Paper Presentation

“Retrieval-based Neural Source Code Summarization”

Presented by

Md Mynuddin
ID : ASH1825007M
Mahbub Alam
ID : ASH1825003M
NSTU, IIT

Published on

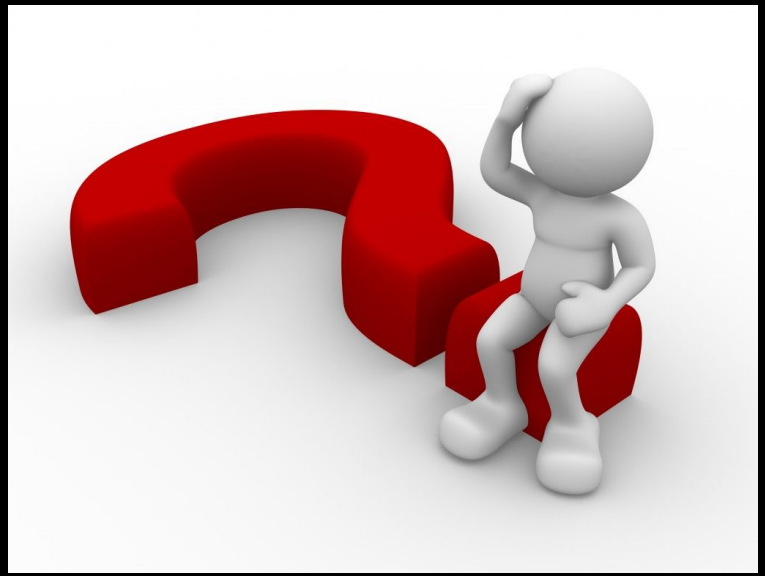
“International
Conference on Software Engineering”
Seoul, Republic of Korea
Accepted: 23-29 May 2020

Introduction

- Source code summarization is the process of generating concise descriptions of source code, often in the form of code comments.
- This is important for understanding and maintaining source code, as developers often spend a lot of time reading and comprehending programs without good software documentation.
- There have some automatic source code summarization techniques
  - Information Retrieval(IR) for Term-based summaries
  - Neural Machine Translation(NMT) for generate summaries word-by-word from code snippets

Problem

- NMT-based approaches have difficulty in handling low-frequency words.
- It may result in incorrect summaries.

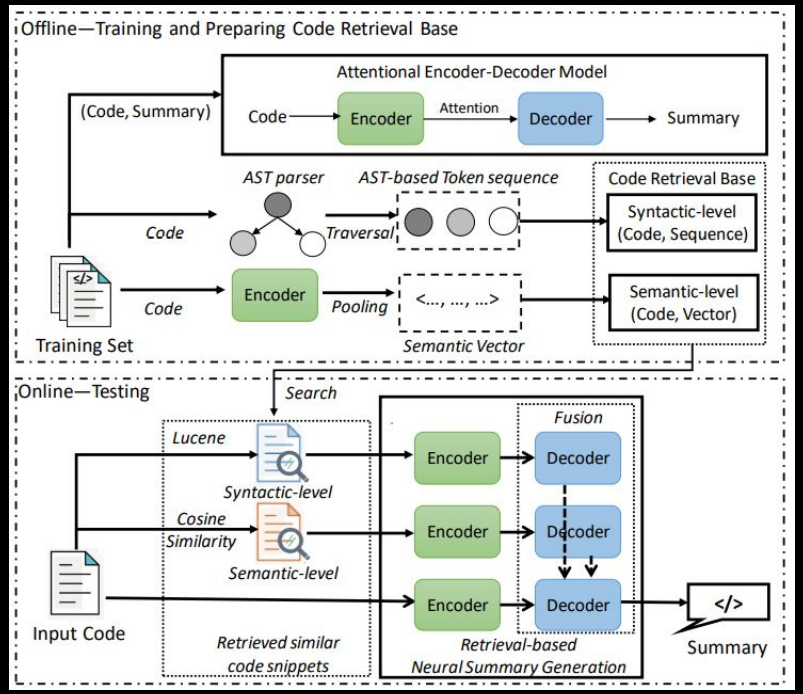


Solution

- The authors proposes a solution called Retrieval-based Neural Source Code Summarizer (Rencos)
- They conducted extensive experiments on two real-world datasets and performed a human evaluation through Amazon Mechanical Turk (AMT) to evaluate their proposed approach.
- It combines the advantages of both NMT-based and retrieval-based methods for source code summarization.
- The proposed architecture first trains an attentional encoder-decoder model to obtain an encoder for all code samples and a decoder for generating natural language summaries.

