Author
Reinier Schep
R.J.H.Schep@student.tudelft.nl

Evaluating the Effectiveness of Meta Llama 3 70B for Unit Test Generation

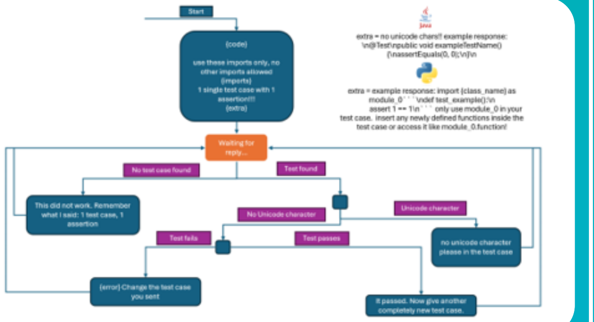
Supervisors
Annibale Panichella
Mitchell Olsthoorn

Background

- Test suites play a crucial role in software development
- Manually writing tests is time intensive [2]
- Automatically generated tests are not comprehensible [1]
- Thus, we need a new way of generating tests
- Generating tests with LLMs could be the solution

Approach

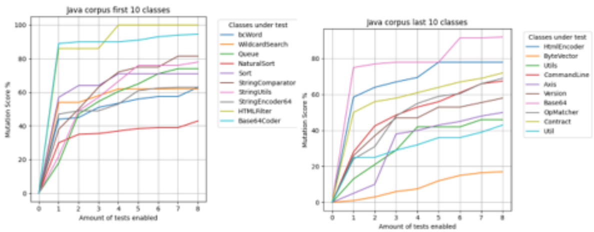
→Acquire corpus of diverse classes with high cyclomatic complexity
→Use Llama3 70B and an automatic tool to generate test suites
→Take multiple samples per class to combat randomness
→Mutation score to quantify performance each test suite
→Run statistical tests to determine any difference between scores

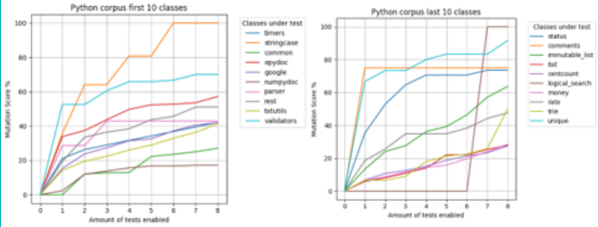


Study Design

→How effective is Llama3 70B at generating unit tests with regards to mutation score?
→Acquire Java and Python corpus of 20 classes each
→Generate 12 test suites per class for both Llama3 and EvoSuite or Pynguin depending on the programming language
→Llama3 test suite consists of exactly 8 tests
→Wilcoxon signed-rank test to determine significant difference in distributions
→Vargha-Delaney effect size to determine how large the difference is

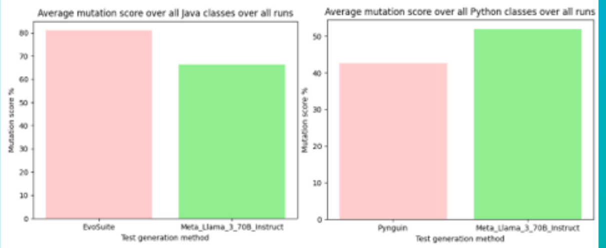
	Project	Class	Pynguin 30s Median Mutation Score	Pynguin 60s Mutation Score	Pynguin 90s Mutation Score
0	codetiming	timers	39.45	47.4	42.1
1	dataclasses_json	stringcase	100.0	90.0	100.0
2	docstring_parser	common	50.0	44.0	38.5
3	docstring_parser	epydoc	16.75	18.1	53.9
4	docstring_parser	google	14.9	14.2	15.6
5	docstring_parser	numpydoc	13.05	14.8	15.6
6	docstring_parser	parser	50.0	16.7	16.7
7	docstring_parser	rest	15.2	21.8	37.0
8	flutils	txtutils	49.55	43.1	68.0
9	flutils	validators	65.85	75.0	70.6
10	httpie	status	64.7	76.5	58.8
11	isort	comments	37.5	70.0	70.0
12	pymonet	immutable_list	33.3	21.9	25.8
13	pyutils	bst	12.9	100.0	99.2
14	pyutils	centcount	54.5	50.0	48.6
15	pyutils	logical_search	0.0	0.0	0.0
16	pyutils	money	57.2	51.4	53.5
17	pyutils	rate	47.65	39.5	32.6
18	pyutils	trie	40.9	96.1	18.2
19	typesystem	unique	81.65	60.0	85.7





