三篇论文

SCIENCE AND TECHNOLOGY

《code2seq: Generating Sequences from Structured Representations of Code》	2019	ICLR
《Code2vec : learning distributed representations of code》	2019	POPL
《Assessing the generalizability of code2vec token embeddings》	2019	ASE
《Can pre-trained code embeddings improve model performance? Revisiting the use of code embeddings in software engineering tasks》	2022	EMSE

Code2vec : learning distributed representations of code

2019 POPL

Goal: The goal of this paper is to learn code embeddings, continuous vectors for representing snippets of code.

> *Methods for learning distributed representations produce **low-dimensional vector representations for objects**, referred to as **embeddings**. In these vectors, the "meaning" of an element is distributed across multiple vector components.*

```
String[] f(final String[] array) {
    final String[] newArray = new String[array.length];
    for (int index = 0; index < array.length; index++) {
        newArray[array.length - index - 1] = array[index];
    }
    return newArray;
}
```

Predictions

reverseArray	77.34%
reverse	18.18%
subArray	1.45%
copyArray	0.74%

Fig. 1. A code snippet and its predicted labels as computed by our model.

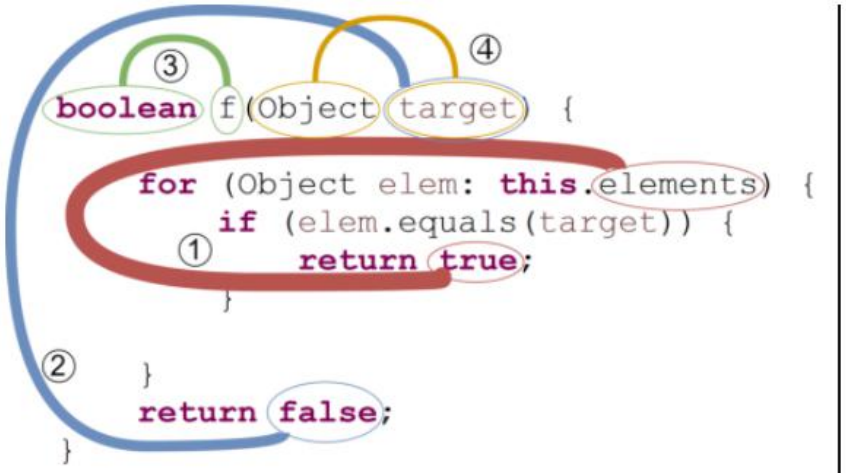
Code2vec : learning distributed representations of code

2019 POPL

Path-attention Interpretation

https://code2vec.org/

$$\boldsymbol{c_i} = embedding\left(\langle x_s, p_j, x_t \rangle\right) = \left[\ value\_vocab_s\ ;\ path\_vocab_j\ ;\ value\_vocab_t\ \right] \in \mathbb{R}^{3d} \quad (1)$$



(a)

Predictions:

