Retrieval-based Neural Source Code Summarization

Jian Zhang[1] Xu Wang[2] Hailong Sun[4] Xudong Liu[5]
SKLSDE Lab, Beihang University, China

Hongyu Zhang[3]
The University of Newcastle, Australia

No Photo

Name: Jian Zhang

Affiliation: Beihang University

Contributions

ISSTA 2019	Author of Androlic: An Extensible Flow, Context, Object, Field, and Path-Sensitive Static Analysis Framework for Android within the Tool Demonstration-track
ICSE 2020	Author of Retrieval-based Neural Source Code Summarization within the Technical Papers-track
ICSE 2019	Author of A Novel Neural Source Code Representation based on Abstract Syntax Tree within the Technical Track-track
ISSTA 2018	Author of LAND: A User-Friendly and Customizable Test Generation Tool for Android Apps within the ISSTA Tool Demonstrations-track
* **ICSE 2018** *	Author of Poster T39: Semantically Enhanced Tag Recommendation for Software CQAs via Deep Learning within the Posters -track

Share

Xu Wang (王旭)

Assistant Professor

Research Directions:
Software Engineering, Software Reliability and Scalability, Distributed Systems

Research Interests:
Improve software development efficiency and software quality through AI techniques, program analysis and algorithm optimization.

Hongyu Zhang

Contact Email: hongyujohn@gmail.com

Research Area:

My research area is software engineering, in particular:

- software analytics, mining software repository, data-driven software engineering
- software measurement and empirical software engineering
- software quality assurance, testing, debugging
- software reuse (generative programming and software product lines)
- software maintenance

孙海龙 Hailong Sun

Associate Professor, School of Computer Science and Engineering, Beihang University

Intelligent Software Engineering (combining AI/big data with software engineering),
Crowd Intelligence (crowdsourcing, human computation, human-AI collaboration),
Distributed Systems,
Service Oriented Computing

Xudong Liu

Beihang University

ICSE 2020	Author of Retrieval-based Neural Source Code Summarization within the Technical Papers-track
ICSE 2019	Author of A Novel Neural Source Code Representation based on Abstract Syntax Tree within the Technical Track-track
*** ICSE 2018 ***	Author of Poster T39: Semantically Enhanced Tag Recommendation for Software CQAs via Deep Learning within the Posters -track

Related Work

- Information Retrieval
  - On the use of automated text summarization techniques for summarizing source code.
  - Clocom: Mining existing source code for automatic comment generation

- Neural Machine Translation
  - Summarizing source code using a neural attention model
  - Deep code comment generation

Motivation

- The NMT-based methods generally prefer high-frequency words in the corpus and may have trouble with low-frequency words
- Combine the NMT-based and IR-based methods
  - the words in the expected summaries (including the low-frequency ones) are also highly probable to appear in similar code snippets

Motivation

```
def create_app(name, site, sourcepath, apppool=None):
    pscmd = list()
    pscmd.append("New-WebApplication -Name '{0}' -
                        Site'{1}'".format(name,site))
    pscmd.append(" -PhysicalPath '{0}'".format(sourcepath))
    if apppool:
       pscmd.append(" -applicationPool '{0}'".format(apppool))
    cmd_ret = _srvmgr(str().join(pscmd))
    if  cmd_ret['retcode'] == 0:
        if name in list_apps(site): return True
    return False
```

