

Delving into Commit-Issue Correlation to Enhance Commit Message Generation Models

1st Liran Wang
Beihang University
Beijing, China
wanglr@buaa.edu.cn

2nd Xunzhu Tang
University of Luxembourg
Luxembourg City, Luxembourg
xunzhu.tang@uni.lu

3rd Yichen He
Beihang University
Beijing, China
hyc2026@buaa.edu.cn

4th Changyu Ren
Beihang University
Beijing, China
cyren@buaa.edu.cn

5th Shuhua Shi
Beihang University
Beijing, China
shishuhua@buaa.edu.cn

6th Chaoran Yan
Beihang University
Beijing, China
ycr2345@buaa.edu.cn

7th Zhoujun Li*
Beihang University
Beijing, China
lizj@buaa.edu.cn

Proceedings of the 38th IEEE/ACM International Conference on Automated Software Engineering (ASE2023)

発表者：松尾 和輝

背景

- コミットメッセージ自動生成研究の増加[2][8][12]
 - 記述の手間を減らす[9]
 - ミスを起こしにくくする[4]
- 既存手法は変更内容と変更理由の両方を含む優れたコミットメッセージを自動生成しない傾向[20]
 - 変更差分は変更内容しか説明しない[4]
- コミットメッセージの多くはIssueと関連していることから、Issueを自動生成に活用できるのでは

[2] L. F. Cortés-Coy, M. Linares-Vásquez, J. Aponte, and D. Poshyvanyk, "On automatically generating commit messages via summarization of source code changes," in 2014 IEEE 14th International Working Conference on Source Code Analysis and Manipulation, 2014, pp. 275–284.

[4] Y. Tian, Y. Zhang, K.-J. Stol, L. Jiang, and H. Liu, "What makes a good commit message?" in Proceedings of the 44th International Conference on Software Engineering, ser. ICSE '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 2389–2401.

[8] S. Jiang, A. Armaly, and C. McMillan, "Automatically generating commit messages from diffs using neural machine translation," in 2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE), 2017, pp. 135–146.

[9] W. Tao, Y. Wang, E. Shi, L. Du, S. Han, H. Zhang, D. Zhang, and W. Zhang, "On the evaluation of commit message generation models: An experimental study," in 2021 IEEE International Conference on Software Maintenance and Evolution (ICSME), 2021, pp. 126–136.

[12] Z. Liu, X. Xia, A. E. Hassan, D. Lo, Z. Xing, and X. Wang, "Neural-machine-translation-based commit message generation: How far are we?" in Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, ser. ASE '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 373–384.

[20] H. Wang, X. Xia, D. Lo, Q. He, X. Wang, and J. Grundy, "Contextaware retrieval-based deep commit message generation," ACM Trans. Softw. Eng. Methodol., vol. 30, no. 4, jul 2021.

背景

- Issueには理由が記述されていることが多い[26][27]

「eval による MATCH の結果が正しくない」ために修正すべきバグがあることを理解することができる

Commit	
Sha:	319c868
Message:	Fix evaluation of MATCH expressions with eval()
Issue	
Number:	7160
Title:	Incorrect MATCH result with if/eval on a datetime value
Body:	Except for the first query (without any if/eval), all subsequent queries incorrectly return no results. In the repro section, all MATCH queries are equivalent and should produce identical results: returning one row with all of the created vertices in it.
Event:	<developer> closed this as completed in 319c868 on <date>.

Fig. 1: Correlation between commit and issue.

目的

コミットメッセージ自動生成にIssueを組み込む手法ExGriFiの構築

[26] J. Anvik, L. Hiew, and G. C. Murphy, "Who should fix this bug?" in Proceedings of the 28th International Conference on Software Engineering, ser. ICSE '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 361–370.
[27] J. Zhu, M. Zhou, and A. Mockus, "Effectiveness of code contribution: From patch-based to pull-request-based tools," in Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering, ser. FSE 2016. New York, NY, USA: Association for Computing Machinery, 2016, p. 871–882.

