

Demystifying LLM-Based Software Engineering Agents

Chunqiu Steven Xia, Yinlin Deng, Soren Dunn,
Lingming Zhang

Proceedings of the ACM on Software Engineering, Volume 2,
Issue FSE(FSE2025), Pages 801 – 824

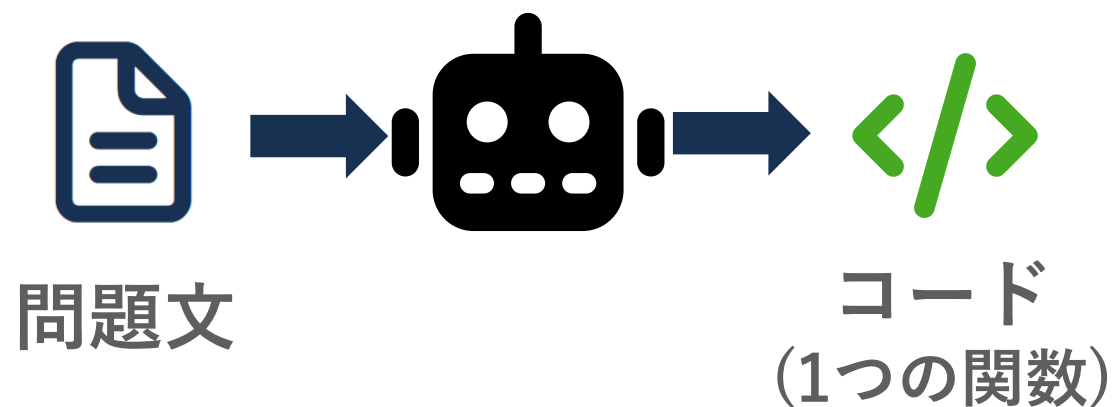
S2310157 谷口

背景：コード生成のベンチマーク

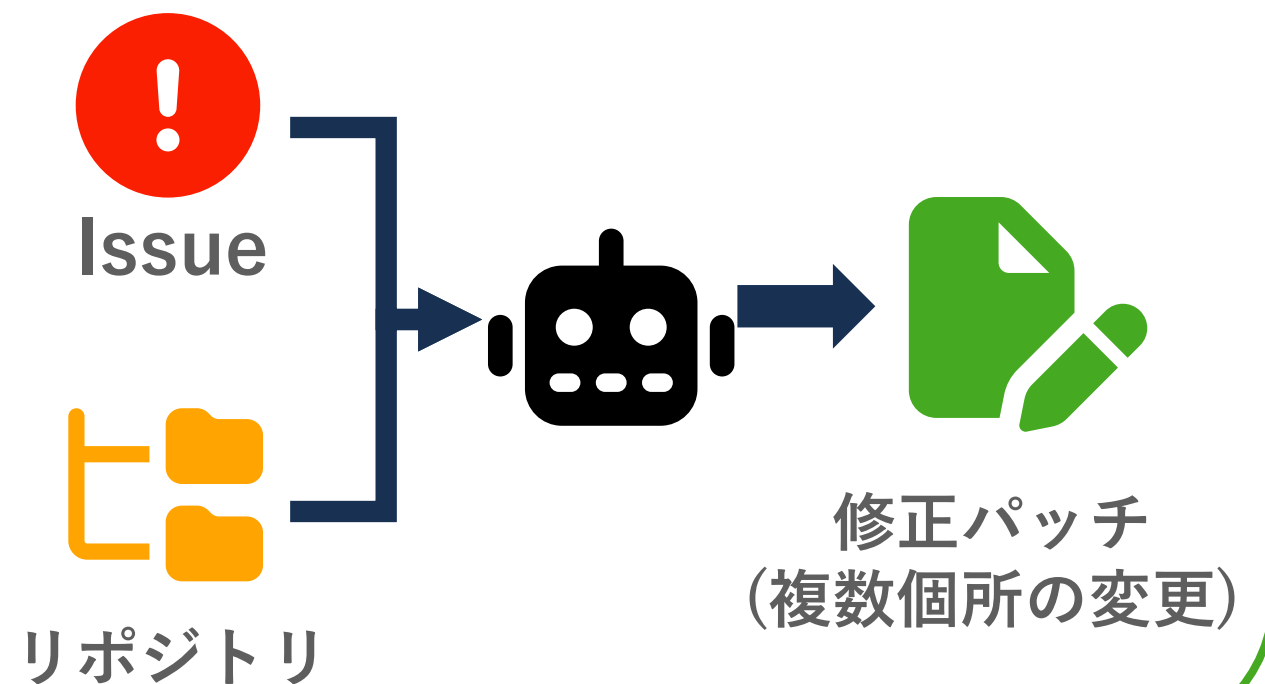
関数レベル⇒リポジトリレベル

- LLMのコード生成能力の進歩

関数レベルのベンチマーク
(HumanEval, MBPPなど)



レポジトリレベルのベンチマーク
(SWE-benchなど)

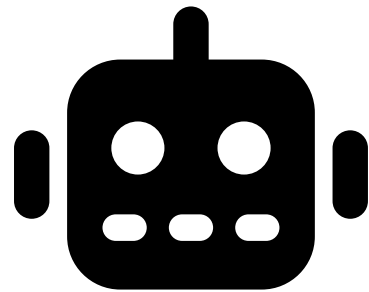


- より現実の問題に近い、複雑なベンチマークでの評価に移行

背景：エージェント方式

agent

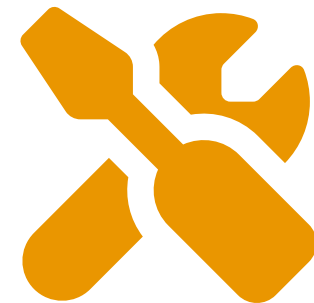
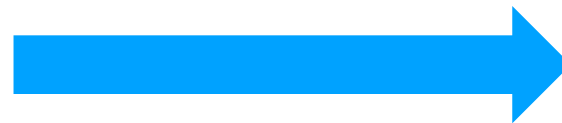
- ・ LLMと外部ツールを連携させ自律的に判断行動させる仕組み