contains	90.93%
matches	3.54%
canHandle	1.15%
equals	0.87%
containsExact	0.77%

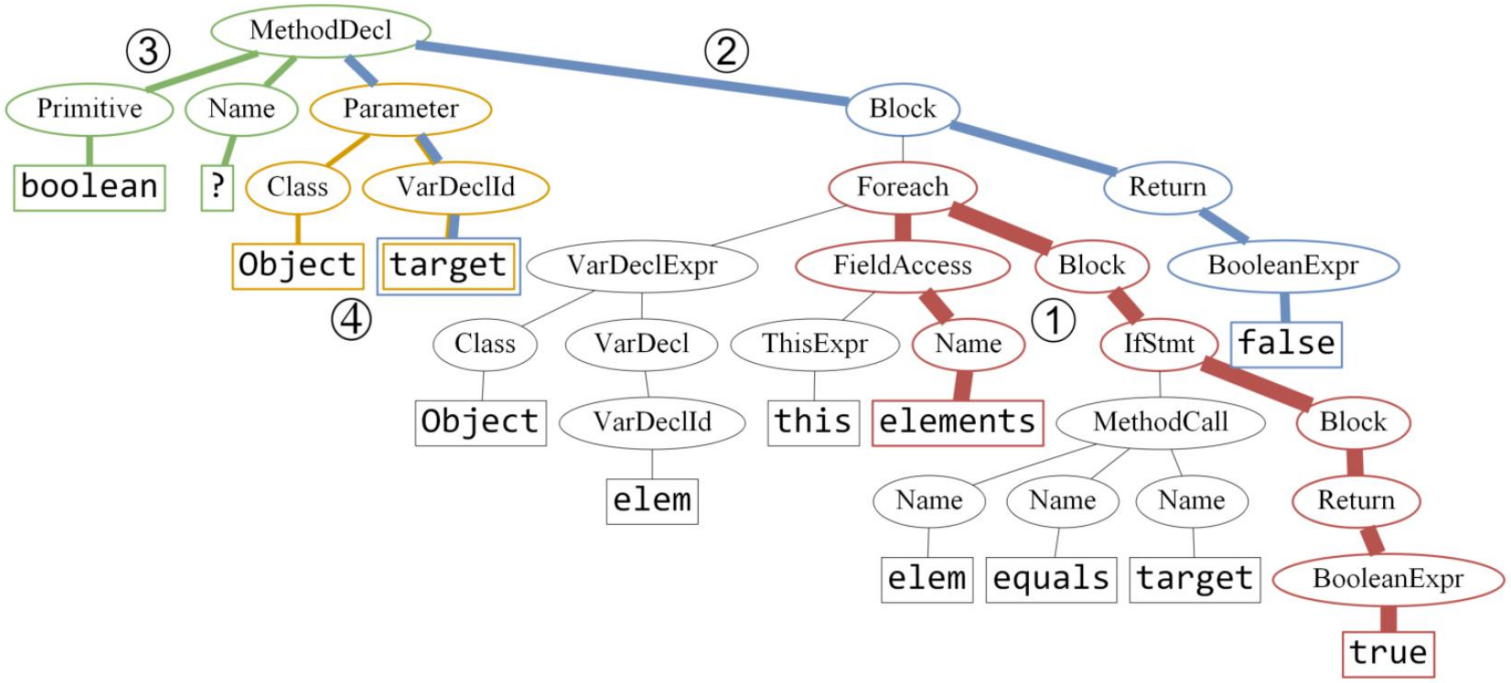


Fig. 3. The top-4 attended paths of Figure 2a, as were learned by the model, shown on the AST of the same snippet. The width of each colored path is proportional to the attention it was given (**red** ①: **0.23**, **blue** ②: **0.14**, **green** ③: **0.09**, **orange** ④: **0.07**).