提案手法 | ExGroFi

- 概要図

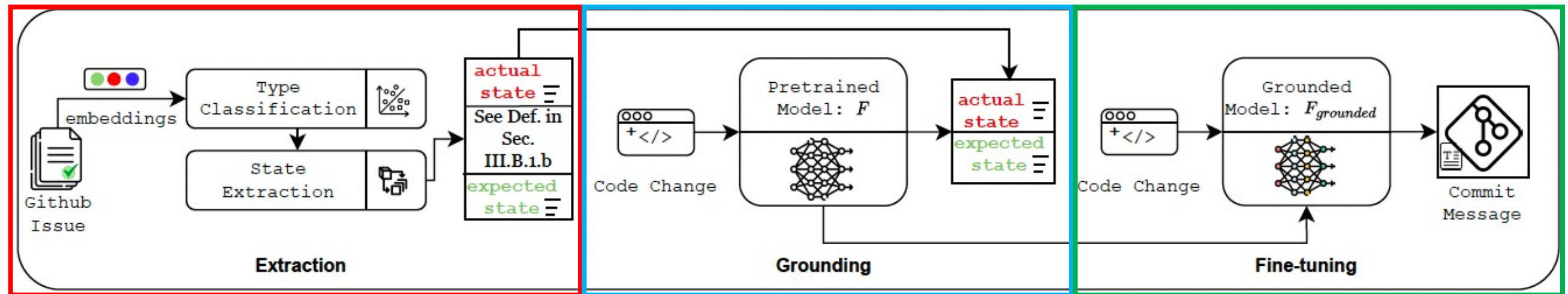
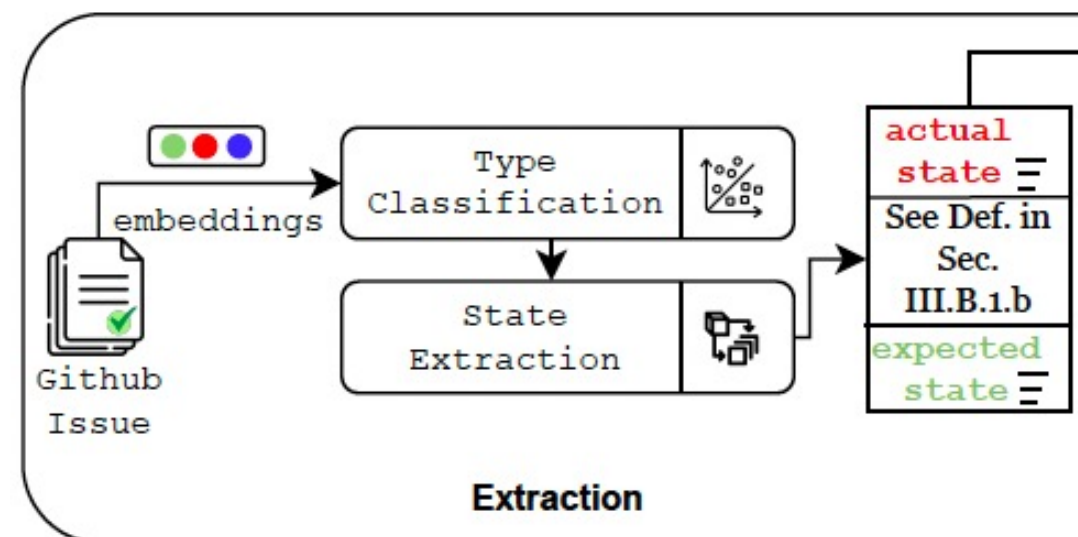


Fig. 3: Overview of *ExGroFi*.

