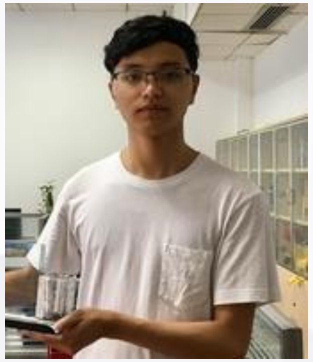
Why My Code Summarization Model Does Not Work?

Code Comment Improvement with Category Prediction

Qiuyuan Chen
Zhejiang University

Xin Xia
HUAWEI



Han Hu
Tsinghua University

David Lo
Singapore Management University

Shanping Li
Zhejiang University

Qiuyuan Chen, Xin Xia, Han Hu, David Lo, Shanping Li. In ACM Transactions on Software Engineering and Methodology (TOSEM).

Index

1

Background

Background

Two things that developers hate most:

1) Comment My Code

2) Others Don’t Comment their Code

What is Comment Generation?

Comment Generation（Code Summarization）

- Given a piece of code, Code Summarization generates a descriptive comment automatically (with template-, retrieval-, or learning-based approach).

Background

Two things that developer hate most:

1) Comment My Code

2) Others Don’t Comment their Code

Example

Source code:

```
public int countCharacter(String str, char c){
    int num=0;
    for (int i=0;i<str.length();i++){
            if (str.charAt(i)==c)
                num++;
        }
        return num;
    }
```

Summary：

Return character count in a string.

Background

Two things that developer hate most:

1) Comment My Code

2) Others Don’t Comment their Code

Advantages

Comment Generation（Code Summarization）

- Code Summarization does not depend on a specific developer and is compatible with different codes.
- It can ease the burden of developers by automating the generation of code comments.

Background

Can we generate all kinds of code comments?

Many Types of Code Comments

Different code comments have different intentions

01

Information Inside the Code

It is possible to learn the semantics of the code to generate the corresponding code comments

02

Information Outside the Code

Code comments are also a key bridge between code and business

03

2

Research Questions

What kind of code comments can be generated, or are suitable for generation?

