

LLM-based Agents for Automated Bug Fixing: How Far Are We?

Xiangxin Meng
ByteDance
Beijing, China
mengxiangxin.1219@bytedance.com

Pengfei Gao
ByteDance
Beijing, China
gaopengfei.se@bytedance.com

Zexiong Ma
ByteDance
Beijing, China mazexiong@bytedance.com

Chao Peng (Corresponding author)
ByteDance
Beijing, China pengchao.x@bytedance.com

背景

LLMベースエージェントの登場

- 環境と相互作用しながら特定タスクを遂行



複雑なタスクに対応可能に

リポジトリの自動バグ修正で大きな影響力を持つようになった

エージェント開発の現状

多様な設計思想

ReActベース・モンテカルロ木探索などの問題解決のスタイル
産業界・学界の両方で活発に開発されている

産業界

- W&B Programmer
- Blackbox AI Agnet
- MarsCode Agen など

学界

- Learn-by-interact
- AutoCOdeRover
- SWE-agent など

評価

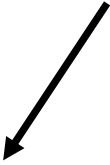
SWE-benchがよく利用される

リーダーボードに毎週新しいシステムが投稿される

エージェント開発の現状

上位のシステムの能力差を体系的に分析されていない

- ・ どの問題を各システムが解け、どれを共通して解けないのか
- ・ システム間の能力差を生んでいる要因は何か
- ・ 修正成功の前提となるサブタスクで各システムがどの程度の性能か
サブタスク：欠陥位置予測・バグ再現プロセスなど



リーダーボードに毎週新しいシステムが投稿される

目的・RQs

目的

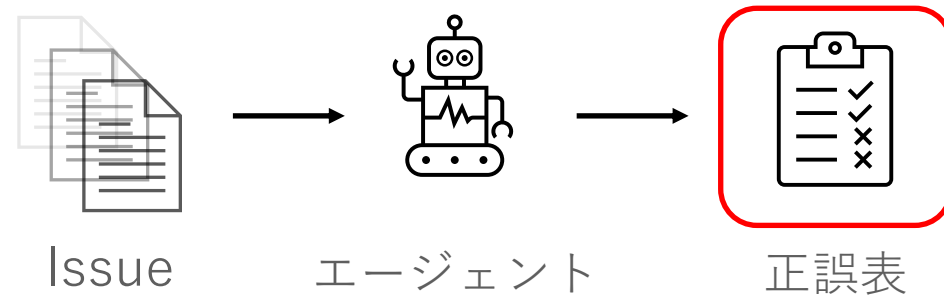
LLMベースエージェントによる自動バグ修正における
現状の到達点と限界を多面的に明らかにする

RQs

- 各システムが解ける問題はどのように異なり，その差はissue記述の品質とどう関係しているのか
- 各システムはFLでどの程度の性能を出しているのか
- バグ再現タスクはどの程度うまく機能しているか

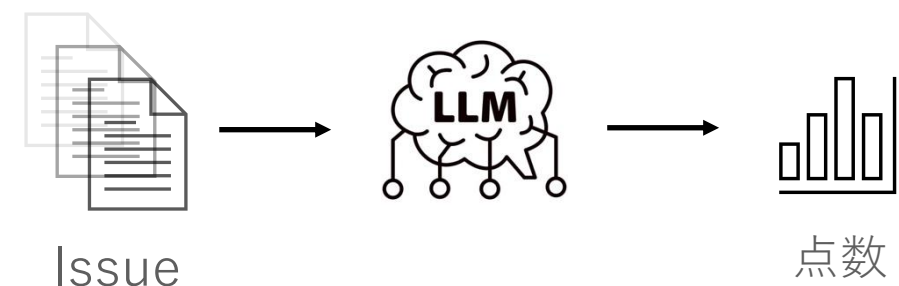
RQ1：システムの有効性 | 実験手順

1, 解決可否の集計



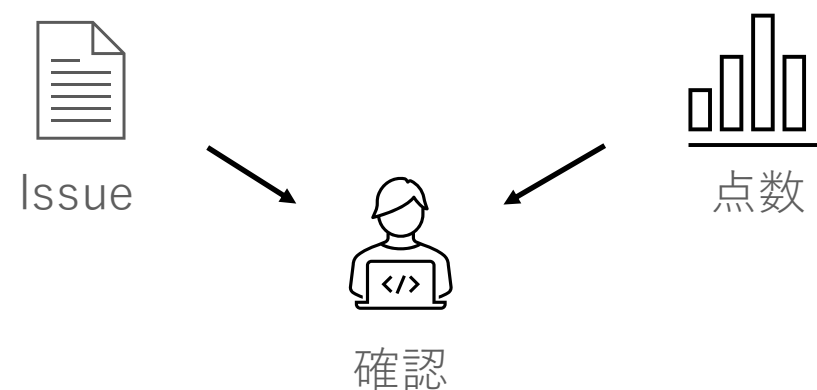
- ・ **どの問題を解けたか** 収集
- ・ 上位6エージェントを利用
(SWE-bench リーダーボードより)