Solution (Continue…)

- The Figure shows the overall framework of the solution



Solution

- Then, given an input source code snippet, it retrieves the most similar code snippets from the training set based on both syntax-level and semantics-level information.
- These retrieved code snippets are used as context vectors and their similarity values are used to adjust the conditional probability of the next word during decoding. This helps to enhance the prediction results of the NMT model.

Experiments

- They conduct experiment to evaluate the effectiveness of the proposed approach.
- Compare it with several state of the art methods from Software Engineering (SE) .
- And Natural Language Processing (NLP) communities.

Experimental Setup

Two public large-scale datasets :

- PCSD(Python Code Summarization Dataset) : Provided by Barone
  - Contains Python Function and Comment
  - This dataset includes 108,726 code-comment pairs.
  - Used for training and evaluation
- JCSD(Java Code Summarization Dataset) : Collected by Hu
  - Contains Java methods and comments.
  - This dataset includes 69,708 code and comment pairs.
  - Used for training and evaluation

Evaluation Metrics

They evaluate the performance of different approaches using common metrics including:

- ➔ BLEU
- ➔ METEOR
- ➔ ROUGE-L
- ➔ CIDER

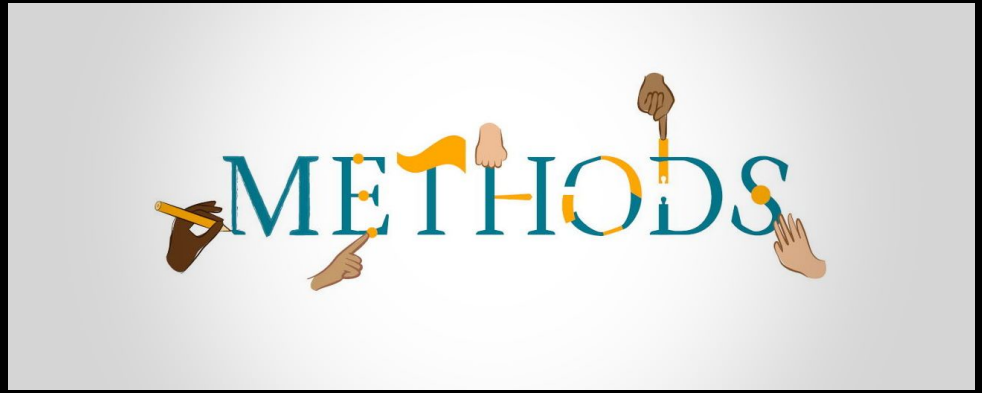
Baseline Methods

Retrieval-based approaches :

- LSI
- VSM
- NNGen

NMT-based approaches :

- CODE-NN
- TL-CodeSum
- Hybrid-DRL



Result

Dataset	Source Code Length (#words)			Summary Length (#words)			Word Frequency ≤10		Word Frequency ≤100	
	MaxL	AvgL	UniT	MaxL	AvgL	UniT	NumW	NumS	NumW	NumS
PSCD	157,116	133.1	481,756	333	9.9	37,111	32,093(86.5%)	46,481(42.8%)	36,003(97.0%)	87,626(80.6%)
JSCD	4842	99.9	230,336	670	17.1	35,535	30,342(85.4%)	34,207(41.4%)	34,223(96.3%)	63,954(77.3%)