RQ1

Research Question

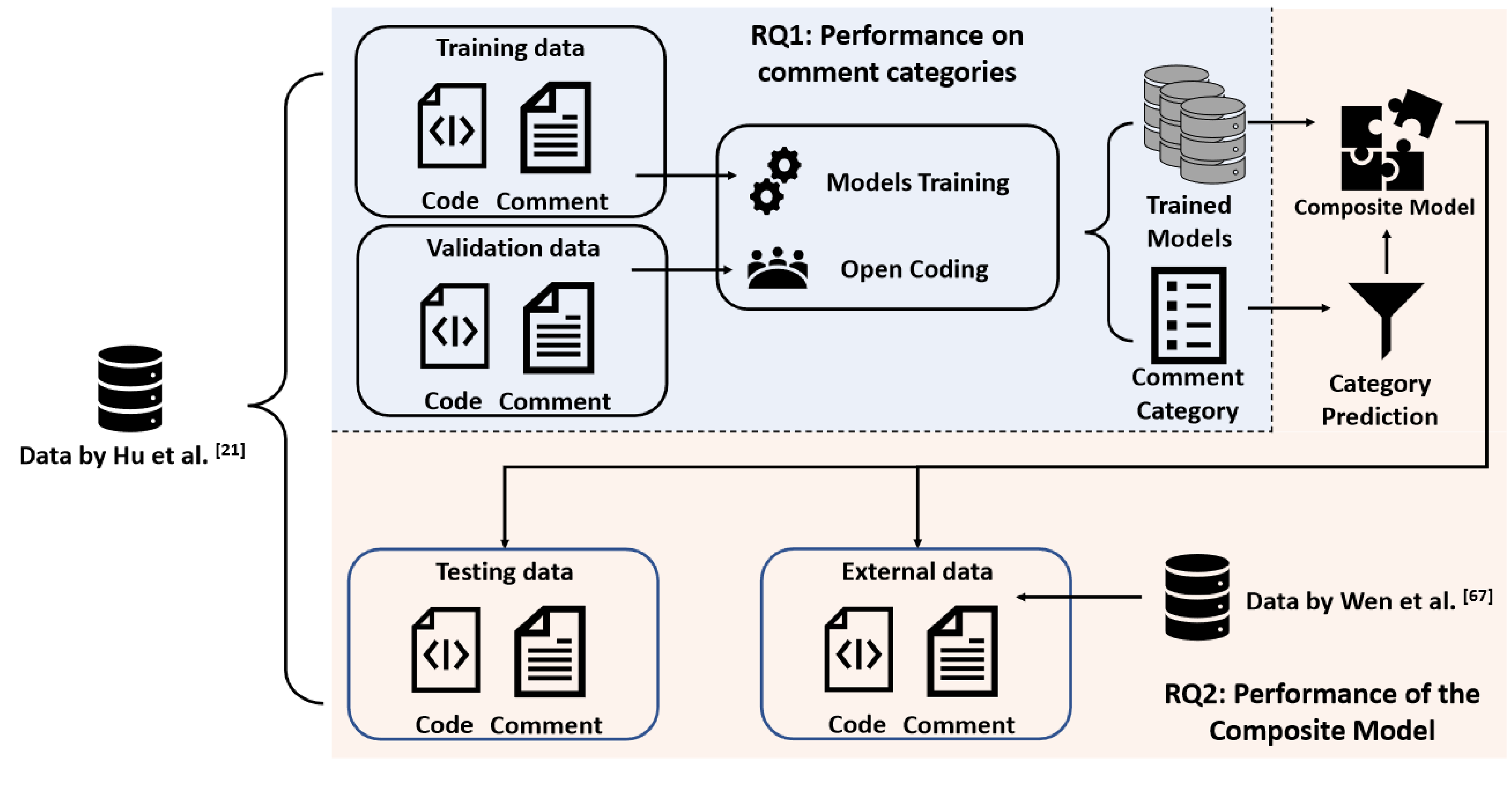
How do different comment categories impact the code summarization performance?

RQ2

Research Question

How can we improve the code summarization performance using the comment categories?

Research Overview



3

Empirical Study

Empirical Study

What

Description of the functionality

01

Why

Why the code is provided or the design rationale

02

How-to-use

Description of the usage of the code

03

How-it-is-done

Implementation details of the functionality

04

Property

Explain properties of the code

05

Others

Unspecified categories

06

[1] J. Zhai *et al.*, "CPC: Automatically classifying and propagating natural language comments via program analysis," In International Conference on Software Engineering (ICSE 2020)

Empirical Study

- We manually classify 20,000 <code, comments> pairs
- Labor-intensive: on average, label 57 pairs every hour

Category	Count	Proportion
What	4106	20.53%
Why	2493	12.47%
How-to-use	10190	50.95%
How-it-is-done	2828	14.14%
Property	291	1.45%
Others	92	0.46%

Category	Description	Example
What	Gives a description of functionality of the method.	*"A helper function that process the stack."*
Why	Explains the reason why the method is provided or the design rationale of the method.	*"Get a copy of the map (for diagnostics)."*
How-to-use	Describes the usage or the expected set-up of using the method.	*"Should be called before the object is used."*
How-it-is-done	Describes the implementation details of the method.	*"Convert the byte[] to a secret key."*
Property	Asserts properties of the method including pre-conditions or post-conditions of a method.	*"Wait until seqno is greater than or equal to the desired value or we exceed the timeout."*
Others	Unspecified or ambiguous comments.	*"The implementation is awesome."*

RQ1

Research Question

How do different comment categories impact the code summarization performance?

Different Code Summarization Model

- CodeNN： *original code sequence*
- Code2Seq： *random AST paths*
- DeepCom： serializing AST with SBT
- NNGen： *recommend comments based on similar code*
- Transformer: *attention mechanism*
- 2-Layer BiLSTM： *Bi-directional LSTM*

RQ1

How effective are the different methods?

