

Leveraging GPT-like LLMs to Automate Issue Labeling

Giuseppe Colavito
giuseppe.colavito@uniba.it
University of Bari
Italy

Nicole Novielli
nicole.novielli@uniba.it
University of Bari
Italy

Filippo Lanubile
filippo.lanubile@uniba.it
University of Bari
Italy

Luigi Quaranta
luigi.quaranta@uniba.it
University of Bari
Italy

The 21st International Conference on Mining Software Repositories (MSR24)

発表者：松尾 和輝

背景

- ソフトウェア開発では課題追跡システムが広く利用
 - バグの報告, 新機能提案などの課題を課題票に記載し報告
 - 課題票解決の優先順位を付け, 効率的に管理するために, 課題票にラベルが付与される
- 手動で課題票にラベルを付与することの課題
 - ラベルを付与する労力が必要^[24]
 - 間違ったラベルを付与してしまう^{[4][32]}
- 自動でラベルを付与する手法を提案する研究の増加

[4] Giuliano Antoniol, Kamel Ayari, Massimiliano Di Penta, Foutse Khomh, and YannGaël Guéhéneuc. 2008. Is it a bug or an enhancement? a text-based approach to classify change requests. In Proceedings of the 2008 Conf. of the center for advanced studies on collaborative research: meeting of minds. ACM, New York, NY, USA. <https://doi.org/10.1145/1463788.1463819>

[24] Qiang Fan, Yue Yu, Gang Yin, Tao Wang, and Huaimin Wang. 2017. Where Is the Road for Issue Reports Classification Based on Text Mining?. In 2017 ACM/IEEE Int'l Symposium on Empirical Software Engineering and Measurement (ESEM). <https://doi.org/10.1109/ESEM.2017.19>

[32] Kim Herzig, Sascha Just, and Andreas Zeller. 2013. It's not a bug, it's a feature: How misclassification impacts bug prediction. In 2013 35th International Conference on Software Engineering (ICSE). 392–401. <https://doi.org/10.1109/ICSE.2013.6606585>

背景

- 従来研究
 - BERTを用いた自動ラベリング
 - 課題
 - 一貫性のないノイズとなるラベルに影響を受ける^{[17][18]}
 - データセットを自作するにしても多大な労力がかかる
- 少数のデータで高い精度を出せるGPTに着目
 - 自動ラベリングにおいて，GPTはどれほど有効かは不明であるため調査を行う

[17] Giuseppe Colavito, Filippo Lanubile, and Nicole Novielli. 2023. Few-Shot Learning for Issue Report Classification. In 2023 IEEE/ACM 2nd International Workshop on Natural Language-Based Software Engineering (NLBSE). 16–19. <https://doi.org/10.1109/NLBSE59153.2023.00011>

[18] Giuseppe Colavito, Filippo Lanubile, and Nicole Novielli. 2023. Issue Report Classification Using Pre-Trained Language Models. In Proceedings of the 1st International Workshop on Natural Language-Based Software Engineering (Pittsburgh, Pennsylvania) (NLBSE '22). Association for Computing Machinery, New York, NY, USA, 29–32. <https://doi.org/10.1145/3528588.3528659>

目的・アプローチ

- 目的

- GPTベースのモデルが、課題票の自動ラベリングにどの程度有効か調査

- アプローチ

- BERTベースのモデルとGPTベースのモデルの比較

データセット

- [17]で作成されたデータセット
 - 手動でラベル付けされたもの
 - 訓練データ, テストデータそれぞれ200件ずつ抽出

Label	Hand labeling			
	Train set		Test set	
Bug	47	24%	53	27%
Feature	60	30%	55	28%
Question	44	22%	47	24%
Documentation	33	17%	32	16%
Discarded	16	8%	13	7%
Total	200		200	

