

Prompting in the Wild: An Empirical Study of Prompt Evolution in Software Repositories

Mahan Tafreshipour
University of California, Irvine
Irvine, USA
mtafresh@uci.edu

Aaron Imani
University of California, Irvine
Irvine, USA
aaron.imani@uci.edu

Eric Huang
University of California, Irvine
Irvine, USA
jianch2@uci.edu

Eduardo Almeida
Federal University of Bahia (UFBA)
Brazil
esa@rise.com.br

Thomas Zimmermann
University of California, Irvine
Irvine, USA
tzimmer@uci.edu

Iftekhhar Ahmed
University of California, Irvine
Irvine, USA
iftekh@uci.edu

2025 IEEE/ACM 22nd International Conference on Mining Software Repositories (MSR)

背景 | LLM統合アプリケーションの登場

LLMを組み込むアプリが増加

- 300万以上のカスタムGPTが存在（2024年1月時点）

プロンプト=新しいソフトウェア成果物

```
SEARCH_TEMPLATE = """
Given a Query and a list of Google Search Results,
return the link from a reputable website which
contains the data set to answer the
question. {columns}

Query:{query}

Google Search Results:
```
{search_results}
```

The answer MUST contain the url link only
"""
```



コードでも、
ドキュメントでもない、
新しいソフトウェア部品

開発者が記述したプロンプト例
(pyspark-ai)

背景 | LLM統合アプリケーションの登場

LLMを組み込むアプリが増加

- 300万以上のカスタムGPTが存在（2024年1月時点）

プロンプト=新しいソフトウェア成果物

```
SEARCH_TEMPLATE = """
Given a Query and a list of Google Search Results,
return the link from a reputable website which
```

だからこそ、

- プロンプトをどう管理・進化させる？
- 適切な管理方法が確立されていない

```
The answer MUST contain the url link only
"""
```

開発者が記述したプロンプト例
(pyspark-ai)

コードでも、
ドキュメントでもない、

新しいソフトウェア部品

問題 | プロンプト管理の実態

LLMアプリにおけるプロンプト管理の実態が不明

- どんな変更をしているのか？
- いつ変更しているのか？
- 変更は十分に記録されているのか？
- 意図通りに動いているのか？

既存のソフトウェア成果物との性質の違い

- 自然言語の曖昧さ
- LLMの非決定性
- 既存のソフトウェア工学の枠組みでは対応できない可能性？

目的・アプローチ

LLM統合アプリにおけるプロンプト進化の実態を解明

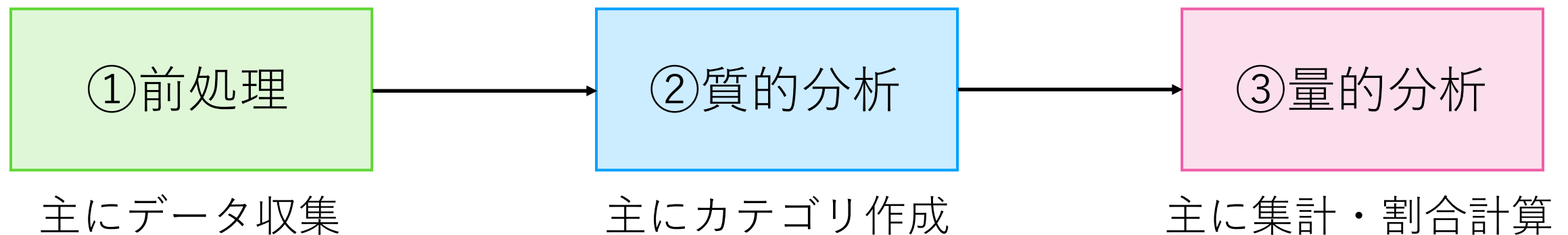
- 変更のタイプとパターン
- コード変更との関係
- ドキュメント実践
- 変更の影響と有効性