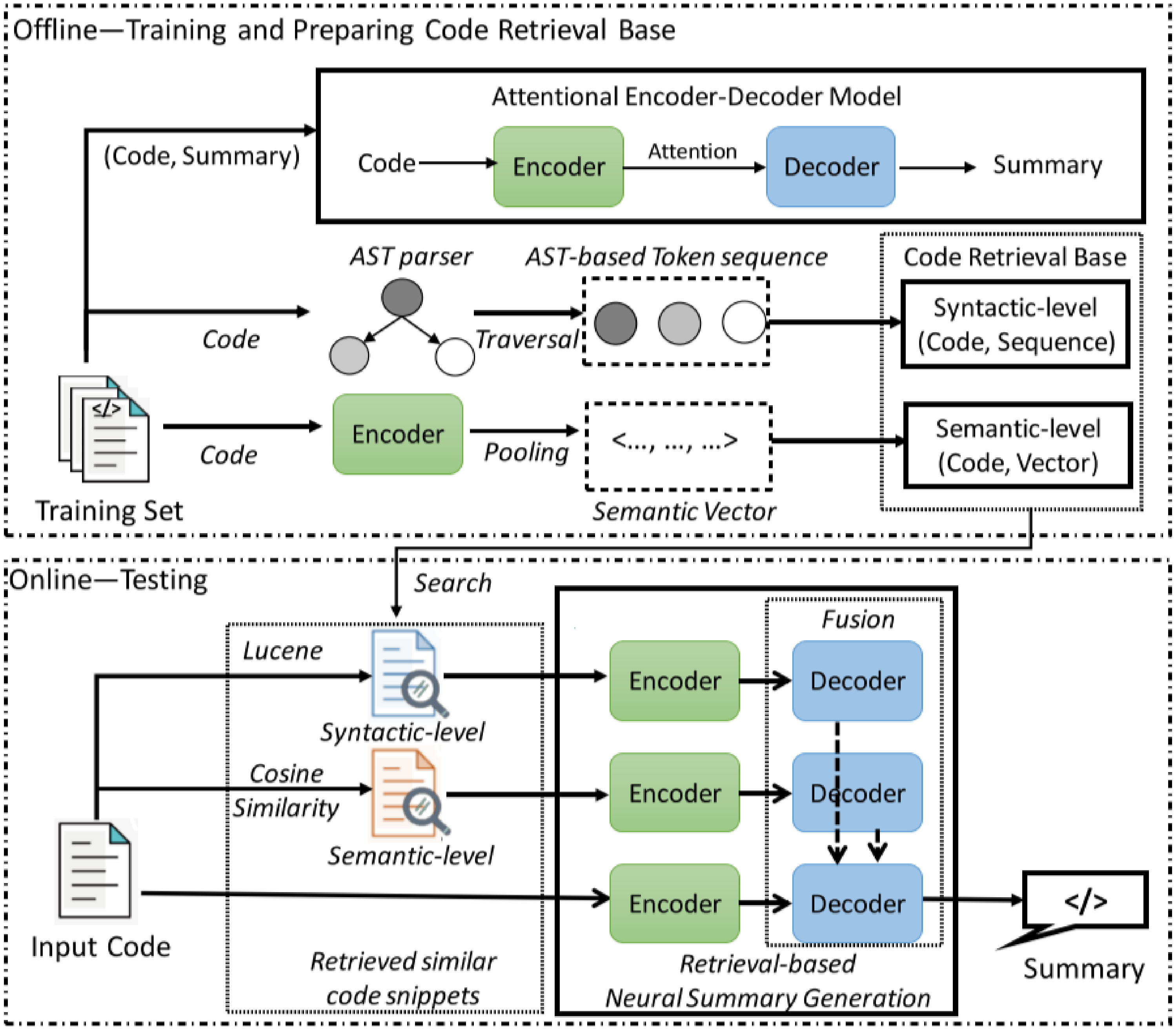
Ground truth: create an iis application .

Syntactic retrieval: remove an iis application .

Semantic retrieval: create an iis virtual directory .