Different Code Summarization Model

- CodeNN： *original code sequence*
- Code2Seq： *random AST paths*
- DeepCom： serializing AST with SBT
- NNGen： *recommend comments based on similar code*
- Transformer: *attention mechanism*
- 2-Layer BiLSTM： *Bi-directional LSTM*

Approach	Category	ROUGE-L (%)	BLEU-1 (%)	BLEU-2 (%)	BLEU-3 (%)	BLEU-4 (%)
CodeNN	What	14.36%	13.64%	3.68%	1.54%	0.90%
	Why	6.52%	6.37%	1.31%	0.42%	0.19%
	How-to-use	8.62%	8.98%	2.23%	1.01%	0.63%
	How-it-is-done	9.21%	8.08%	2.38%	0.91%	0.45%
	Property	13.34%	13.17%	4.13%	1.69%	0.00%
	Others	7.01%	7.26%	1.66%	0.00%	0.00%
	All	9.72%	9.33%	2.44%	1.01%	0.58%
Code2Seq	What	30.31%	31.66%	21.68%	17.23%	14.70%
	Why	26.71%	24.28%	15.70%	11.83%	9.91%
	How-to-use ↗	34.30%	36.76%	26.79%	21.85%	19.14%
	How-it-is-done	30.78%	30.14%	21.12%	16.98%	14.80%
	Property	29.71%	33.36%	23.91%	19.46%	17.22%
	Others	25.36%	25.48%	17.28%	14.28%	12.82%
	All	31.60%	32.25%	22.76%	18.26%	15.84%
DeepCom	What ↗	36.59%	34.51%	28.26%	24.30%	21.44%
	Why	27.48%	26.47%	20.60%	18.13%	16.89%
	How-to-use	33.28%	33.83%	26.73%	23.42%	21.58%
	How-it-is-done ↗	33.99%	32.51%	26.61%	23.99%	22.63%
	Property ↗	30.89%	31.66%	25.46%	22.07%	19.89%
	Others	27.38%	29.09%	22.79%	20.22%	18.62%
	All	33.10%	32.42%	25.94%	22.81%	21.01%
NNGen	What	35.55%	34.87%	26.26%	23.06%	21.33%
	Why ↗	29.65%	28.33%	22.09%	20.34%	19.79%
	How-to-use	32.52%	33.16%	25.46%	21.10%	18.83%
	How-it-is-done	33.39%	32.25%	27.34%	22.42%	21.53%
	Property	30.49%	28.39%	21.29%	18.50%	16.68%
	Others ↗	32.45%	32.45%	25.93%	23.49%	21.37%
	All	34.04%	33.75%	24.98%	21.87%	21.07%
Transformer	What	19.06%	11.81%	7.21%	5.27%	4.23%
	Why	20.36%	14.31%	8.84%	6.70%	5.53%
	How-to-use	20.59%	13.03%	8.22%	6.24%	5.13%
	How-it-is-done	23.14%	15.86%	10.56%	8.26%	6.98%
	Property	18.55%	11.70%	6.84%	4.87%	3.91%
	Others	18.50%	11.96%	7.84%	5.84%	4.84%
	All	20.61%	13.42%	8.48%	6.43%	5.30%
2-Layer BiLSTM	What	15.99%	9.75%	4.91%	2.96%	1.99%
	Why	17.27%	12.13%	6.37%	4.02%	2.87%
	How-to-use	18.44%	11.60%	6.40%	4.23%	3.09%
	How-it-is-done	18.85%	12.91%	7.22%	4.81%	3.60%
	Property	15.13%	9.28%	4.47%	2.46%	1.60%
	Others	13.10%	8.44%	4.19%	2.18%	1.28%
	All	17.58%	11.33%	6.11%	3.95%	2.86%

RQ1

Experimental Results (1)