Code2vec : learning distributed representations of code

Path-attention Model

2019 POPL

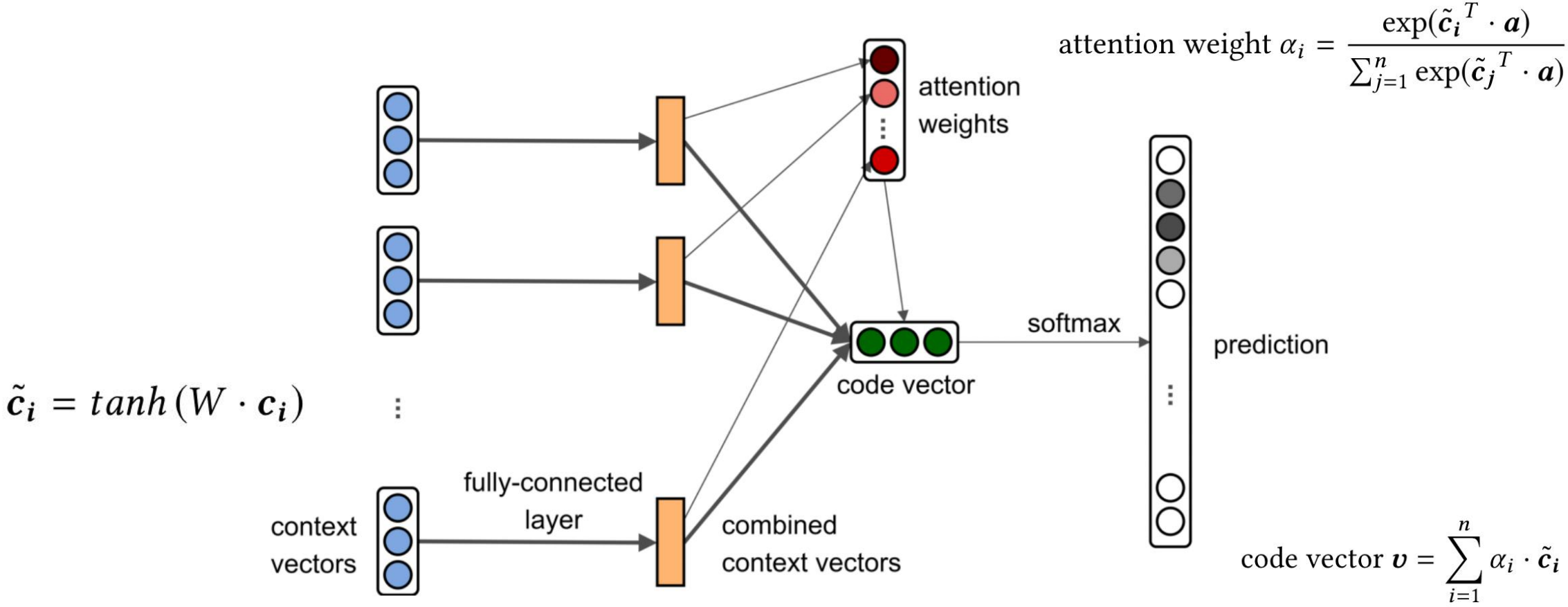
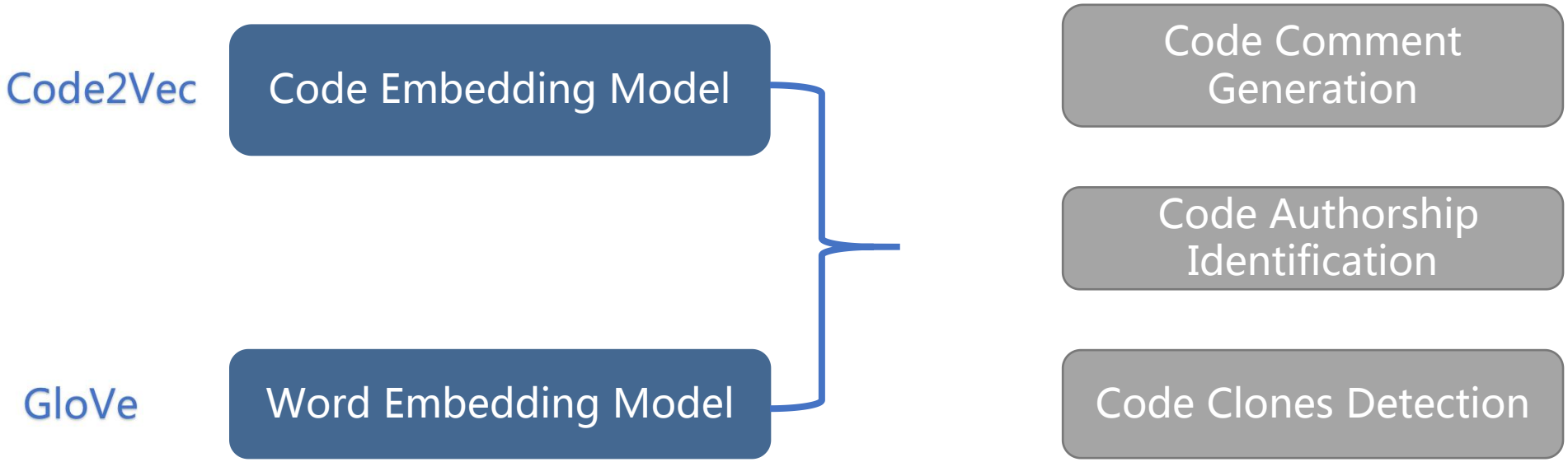


Fig. 4. The architecture of our path-attention network. A *fully connected layer* learns to combine embeddings of each path-context with itself; attention weights are learned using the combined context vectors and used to compute a *code vector*. The code vector is used to predict the label.