- ①抽出
- ②グラウンディング
- ③ファインチューニング

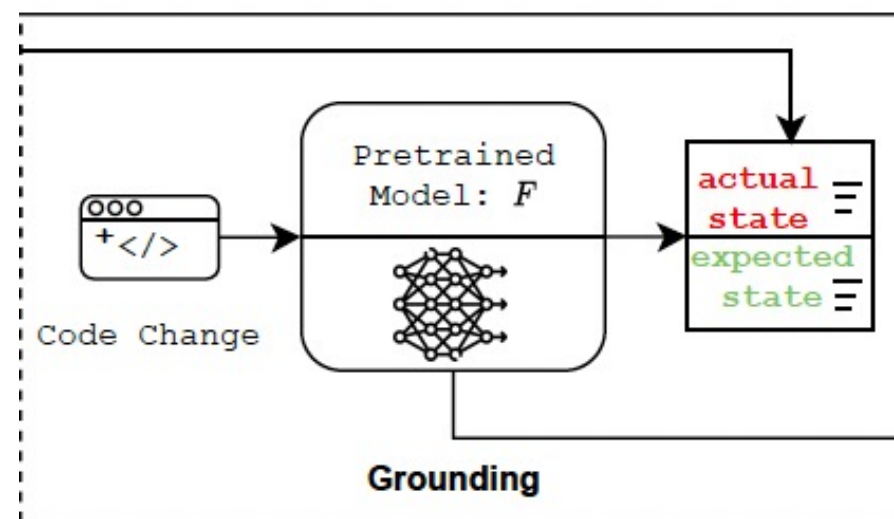
提案手法 | 抽出

- Issueを入力し以下の2つを分類するモデルを構築
 - Issueタイプ：Issueのラベル
 - バグ報告
 - 機能リクエスト
 - 機能強化
 - 状態情報：現在の状況や期待される結果に関する記述
 - 実際状態：e.g.) 「現状〇〇な問題が発生している」
 - 理想状態：e.g.) 「〇〇は□□であるべき」



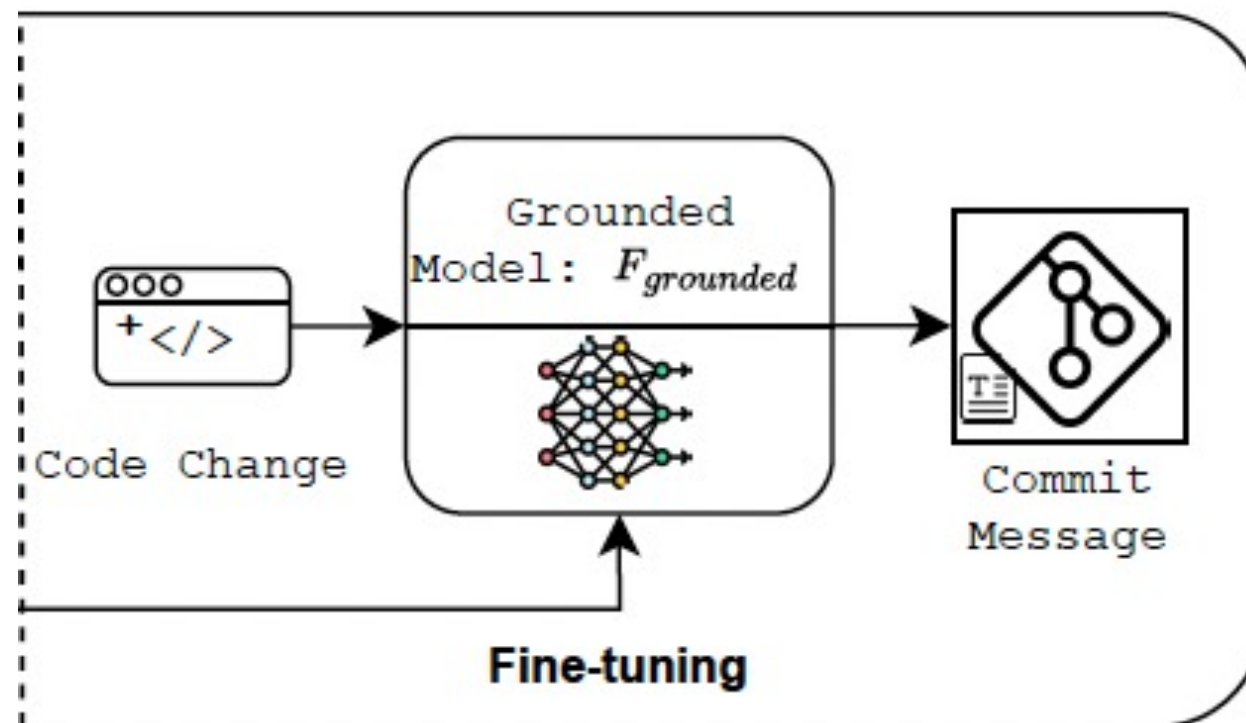
提案手法 | グラウンディング[55]