243リポジトリ/1,262プロンプト変更を分析

- GitHub上のオープンソースリポジトリ
- 実世界の変更履歴
- 質的・量的の両面で分析

分析手順

全体の流れ



①前処理

- 既存データセットから、対象プロジェクトを選定

20,598プロジェクト $\xrightarrow{\text{クリーニング, フィルタリング}}$ 1,568プロジェクト

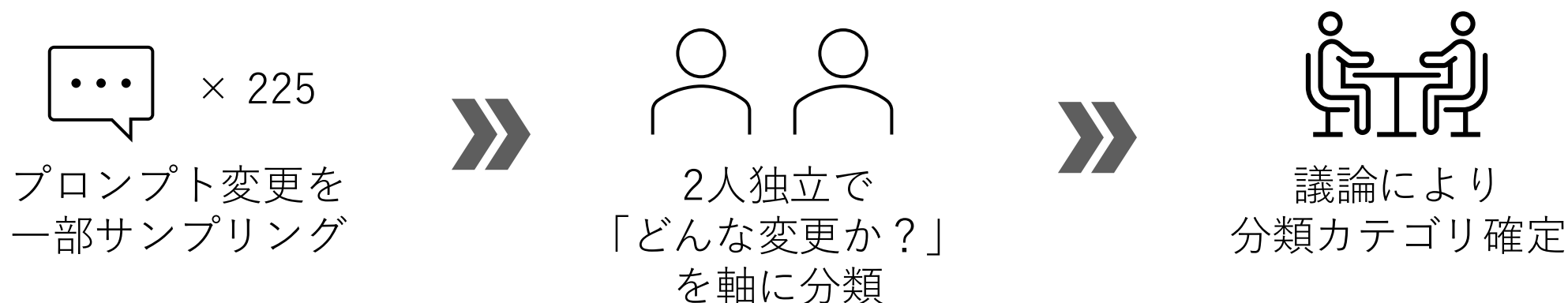
- 変更のあったプロンプトのみを抽出

1,568プロジェクト $\xrightarrow{\text{抽出}}$ 1,262プロンプト変更
(243プロジェクト)

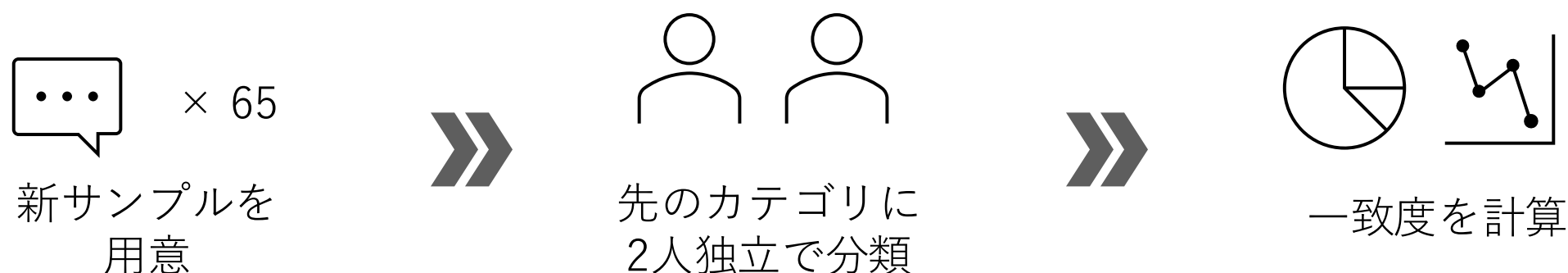
分析② | 質的分析 (RQ1~3)

プロンプト変更を体系的に分類 (コーディング)

- Phase1 : 分類カテゴリ作成



- Phase2 : 検証



- Phase3 : 全データコーディング

- 全1,262件の残りを先のカテゴリに分類

分析② | 質的分析 (RQ1～3)