Assessing the generalizability of code2vec token embeddings

2019 ASE

Research Question: *Are embeddings of source code tokens generalizable for use in tasks that they are not trained for?*



Assessing the generalizability of code2vec token embeddings

2019 ASE

- **Code Comment Generation** Seq2Seq Model

TABLE I
QUALITY OF COMMENTS GENERATED, WITH SBT PREPROCESSING

Preprocessing	Embedding model	BLEU-4
Lowercased	GloVe	27.4
Lowercased	code2vec	29.9
Lowercased	No pretrained embeddings	28.1
Non-lowercased	GloVe	28.1
Non-lowercased	code2vec	29.3
Non-lowercased	**No pretrained embeddings**	**33.5**

TABLE II
JALITY OF COMMENTS GENERATED, WITHOUT SBT PREPROCESSING

Preprocessing	Embedding model	BLEU-4
Lowercased	GloVe	29.7
Lowercased	code2vec	29.3
Lowercased	**No pretrained embeddings**	**31.3**
Non-lowercased	GloVe	22.0
Non-lowercased	code2vec	31.0
Non-lowercased	No pretrained embeddings	26.7

Assessing the generalizability of code2vec token embeddings

2019 ASE

- Code Authorship Identification

TABLE III
ACCURACY FOR IDENTIFICATION OF CODE AUTHORSHIP

Setting	Accuracy
LSTM, code2vec	39
LSTM, GloVe	50
LSTM, randomly initialized	69
Fully connected layers, TF-IDF	**77**

- Code Clones Detection

TABLE V
RECALL AND PRECISION ON THE OJCLONE DATASET

Setting	Precision	Recall	F1
code2vec	0.03	0.45	**0.06**
GloVe	0.03	**0.67**	**0.06**
SourcererCC	**0.87**	0.01	0.01
Random	0.01	0.02	0.01

Assessing the generalizability of code2vec token embeddings

2019 ASE

- From the 3 tasks above, we see that code embeddings cannot be used readily to improve simpler models.
- Code embeddings may not be a silver bullet to boost the performance of deep learning models; other considerations may have more impact.
- It may indicate that token embeddings learned over source code may not encode a significant amount of either semantic or syntactic information usable in different downstream tasks.

GRADUATION DEFENSE

Can pre-trained code embeddings improve model performance?