- 変更差分とIssueの情報を対応付け（やと思う）
 - 「りんご」と「Apple」を対応付けるイメージ
- 入力された変更差分をIssue情報に変換するようなモデルを学習



提案手法 | ファインチューニング

- グラウンディングで得られたモデルをコミットメッセージ自動生成用にファインチューニング
 - コード変更を入力とし，コミットメッセージを出力させるように学習



評価 | リサーチクエスチョン

- RQ1 : ExGroFiは既存のCMGモデルの性能をどの程度向上させることができるか？
- RQ2 : ExGroFiの第一段階はどの程度有効か？
- RQ3 : グラウンディングステージを通じてモデルは本当に有用な相関知識を学習するのか？
- RQ4 : ExGroFiで強化されたモデルは人間評価の観点からどのような性能を示すか？

評価 | データセット

- 対象リポジトリ
 - GitHubにあるスター数上位1,000のJavaリポジトリ
- 以下のデータを収集
 - コミットメッセージ
 - コミットに紐付くIssueタイトル, Issue本文
 - 変更差分
- フィルタリング
 - 情報をほとんど含まないコミットメッセージ
 - ボットが記述したコミットメッセージ
 - 1,024トークン以上のコミットメッセージ

評価 | データセット

- アノテーション
 - Issueタイプ：Issueのラベルに基づき分類
 - バグ報告
 - 機能リクエスト
 - 機能強化
 - 状態情報：Issueのテキストを分類
 - 実際状態：e.g.) 「現状〇〇な問題が発生している」
 - 理想状態：e.g.) 「〇〇は□□であるべき」
- アノテーション前の19,262件, アノテーション後の960件をデータセットとする

評価 | 評価指標

- RQ1：全体評価
 - BLEU^[61]
 - ROUGE-L^[62]
 - METEOR^[64]
 - CIDEr^[65]：BLEUに TF-IDFの考えを混ぜたもの？
- RQ4：人間評価
 - 合理性：コミットの背後にある”Why”を取り上げ，変更の論理的な説明を提供しているか
 - 包括性：コミットの背後にある”What”に対処するため，重要な詳細をすべて網羅し，行われた変更の全体像を提供しているか
 - 簡潔さ：情報を簡潔に伝え，読みやすさと迅速な理解を保証しているか
 - 表現力：内容が文法的に正しく流暢か

[61] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in Proceedings of the 40th annual meeting of the Association for Computational Linguistics, 2002, pp. 311–318.

[62] C.-Y. Lin, “Rouge: A package for automatic evaluation of summaries,” in Text summarization branches out, 2004, pp. 74–81.

[64] S. Banerjee and A. Lavie, “Meteor: An automatic metric for mt evaluation with improved correlation with human judgments,” in Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, 2005, pp. 65–72.

[65] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, “Cider: Consensusbased image description evaluation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 4566–4575.

結果 | RQ1 : 全体評価

- Issueを利用することで、生成性能を向上可能
 - Issueの知識を獲得しているため、入力の長さ制限に強いからでは？

TABLE I: Generation performance for *ExGroFi*-enhanced commit message generation models.

Model	BLEU			ROUGE-L			METEOR			CIDEr		
	Ori.	Enh.	Imp.	Ori.	Enh.	Imp.	Ori.	Enh.	Imp.	Ori.	Enh.	Imp.
CodeBERT [23]	8.81	9.09	3%	15.85	16.05	1%	4.75	5.0	5%	0.09	0.11	22%
CodeBERTa [57]	8.43	8.50	1%	14.86	15.66	5%	4.92	5.12	4%	0.09	0.11	22%
unixcoder [58]	9.46	9.54	1%	17.04	17.59	3%	5.57	6.18	11%	0.14	0.16	14%
encoder-only average	8.9	9.04	2%	15.92	16.43	3%	5.08	5.43	7%	0.11	0.13	18%
CodeTrans-small [24]	11.08	12.66	14%	18.62	20.84	12%	7.98	9.07	14%	0.3	0.46	53%
CodeTrans-base [24]	19.27	20.37	6%	27.34	28.53	4%	16.37	18.31	12%	1.0	1.08	8%
CodeT5-small [25]	11.40	12.59	10%	20.70	22.68	10%	8.86	10.02	13%	0.36	0.49	37%
CodeT5-base [25]	19.96	20.94	5%	30.07	31.46	5%	16.99	18.39	8%	1.04	1.11	7%
RACE [19]	21.06	22.47	7%	30.17	36.09	20%	17.23	18.64	8%	1.14	1.51	32%
encoder-decoder + hybrid average	16.55	17.81	8%	25.38	27.92	10%	13.486	14.886	10%	0.768	0.93	21%