- 4種類のラベル（Bug, Feature, Question, Documentation）と, 分類できなかったもの（Discarded）で構成

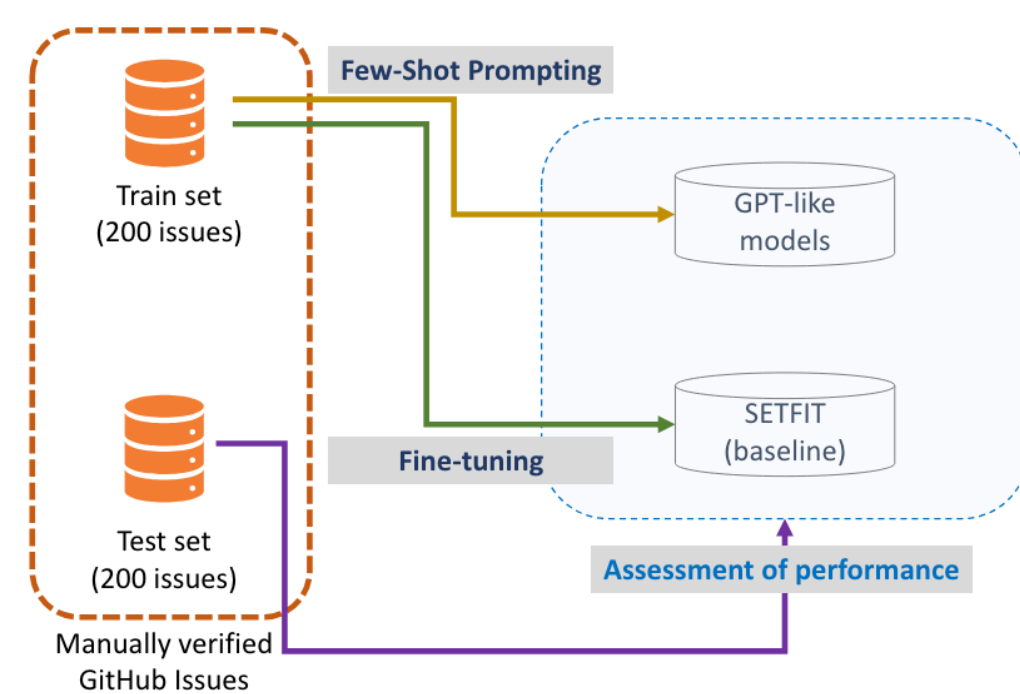
使用LLM

- ChatGPT

- gpt-3.5-turbo0301 (4k-0301)
- gpt-3.5-turbo-0613 (4k-0613)
- gpt-3.5-turbo-16k-0613 (16k0613)
- Zero-shot, One-shot, Two-shotで実験

- SETFIT^[68] (ベースライン)

- Sentence-BERT (SBERT) ^[59]のような変形モデルの微調整のために最適化されたフレームワーク
- 少ないデータでも精度が高い



[59] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 3982–3992. <https://doi.org/10.18653/v1/D19-1410>

[68] Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. E\$cient Few-Shot Learning Without Prompts. <https://doi.org/10.48550/arXiv.2209.11055>

プロンプトテンプレート

Prompt template with label explanation

入力フォーマット

You are provided with a GitHub issue in this format:

Title : ""the title of the issue""

Body : ""the body of the issue""

タスクの説明

You have to assign it a label.

Possible labels are: {label_list}

各ラベルの説明

The "bug" label is used to identify an issue report that describes a problem or error within the software or codebase. It indicates that something is not functioning as intended or producing unexpected results. Bug reports help developers identify and fix issues to improve the overall quality and reliability of the software.

The "feature" label is applied to an issue report that suggests the addition or enhancement of a new capability or functionality within the software. It indicates a request for the implementation of a specific feature that does not currently exist. Feature requests help the development team understand the needs and requirements of the users and can influence the future development roadmap.

The "question" label is used when a user or contributor has a query or needs clarification regarding the software or codebase. It indicates that the issue report is seeking information, guidance, or assistance. Questions can be related to functionality, usage, best practices, or any other topic that requires clarification.

The "documentation" label is applied to an issue report that suggests improvements, updates, or corrections to the software's documentation. It indicates that the issue is focused on the written instructions, guides, or explanations provided alongside the codebase. Documentation issues help ensure that the information available to users and contributors is accurate, comprehensive, and up-to-date, thus facilitating better understanding and adoption of the software.

The "discard" label is used when you can't decide due to lack of information contained in the issue.

