Discovering the topics of Continuous Integration projects on GitHub

Author: Lukas Ostrovskis (l.ostrovskis@student.tudelft.nl)
Supervisors: Sebastian Proksch, Shujun Huang

TU Delft

Introduction ①

- Continuous Integration (CI) - software development practice, automating the building and testing of code changes in a shared repository.
- Studies expressed concerns about the impact of project contexts on the effectiveness of CI implementations and the potential for enhancements – no "one-size-fits-all" implementation [1,2]
- Clustering software projects by their topics would enable the analysis of the correlation between CI implementations and different topics, and the discovery of emerging best practices in various domains

Research question ②

What data can be extracted from GitHub to effectively classify CI projects by topic and what CI practices emerge from these topic clusters?

Methodology ③

Explore the definition of a software project topic

Identify existing topic classification tools

Compare different textual data effectiveness

Build a tool to mine GitHub CI project metrics and data and implement categorization

Mine projects and analyze correlations

Conclusions ⑤

- Developed a tool to mine CI projects on GitHub, utilizing GitHub topic labels, a Multi-label LR classifier, ChatGPT and GitHub Search to cluster them by topics, using the name, description, and README of a project for topic modeling
- Conducted a brief analysis of context-dependent CI metrics. Interesting insights suggest that the tool could be successfully employed for further research in context-dependent CI implementation analysis

Results ④

Definition of a software topic (1)

Related work
Definition ranges from broad application domains to highly specific topic labels.

Taxonomies
Multiple attempts to create coherent software topic taxonomies. No existing integrations into topic modeling tools limits applicability.

GitHub topic labels
- Built-in feature – convenient
- Number of topics is a concern – over 1M unique topic labels on GitHub
- Authors of Repologue [3] condensed the set into 228 topics
- There is a community-curated list of topic labels (~850), which follow the power-law distribution, illustrated in *Figure 1*.

Therefore, GitHub topic labels is a suitable set of topics with manageable granularity that we can use for categorizing CI projects.

Tools and approaches (2)

3 tools compared on a dataset of 103 projects: **LASCAD** [4], **Multi-label LR classifier** [3], **ChatGPT.** Results in *Table 1*.

Multi-label LR classifier and **ChatGPT** API integrated into our CI project mining tool, LASCAD's processing time deemed a bottleneck

2 approaches not relying on topic modeling tools:
1. Utilizing existing GitHub topic labels
2. Utilizing GitHub Search with the topic filter

Types of data for topic modeling (3)

Source code, commits, pull requests and **issues** proved to be the most resource-intensive data types (*Table 2*)

We decided to use a combination of **repository name, description, and README** for topic modeling due to the best balance between performance and time

Brief analysis of context-dependent CI metrics (4)

Focusing on 6 arbitrary topics, we collected data from 4899 public repositories utilizing GitHub Action Workflows, corresponding CI metric results presented in *Table 3*.

Analysis offers valuable insights: API projects exhibit highest mean workflow count per project and run workflows significantly more than other categories; TypeScript projects are more likely to utilize CI than JavaScript ones, but have fewer workflow runs on average; Docker projects contradict the trend of majority workflow triggers being pull requests, with a push to pull request trigger ratio of 2:1

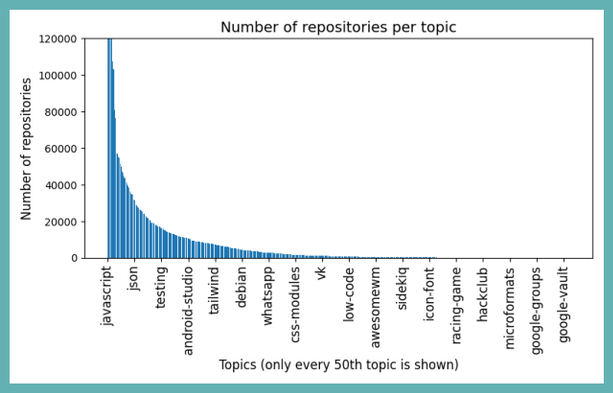

Figure 1: Power law-like distribution of repositories associated with GitHub community-curated topic labels

	LASCAD	Multi-label LR classifier	ChatGPT
Accuracy	0.72	0.57	0.95
Macro-Precision	0.72	0.78	0.95
Macro-Recall	0.62	0.62	0.92
Macro-F1 score	0.67	0.69	0.94
Time (s)	2079.1	11.9	186.4

Table 1: Comparison of 3 tools used for software categorization by topics with the LASCAD dataset of 103 GitHub repositories

	Source code	Repository Name, Description	README
Avg. Retrieval Time (s)	89.5	1.2×10^{-5}	0.23
Avg. # of used API requests	0	1	1
	Commits	**Pull Requests**	**Issues**
Avg. Retrieval Time (s)	89.2	98.4	115.6
Avg. # of used API requests	406.8	76.9	164.8

Table 2: Resource comparison of different textual data from GitHub repositories

	android	api	ios
Projects with workflows (%)	59.8	64.1	53.1
Avg. # of Workflows	0.82	1.88	0.65
Avg. Pull Request vs. Push runs	216 / 63	1437 / 586	7 / 7
	javascript	**typescript**	**docker**
Projects with workflows (%)	56.1	73.8	80
Avg. Workflow count	1	1.54	1.04
Avg. Pull Request vs. Push runs	179 / 148	122 / 127	30 / 61

Table 3: GitHub Actions workflow metrics of repositories with different topics

References

[1] Omar Elazhary et al. "Uncovering the benefits and challenges of continuous integration practices" (2021).
[2] Daniel Stahl and Jan Bosch. "Modeling continuous integration practice differences in industry software development" (2014).
[3] Maliheh Izadi, Abbas Heydarnoori, and Georgios Gousios. "Topic recommendation for software repositories using multi-label classification algorithms" (2021).
[4] Doaa Altarawy et al. "Lascad: Language-agnostic software categorization and similar application detection" (2018).