The statistics of two datasets

Methods	PCSD							JCSD						
	BLEU-1/2/3/4(%)				METEOR(%)	ROUGE-L(%)	CIDER	BLEU-1/2/3/4(%)				METEOR(%)	ROUGE-L(%)	CIDER
LSI	36.3	23.6	20.1	17.6	17.2	40.0	1.982	31.4	22.5	19.3	17.3	14.4	34.8	1.803
VSM	38.9	26.1	22.1	19.3	19.0	42.7	2.216	33.3	24.4	21.1	19.0	15.4	36.6	1.983
NNGen	36.5	23.8	20.1	17.4	17.1	40.2	1.967	33.0	24.4	20.9	18.7	15.0	36.3	1.933
CODE-NN	30.8	15.4	10.7	8.1	13.4	35.1	1.229	23.9	12.8	8.6	6.3	9.1	28.9	0.978
TL-CodeSum	31.1	16.5	12.5	10.4	13.6	35.3	1.335	29.9	21.3	18.1	16.1	13.7	33.2	1.66
Hybrid-DRL	41.1	26.2	19.5	15.0	17.9	42.2	2.042	32.4	22.6	16.3	13.3	13.5	36.5	1.656
GRNMT	38.6	24.0	18.8	15.8	18.5	43.4	1.978	32.6	22.6	17.9	15.5	15.0	37.6	1.732
Rencos	**43.1**	**29.5**	**24.2**	**20.7**	**21.1**	**47.5**	**2.449**	**37.5**	**27.9**	**23.4**	**20.6**	**17.3**	**42.0**	**2.209**

Method comparison for source code summarization

Continue..

Descriptions	PCSD							JCSD						
	BLEU-1/2/3/4(%)				METEOR(%)	ROUGE-L(%)	CIDER	BLEU-1/2/3/4(%)				METEOR(%)	ROUGE-L(%)	CIDER
Only Syntactic Retrieval	39.8	27.4	23.3	20.2	19.5	43.5	2.296	33.9	25.2	21.7	19.5	15.9	37.4	2.020
Only Semantic Retrieval	39.5	27.1	23.1	20.1	19.1	43.1	2.270	33.7	25.3	22.1	19.9	15.4	37.0	2.049
NMT	37.5	22.5	17.1	14.2	17.3	42.3	1.871	31.1	20.7	16.0	13.8	13.8	36.3	1.633
NMT+Syntactic Retrieval	41.9	28.2	22.8	19.5	20.4	46.5	2.344	36.3	26.7	22.1	19.5	16.7	40.9	2.106
NMT+Semantic Retrieval	42.2	28.4	23.2	19.8	20.6	46.6	2.362	36.8	27.2	22.6	19.9	17.0	41.3	2.164
NMT+Both Retrieval	**43.1**	**29.5**	**24.2**	**20.7**	**21.1**	**47.5**	**2.449**	**37.5**	**27.9**	**23.4**	**20.6**	**17.3**	**42.0**	**2.209**

Effectiveness of each component of the proposed approach

Related Works

Information Retrieval techniques are widely used for automatic source code summarization.

1. Haiduc use IR methods including LSI and VSM to choose top-k terms from a code snippet.
2. Rodeghero et al. improve the process of selecting terms by eye-tracking and modify the weights of VSM for better code summarization
3. Sridhara et al. [52] design Software Word Usage Model (SWUM) to identify keywords from those statements and create summaries though manually-crafted templates.

Limitation

- Proposed approach only for Python and Java code - comment dataset.
- when the code base is small proposed approach is not work well.
- For the low frequency of words need more large scale dataset.

Future Work

- Experiment with even larger-scale datasets.
- Evaluate the effectiveness of “Rencos” in handling low-frequency words.

Interesting Part

Their experimental data and results are publicly available

- https://github.com/zhangj111/rencos
- https://github.com/xing-
- https://github.com/OpenNMT/OpenNMT-
- https://github.com/sriniiyer/codenn
- https://github.com/wanyao1992/code_summarization_public

Conclusion

- They propose a novel retrieval-based neural approach named “Rencos”
- Used an attentional encoder-decoder model to find the two most comparable code snippets for enhanced source code summary.
- To accommodate more information about the code, they also develop two code retrieval methods from the perspectives of syntax and semantics.
- Syntactic level (Lucene) and Semantic level (Cosine similarity).
- And finally evaluated the effectiveness of their approach.


A Big
thanks
to you All!