例 (Few-shotの場合)

{examples}

入力 (課題票)

Title: ""{title}""

Body: ""{body}""

出力 (ラベルの種類)

{output_format_instructions}

結果 | ChatGPT間の比較

- Zero-shot

	4k-0301			4k-0613			16k-0613			Support
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	
Bug	0,6684	0,9509	0,7850	0,7100	0,9792	0,8232	0,7133	0,9811	0,8261	53
Documentation	0,9488	0,5511	0,6971	0,8627	0,6149	0,7180	0,8853	0,6191	0,7285	32
Feature	0,8916	0,8018	0,8442	0,8883	0,8527	0,8701	0,8861	0,8491	0,8672	55
Question	0,7849	0,8438	0,8129	0,8869	0,7594	0,8181	0,8668	0,7719	0,8164	47
Micro avg	0,7899	0,7882	0,7891	0,8137	0,8128	0,8133	0,8155	0,8155	0,8155	187
Macro avg	0,8234	0,7869	0,7848	0,8370	0,8016	0,8073	0,8379	0,8053	0,8095	187
Discarded prediction	0,2%			0,11%			-			

- 新しいモデル（16k-**0613**）の方が精度高
 - 差は小さめ
- 入力トークン数が多いモデル（**16k**-0613）の方が精度高

結果 | ChatGPT間の比較

- One-shot

		4k-0301			4k-0613			16k-0613			Support
		Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	
「各ラベルの説明」 (p.7) を含めない	(a) Without label explanations in the prompt										
	Bug	0,7212	0,9792	0,8305	0,7115	0,8302	0,7658	0,7049	0,8358	0,7643	53
	Documentation	0,9360	0,6128	0,7400	0,8500	0,5745	0,6839	0,8380	0,5553	0,6674	32
	Feature	0,8541	0,8491	0,8513	0,8862	0,6564	0,7524	0,9105	0,6582	0,7625	55
	Question	0,8316	0,7375	0,7813	0,5204	0,8438	0,6416	0,5296	0,8688	0,6569	47
	Micro avg	0,8123	0,8075	0,8099	0,7175	0,7171	0,7173	0,7195	0,7187	0,7191	187
	Macro avg	0,8357	0,7947	0,8008	0,7420	0,7262	0,7109	0,7458	0,7295	0,7128	187
Discarded prediction	0,59%			0,05%			0,11%				
「各ラベルの説明」 (p.7) を含める	(b) With label explanations in the prompt										
	Bug	0,7332	0,9792	0,8385	0,7018	0,8774	0,7791	0,7006	0,8962	0,7863	53
	Documentation	0,9208	0,6426	0,7566	0,8196	0,6234	0,7074	0,8310	0,6064	0,7010	32
	Feature	0,8664	0,8255	0,8453	0,8750	0,6709	0,7582	0,8687	0,6909	0,7689	55
	Question	0,8570	0,7406	0,7939	0,6454	0,7875	0,7066	0,6516	0,7594	0,7006	47
	Micro avg	0,8231	0,8086	0,8158	0,7502	0,7374	0,7437	0,7546	0,7396	0,7470	187
	Macro avg	0,8443	0,7970	0,8086	0,7604	0,7398	0,7378	0,7630	0,7382	0,7392	187
Discarded prediction	1,77%			1,71%			1,98%				

- Zero-shotとは異なり古いモデル（4k-0301）の方が精度高
- ラベルの説明はあった方が良い
- One-shotよりもZero-shotの方が精度高

結果 | ChatGPT vs SETFIT

Label	SETFIT			GPT-3.5 (16k-0613), zero-shot			Support
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	
Bug	0.8723	0.8472	0.8590	0,7133	0,9811	0,8261	53
Documentation	0.9039	0.6594	0.7616	0,8853	0,6191	0,7285	32
Feature	0.7494	0.9182	0.8251	0,8861	0,8491	0,8672	55
Question	0.8754	0.8319	0.8528	0,8668	0,7719	0,8164	47
Micro average	0.8321	0.8321	0.8321	0,8155	0,8155	0,8155	187
Macro average	0.8502	0.8142	0.8246	0,8379	0,8053	0,8095	187

- ほとんど同等の精度を達成
 - ・ ファインチューニングなしで（訓練データがなくても），課題票のラベリングを行うことが可能

まとめ

- 目的
 - GPTベースのモデルが、課題票の自動ラベリングにどの程度有効か調査
- 結果
 - GPTモデルはZero-shotで十分な精度を発揮、ラベリングの労力の削減につながる可能性
- 所感
 - ラベルの違いによる影響とか、モデルの違いによる影響とかの議論がもっと欲しかった
 - いろんなプロンプトをためしてもいいんじゃないかなと思う（というかなぜしなかったの？）