Upward arrow：↗

The Best Performing method in the category

Approach	Category	ROUGE-L (%)	BLEU-1 (%)	BLEU-2 (%)	BLEU-3 (%)	BLEU-4 (%)
CodeNN	What	14.36%	13.64%	3.68%	1.54%	0.90%
	Why	6.52%	6.37%	1.31%	0.42%	0.19%
	How-to-use	8.62%	8.98%	2.23%	1.01%	0.63%
	How-it-is-done	9.21%	8.08%	2.38%	0.91%	0.45%
	Property	13.34%	13.17%	4.13%	1.69%	0.00%
	Others	7.01%	7.26%	1.66%	0.00%	0.00%
	All	9.72%	9.33%	2.44%	1.01%	0.58%
Code2Seq	What	30.31%	31.66%	21.68%	17.23%	14.70%
	Why	26.71%	24.28%	15.70%	11.83%	9.91%
	How-to-use ↗	34.30%	36.76%	26.79%	21.85%	19.14%
	How-it-is-done	30.78%	30.14%	21.12%	16.98%	14.80%
	Property	29.71%	33.36%	23.91%	19.46%	17.22%
	Others	25.36%	25.48%	17.28%	14.28%	12.82%
	All	31.60%	32.25%	22.76%	18.26%	15.84%
DeepCom	What ↗	36.59%	34.51%	28.26%	24.30%	21.44%
	Why	27.48%	26.47%	20.60%	18.13%	16.89%
	How-to-use	33.28%	33.83%	26.73%	23.42%	21.58%
	How-it-is-done ↗	33.99%	32.51%	26.61%	23.99%	22.63%
	Property ↗	30.89%	31.66%	25.46%	22.07%	19.89%
	Others	27.38%	29.09%	22.79%	20.22%	18.62%
	All	33.10%	32.42%	25.94%	22.81%	21.01%

RQ1

Experimental Results (2)

Upward arrow： ↗

The Best Performing method in the category

Approach	Category	ROUGE-L (%)	BLEU-1 (%)	BLEU-2 (%)	BLEU-3 (%)	BLEU-4 (%)
NNGen	What	35.55%	34.87%	26.26%	23.06%	21.33%
	Why ↗	29.65%	28.33%	22.09%	20.34%	19.79%
	How-to-use	32.52%	33.16%	25.46%	21.10%	18.83%
	How-it-is-done	33.39%	32.25%	27.34%	22.42%	21.53%
	Property	30.49%	28.39%	21.29%	18.50%	16.68%
	Others ↗	32.45%	32.45%	25.93%	23.49%	21.37%
	All	34.04%	33.75%	24.98%	21.87%	21.07%
Transformer	What	19.06%	11.81%	7.21%	5.27%	4.23%
	Why	20.36%	14.31%	8.84%	6.70%	5.53%
	How-to-use	20.59%	13.03%	8.22%	6.24%	5.13%
	How-it-is-done	23.14%	15.86%	10.56%	8.26%	6.98%
	Property	18.55%	11.70%	6.84%	4.87%	3.91%
	Others	18.50%	11.96%	7.84%	5.84%	4.84%
	All	20.61%	13.42%	8.48%	6.43%	5.30%
2-Layer BiLSTM	What	15.99%	9.75%	4.91%	2.96%	1.99%
	Why	17.27%	12.13%	6.37%	4.02%	2.87%
	How-to-use	18.44%	11.60%	6.40%	4.23%	3.09%
	How-it-is-done	18.85%	12.91%	7.22%	4.81%	3.60%
	Property	15.13%	9.28%	4.47%	2.46%	1.60%
	Others	13.10%	8.44%	4.19%	2.18%	1.28%
	All	17.58%	11.33%	6.11%	3.95%	2.86%

Empirical Study

Empirical

Findings

Analysis

01 No model can perform well on the “Why" category

Because the model cannot generate information that is not in the code
But NNGen has a relatively good performance in this category
-> NNGen is retrieval-based and can find similar patterns

02 No model can perform well on the "Property" category

The search space for property is too large (e.g., many parameters)
Difficult to locate precisely

03 Each model has advantages in different categories

The results are caused by differences in the generation mechanism
E.g., Learning based/Retrieval Based

➢ Can we combine different models to take advantage of different approaches?

4

Approach

RQ2

Research Question

How can we improve the code summarization performance using the comment categories?

