

# コード差分に加え課題追跡システムにおける投稿と一連の返信を活用したコミットメッセージ生成

松田蒼唯\* 早瀬康裕† 天笠俊之‡

\* 筑波大学情報学群情報科学類 † 筑波大学計算科学研究センター

## 1 序論

コミットメッセージは開発者にとってソフトウェアを理解するのに役に立つ。ソフトウェア開発をする際、開発者はソースコードを変更しバージョン管理システムにコミットという単位で記録する。また開発者はコミットに対してコミットメッセージを付与することができる。コミットメッセージは自然言語のテキストであり、その変更が何をしたもので(変更の概要)、なぜその変更が行われたのか(変更の理由)を説明することが期待されている[1]。開発者はコミットメッセージを読むことで、変更内容を細かく見なくても変更の概要と理由を素早く知ることができる。

しかし、開発者にとってコミットメッセージを書くことは難しいタスクだとされている。開発者はコミットメッセージを読んだ時に変更の理由がわからないことがある。その原因としてコミットを記録する開発者がコミットメッセージを書く時に変更の理由が思いつかないことが挙げられる。

本研究では、開発者がコミットメッセージを読んで変更の理由がわかるようになることを目的として、コード差分に加え課題システムにおける投稿と一連の返信を入力として大規模言語モデルに与え、変更の理由を説明するコミットメッセージを生成させる手法を提案する。課題追跡システムとはソフトウェア開発において発生する課題をリスト化し、進捗を管理するシステムである。開発者は課題を掲示板のようにして議論し、議論を動機に変更を行うことがある。そこで、動機となる可能性の高い投稿とその一連の返信を大規模言語モデルの入力に加えることで、変更の理由を説明するコミットメッセージが生成できると考えられる。

## 2 関連研究

開発者のコミットメッセージ作成支援を目的として、コード差分を入力として機械学習モデルを構築し、コミットメッセージを自動生成する手法が提案されてきた。代表的な例として Shi ら [2] が提案した検索拡張型

Commit message generation utilizing submitted issues and their series of replies in Issue Tracking System

Aoi MATSUDA\*, Yasuhiro HAYASE †, Toshiyuki AMAGASA ‡

\*College of Information Science, University of Tsukuba

‡Center for Computational Sciences, University of Tsukuba

のニューラル・コミットメッセージ生成手法、RACE が挙げられる。この手法では、検索された類似コミットを模範として扱い、それを利用して正確なコミットメッセージを生成する。

さらに、課題追跡システムの投稿を機械学習モデルの学習時に活用することで、コミットメッセージ生成の性能を向上させる研究も行われている。その代表的な例が Lian ら [3] が提案した ExGroFi である。ExGroFi は、まず課題追跡システムの投稿からソフトウェアに関する情報を抽出し、この抽出した情報を出力目標、コード差分を入力として機械学習モデルを訓練する。さらにこの機械学習モデルをコード差分を入力、コミットメッセージを出力目標としてファインチューニングする。これにより課題追跡システムの投稿に関する知識をもった機械学習モデルが得られる。

## 3 提案手法

本研究では、開発者がコミットメッセージを読んで変更の理由がわかるようになることを目的として、コード差分に加え課題システムにおける投稿と一連の返信を入力として大規模言語モデルに与え、変更の理由を説明するコミットメッセージを生成させる手法を提案する。提案手法の全体像を図1で示す。まず、大規模言語モデルにコミットメッセージを生成させるための準備として開発者から今コミットしようとしているコード差分(diff)とコミットの直接の背景となる課題追跡システム上の議論(投稿と一連の返信)の2つの要素を受け取り、この2つの要素と雛型を組み合わせ、コミットメッセージ生成の指示を作成する。そして、作成したコミットメッセージ生成の指示を大規模言語モデルに入力し、指示に基づいて変更の理由を説明するコミットメッセージ候補を生成させる。

図1の提案手法の全体像に沿って、各段階の処理内容を説明する。

開発者から与えられたコード差分と課題追跡システムにおける投稿と一連の返信を受け取り、これらを雛型に組み込み、コミットメッセージ生成の指示を作成する(図1①)。雛型はコミットメッセージ生成の指示の見本である。この雛形は、大規模言語モデルが実行すべきタスクの説明、タスクの補足情報、タスクを実行させるために必要な入力、実行した結果の出力形式の4つの章で構成されている。受け取った要素のうちコード差分を、タスクを実行させるために必要な入力

