TOO NOISY TO LEARN: ENHANCING DATA QUALITY FOR CODE REVIEW COMMENT GENERATION

10/8/25

Chunhua Liu, Hong Yi Lin, Patanamon Thongtanunam,

The University of Melbourne

Xiangrui Ke,

David R. Cheriton School of Computer Science

UNIVERSITY OF WATERLOO

BACKGROUND

- Code review is essential but time-consuming and inconsistent.
- AI models can help generate review comments.
- Training datasets are **noisy** (vague, difficult-to-understand) → poor model outputs.

UNIVERSITY OF WATERLOO

EXSITI

- Existi
  or **no**
- Model
  comm
- Reviev

```
@@ -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?*
Label: Noisy Comment

```
@@ -157,7 +157,7 @@ public class ProviderConfig extends
AbstractServiceConfig {

    @Deprecated
    public void setProtocol(String protocol) {
-       this.protocols = Arrays.asList(new ProtocolConfig[]{new
        ProtocolConfig(protocol)});
+       this.protocols = new ArrayList<>(Arrays.asList(new
        ProtocolConfig[]{new ProtocolConfig(protocol)}));
    }
```

Reviewer's Comment: *This can be simplified as* `new ArrayList<>(Arrays.asList(new ProtocolConfig(protocol)))`

Label: Valid Comment

UNIVERSITY OF WATERLOO

IDEA

Use **Large Language Models (LLMs)** to semantically detect and filter out noisy comments.

- Manually label a sample (valid or noisy?)
- Use LLMs for classification
- Train review generation models with high-quality cleaned data

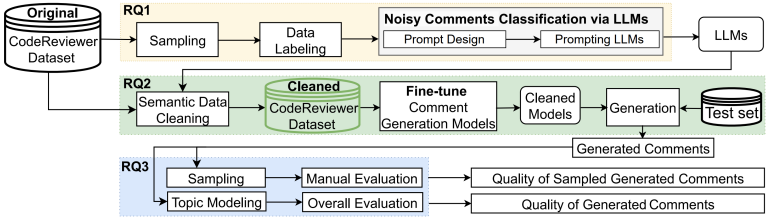
Gemini

Grok

deepseek

UNIVERSITY OF WATERLOO

STUDY DESIGN



RQ.1

To what degree can large language models semantically clean code review comments?

RQ.2

Does semantic data cleaning impact the accuracy of code review comment generation models?

RQ.3

Does semantic data cleaning improve the quality of code review comment generation models?

RQ1: LLMS Semantically Clean Code Review Comments

Prompt	Model	Input	Overall (weighted)			Valid (172)				Noisy (98)			
			Prec	Rec	F1	Prec	Rec	F1	#	Prec	Rec	F1	#
	Baseline [5]	-	40.6	63.7	49.6	63.7	100	77.8	270	0	0	0	0
$P_{\text{DEFINITION}}$	**GPT-3.5**	R_{NL}	70.3	54.1	55.7	**85.1**	36.6	51.2	74	44.4	**88.8**	**59.2**	196
	CodeLlama	R_{NL}	64.1	65.6	58.0	66.0	**94.8**	77.8	247	60.9	14.3	23.1	23
	Llama3	R_{NL}	**71.8**	**72.6**	**71.7**	75.3	84.9	**79.8**	194	65.8	51	57.5	76
	GPT-3.5	R_{NL}+ C_{DIFF}	65.6	61.5	62.2	75.8	58.1	65.8	132	47.8	67.3	55.9	138
	CodeLlama	R_{NL}+ C_{DIFF}	54.2	62.2	52.3	63.8	94.2	76.1	254	37.5	6.1	10.5	16
	Llama3	R_{NL}+ C_{DIFF}	62.6	65.2	59.8	66.7	90.7	76.8	234	55.6	20.4	29.9	36
$P_{\text{AUXILIARY}}$	**GPT-3.5**	R_{NL}	66.8	59.2	59.5	49.7	60.6	54.6	107	46.2	76.3	57.6	160
	CodeLlama	R_{NL}	71.0	71.7	70.1	73.2	87.7	79.8	205	**67.2**	43.9	53.1	64
	Llama3	R_{NL}	71.0	71.9	70.6	74.0	86.0	79.6	200	65.7	46.9	54.8	70
	GPT-3.5	R_{NL}+ C_{DIFF}	60.4	55.9	56.7	70.5	52.9	60.5	129	42.6	61.2	50.2	141
	CodeLlama	R_{NL}+ C_{DIFF}	47.0	62.2	55.7	64.1	92.4	75.7	248	40.9	9.2	15.0	22
	Llama3	R_{NL}+ C_{DIFF}	63.3	65.6	62.2	68.0	86.6	76.2	219	54.9	28.6	37.6	51

The highest and second-highest results are in bold and underlined. # represents the number of instances predicted in each class.

LLMs is a good classifier.

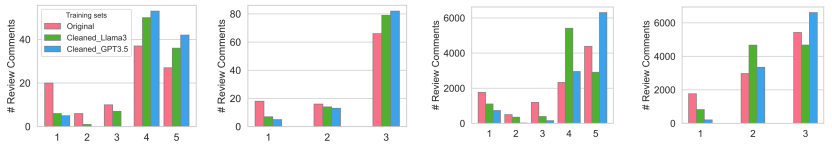
RQ2: Clean Data Impact the Accuracy of Generation

TABLE III: Model Performance (BLEU-4) on Comment Generation Models.

M	Dataset	Test		$\text{Valid}_{\text{Our\&Tufano}}$		$\text{Noisy}_{\text{Our\&Tufano}}$		$\text{Valid}_{\text{Our}}$		$\text{Noisy}_{\text{Our}}$		$\text{Valid}_{\text{Tufano}}$		$\text{Noisy}_{\text{Tufano}}$	
CodeReviewer	ORIGINAL	5.73		6.17		5.41		5.45		5.17		7.12		5.60	
	$\text{CLEANED}_{\text{GPT-3.5}}$	**6.04**	5.4%↑*	**6.97**	**13.0%**↑*	5.02	7.2%↓	**5.93**	8.8%↑	5.19	0.4%↑	**7.99**	**12.2%**↑*	4.83	13.8%↓
	$\text{CONTROLLED}_{\text{GPT-3.5}}$	5.63		6.20		5.43		5.21		5.13		7.39		5.70	
	$\text{CLEANED}_{\text{LLAMA3}}$	5.97	4.2%↑*	6.63	7.5%↑*	5.18	4.3%↓	5.64	3.5%↑	5.11	1.2%↓	7.71	8.3%↑*	5.14	8.2%↓
	$\text{CONTROLLED}_{\text{LLAMA3}}$	5.63		6.18		5.66		5.12		5.36		7.45		5.86	
CodeT5	ORIGINAL	5.19		5.34		5.04		4.84		5.09		5.85		6.03	
	$\text{CLEANED}_{\text{GPT-3.5}}$	5.67	**9.2%**↑*	6.00	12.4%↑*	5.23	3.8%↑*	5.88	**21.5%**↑*	5.27	3.5%↑	6.06	3.6%↑	5.15	14.6%↓
	$\text{CONTROLLED}_{\text{GPT-3.5}}$	5.20		5.34		5.30		5.17		5.39		5.45		5.41	
	$\text{CLEANED}_{\text{LLAMA3}}$	5.54	6.7%↑*	5.74	7.5%↑*	5.33	5.8%↑	5.32	9.9%↑*	5.14	1.0%↑	6.09	4.1%↑	5.46	9.5%↓
	$\text{CONTROLLED}_{\text{LLAMA3}}$	5.21		5.19		5.12		4.95		5.26		5.38		5.01	

The highest and second-highest results are in bold and underlined, respectively. * indicates the statistical significance (p-value < 0.05).

RQ3: Clean Data Improve the Quality of Generation



(a) Information score (manual) (b) Relevance Score (manual) (c) Information Score (overall) (d) Relevance Score (overall)

Fig. 4: Distribution of information and relevance scores on tests across CodeReviewer models trained on different training sets.

The cleaned models can generate more informative and relevant comments

