

Too Noisy To Learn: Enhancing Data Quality for Code Review Comment Generation

Chunhua Liu
The University of Melbourne
chunhua.liu1@unimelb.edu.au

Hong Yi Lin
The University of Melbourne
holin2@student.unimelb.edu.au

Patanamon Thongtanunam
The University of Melbourne
patanamon.t@unimelb.edu.au

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

背景 | コードレビュー

コードレビューの重要性と課題

- 論理エラーの特定やコード品質向上に寄与
- 時間・リソースの大量消費, 経験豊富なレビュアーへの負担
- コードレビュー自動化の研究が盛ん

コードレビュー自動生成

- GitHub等から大規模なデータセットを取得
- ニューラルモデルの学習・生成により, 一定の有望性
- レビューアーの負担軽減へ

問題 | 学習データの“雑音”

学習データには、低品質なレビュー（Noisy）が含まれる

- 曖昧で理解困難なコメント
- 改善提案ではなく、明確化を求めるだけのコメント など

```
@@ -80,6 +80,7 @@ public class HoodieCreateHandle<T extends  
HoodieRecordPayload> extends HoodieIOH  
String partitionPath, String fileId, Iterator<HoodieRecord<T>>  
recordIterator) {  
    this(config, commitTime, hoodieTable, partitionPath, fileId);  
    this.recordIterator = recordIterator;  
+    this.useWriterSchema = true;
```

Reviewer's Comment:

Why do we have this flag?

→単なる質問になっており、
改善提案がない

生成レビューコメントへ影響する可能性

- 学習データの品質低 ≡ 生成物の品質低
 - 低品質なレビューは有用性が低く、悪影響をもたらす可能性
- 学習データの品質向上が必要

先行研究とその課題

Liら^[1]のクリーニング手法

- ルールベースの除去
- 手動ラベル付きデータでSVMの訓練・分類
- 低品質データを除去し、品質の向上を目指す

一方, Tufanoら^[2]の指摘

- Liらのクリーニング後のデータセットのうち、**32%**が依然として低品質なコメントが含まれる
- 残存雑音の特徴：意味理解の不足、文脈推論を要する など
- 表面的な特徴を拾うだけの既存手法では不十分

[1] Z. Li, S. Lu, D. Guo, N. Duan, S. Jannu, G. Jenks, D. Majumder, J. Green, A. Svyatkovskiy, S. Fu, and N. Sundaresan, "Automating code review activities by large-scale pre-training," in Proceedings of ESEC/FSE, 2022

[2] R. Tufano, S. Masiero, A. Mastropaolo, L. Pascarella, D. Poshyanyk, and G. Bavota, "Using pre-trained models to boost code review automation," in Proceedings of ICSE, 2022

目的・アプローチ

コードレビューデータセットの品質を向上させる手法の提案

- 意味理解・文脈推論を可能にした識別手法
- 学習データ品質の，生成物への影響の調査

アプローチ：LLMを利用した，コメント品質の識別

- LLMは，コード理解・自然言語理解の両方に有望な性能
 - 課題だった意味理解・文脈推論が可能に
 - LLMでクリーニングしたデータと，そうでないデータで学習・生成を行った際の生成物の品質を評価
- 高品質データのみを収集することで，生成物の品質向上へ

有効(高品質)/雑音(低品質)の定義

有効レビューコメント

- “有効コメントとは、ソースコードの改善を目的とした明確な提案を提供すべきレビューコメントを指す。”
- 核心：実行可能で具体的な改善指示
- 特徴：問題の明示 + 解決策の提示
- 例：「変数名をcamelCaseに変更」，「try-catch文を追加」など