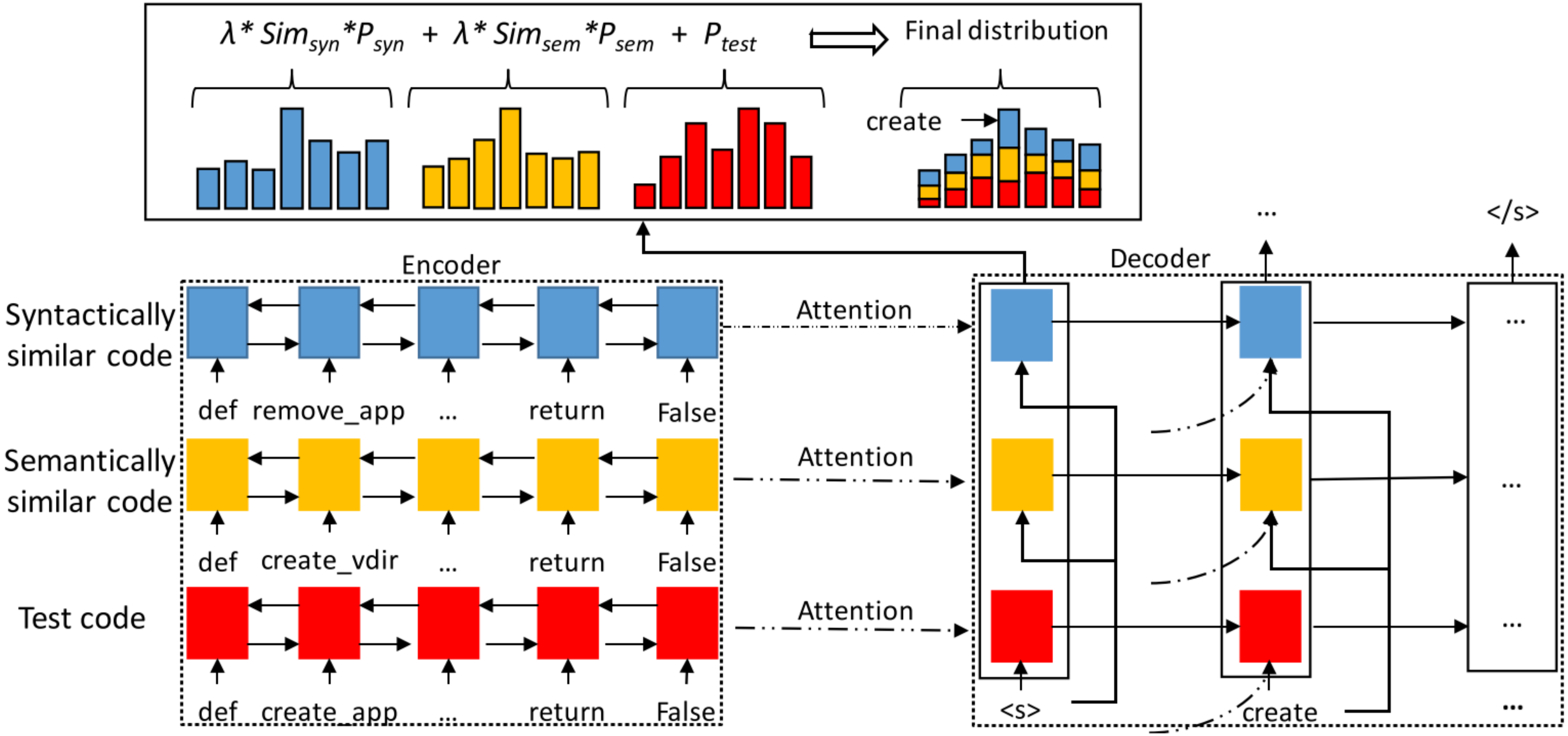
NMT: create the new app .

Overview



Offline—Training and Preparing Code Retrieval Base
Attentional Encoder-Decoder Model
Code
Encoder
Attention
Decoder
Summary
(Code, Summary)
AST parser
AST-based Token sequence
Code Retrieval Base
Code
Traversal
Syntactic-level (Code, Sequence)
Encoder
Code
Pooling
<..., ..., ...>
Semantic-level (Code, Vector)
Training Set
Semantic Vector
Online—Testing
Search
Fusion
Lucene
Encoder
Decoder
Syntactic-level
Cosine Similarity
Encoder
Decoder
Semantic-level
Encoder
Decoder
</>
Input Code
Retrieved similar code snippets
Retrieval-based Neural Summary Generation
Summary

Retrieval-based Neural Summary Generation



Evaluation

- Dataset
  - PCSD： 10872 pairs、
  - JCSD: 69708 pairs

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%)

Evaluation

- Baseline

	Method	Description
IR	LSI	Key world similarity
	VSM	Word frequence similarity
	NNGen	Vector similarity
NMT	CODE-NN	LSTM+attention
	TL-CODESum	API+LSTM+attention
	Hybrid-DRL	AST+LSTM+attention
IR&NMT	GRNMT	N-gram represented

Evaluation

- RQ1:Howdoes our proposed approach perform compared to the baselines?