Revisiting the use of code embeddings in software engineering tasks

作 者 ： Zishuo Ding，Heng Li，Weiyi Shang，Tse-Hsun (Peter) Chen

汇报人：陈冰婷

导师：邹卫琴

浙江工业大学
ZHE JIANG UNIVERSITY OF TECHNOLOGY

Content

SCIENCE AND TECHNOLOGY

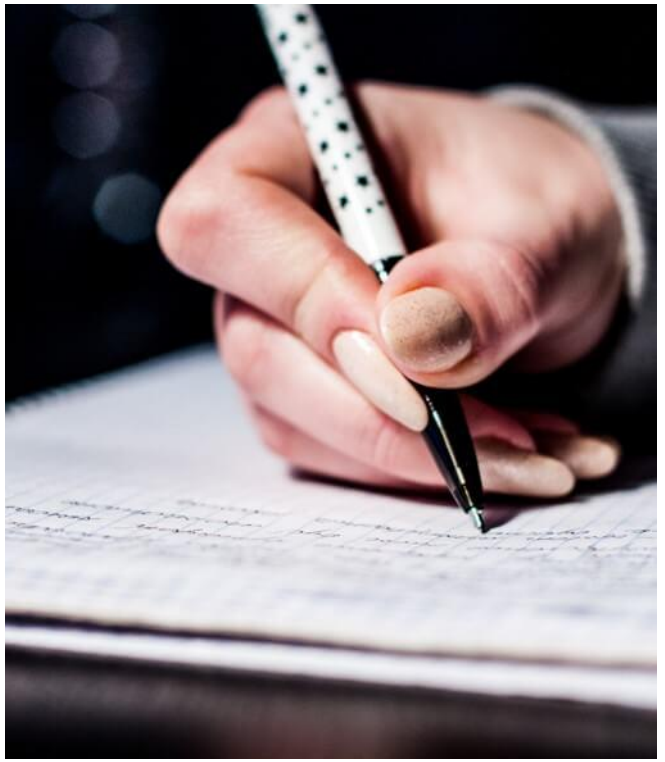
GRADUATION DEFENSE

Background

SCIENCE AND TECHNOLOGY

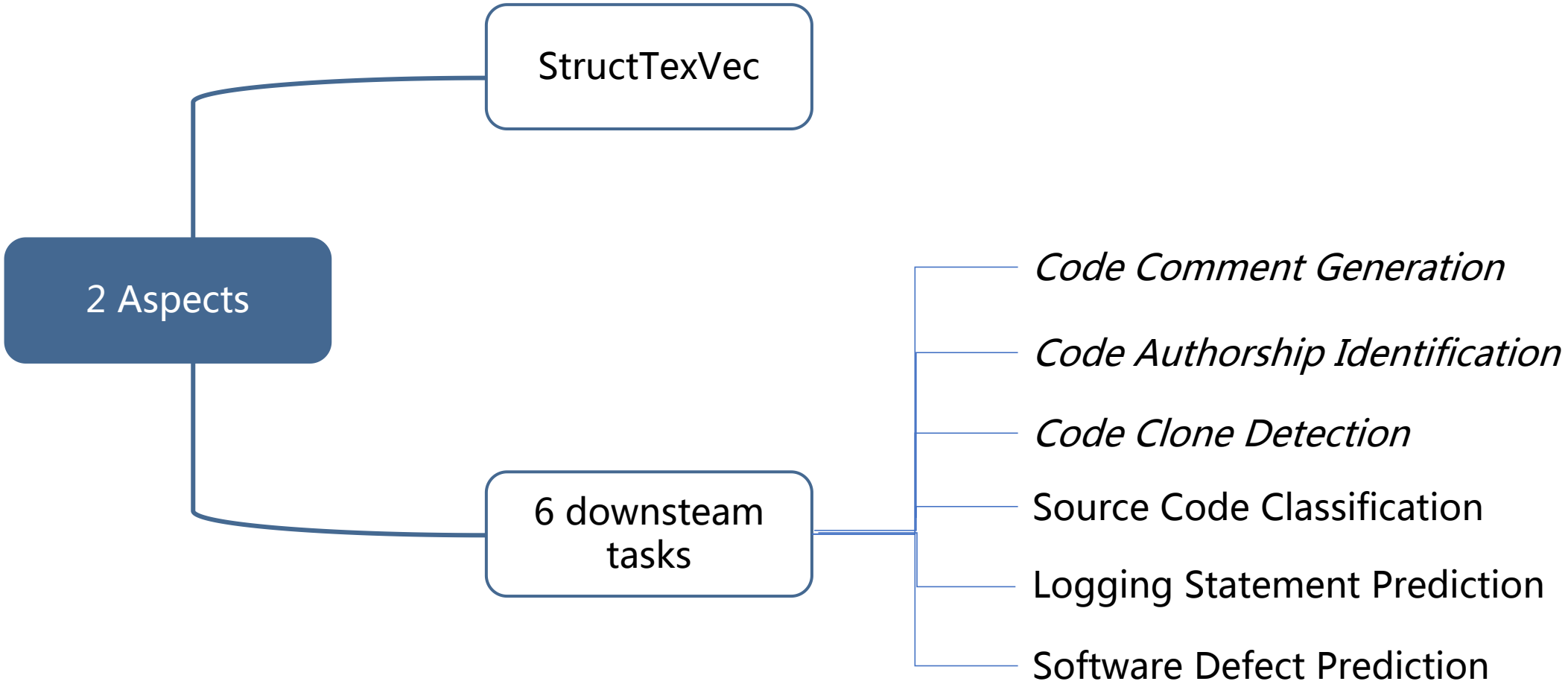
A recent study by Kang et al. (2019) evaluates two code embedding approaches(i.e., GloVe (Pennington et al. 2014) and code2vec (Alon et al. 2019) on three downstream SE tasks.