Apporach：
Combining the advantages of different models

Composite Model

- Comment Category Prediction
  - Construct a classifier that predicts the category to which the comment of a function code belongs.
- Assign the appropriate model according to the predicted category
- Combining the advantages, e.g.:
  - Code2Seq performs well in the “How-to-use”
  - DeepCom performs well in the “What”

Core Task:

Comment Category Prediction

- Feature engineering on source code
  - Textual Features
  - Lexical features (number of lines, number of symbols, variable names, etc.)
- Constructing a classifier using the labeled data
- Comparing different classifiers

Classifier Selection

Random Forest

- LightGBM
- Decision Tree
- Naïve Bayes
- BiLSTM

5

Experiment

Comment Category Prediction

- **Random Forest (Best Results)**
- LightGBM
- Decision Tree
- Naïve Bayes
- BiLSTM

Classifier	Precision	Recall	F1
Random Forest	78.49% ± 0.64%	78.04% ± 0.52%	**76.91% ± 0.56%**
LigthGBM	74.14% ± 1.16%	74.53% ± 1.17%	74.19% ± 1.12%
Decision Tree	73.45% ± 0.84%	73.78% ± 0.84%	72.40% ± 0.88%
Naïve Bayes	69.67% ± 1.22%	57.31% ± 0.39%	46.73% ± 0.61%
BiLSTM	73.81% ± 1.15%	74.19% ± 0.99%	73.37% ± 1.01%

Composite Model

- Using Random Forest as a selector based on the priori experiments
- A new test dataset: without any priori knowledge
- Experimental results: our composite model **outperforms all benchmarks**
  - Demonstrates that using **Comment Classification** can improve code summarization

Approach	ROUGE-L	BLEU-1	BLEU-2	BLEU-3	BLEU-4
DeepCom	32.16% ± 1.77%	31.91% ± 2.77%	20.79% ± 1.84%	17.35% ± 0.17%	16.44% ± 0.10%
Code2Seq	**32.22% ± 0.89%**	30.99% ± 0.10%	24.11% ± 0.23%	17.76% ± 0.51%	16.07% ± 0.86%
NNgen	30.57%	29.72%	24.56%	20.32%	**17.14%**
Ours	**34.98% ± 2.09%**	32.66% ± 2.41%	25.76% ± 0.12%	21.58% ± 0.17%	**19.94% ± 0.22%**

6

Conclusion

Conclusion

Core Question:

Are all kinds of code suitable for comment generation?

Six Categories

"What", "Why", "How-to-use", "How-it-is-done", "Property", "Others"

