The Vault: A Comprehensive Multilingual Dataset for Advancing Code Understanding and Generation

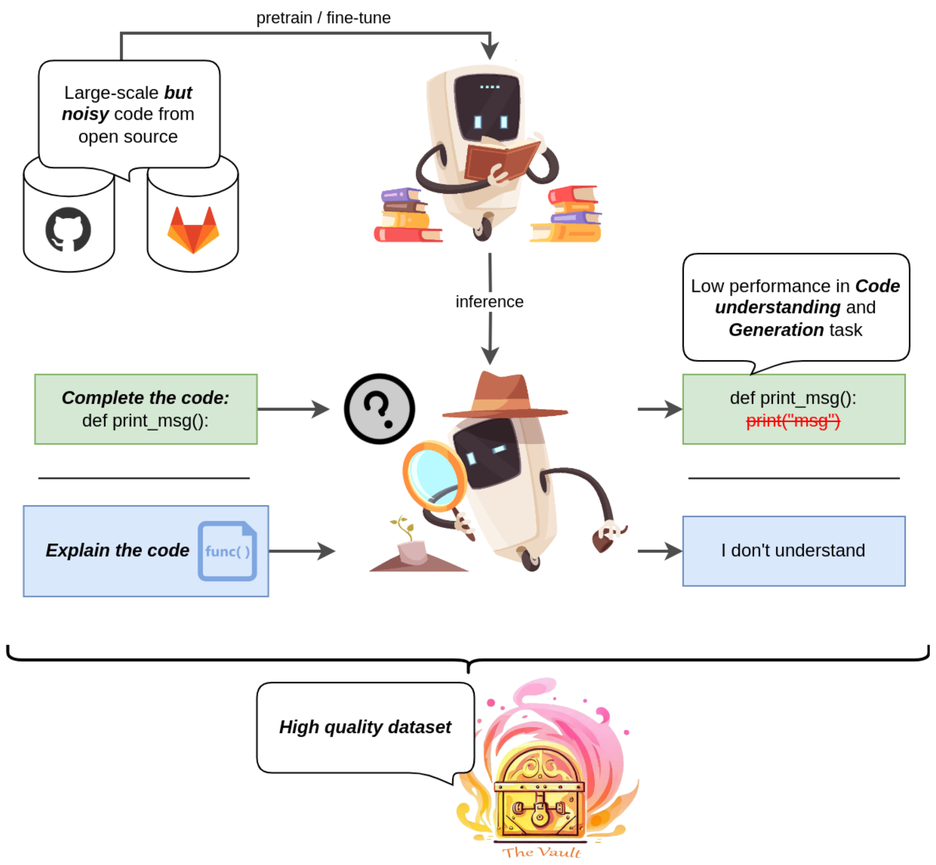
Dung Nguyen Manh[1]*, Nam Le Hai[1,3]*, Anh T. V. Dau[1,3], Anh Minh Nguyen[1], Khanh Nghiem[1], Jin Guo[4,5], Nghi D. Q Bui[2]

[1]FPT Software AI Center *{dungnm31, namlh35, anhdtv7, minhna4, khanhnv22}@fpt.com*

[2]Fulbright University, Vietnam, *nghi.bui@fulbright.edu.vn*; [3]Hanoi University of Science and Technology, Vietnam

[4]School of Computer Science, McGill University, Canada; [5]Mila - Quebec AI Institute

MOTIVATION



Demonstrate the target of TheVault motivation

MAIN CONTRIBUTIONS

- We release a dataset with **43 million samples of high-quality code-text pairs**, 243 million uni-modal samples, and 69 million pairs of line comments with context from 10 popular programming languages.
- We propose a **novel pipeline for obtaining high-quality pairs** of code from large, noisy corpora.
- We publish an open-source toolkit used for **converting raw source code into code-text pairs** and filtering noisy samples in various programming languages.