ツール使用を判断

フィードバックから
更なる意思決定

ツール呼び出し



検索・編集・コマンド実行

実行結果をフィードバックする

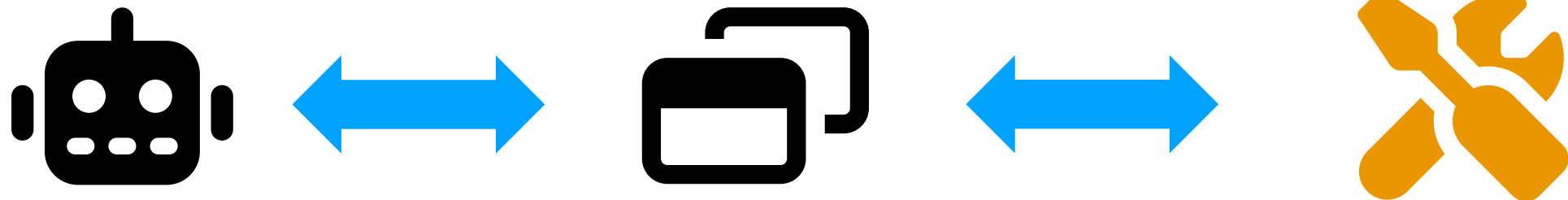


フィードバック

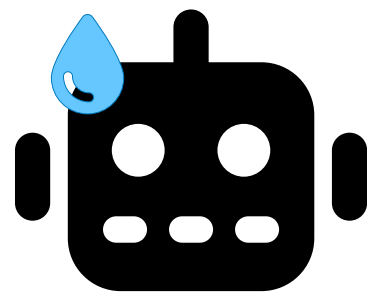