	Project	Class	EvoSuite Median Mutation Score	Meta_Llama_3_70B_Instruct Median Mutation Score	p-value Wilcoxon	Vargha-Delaney effect size
0	battlecry_72	bcWord	79.0	63.0	0.0049	L (0.8438)
1	beanbin_15	WildcardSearch	96.0	62.0	0.0005	L (0.9896)
2	biblestudy_68	Queue	82.0	74.0	0.0342	M (0.7118)
3	corina_35	NaturalSort	75.0	43.0	0.001	L (0.9722)
4	corina_35	Sort	29.0	71.0	0.0005	L (0.0)
5	corina_35	StringComparator	100.0	81.5	0.0005	L (1.0)
6	corina_35	StringUtils	87.5	78.0	0.064	M (0.7326)
7	fim1_73	StringEncoder64	79.5	63.0	0.0005	L (1.0)
8	imsmart_11	HTMLFilter	100.0	100.0	1.0	- (0.5)
9	javaviewcontrol_33	Base64Coder	94.0	94.5	0.3804	S (0.3993)
10	javaviewcontrol_33	HtmlEncoder	69.5	78.0	0.0342	L (0.2326)
11	jiprof_51	ByteVector	31.0	17.0	0.0005	L (1.0)
12	lagoon_52	Utils	83.0	46.0	0.0005	L (1.0)
13	openjms_66	CommandLine	88.0	67.5	0.0005	L (1.0)
14	saxpath_24	Axis	100.0	50.0	0.0005	L (1.0)
15	schemaspy_36	Version	84.0	58.0	0.0005	L (1.0)
16	sfmis_7	Base64	77.5	92.0	0.001	L (0.0625)
17	templateit_5	OpMatcher	72.0	69.0	0.3013	M (0.6875)
18	tullibee_1	Contract	100.0	72.0	0.0005	L (1.0)
19	tullibee_1	Util	100.0	43.0	0.0005	L (1.0)

	Project	Class	Pynguin Median Mutation Score	Meta_Llama_3_70B_Instruct Median Mutation Score	p-value Wilcoxon	Vargha-Delaney effect size
0	codetiming	timers	39.45	42.1	0.6221	- (0.4688)
1	dataclasses_json	stringcase	100.0	100.0	0.7334	- (0.5347)
2	docstring_parser	common	50.0	27.1	0.0161	L (0.7639)
3	docstring_parser	epydoc	16.75	57.2	0.0024	L (0.1597)
4	docstring_parser	google	14.9	42.0	0.0005	L (0.0)
5	docstring_parser	numpydoc	13.05	17.2	0.0015	L (0.1181)
6	docstring_parser	parser	50.0	42.9	0.9697	M (0.6875)
7	docstring_parser	rest	15.2	51.05	0.0005	L (0.0104)
8	flutils	txtutils	49.55	41.3	0.0068	L (0.7847)
9	flutils	validators	65.85	70.0	0.6772	S (0.3889)
10	httpie	status	64.7	73.55	0.001	L (0.0972)
11	isort	comments	37.5	75.0	0.0034	L (0.1806)
12	pymonet	immutable_list	33.3	63.7	0.0015	L (0.0556)
13	pyutils	bst	12.9	27.95	0.0342	L (0.1111)
14	pyutils	centcount	54.5	27.55	0.0005	L (1.0)
15	pyutils	logical_search	0.0	100.0	0.0034	L (0.1285)
16	pyutils	money	57.2	28.5	0.0005	L (1.0)
17	pyutils	rate	47.65	47.65	0.7334	- (0.5417)
18	pyutils	trie	40.9	50.0	0.5186	- (0.5)
19	typesystem	unique	81.65	91.65	0.064	M (0.2986)



Conclusion

→EvoSuite is more effective than Llama3 in terms of mutation score
→Llama3 is more effective than Pynguin in terms of mutation score
→Overall, Llama3 is a serious competitor to both tools

Future work

→Compare against different LLMs
→Explore different programming languages
→Search for optimal prompting strategies

References

[1] M. Moein Almasi, Hadi Hemmati, Gordon Fraser, Andrea Arcuri, and Janis Benefelds. 2017. An Industrial Evaluation of Unit Test Generation: Finding Real Faults in a Financial Application. In 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP). 263–272. https://doi.org/10.1109/ICSE-SEIP.2017.27

[2] Claus Klammer and Albin Kern. 2015. Writing unit tests: It's now or never!. In 2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW). 1–4. https://doi.org/10.1109/ICSTW.2015.7107469