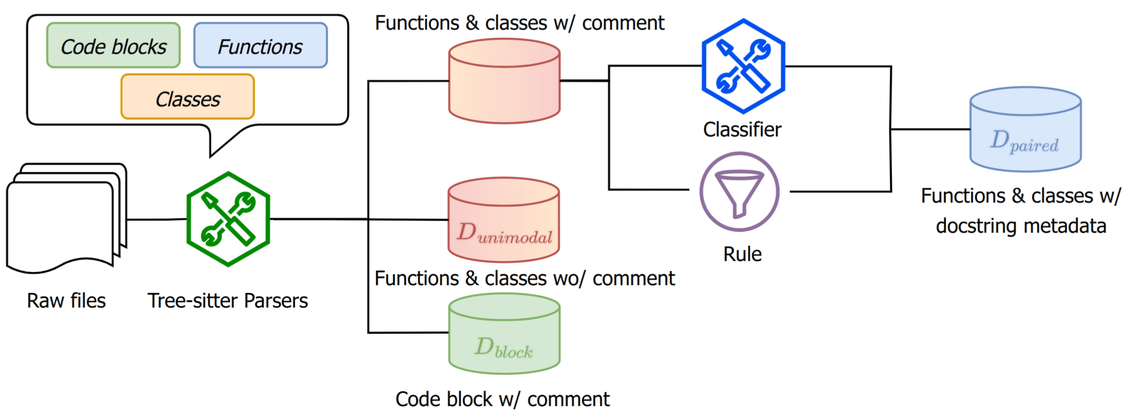
DATASET COMPARISON

Dataset	#PL	#Function w/ docstring	#Function w/o docstring
PyMT5 [Clement et al., 2020]	1	≈ 7,700,000	-
CoDesc [Hasan et al., 2021]	1	4,211,516	-
CodeSearchNet [Husain et al., 2019]	6	2,326,976	4,125,470
CodeXGLUE CSN [Lu et al., 2021]	6	1,005,474	-
Deepcom [Hu et al., 2020]	1	424,028	-
CONCODE [Iyer et al., 2018b]	1	2,184,310	-
Funcom [LeClair et al., 2019]	1	2,149,121	-
CodeT5 [Wang et al., 2021]	8	3,158,313	5,189,321
THEVAULT	**10**	**34,098,775**	**205,151,985**

A comparison between programming language of current available parallel dataset on function level

DATA CREATION

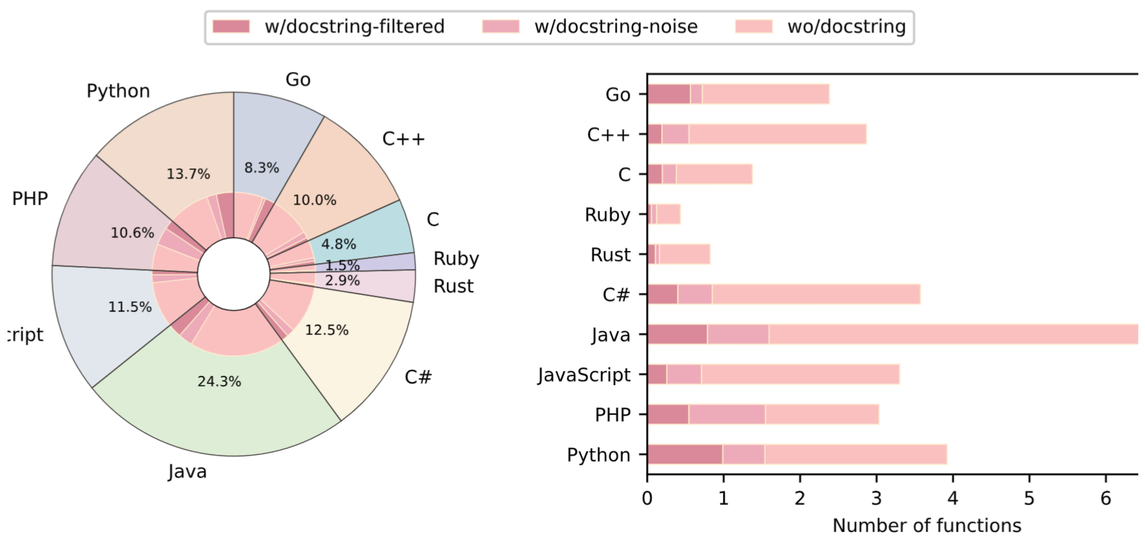
In The Vault, we leverage a subset of The Stack, recognized as the largest publicly accessible, multilingual, permissive licensed source code dataset. We comprise code in 10 prevalent programming languages and use a customized `tree-sitter`, an abstract syntax tree parser, to extract function, method, class, etc and their docstring.