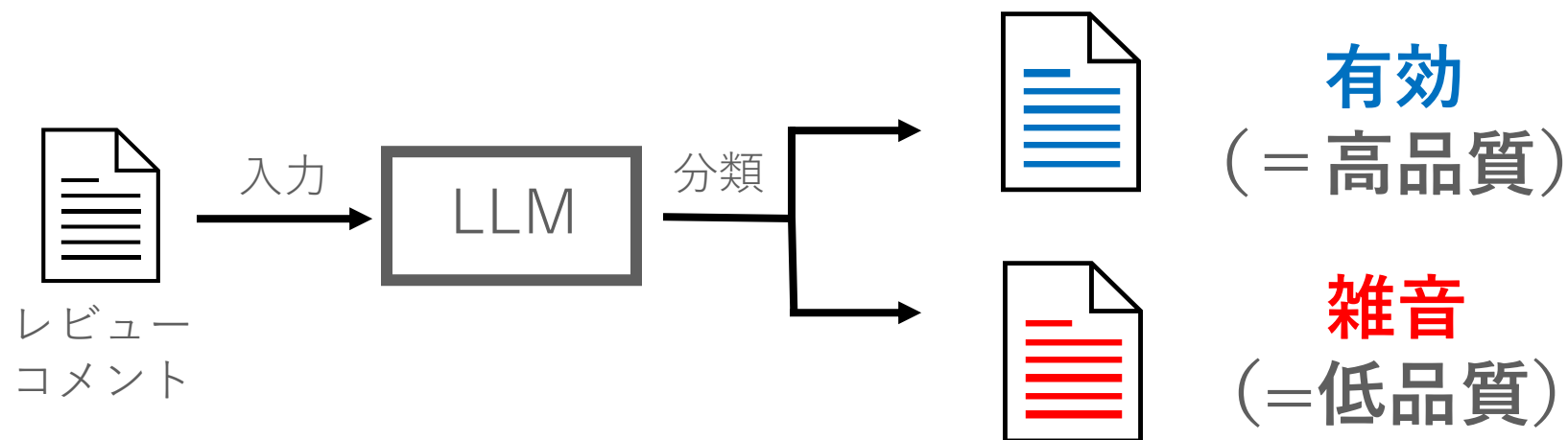
雑音レビューコメント

- “雑音コメントとは、コード改良のための直接的で適用可能な行動を要求しない、または表現されたメッセージが不明瞭で理解困難なレビューコメントを指す。”
- 核心：実行不可能または曖昧な内容
- 特徴：質問のみ、感想、不明確な指摘
- 例：「なぜこの変更？」「良いアイデア」など

RQ1 | LLMでの品質分類の精度

LLMはどのくらいの精度でクリーニングできるか？

- LLMでレビューコメントが有効 or 雑音を判定, その精度を算出



2×2種類のプロンプト条件で比較実験

- DEFINITION (基本定義のみ), AUXILIARY (7つの補助規則追加)
 - 基本的な指示のみか, ちょっと具体的な分類基準を与えるか
- コメントのみ, diff+コメント
 - レビューコメントのみを与えるか, diffも一緒に与えるか

RQ1 | LLMでの品質分類の精度

データセット

- CodeReviewerデータセット¹を利用 (from GitHubなど)
- うち270件(学習)を目視で有効/雑音のラベル付け
 - 分類精度の算出に利用

元データセット

学習	検証	テスト
117,739件	10,319件	10,169件

ラベル付けセット

有効	雑音	合計
172件	98件	270件

使用モデル

- GPT3.5, CodeLlama(34B), Llama(8B)
- 3LLM × 2プロンプト × 2入力 = 12パターンで精度比較

補足：目視評価の手順

- 人間を3人用意（便宜上A, B, Cとする）
- 一部データをサンプリング（今回は信頼区間90%, $\pm 5\%$ ）
- 結果, 目視対象全部で270件が残る

手順

- ①最初の50件を, AとBで目視評価, 一致度（カッパ係数）算出
 - 論文では0.57でした（中程度の一致）
- ②不一致を議論・修正
 - 0.57→0.83（高い一致）へ向上, ここでの不一致はCが判断
- ③残りの220件は1人目が全部やる
 - 迷う事例・境界事例が出たら, B・Cに相談・議論し決定

RQ1 | 結果（一部抜粋，Validのみ）

モデル	プロンプト	入力	Precision	Recall	F1
GPT-3.5	定義のみ	レビューのみ	85.1	36.6	51.2
CodeLlama	定義＋補助	レビューのみ	73.2	87.7	79.8
Llama3	定義のみ	レビューのみ	75.3	84.9	79.8