背景：エージェント方式は最適？

エージェント方式の問題点

- ・ LLMと外部ツールを繋ぐ抽象層の設計

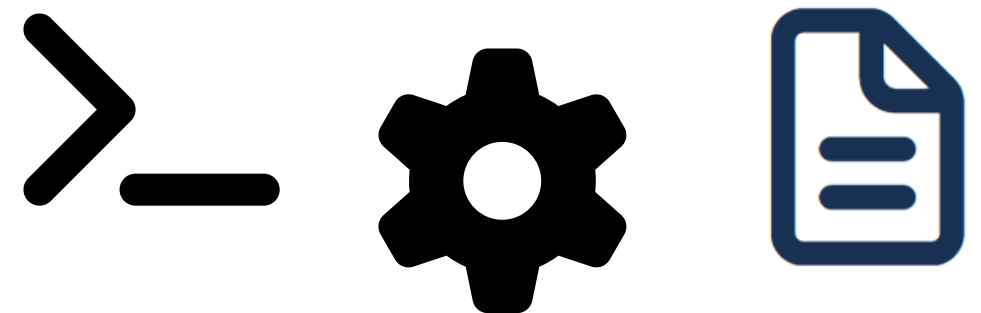


- ・ 不十分な自己制御/自己反省能力



混乱するLLM

大量の情報・選択肢

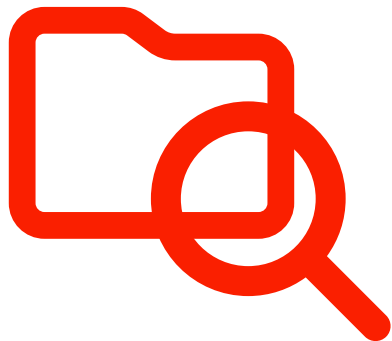