Pipeline to create datasets of code blocks with comments D_{block}, unimodal code $D_{unimodal}$, and code-textpairs D_{paired} from raw source code

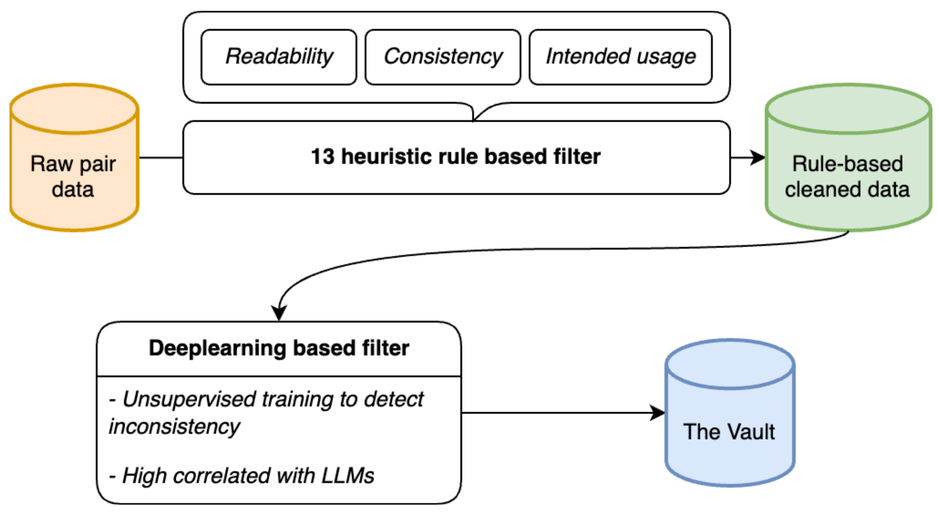
The Vault construction process from scratch:

- ***Step 1:*** Use tree-sitter parsers to parse raw code into ASTs, yielding (D_{paired}) pairs of classes and functions with corresponding block comments (docstring); and uni-modal functions and classes without documentation ($D_{unimodal}$)
- ***Step 2:*** Apply the filtering pipeline, which includes both rules-based and deep-learning based filters, to select high-quality data.



Distribution and the number of functions by the presence of docstrings.

DATA CLEANING



Demonstrate cleaning pipeline using both rule-based and deep learning-based

EXPERIMENTS

Model	Fine-tune dataset	pass@1	pass@10	pass@100
	HUMANEVAL			
CodeGen 350M	-	6.67	10.61	16.84
	Py/CodeSearchNet	2.76	8.76	14.72
	(250K) Py/TheVault	3.74	10.57	16.26
	raw/PyTheStack	6.64	15.42	24.80
	Py/TheVault	**8.14**	**18.12**	**30.07**
CodeGen 2B	-	**14.51**	24.67	38.56
	Py/TheVault	14.00	**25.74**	**41.72**
	MBPP			
CodeGen 350M	-	7.46	24.18	46.37
	Py/TheVault	**10.13**	**33.96**	**53.20**
CodeGen 2B	-	18.06	45.80	**65.34**
	Py/TheVault	**27.82**	**50.06**	65.06

Code generation benchmarks running on CodeGen Multi model.

- CodeT5 and PLBART fine-tuned on (similar size) The Vault significantly better performance than on CodeSearchNet on task ***Code Summarization***.
- On ***Code Search***, superior results are observed in most languages when using (similar size) The Vault than on CodeSearchNet when fine-tuning CodeBERT, RoBERTa, UniXCoder.

Check our Github repository