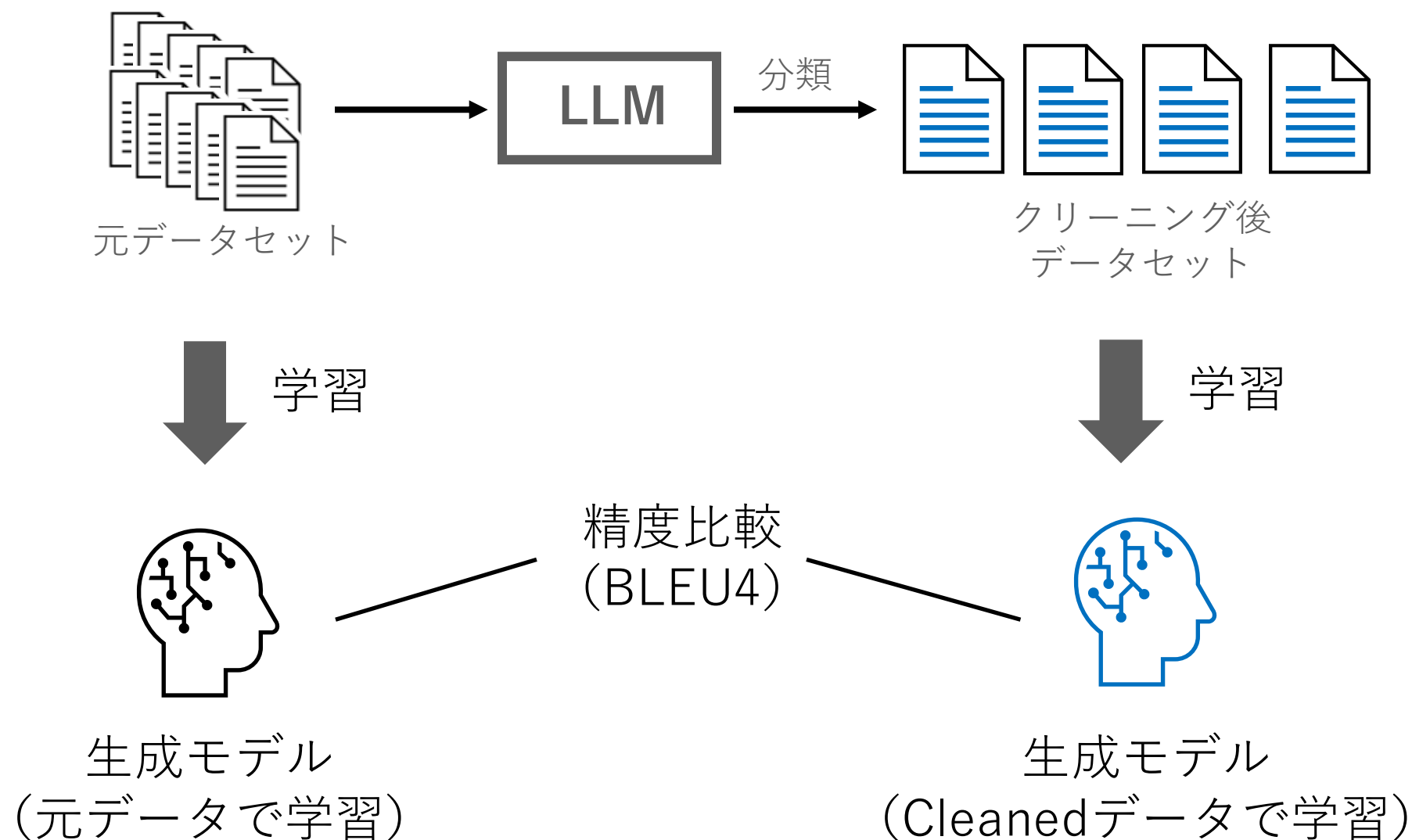
特徴

- GPT＋定義のみ＋レビューのみで最高Prec：85.1%
- 全体的に，定義のみ＞定義＋補助
- 全体的に，レビューのみ＞レビュー＋diff
- 余計な情報追加しない方が高精度