Labeling 20,000 pairs

Labor-intensive:
57 pairs every hour on average

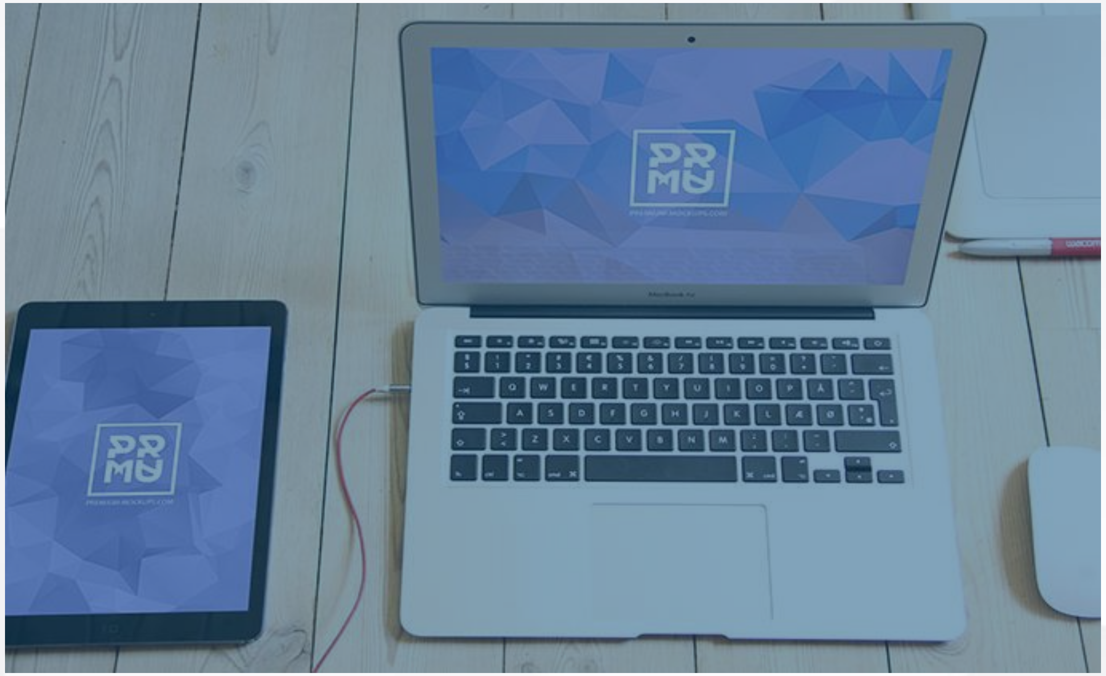
Comment Category Prediction

Random Forest (Best Results)

Composite Model

Assigning optimal generators

Conclusion



Implications on Practice

More Specific Category

More specific than "What", "How"
Associated with specific business

Collect More Labeled Data

Code comments in areas such as games, communications

Explore Code Intention

Consider more information
e.g., Context information

Code Documentation

Not only code comment

Why My Code Summarization Model Does Not Work?

Thank you for listening!

Qiuyuan Chen

2022.04.17

GitHub: https://github.com/chenqiuyuan/TOSEM_CodeSum

Qiuyuan Chen, Xin Xia, Han Hu, David Lo, Shanping Li. In ACM Transactions on Software Engineering and Methodology (TOSEM).

Why My Code Summarization Model Does Not Work?

Code Comment Improvement with Category Prediction





Qiuyuan Chen Zhejiang University

Xin Xia HUAWEI

Han Hu Tsinghua University

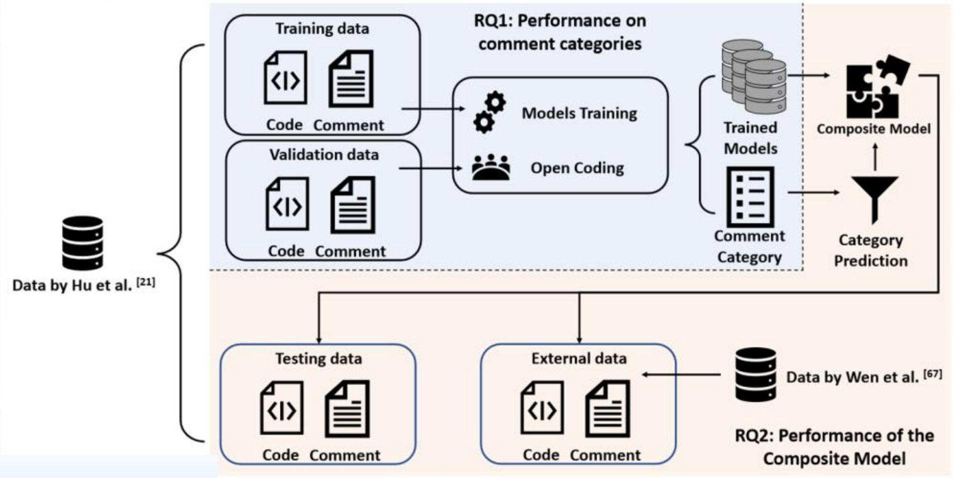
David Lo Singapore Management University

Shanping Li Zhejiang University

Qiuyuan Chen, Xin Xia, Han Hu, David Lo, Shanping Li. In ACM Transac

Research Question

Research Overview



Q&A

Empirical Study

RQ1

How effective are the different methods?