結果① : 8つの変更タイプを特定

変更タイプ	定義	変更数	割合
追加	新しい指示や例などの導入	440	30.1%
修正	既存の構成要素の変更	373	25.5%
削除	既存の構成要素の除去	129	8.8%
言い換え	元の意味を維持しながら表現を変更	254	17.4%
フォーマット	見た目の整理	116	7.9%
校正	スペル・文法ミスの修正	71	4.9%
一般化	一部を変数などに置き換え	34	2.3%
再構築	既存の構成要素を再編成	29	2.0%
未分類	その他	17	1.1%

追加・修正が多く、プロンプトは拡張されていく傾向

分析② | 質的分析 (RQ1～3)

結果②：変更されるプロンプトの構成要素

- 既存研究で定義された、プロンプトの構成要素を利用
- プロンプト内のどんな要素が最も変更されやすいか？

変更タイプ	変更数	割合
動作指示・制約	569	60.4%
追加情報・文脈	154	16.4%
出力形式・スタイル	112	11.9%
例示	75	8.0%
役割設定	30	3.2%
主タスク定義	2	0.2%

モデル動作・出力の調整に最も力を入れる傾向

分析③ | 量的分析 (RQ4~9)

3つの観点でプロンプト変更の分布を分析

テーマA：いつ、どんなプロンプト変更を行うか？

3種のコミット (Feature, Fix, Refactor) それぞれの際に、
どのようなプロンプト変更が行われていたか？を分析

テーマB：プロンプト変更の文書化の頻度と、その内容は？

プロンプト変更があった際のコミットメッセージを収集、
その数と内容を分析

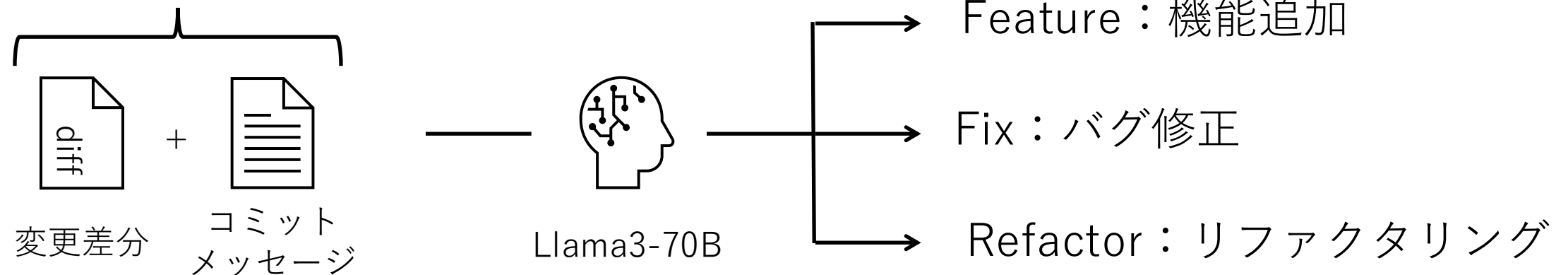
テーマC：プロンプト変更による実際の影響や効果は？

プロンプト変更によって、プロンプト自体やLLMの出力は
改善・悪化したのか？を分析

テーマA | いつ, どんなプロンプト変更?

プロンプト変更を含む931件のコミットをLLMで分類

プロンプト変更を含むコミット



結果① : メンテナンス活動の分布

Feature	59.7%
Fix	18.9%
Refactor	14.5%
その他	6.9%

Featureが約60%

プロンプト変更は
主に機能開発中に発生する

テーマA | いつ, どんなプロンプト変更?