RQ2 | クリーニング後の生成精度

データクリーニングは生成モデルの精度に影響するか

- クリーニングすると、データ数は削減される
- 学習データ量が減ることによって、生成精度に影響はあるか？を検証



RQ2 | クリーニング後の生成精度

元データ vs クリーニング後データ

- ・ クリーニングモデルとして, GPT3.5, Llama3 (RQ1で好成績) を使用

元データ

学習	検証
117,739件	10,319件

→

クリーニング後データ

	学習	検証
GPT3.5	39,625件	3,395件
Llama3	87,872件	7,571件

使用生成モデル

- ・ CodeT5, CodeReviewerモデル¹⁾
- ・ それぞれを元データ, クリーン後データで学習, BLEUスコアを比較

RQ2 | 結果

<i>M</i>	Dataset	Test	
CodeReviewer	ORIGINAL	5.73	
	CLEANED _{GPT-3.5}	6.04	5.4%↑*
	CONTROLLED _{GPT-3.5}	5.63	
	CLEANED _{LLAMA3}	<u>5.97</u>	4.2%↑*
	CONTROLLED _{LLAMA3}	5.63	
CodeT5	ORIGINAL	5.19	
	CLEANED _{GPT-3.5}	<u>5.67</u>	9.2% ↑*
	CONTROLLED _{GPT-3.5}	5.20	
	CLEANED _{LLAMA3}	<u>5.54</u>	<u>6.7%</u> ↑*
	CONTROLLED _{LLAMA3}	5.21	

