

The Fact Selection Problem in LLM-Based Program Repair

Nikhil Parasaram and Huijie Yan and Boyu Yang and Zineb
Flahy and Abriele Qudsi and Damian Ziaber and Earl Barr and
Sergey Mechtaev

lcse 2025

背景と課題

自動プログラム修正(APR)

- バグを修正する際は様々な情報を活用する
- 精度にはプロンプトの質が大きく影響する
- LLMの発展によりコンテキストウィンドウは増大している

最適なプロンプトはどんなものか？

- バグを正しく修正するために必要な情報（事実）はなにか
- どれぐらいの情報（事実）が必要なのか
- 事実選択問題

先行研究と課題

個別の成果

- コードの文脈, GitHubのIssue, スタックトレースなど, それぞれの事実が精度を向上させることは確認済み

欠如している点

- 複数の事実を与えた際の効果の変化
- 選択（複数の事実から一部のみを与える）という視点
- Angelic Valuesなどの, 経験的に有用とされている情報の効果

研究内容

目的

- APRの際にLLMの性能を最大限発揮するためのプロンプトを自動的に作成することを可能にする
- 事実選択問題を定義する

アプローチ

- ARPに活用できる「事実」及び、事実選択問題の定義
- 「MANIPLE」の開発

事実選択問題

バグ修正に役立つ7つの事実

- バグのあるクラス, 使用されたメソッド, 失敗するテスト
- エラー情報, 実行時情報, Angelic Forest
- GitHub Issue

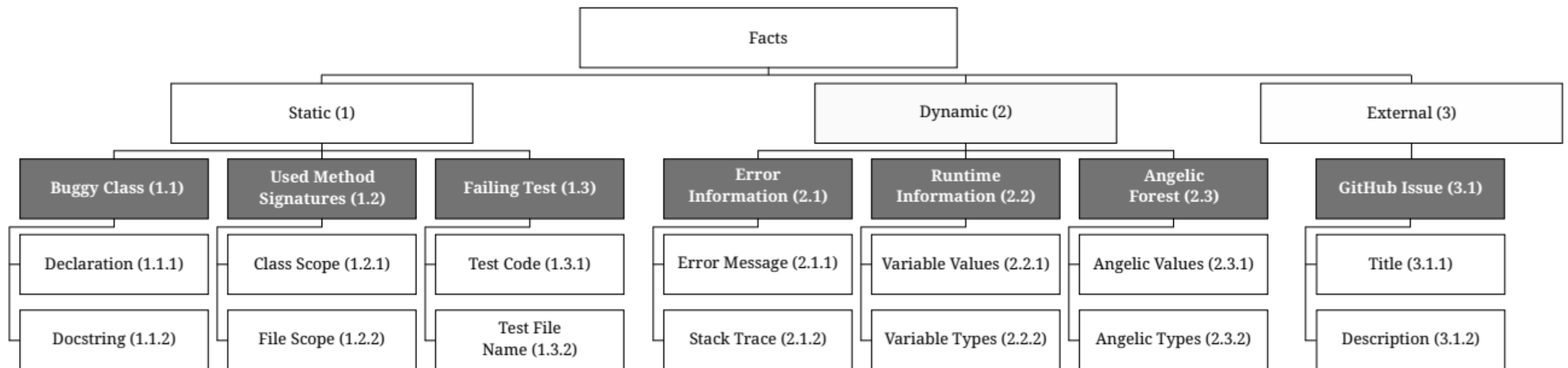


Fig. 3: This work uses seven facts (dark rectangles) across three categories for constructing program repair prompts. Each fact is composed of related pieces of information. Each prompt contains the buggy function to be repaired; the facts can be included based on the employed fact selection strategy.