- ・ エージェント方式は最適なのか

エージェントを用いない手法

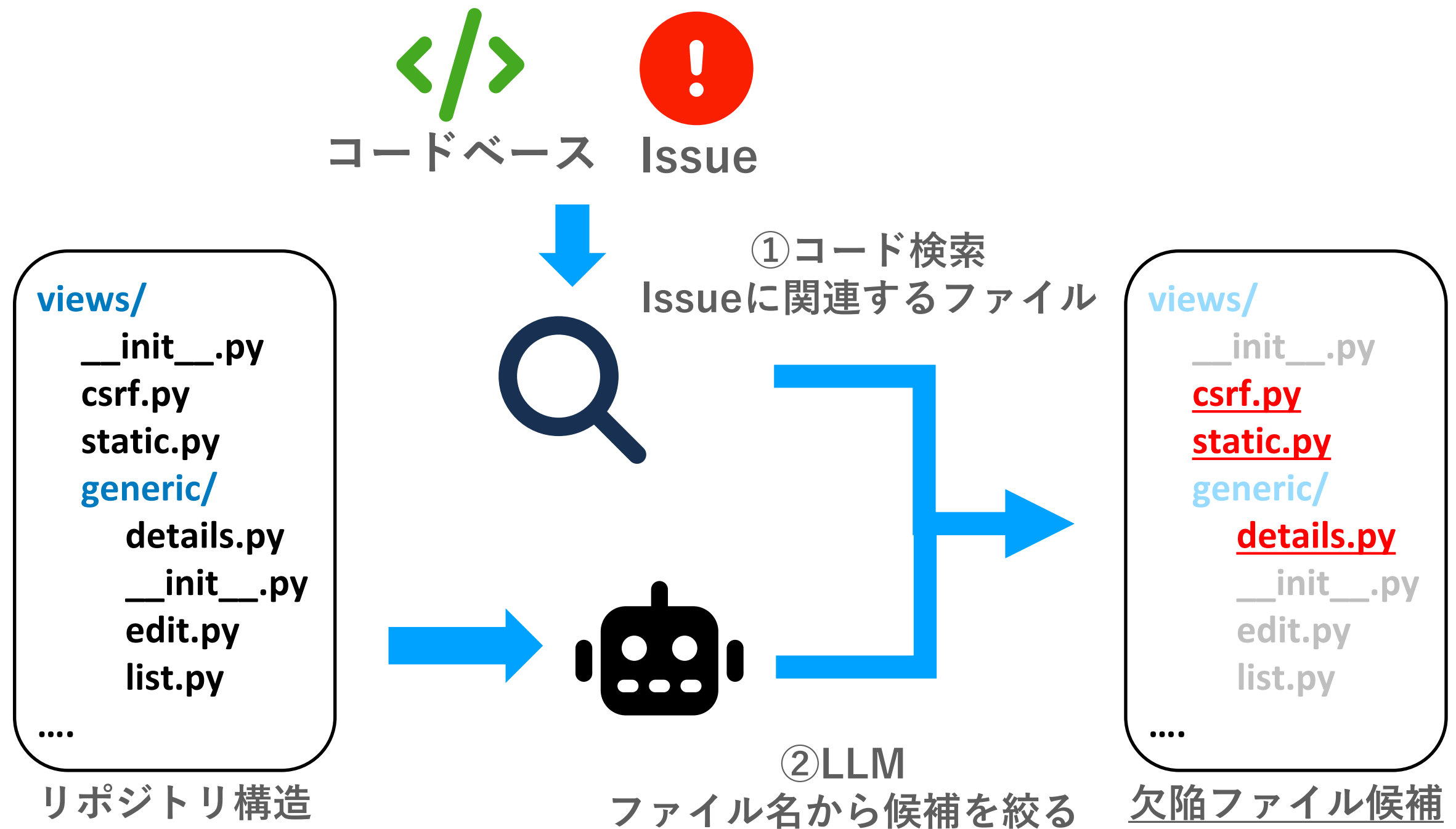
提案手法 : Agentless

- ・ 単純な3プロセスに従った問題解決
 - 局所化 ⇒ 修正 ⇒ パッチ検証