2, Issueの品質評価



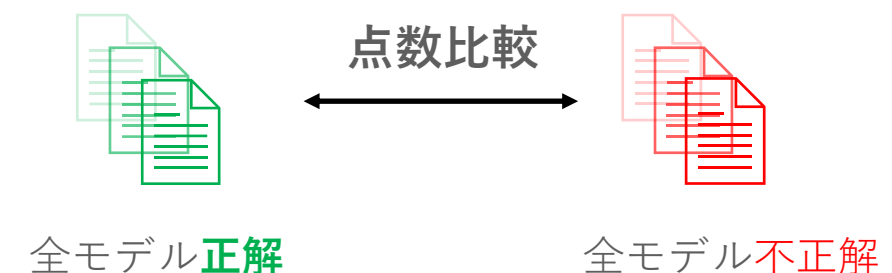
- ・ **各問題記述の品質** を評価
- ・ 5つの観点（後述）で採点
LLMを用いて採点

3, 品質評価を人手確認



- ・ LLMの採点が正しいか確認

4, 正解率ごとの点数を比較



- ・ 全モデルで**正解した**問題と
全モデルで**不正解だった**問題で
品質を比較

RQ1：システムの有効性 | Issueの評価

5つの観点

- 再現例の質（0～3）

バグを再現するコード例が含まれているか

- 解決策の質（-1～3）

修正方針の示唆が含まれているか

的外れなことが書いてあれば-1点

- ファイルレベルのバグ位置の質（0～3）

修正対象のファイル名が具体的に含まれているか

- コードシンボルレベルのバグ位置の質（0～3）

修正対象のコードシンボル（クラス・関数など）が含まれるか

- 行レベルのバグ位置の質（0～3）

修正対象の行番号がどの程度具体的に含まれているか

RQ1：システムの有効性 | 結果

全システムで失敗/成功した件数（全：500件）

全成功：181件 全失敗：96件 一部で成功：223件

2割の問題は誰も解けない

Issue品質の差

	再現例の質	解決策の質	バグ位置 (ファイル)	バグ位置 (コードシンボル)	バグ位置 (行レベル)
全失敗	2.01	0.656	1.052	0.906	0.813
全成功	2.022	1.276	1.326	1.177	0.994
スコア比	101%	195%	126%	130%	122%

RQ1：システムの有効性 | 結果

解決策の質（修正方針が示唆されているか）が成功率に最も影響する
バグ再現例の質はほとんど効いていない

	再現例の質	解決策の質	バグ位置 (ファイル)	バグ位置 (コードシンボル)	バグ位置 (行レベル)
全失敗	2.01	0.656	1.052	0.906	0.813
全成功	2.022	1.276	1.326	1.177	0.994
スコア比	101%	195%	126%	130%	122%

RQ2：FL能力と解決率

収集方法

最終提出パッチを見て，そのシステムが特定したバグ箇所を収集

結果

FLのF1スコア順位（ファイル）		修正成功件数順位
1, W&B Programmer	↔	1, W&B Programmer
2, Blackbox AI Agent	↔	2, Blackbox AI Agent
3, Midwit Agent	↔	3, Midwit Agent
4, Devlo	↘	4, Learn-by-interact
5, Emergent E1	↘	5, Devlo
6, Learn-by-interact		6, Emergent E1

RQ2：FL能力と解決率

欠陥位置予測が得意なシステムほど修正にも成功している

FLのF1スコア順位（ファイル）		修正成功件数順位
1, W&B Programmer	↔	1, W&B Programmer
2, Blackbox AI Agent	↔	2, Blackbox AI Agent
3, Midwit Agent	↔	3, Midwit Agent
4, Devlo	↘	4, Learn-by-interact
5, Emergent E1	↘	5, Devlo
6, Learn-by-interact		6, Emergent E1

RQ2：FL能力と解決率

ファイルレベルとコードシンボルレベル

システム	ファイル命中	シンボル命中	差
W&B Programmer	448	396	-52
Blackbox AI Agent	435	348	-87
Devlo	439	354	-85
Midwit Agent	408	309	-99
Emergent E1	424	374	-50
Learn-by-interact	342	308	-34

RQ2：FL能力と解決率

ファイルレベルとコードシンボルレベル

システム	ファイル命中	「どのファイルか」までは 87~90%当てられる
W&B Programmer	448	
Blackbox AI Agent	435	
Devlo	439	
Midwit Agent	408	
Emergent E1	424	
Learn-by-interact	342	

RQ2：FL能力と解決率

ファイルレベルとコードシンボルレベル

シン	シンボル命中	差
W&B	396	-52
Black	348	-87
	354	-85
Mic	309	-99
Em	374	-50
Lear	308	-34

「どの関数か」
「どのメソッドか」までは
62~79%に落ちる
→改善余地

RQ3：バグ再現タスクの有効性

実験方法については割愛

独自の方法でバグ再現スクリプトを作成.

Issueの情報量とバグ再現スクリプトの品質の関係を調査

結論

- 上位システムはバグ再現プロセスを採用
→ バグ再現プロセスは必要である
- Issueに**情報が多いほど品質の高いスクリプト**ができる

まとめ

エージェントシステムの現状を多面的に分析

上位エージェントについて

- Issue品質と解決率の関係
- 欠陥位置予測能力の比較
- バグ再現プロセスの品質とIssue情報量の関係 を調査

Issue内の解決策示唆の有無が解決率に最も影響

システムのFL能力と解決率には正の相関

再現スクリプトの品質はissue内の再現情報の質に依存する