結果②: コミットタイプごとの変更分布

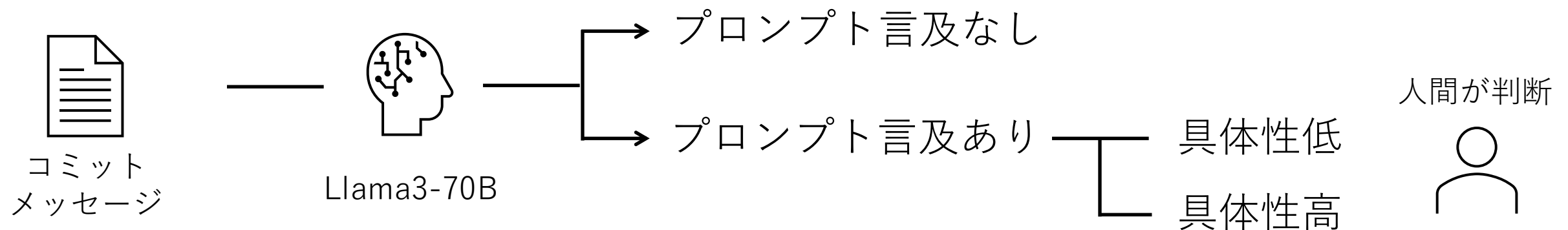
- ✓ Featureコミット (機能追加)
 - “動作指示・制約”の追加, “言い換え”が多かった
 - 開発者は新機能実装の際, 指示の追加とその明確性の確保を優先
- ✓ Fixコミット (バグ修正)
 - “動作指示・制約”の修正が多かった
 - LLMの出力ミスもある種バグ扱い?
- ✓ Refactorコミット (リファクタリング)
 - “言い換え”が最も多かった
 - プロンプトの明確性や表現の改善を優先

➤ それぞれの活動タイプで, 変更の傾向が異なる

テーマB | プロンプト変更の文書化と内容

931のコミットメッセージをLLMで分類

- プロンプト変更が言及されているかどうかを判別
- その後その言及の具体度を人間が判別



結果①：言及有無の分布

- 言及なし：78.1%
- 言及あり：21.9%

(具体性低：15.2%，具体性高：6.7%)

➡ 8割は文書化されておらず，
具体的な記録はたった7%

テーマB | プロンプト変更の文書化と内容

コミットメッセージから読み取れる変更意図を分析

- 具体性高と判別された62件を対象
- 目視で意図ごとにコーディング・分類

結果②

Category	Count
<u>Task Specific Change</u>	40
Typo Fix & Grammar Correction	7
Output Modification	5
Template Update	3
Technology Change/Adaptation	3
Reformat (moving parts of the prompt)	2
Handle Hallucination	2
Total	62

タスク固有の変更
(タスクの精度向上や文脈追加)
が大半



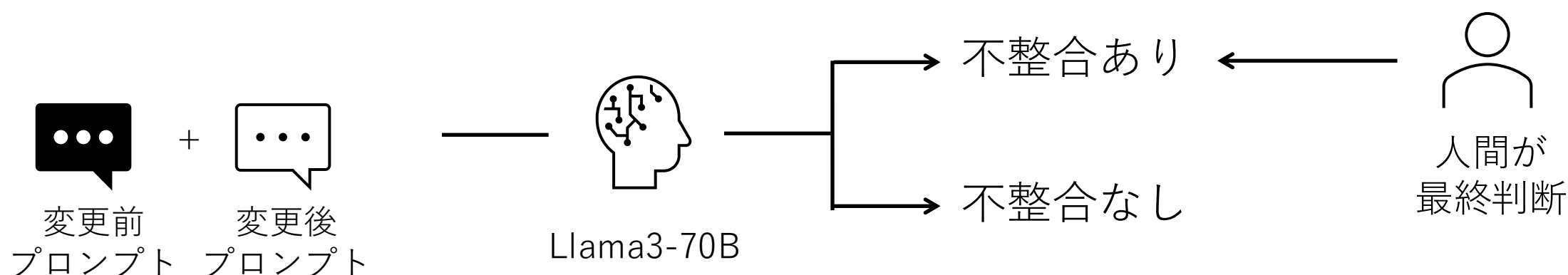
タスク改善が主目的
(保守や高度な処理は少ない)

テーマC | プロンプト変更の影響

1,262の変更の不整合の有無をLLMに分類

不整合：

- 変更後の文が，他の部分と食い違う
- 指示が互いに矛盾する，構造が壊れる など



結果①

Category	Count
Contradictory Considerations	8
Structural Conflict	4
Format Misalignment	3
Total	15