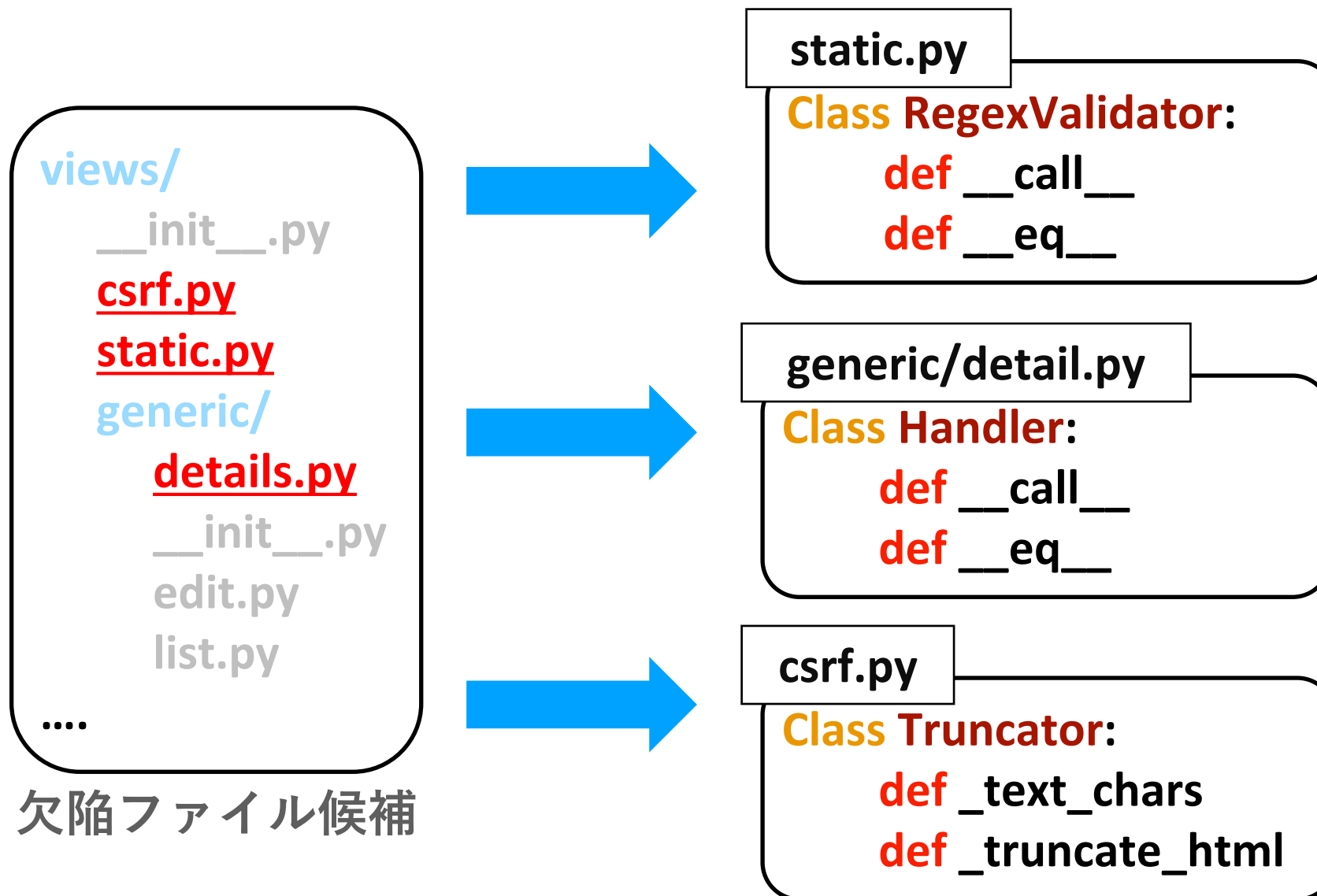
局所化

ファイルの特定

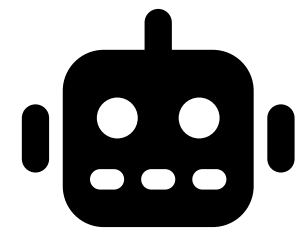


局所化

編集個所の特定



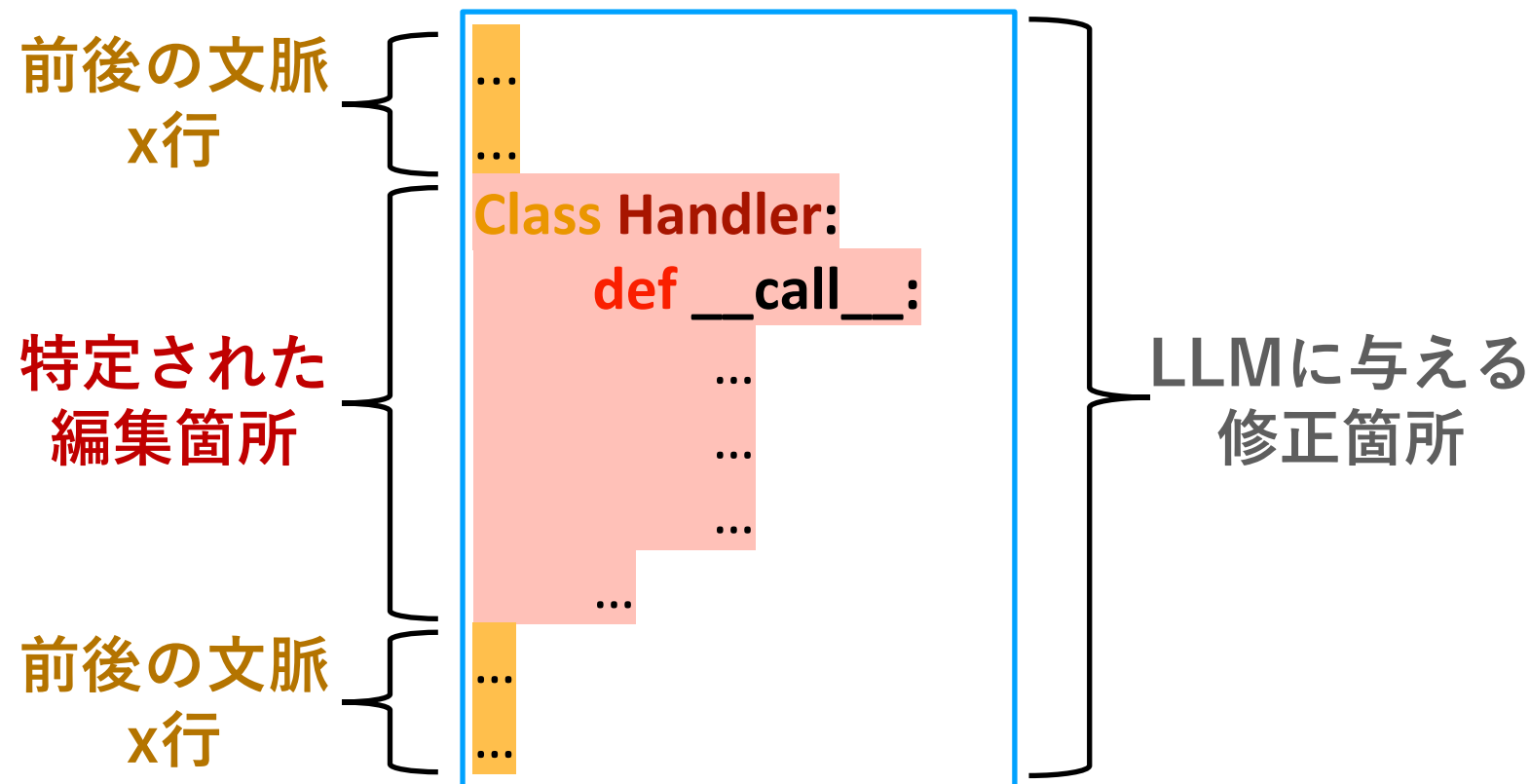
クラス名、関数名、変数宣言のみを抽出
(スケルトン形式)