ORIGINAL：元データ

CLEANED：クリーン後

CONTROLLED：元データをCLEANEDと同件数になるようランダム削減

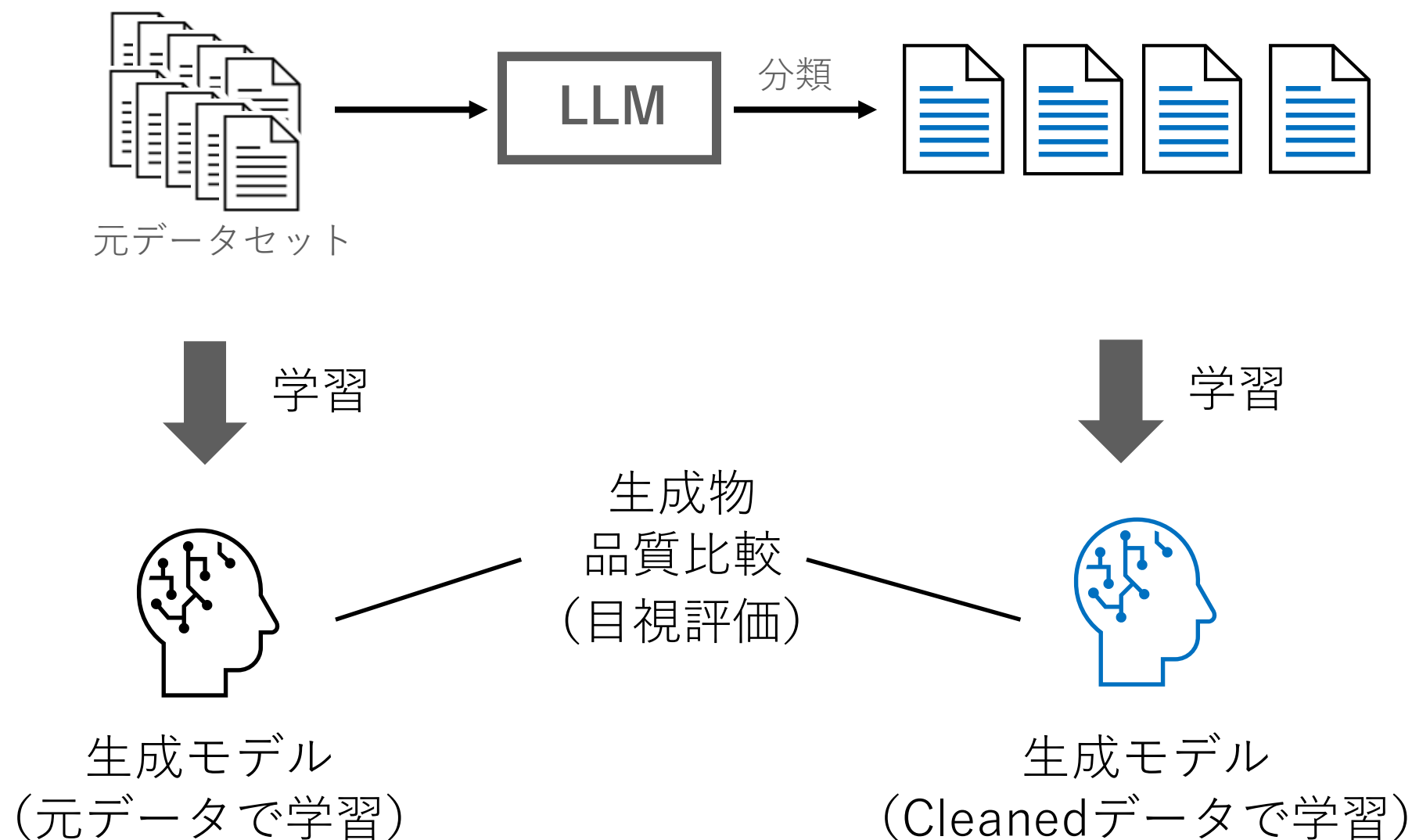
結論：大きな悪影響なし

- ORIGINAL→CLEANEDでデータ数が削減されたが、数値は改善
- ランダム削減よりも数値は良い
- クリーニングによるデータ数削減は生成精度に悪影響を及ぼさない

RQ3 | クリーニング後の生成品質

データクリーニングは生成モデルの品質に影響するか

- クリーニングすると、雑音データの割合が削減される
- 生成モデルが生成する



RQ3 | クリーニング後の生成品質

目視評価指標（[]参考）

- 情報
 - コメントがコード作成者にとってコード変更修正にどの程度情報豊富か
 - 1-5点の5段階評価
- 関連性
 - レビューコメントが対応するコード変更にどの程度関連しているか
 - 1-3点の3段階評価

評価対象

- 計2人で以下の100×3=300件の生成文をスコア付け・比較評価

ORIGINAL	CLEANED-GPT3.5	CLEANED-Llama
100件	100件	100件

補足：目視評価の手順

- 人間を2人用意（便宜上A, Bとする）
- 生成データをサンプリング（今回は100件ずつ）

手順

- ①最初の50件を, AとBで目視評価, 一致度（カッパ係数）算出
 - 情報：0.71（相当一致）、関連性：0.42（中程度一致）
- ②不一致を議論, 次の50件で再度一致度を算出
 - 関連性み：0.42→0.60（相当一致）へ向上
- ③残りの200件も2人で全部やる
 - 不一致は議論で解決