```
@@ -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
CLEANED$_{\text{LLAMA3}}$: *Why not use `$event->getRef()`?* Info: 4 Rel: 2
CLEANED$_{\text{GPT3.5}}$: *Please rename `$prev_ref` to `$previousRef`.* Info: 5 Rel: 3

Fig. 5: Example comments generated by original and cleaned models with information (Info) and Relevance (Rel) scores.

CONCLUSION

- To RQ1: LLM-based approach achieves 66-85% precision in detecting valid comments
- To RQ2: Using the predicted valid comments to fine-tune the state-of-the-art code review models (cleaned models)
- To RQ3: Cleaned models can generate more informative and relevant comments than the original models

Quality > Quantity

PROS

- First to use LLMs for semantic cleaning of code review datasets
- Comprehensive experiment with different models and metrics.
- **Insightful Resulte**: better data quality improves model performance.
- Pioneeringly investigate the feasibility of using LLMs .

CONS

- Bias exists
- Manual job still require
- Limited generalizability

RATING 4/5

- LLM-based cleaning significantly improve model quality
- Come out with the key insight: Quality > Quantity
- Advanced LLM-assisted exploration in software engineering

FUTURE WORK

- Expand to both validity and correctness.
- Bring more LLM-driven work to reduce manual effort
- More dataset and ablation experiments
- Domain generalization

DISCUSSION

UNIVERSITY OF WATERLOO

Thank you