編集すべき個所を
LLMが判断

修正

パッチ生成



- Search/Replace Edit
- 小さな修正に集中させ幻覚防止

Search/Replace Edit

Search (置き換え元のコード)

```
Class Handler:
    def __call__:
        ...
        ...
    ...
```



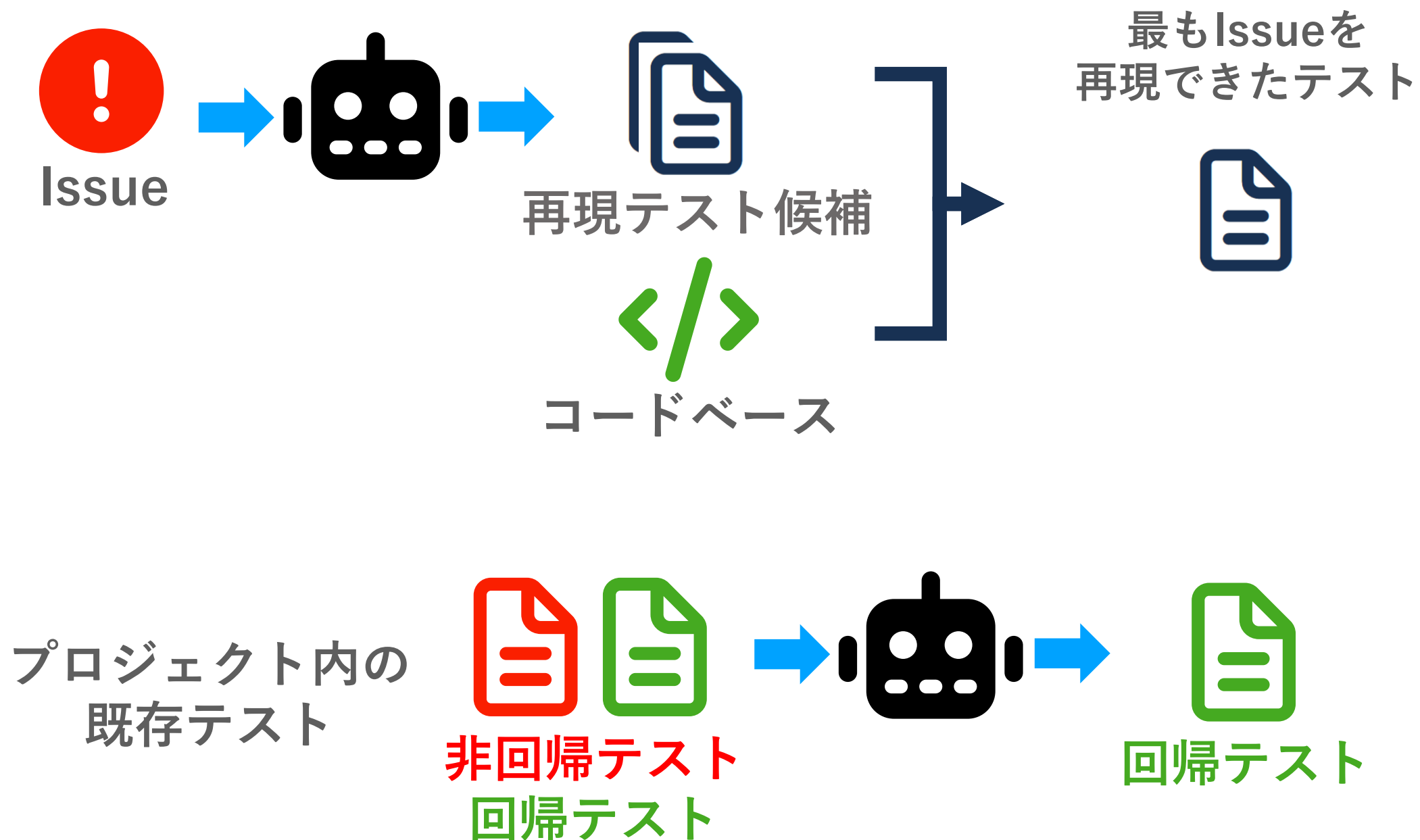
Replace (置き換え後のコード)

```
Class RepairedHandler:
    def __repairedcall__:
        ...
        ...
    ...
```