“Ori.” means Original; “Enh.” is Enhanced; “Imp.” indicates Improvement.

結果 | RQ4：人間評価

- ExGroFiで生成されたものと既存手法で生成されたものを4人の開発者に示す
 - ExGroFiで生成された方が良いと判断したなら「Win」，そうでなければ「Lose」，変わらないなら「Tie」

TABLE III: Result of human evaluation.

Indicator	Win (N-%)	Lose (N-%)	Tie (N-%)	Kappa
合理性 (Why)	35 (70%)	10 (20%)	5 (10%)	0.70
包括性 (What)	25 (50%)	14 (28%)	11 (22%)	0.67
簡潔さ	21 (42%)	20 (40%)	9 (18%)	0.65
表現力	17 (34%)	17 (34%)	16 (32%)	0.66

N is short for number; % indicates percentage.

Rat. means Rationality. **Comp.** is Comprehensiveness.

Conc. is Conciseness. **Expr.** represents Expressiveness.

- 合理性については既存手法を凌駕

結果 | RQ4：人間評価

- 生成されるコミットメッセージの例
 - Before：既存手法で自動生成したコミットメッセージ
 - After：ExGroFiで自動生成したコミットメッセージ
 - Issue：コミットに関連するIssueのテキスト

TABLE IV: Examples that demonstrate *ExGroFi*'s effect.⁹

1	Before	Add support for custom field.
	After	Fix TimingKey to reclaim the memory.
	Issue	... There is no mechanism to reclaim the memory after the filters are unreachable ...
2	Before	Reduce visualisation to fix issue.
	After	Fix exception with error handler in Mono.subscribe().
	Issue	... Error handler in Mono.subscribe() should handle exceptions thrown by subscriber ...
3	Before	Update process instance.
	After	Fix bug when starting process instance.
	Issue	... No tenant is set when starting process instance Expect a tenant user can start a process instance in task application ...

“Before” and “After” represent the commit messages generated by the model before and after using *ExGroFi*, respectively.

“Issue” provides part of the text in the related issue; among them, the **red text** indicates the actual state, and the **green text** indicates the expected state.

まとめ

- 目的
 - コミットメッセージ自動生成にIssueを組み込む手法の構築
- 結果
 - 既存手法より性能向上
 - 変更理由を含むコミットメッセージの自動生成にも対応

所感

- 機械学習の知識が不足してるのでちよくちよく理解できない所があった
 - 単語のベクトル化とかもよーわからん
- Issueのコメント情報とかは使わないのか…
 - むしろノイズになるのかも？

結果 | RQ2 : 抽出性能

- 抽出した情報を使わないケース(表左列)と, 使うケース(表右列)
 - 抽出した情報を使う方が精度が上がる

TABLE II: State information extraction performance

Approach		codebert	+ issue type
Actual State	P	56.90	58.93
	R	63.46	64.71
	F1	60.00	61.68
Expected State	P	80.77	84.91
	R	72.41	77.59
	F1	76.36	81.08
Micro-F1		68.18	71.56

結果 | RQ3 : グラウンディング性能

- 既存のCodeT5-baseと, ExGroFiで学習させたあとのCodeT5-baseのエンコーダを使用
- コミットメッセージと変更差分をそれぞれ埋め込み, ベクトルを取得
- 標準化したのちユークリッド距離を計算
 - 距離が近いほど上手に対応付けできている
- 既存(Before)とExGroFi(After)の全テストデータにおけるユークリッド距離を比較