- *code comment generation*
- *code authorship identification*
- *code clone detection*



Introduction

SCIENCE AND TECHNOLOGY



Introduction

SCIENCE AND TECHNOLOGY

Textual Context

— which is the plain text of the source code

Consider source code as plain text and directly apply existing word embedding techniques to source code.

Structual Context

— which refers to the abstract syntax trees (ASTs) of the source code.

Due to its ability of capturing not only the lexical information but also the syntactic structure of source code.

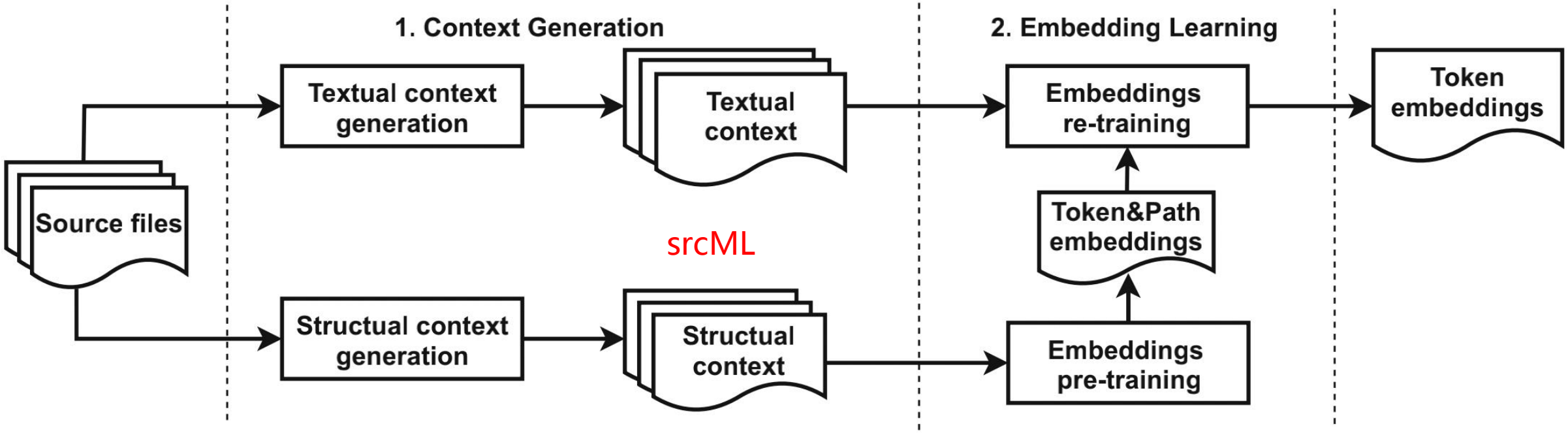


Fig. 4 The overall framework of StrucTexVec

(1) AST paths
(2) method calls
(3) variable references

Dataset

Java-Small dataset

This dataset is collected from Java projects hosted on GitHub.

Table 2 Parameter settings for different embedding techniques

	Non-contextual embeddings					Contextual embeddings	
	Word2vec	GloVe	fastText	code2vec	StrucTexVec	CodeBERT	CuBERT
Vocabulary	109,743	192,363	109,743	507,271	192,362	50,265	50,297
Epoch	5	5	5	20	10 & 5	–	2
Window	5	5	5	5	5	–	–
Negative	5	–	5	–	4 & 5	–	–
Dimension	128	128	128	128	128	768	1024

https://s3.amazonaws.com/code2vec/data/java-small data.tar.gz

RQ1: How Effective Are Pre-trained Embeddings in Improving the Performanceof Downstream SE Tasks

Models using pre-trained embeddings can perform better than models without pre-trained embeddings.