修正前と修正後の両方を
出力させる

Agentless : パッチ検証

再現テスト生成



Agentless : パッチ検証

再現テスト生成



実験

実験設定

- LLM: GPT-4o (gpt-4o-2024-05-13)
 - temperature = 0 (サンプリング時のみ0.8)
- 埋め込み : text-embedding-3-small [OpenAI 2024b]
- データセット : SWE-bench Lite
- 26件のエージェントベース手法と比較
 - 2024年10月時点の公式SWE-bench リーダーボード上位
- 解決率とコストで評価

実験結果 + 比較(一部省略)

オープンソースの中では一位

ツール	LLM	解決率	平均コスト(\$)	平均トークン
Agentless	GPT-4o	96 (32.00%)	\$0.70	78,165
AutoCodeRover-v2	GPT-4o	92 (30.67%)	-	-
RepoGraph	GPT-4o	89 (29.67%)	-	-
Moatless	Claude 3.5 S	80 (26.67%)	\$0.17	
...(2件省略)...	...	...	...	...
SWE-agent	Claude 3.5 S	69 (23.00%)	\$1.62	521,208
AppMap Navie	GPT-4o	65 (21.67%)	-	-
AutoCodeRover	GPT-4	57 (19.00%)	\$0.45	38,663

実験結果 + 比較(一部省略)

クローズドソースでは中位

ツール	LLM	解決率	平均コスト(\$)	平均トークン
CodeStory Aide	GPT-4o + Claude 3.5 S	129 (43.00%)	-	-
...(4件省略)...	...	...	...	...
Isoform	-	105 (35.00%)	-	41,963
...(2件省略)...	...	...	-	-
Agentless	GPT-4o	96 (32.00%)	\$0.70	78,165
SpecRover	GPT-4o + Claude 3.5 S	93 (31.00%)	\$0.65	-
CodeR	GPT-4	85 (28.33%)	\$3.34	323,802
...(1件省略)...	...	...	...	...
SIMA	GPT-4o	83 (27.67%)	\$0.82	-
...(4件省略)...	...	...	...	...

解決率一位ではないが、多くのエージェントのコストより低い

Agentlessの採用

商用ツールでの採用

- 公開後、様々なツールがAgentlessの仕組みを採用
- Agentlessの初期版に基づくツール : RepoGraph, Isoform

モデルのデモンストレーション

- SWE-bench上でのモデル性能評価にAgentlessを使用
 - OpenAI : GPT-4o, o1
 - DeepSeek : V3, R1

まとめ

エージェントに頼らないAgentless

- ・ エージェントは自律的判断で混乱しがち
- ・ 決まったプロセスで問題を解決するAgentless
 - 局所化⇒修正⇒パッチ検証の決まったプロセス

低コスト高パフォーマンス

- ・ オープンソースの中では最も高い解決率(SWE-bench Lite)
- ・ エージェントベースの多くにコスト面で勝っている
- ・ 新モデルの性能評価にも使用されている

所感

スケルトン形式が面白い

- ・ 保守が為されていること前提ではあるが
為されてないものは
そもそも大した規模ではない

```
Class RegexValidator:  
    def __call__  
    def __eq__
```

クラス、関数、変数宣言のみを
抽出した形

LLMにとっても全文読むのは負担

- ・ 人間の開発者も大まかにあたりを付けてから読むと思われる
- ・ LLMにもあたりを付ける作業が必要
- ・ LLMも他ツールによる作業工程の自動化の恩恵を受ける
- ・ エージェントが真に必要な場面とは