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**

Example

```
public void removeColumn(final String columnName){
  if(columnName == null) {  return; }
  List<String> cols = Arrays.asList(getInfo().headers);
  final int colIndex = cols.indexOf(columnName);
  removeColumn(colIndex);
}
```

Reference: remove the column represented by its name
LSI: get index of this column name
VSM: adds the given column to this table
NNGen: get index of this column name
CODE-NN: remove a column from the table .
TL-CodeSum: remove column at specified index .
Hybrid-DRL: removes a column from the column .
GRNMT: remove a column from the table .
Our approach: remove the column represented by the index

Evaluation

- RQ2: How effective are the main components of our approach?

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**

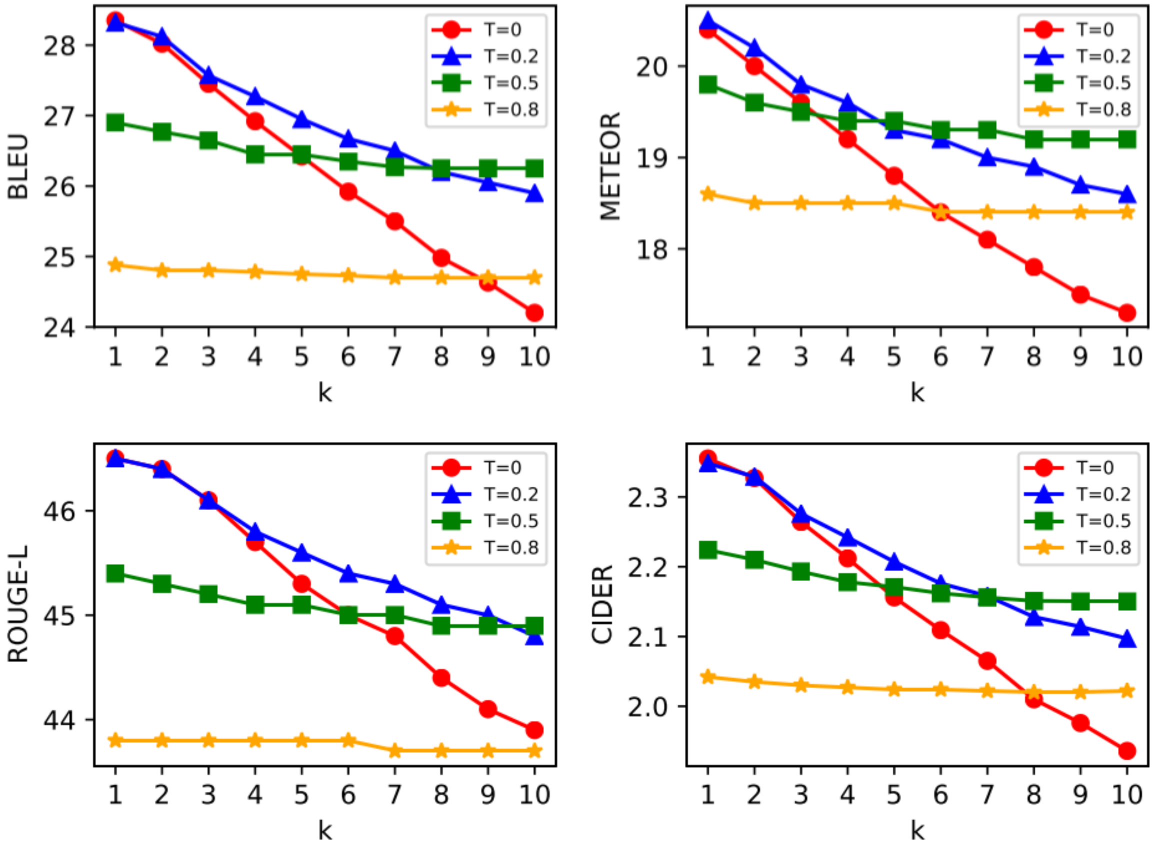
Evaluation

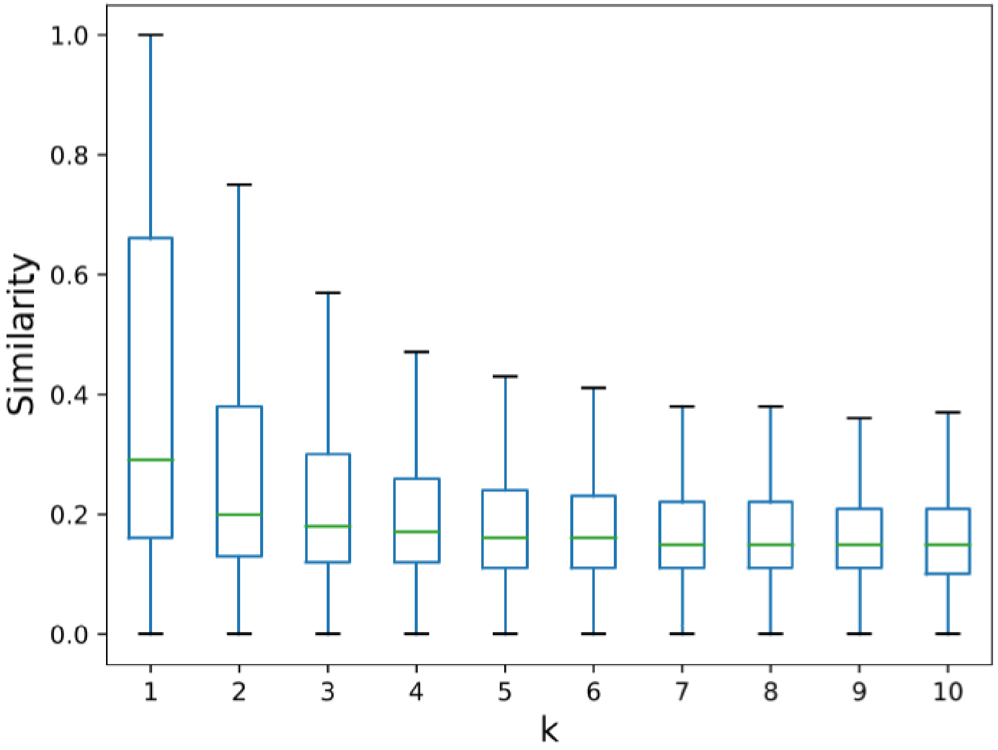
- RQ3: Does our approach perform better than NMT-based methods for tackling the low-frequency word problem?

Word Frequency		1	2	5	10	20	50	100
PCSD	NMT	452	376	272	176	145	84	82
	Rencos	799	515	344	223	184	88	109
	Ratio	1.77	1.37	1.26	1.27	1.27	1.05	1.33
JCSD	NMT	126	75	45	27	38	28	16
	Rencos	243	138	73	38	49	37	18
	Ratio	1.93	1.84	1.62	1.41	1.29	1.32	1.11

Evaluation

- RQ4: Will our approach perform better if we retrieve top k (k>1) similar code snippets and filter them according to a similarity threshold





Evaluation

- Human Evaluation: Amazon Mechanical Turk (AMT)

Score	1	2	3	4	5	Avg	≥4	≥3	≤2
VSM	24	18	31	17	10	2.71	27	58	42
Hybrid-DRL	0	26	48	23	3	3.03	26	74	26
Rencos	2	13	39	30	16	3.45	46	85	15

Conclusion

- We propose a novel retrieval-based neural architecture to enhance the NMT model for summarizing source code with the help of most similar code snippets. To the best of our knowledge, this is the first work that combines retrieval-based and NMT-based methods in source code summarization

- We conduct extensive experiments to evaluate our approach on two real-world datasets. We also perform a human evaluation through Amazon Mechanical Turk (AMT). All results confirm that the proposed approach is effective and outperforms the state-of-the-art methods.