Different Code Summarization Model

- CodeNN: *original code sequence*
- Code2Seq: *random AST paths*
- DeepCom: serializing AST with SBT
- NNGen: *recommend comments based on similar code*
- Transformer: *attention mechanism*
- 2-Layer BiLSTM: *Bi-directional LSTM*

Approach	Category	R[illegible] (%)	[illegible] (%)	[illegible] (%)	[illegible] (%)	[illegible] (%)
CodeNN	What	14.36%	13.64%	3.68%	1.54%	0.90%
	Why	6.52%	6.37%	1.31%	0.42%	0.19%
	How-to-use	8.62%	8.98%	2.23%	1.01%	0.63%
	How-it-is-done	9.21%	8.08%	2.38%	0.91%	0.45%
	Property	13.34%	13.17%	4.13%	1.69%	0.00%
	Others	7.01%	7.26%	1.66%	0.00%	0.00%
	All	9.72%	9.33%	2.44%	1.01%	0.58%
Code2Seq	What	30.31%	31.66%	21.68%	17.23%	14.70%
	Why	26.71%	24.28%	15.70%	11.83%	9.91%
	How-to-use ↗	34.30%	36.76%	26.79%	21.85%	19.14%
	How-it-is-done	30.78%	30.14%	21.12%	16.98%	14.80%
	Property	29.71%	33.36%	23.91%	19.46%	17.22%
	Others	25.36%	25.48%	17.28%	14.28%	12.82%
	All	31.60%	32.25%	22.76%	18.26%	15.84%
DeepCom	What ↗	36.59%	34.51%	28.26%	24.30%	21.44%
	Why	27.48%	26.47%	20.60%	18.13%	16.89%
	How-to-use	33.28%	33.83%	26.73%	23.42%	21.58%
	How-it-is-done ↗	33.99%	32.51%	26.61%	23.99%	22.63%
	Property ↗	30.89%	31.66%	25.46%	22.07%	19.89%
	Others	27.38%	29.09%	22.79%	20.22%	18.62%
	All	33.10%	32.42%	25.94%	22.81%	21.01%
NNGen	What	35.55%	34.87%	26.26%	23.06%	21.33%
	Why ↗	29.65%	28.33%	22.09%	20.34%	19.79%
	How-to-use	32.52%	33.16%	25.46%	21.10%	18.83%
	How-it-is-done	33.39%	32.25%	27.34%	22.42%	21.53%
	Property	30.49%	28.39%	21.29%	18.50%	16.68%
	Others ↗	32.45%	32.45%	25.93%	23.49%	21.37%
	All	34.04%	33.75%	24.98%	21.87%	21.07%
Transformer	What	19.06%	11.81%	7.21%	5.27%	4.23%
	Why	20.36%	14.31%	8.84%	6.70%	5.53%
	How-to-use	20.59%	13.03%	8.22%	6.24%	5.13%
	How-it-is-done	23.14%	15.86%	10.56%	8.26%	6.98%
	Property	18.55%	11.70%	6.84%	4.87%	3.91%
	Others	18.50%	11.96%	7.84%	5.84%	4.84%
	All	20.61%	13.42%	8.48%	6.43%	5.30%
2-Layer BiLSTM	What	15.99%	9.75%	4.91%	2.96%	1.99%
	Why	17.27%	12.13%	6.37%	4.02%	2.87%
	How-to-use	18.44%	11.60%	6.40%	4.23%	3.09%
	How-it-is-done	18.85%	12.91%	7.22%	4.81%	3.60%
	Property	15.13%	9.28%	4.47%	2.46%	1.60%
	Others	13.10%	8.44%	4.19%	2.18%	1.28%
	All	17.58%	11.33%	6.11%	3.95%	2.86%

Composite Model

- Using Random Forest as a selector based on the priori experiments
- A new test dataset: without any priori knowledge
- Experimental results: our composite model **outperforms all benchmarks**
  - Demonstrates that using **Comment Classification** can improve code summarization

Approach	ROUGE-L	BLEU-1	BLEU-2	BLEU-3	BLEU-4
DeepCom	32.16% ± 1.77%	31.91% ± 2.77%	20.79% ± 1.84%	17.35% ± 0.17%	16.44% ± 0.10%
Code2Seq	**32.22% ± 0.89%**	30.99% ± 0.10%	24.11% ± 0.23%	17.76% ± 0.51%	16.07% ± 0.86%
NNgen	30.57%	29.72%	24.56%	20.32%	**17.14%**
Ours	**34.98% ± 2.09%**	32.66% ± 2.41%	25.76% ± 0.12%	21.58% ± 0.17%	**19.94% ± 0.22%**