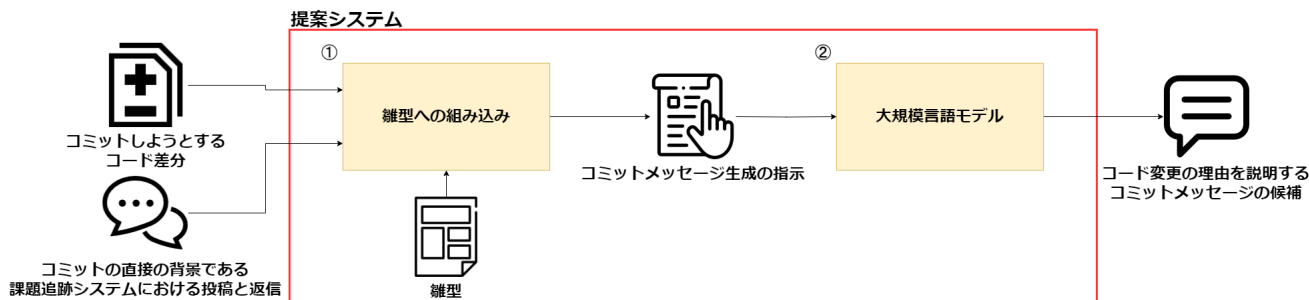


図 1: 提案手法全体像

として組み込む。残りの要素である、課題システムにおける投稿と一連の返信を、作成時刻が早い順に上からタスクの補足情報に組み込む。これによりコミットメッセージ生成の指示が作成される。

コミットメッセージ生成の指示に基づいて大規模言語モデルが変更の理由を説明するコミットメッセージ候補を作成する (図 1 ②)。大規模言語モデルはコミットメッセージ候補を 3 つ作成し、作成した時刻が早い順に開発者にコミットメッセージの候補を提示する。

## 4 実験

2つのコミットメッセージ生成手法のうち、どちらがより良い変更の理由を説明するコミットメッセージを生成できるかを明らかにする。2つの手法とは (a) コード差分を入力として大規模言語モデルにコミットメッセージを生成させる手法と (b) コード差分に加え課題追跡システムにおける投稿とその一連の返信を入力として大規模言語モデルにコミットメッセージを生成させる手法である。各々の手法によって生成されるコミットメッセージのうち、実際に開発者が書いたコミットメッセージ (以後、正解コミットメッセージと呼ぶ) と似た変更の理由が説明されていると判断された割合が高い方が、よりよい変更の理由を説明できていると言える。

そこで、2つの手法を比較するためシングルブラインドなアンケート調査を行う。アンケート調査では、評価者はコミットメッセージのペアを読み、変更の理由が似ているかどうかを評価する。ただし、評価者はペアのうちどちらが正解コミットメッセージでどちらが手法 (a) あるいは手法 (b) によって生成されたコミットメッセージであるのかわからない。そのため評価者はコミットメッセージのペアに対して「変更の理由が似ている」「変更の理由が異なる」「どちらかに変更の理由が欠けている」「両方とも変更の理由が欠けている」のいずれかに分類する。評価対象は 1 人あたり 18 ペアある。18 ペアのうち 9 ペアは正解コミットメッセージと手法 (a) で生成したコミットメッセージのペア、残りの 9 ペアは正解コミットメッセージと手法 (b) で生成したコミットメッセージのペアである。ソフトウェア開発経験が 1 年以上の大学生 2 名と大学院生 4 名に評価をしてもらう。

アンケート調査結果を分析し、表 1 に示した。分類

1 が 18 ペアから 25 ペアに向上し、分類 2 が 21 ペアから 11 ペアに減少した。このことから提案手法の方がより良い変更の理由の説明をするコミットメッセージを生成できていることがわかった。

Table 1: 予備実験結果

分類	手法 (a)	手法 (b)
1. 正解コミットメッセージのコード変更の理由と大規模言語モデルが生成したコミットメッセージのコード変更の理由は似ている。	18	25
2. 正解コミットメッセージのコード変更の理由と大規模言語モデルが生成したコミットメッセージのコード変更の理由は異なる。	21	11
3. 正解コミットメッセージはコード変更の理由が欠けている。	13	17
4. 大規模言語モデルが生成したコミットメッセージはコード変更の理由が欠けている。	1	1
5. 正解コミットメッセージと大規模言語モデルが生成したコミットメッセージはコード変更の理由が欠けている。	1	0

## 5 結論

本稿では、コード差分に加え課題システムにおける投稿と一連の返信を入力として大規模言語モデルに与え、変更の理由を説明するコミットメッセージを生成させる手法を提案した。今後、実際にオープンソースプロジェクトに携わる開発者にアンケート調査を行い本手法の有効性を考察する予定である。

## References

- [1] Yingchen Tian, Yuxia Zhang, Klaas-Jan Stol, Lin Jiang, and Hui Liu. What makes a good commit message? *CoRR*, 2022.
- [2] Ensheng Shi, Yanlin Wang, Wei Tao, Lun Du, Hongyu Zhang, Shi Han, Dongmei Zhang, and Hongbin Sun. Race: Retrieval-augmented commit message generation, 2022.
- [3] Liran Wang, Xunzhu Tang, Yichen He, Changyu Ren, Shuhua Shi, Chaoran Yan, and Zhoujun Li. Delving into commit-issue correlation to enhance commit message generation models, 2023.