Filtered-Instruction

3. **Magicoder-Evol-Instruct-110K**

Description: A decontaminated version of evol-codealpaca-v1. Decontamination was done in the same way as StarCoder (bigcode decontamination process). See Magicoder paper.
Link: https://huggingface.co/datasets/ise-uiuc/Magicoder-Evol-Instruct-110K

4. **Python-code-dataset-500k**

Description: a summary and reformat pulled from GitHub code. 500K examples to be cleaned first. Cleaning can be done using an SLM.
Link: https://huggingface.co/datasets/jtatman/python-code-dataset-500k

5. **Just-write-the-code-Python-GenAI-143k**

Description: The entire dataset of 230k examples of AI and Machine Learning python code retrieved from public repositories on GitHub. It is a prototype and needs to be cleaned.
Link: https://huggingface.co/datasets/guidevit/Just-write-the-code-Python-GenAI-143k and https://huggingface.co/datasets/guidevit/Just-write-the-code-Python-GenAI-230k

6. **Tiny codes**

1.6 M short and clear code snippets that can help LLM models learn how to reason with both natural and programming languages.
Link: https://www.sonarsource.com/learn/llm-code-generation/

Small Language Models

1. **Llama 3** – an advanced language model from Meta considered one of the best open-source models in its category. Description: https://ai.meta.com/blog/meta-llama-3/ . Usage: https://huggingface.co/meta-llama/Meta-Llama-3-8B. CodeLlama-7b-Instruct.
2. **Mixtral** – advanced mix of experts for better reasoning. One of the best small language models out there. It's able to leverage a wide spectrum of knowledge through a blend of various domains. Mixtral creates new models capable of running on local machines while still achieving comparable power to full-scale LLMs. Description: https://mistral.ai/news/mixtral-of-experts/. Usage: https://huggingface.co/docs/transformers/en/model_doc/mixtral.
3. **DeepSeek-Coder-V2** – among the best small language models for code generation. Description and usage examples: https://github.com/deepseek-ai/DeepSeek-Coder-V2.
4. CodeGemma-7B-it (HumanEval – 61%)
5. Codestral Mamba (7B) (Apache 2) (HumanEval – 75%)
6. Mixtral 8 x 7B (Apache 2)
7. Mistral 7B (Apache 2)
8. Ministral 8B (Research only)

Code

- Main repo that I am using: https://github.com/openai/human-eval
- May be helpful: https://github.com/abacaj/code-eval/tree/main
- Leaderboard: https://huggingface.co/spaces/bigcode/bigcode-models-leaderboard
- Pass @k explained: https://deepgram.com/learn/humaneval-llm-benchmark
- Replicate LLMs: https://replicate.com/pricing + deploy your own model.

```
# pip install -q datasets
from datasets import load_dataset
# Languages: "python", "js", "java", "go", "cpp", "rust"
ds = load_dataset("bigcode/humanevalpack", "python")["test"]
ds[0]
```

AGENTS

Qwen agent: https://github.com/QwenLM/Qwen-Agent