事実選択問題

定義

- バグ修正の成功率を最大化するための、プロンプトに含めるべき情報の最適な組み合わせを求める問題

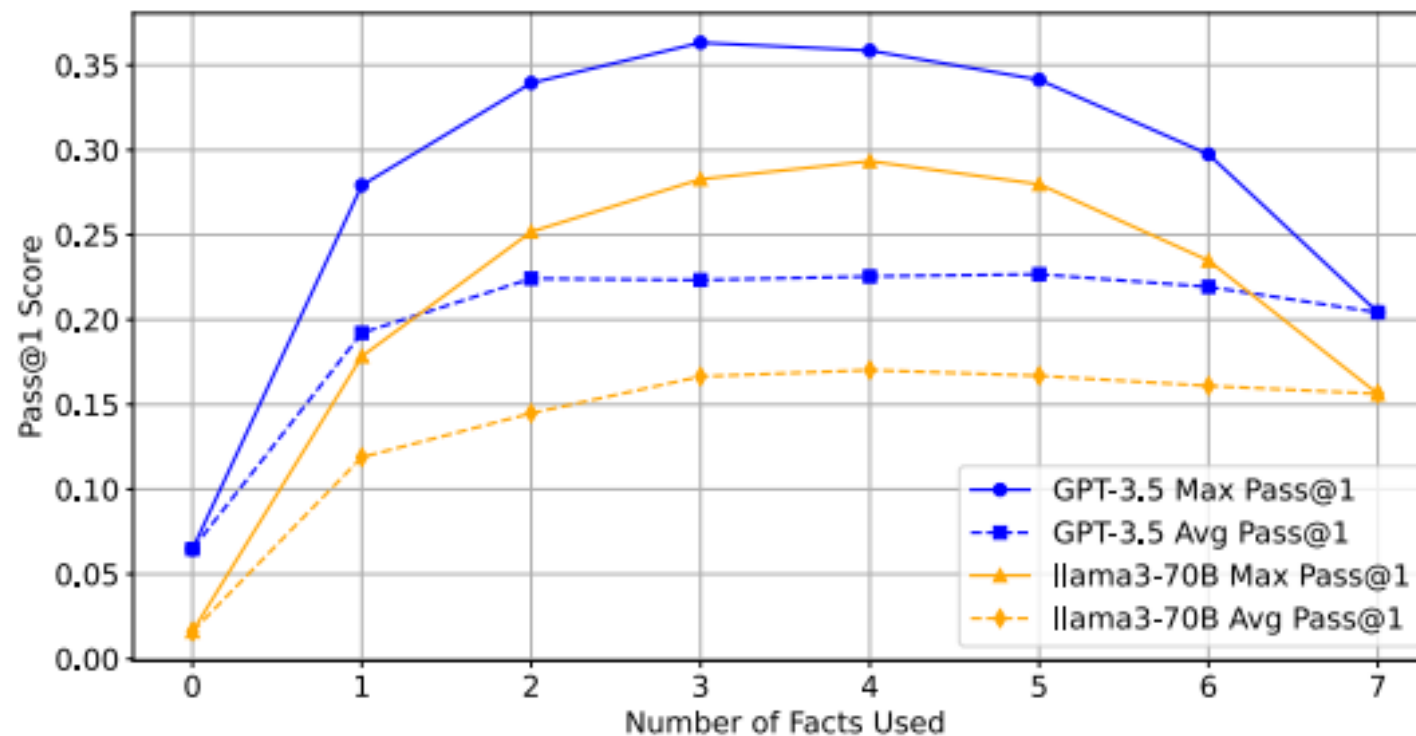
問題の必要性

- 3つの証明
 1. それぞれの事実が有用であること
 2. 全て入れれば良いわけではないこと
 3. 共通の最適解があるわけではないこと

事実選択問題

問題の必要性

- 3つの証明
 1. それぞれの事実が有用であること
 2. 全て入れれば良いわけではないこと
 3. 共通の最適解があるわけではないこと



事実選択問題

問題の必要性

- 3つの証明
 1. それぞれの事実が有用であること
 2. 全て入れれば良いわけではないこと
 3. 共通の最適解があるわけではないこと

Project	GPT-3.5			Llama3-70B		
	Best Fact Set	Project	Total	Best Fact Set	Project	Total
luigi	0011111	0.53	0.26	1010111	0.42	0.16
black	0010111	0.38	0.22	1101011	0.35	0.17
fastapi	0000101	0.19	0.18	0011001	0.11	0.11
httpie	0001111	0.50	0.25	0001000	0.50	0.12
pandas	0011001	0.25	0.25	0101011	0.13	0.17
tornado	0011000	0.37	0.20	1001001	0.22	0.13
ansible	0010101	0.10	0.12	0011011	0.15	0.14
matplotlib	0001001	0.36	0.25	0101010	0.26	0.16
cookiecutter	0001101	0.93	0.20	0001010	0.67	0.14
tqdm	0001111	0.00	0.24	1011001	0.00	0.13
youtube-dl	0011001	0.08	0.25	0001110	0.06	0.14
keras	0101111	0.32	0.23	1011110	0.17	0.15
scrapy	0011111	0.45	0.26	1101110	0.36	0.15
sanic	0010101	0.64	0.21	0110011	0.51	0.15
thefuck	0001001	0.16	0.21	0001001	0.29	0.12

提案手法

MANIPLE

- あるバグとそのバグに関する事実の組み合わせが与えられたときに、バグ修正が成功する確率を求めるモデル
- あるバグに対する最も成功確率が高いと思われる事実の組み合わせを求める

学習データ

- BugsInPyから抽出した157個のバグを基にした約9500件のプロンプトとその実行結果（pass@1スコア）のうち、各バグごとに上位5件
- 四つの特徴量（ビットベクトル、リポジトリID、プロンプト長、サイクロマティック複雑度）を与える

提案手法

MANIPLE

- あるバグとそのバグに関する事実の組み合わせが与えられたときに、バグ修正が成功する確率を求めるモデル
- あるバグに対する最も成功確率が高いと思われる事実の組み合わせを求める

使用時

- 試したいパターンの四つの特徴量を与えると、最も成功確率が高いと思われるパターンが出力される
- 内部ではランダムフォレストによる分類が行われている

既存手法との比較

比較対象

- T0 : バグのあるコードのみを与える手法
- T1 : コード+クラス情報やメソッド情報
- T2 : コード+GitHub Issue
- T3 : コード+エラー情報

Tool	#fixed	Correct	% Correct
T_0	44	7	16%
T_1	37	5	14%
T_2	63	18	29%
T_3	66	31	47%
MANIPLE	88	37	42%

- #fixed : テストをすべて通過したかどうか
- Correct : 人間が正しく修正できていると判断した物

まとめ

バグに関連する「事実」は多すぎても少なすぎても良くない

事実選択問題の定義

バグによって最適な組み合わせがある