100件

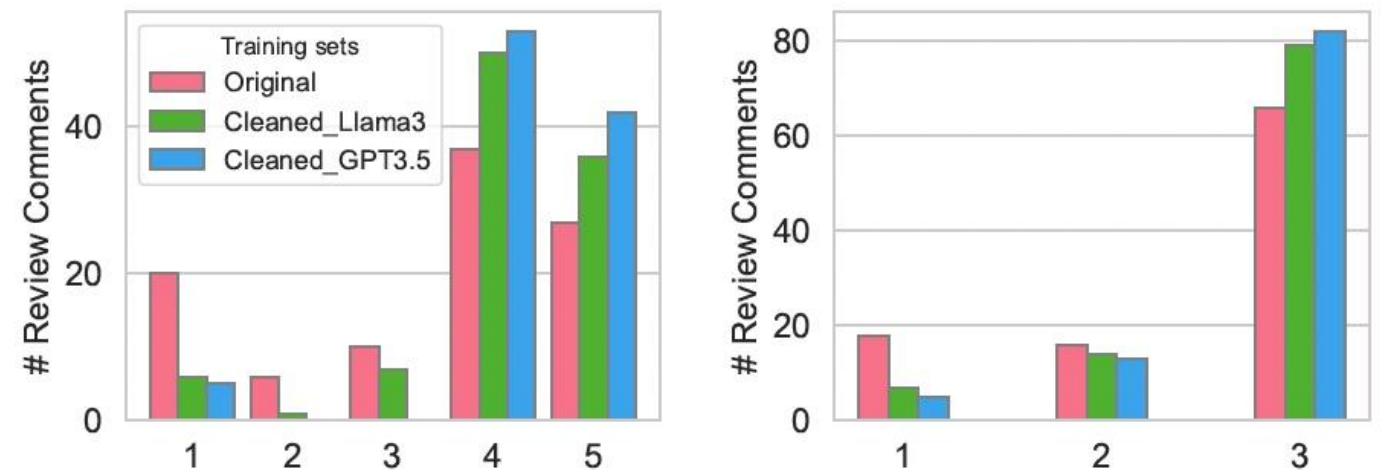
100件

100件

RQ3 | 結果

平均スコア比較表

Training set	Manual	
	Information	Relevance
ORIGINAL	3.44	2.48
CLEANED _{GPT-3.5}	4.27*	2.77*
CLEANED _{LLAMA3}	4.08*	2.72*



(a) Information score (manual)

(b) Relevance Score (manual)

スコア分布比較グラフ

CLEANEDはいずれも元データよりも生成物の品質が向上

- Precisionが最も高いGPT-3.5は、特に高いスコア

✓ クリーニング済みデータセットを学習に用いると、生成レビューコメントの品質を改善可能！

RQ3 | 生成物の比較 (例)

```
@@ -95,6 +95,8 @@ class Product extends BaseAction implements
EventSubscriberInterface
    $con->beginTransaction();
    try {
+        $prev_ref = $product->getRef();
        $product
            ->setDispatcher($event->getDispatcher())
            ->setRef($event->getRef())
```

Ground Truth: *Please use camelCase instead of underscore_case*

Original: *What is the purpose of this line?*

Info: 1 Rel: 1

CLEANEDLLAMA3: *Why not use `\$event->getRef()`?*

Info: 4 Rel: 2

CLEANEDGPT3.5: *Please rename `\$prev\_ref` to `\$previousRef`.*

Info: 5 Rel: 3

元データ生成：質問のみ，改善提案なし

CLEANED生成：具体的な改善提案を提示

まとめ

- | LLMを用いたデータクリーニングによる, コードレビュー生成学習データ品質の向上
- | 学習データの雑音の割合を削減することで, 生成レビューの品質も向上
- ✓ LLMのデータクリーニングの有効性を提示

所感

自分の研究と背景・実験構造が非常に類似

- 非常に参考にはなる
 - 実験構想・設定など
 - 特に目視評価のやり方（労力的に）どうしようかと思ってた(でもこれはちょっと変則的?)

最近コードレビュー論文多い？

- 若干飽きてきた（輪講内だけで今年4回目）
- LLMコーディング時代に重要なタスクだから説

「修士は2年」って、遅いなあと言う話

- 極論，自分が今年～来年にかけて提出するのと同じ発想の論文が，去年の冬頃には提出されていた
- その頃には我々も発想は固まっていたが，修了は2年なのでそれに合わせたペースで進行，結果時代遅れに．
- さらに言うなら，今年すでにfine-tuningは下火

補足：プロンプト

System Prompt (Task Instruction)

Role Assignment

Your task, as an experienced coder reviewer, is to evaluate review comments generated by other developers submitted during the code review process. Your objective is to discern between noisy comments and those are valid.

Definitions

Valid comments are those that request direct and explicit code changes aimed at improving the source code. ...
A comment is considered noisy if it does not request direct and applicable actions to refine the code, ...

Criteria

Your evaluation should be guided by the following criteria:

1. Relevance to Code Change: ...
2. Clarity and Constructiveness: ...
3. Focus on Improvement: ...

User Prompt

Below is a code diff and review comment.
Please evaluate whether this comment is Valid or Noisy.

Context: Code Change

```
@@ -86,6 +86,7 @@ type PipelineManifest struct {  
    Version PipelineSchemaMajorVersion `yaml:"version"`  
    Source *Source                      `yaml:"source"`  
    Stages []PipelineStage             `yaml:"stages"`  
+    // ArtifactBuckets?  
}
```

Input: Review Comment

The buckets are created via the stackset, customers don't need to provide them.

Answer Format

Return the answer as a dict with label and explanation as keys.