15件(全体の1.2%)で
不整合

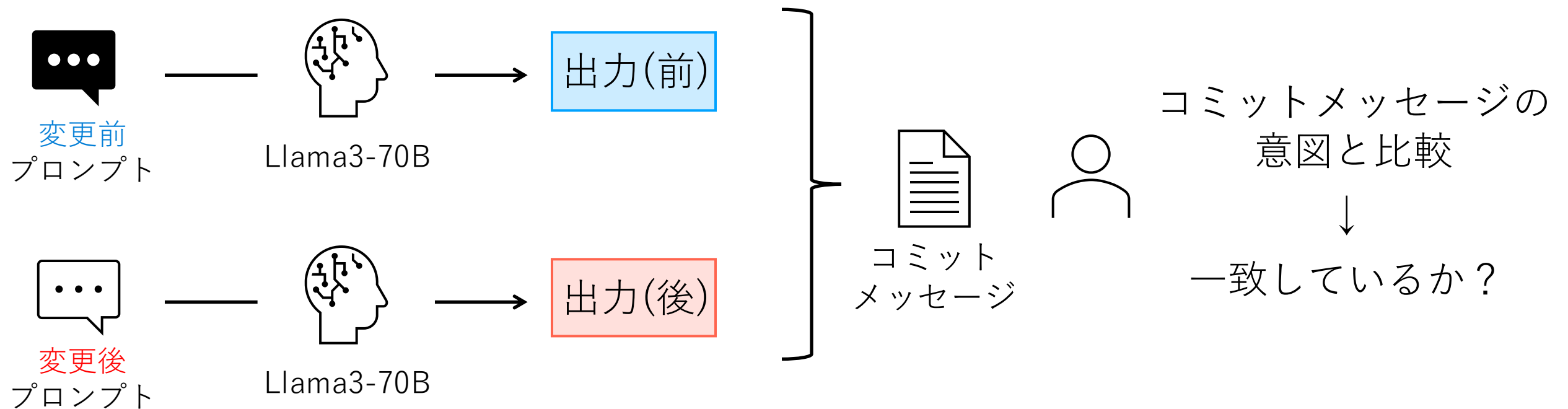


少ないが，検出が必要

テーマC | プロンプト変更の影響

意図が明確なプロンプト変更のうち、実行可能だったプロジェクトでLLMを実行実験

- 変更前のプロンプトでのLLMの出力と、変更後のプロンプトでのLLMの出力を比較
- 出力に変化があるか、または変化が意図に沿っているか？



✓ 計7プロジェクトで検証

テーマC | プロンプト変更の影響

結果②

- ✓ 修正への正確な適応：4件
 - 指示の追加や、既存指示の簡素化など
- ✓ 応答に変化なし：2件
 - few-shot例の追加などの重要な変更
 - にも関わらず応答に変化は見られなかった
- ✓ 意図しない変化：1件
 - 出力に変化はあったものの、意図された修正からは逸脱

➤ LLM統合システムの重要な課題を浮き彫りに

- 予測不可能性
- 従来のソフトウェアとの違い



効果的な検証・テスト・ツール
が必要

総括

LLM統合アプリにおけるプロンプト進化の実証研究

- 1,262件のプロンプト変更を詳細に分析

プロンプトの進化は主に追加・増大傾向

- 中でも動作指示の追加・修正が最重要

プロンプト進化は、約78%が文書化されていない

プロンプト変更後、変更意図通りに動作しないケース

所感

現状、プロンプト内容は重要視されていない説

- 文書化の少なさを問題として取り扱い、保守において重要な役割を果たしうると主張
- 現状、従来の成果物と区別した上で明示的に管理するほどの重要性はあるの？
- LLM自身がプロンプト改善をする、みたいなのが流行り始めているので、その進化を見るのは面白いかも

分析のためのLLMの使い方が大胆

- コミットやプロンプトの性質の分類にLLMを利用
- 一部（おそらく）人間評価なし
- 大まかな分類段階だから別にいいということか？