For a specific downstream task, different embedding methods can result in diverse performance, and there does not exist an embedding technique that outperforms others in all nor even majority of the tasks

Using pre-trained embeddings may not always improve the performance of down stream tasks significantly

Evaluation

Table 3 Evaluation results on the test set of six downstream tasks. The second last row shows the percentage of the best result produced by each approach on 22 datasets and the last row is the weighted averaged percentage of best results on six downstream tasks (i.e., each task's contribution to the percentage is weighted by its number of datasets)

Downstream tasks	Evaluation metrics	Dataset	None	Non-contextual embeddings					Contextual embeddings	
				Word2vec	GloVe	fastText	code2vec	StrucTexVec	CodeBERT	CuBERT
Code comment generation	BLEU	GitHub	14.9	15.4	15.9	14.6	15.3	16.0	**16.7**	16.1
Code authorship identification	Accuracy	Google Code Jam	87.5	80.2	87.5	77.1	85.4	86.5	87.0	**89.1**
Code clone detection	F1	BCB	92.7	**93.8**	**93.8**	**93.8**	93.5	93.6	93.5	93.6
		OJClone	85.1	86.8	81.4	78.0	**89.7**	88.1	85.9	81.6
		Avg.	88.9	90.3	87.6	85.9	**91.6**	90.9	89.7	87.6
Source code classification	Accuracy	OJ dataset	88.5	87.0	89.2	77.7	**91.2**	89.1	79.8	75.8
Logging statement prediction	Balance Accuracy	Airavata	**95.6**	94.3	94.2	93.1	94.8	94.5	**93.8**	93.4
		Camel	76.6	77.8	77.5	76.4	77.4	**79.2**	77.1	75.0
		CloudStack	85.9	86.0	85.5	84.7	86.9	**87.3**	86.0	86.7
		Directory-Server	82.9	84.1	85.6	84.7	84.0	86.6	**88.0**	81.9
		Hadoop	76.7	73.6	71.5	71.7	72.3	71.0	75.4	**77.6**
		Avg.	83.6	83.2	82.8	82.1	83.1	83.7	**84.1**	82.9
Software defect prediction	F1	Ant 1.5–>1.6	28.0	35.5	36.0	32.9	47.6	34.2	36.4	**54.8**
		Ant 1.6–>1.7	33.1	44.9	45.1	39.6	48.4	43.4	51.9	**52.9**
		Camel 1.2–>1.4	23.3	43.3	45.5	43.8	43.2	**46.8**	45.6	44.2
		Camel 1.4–>1.6	26.3	47.0	49.8	46.0	50.0	50.2	**51.2**	50.3
		jEdit 3.2–>4.0	32.7	52.0	56.2	55.9	56.6	59.5	**61.5**	59.4
		jEdit 4.0–>4.1	40.6	60.5	60.1	59.7	57.9	**64.7**	62.4	59.9

RQ2: How do the Structural and the Local Textual Information Affect the Performance of the Pre-trained Embeddings?

The structural information extracted from the source code can improve the performance of the code embeddings.

Code embeddings can benefit from the local textual context.

However, the benefit from the local textual context is limited for some downstream tasks.

The structural information has a stronger impact on the quality of the code embeddings than that of local textual information

Table 4 Evaluation results of embeddings trained with and without the structural and local textual contexts. Word2vec is equivalent to the variant of StrucTexVec which removes the structural information from the training process; StrucTexVec^{-text} only utilizes the structural information for embeddings learning

Downstream tasks	Code comment generation	Code authorship identification	Code clone detection			Source code classification	Logging statement prediction						
Datasets	GitHub	Google Code Jam	BCB	OJClone	**Avg.**	OJ dataset	Airavata	Camel	CloudStack	Directory-Server	Hadoop	**Avg.**	
StrucTexVec	**16.0**	**86.5**	93.6	**88.1**	**90.9**	89.1	94.5	**79.2**	**87.3**	86.6	71.0	**83.7**	
Word2vec	15.4	80.2	**93.8**	86.8	90.3	87.0	94.3	77.8	86.0	84.1	**73.6**	83.2	
StrucTexVec^{-text}	15.7	79.7	93.6	86.9	90.3	**89.7**	**95.3**	76.0	85.2	**90.0**	71.2	83.5	
Downstream tasks	Software defect prediction												
Datasets	Ant 1.5 ->1.6	Ant 1.6 ->1.7	Camel 1.2 ->1.4	Camel 1.4 ->1.6	jEdit 3.2 ->4.0	jEdit 4.0 ->4.1	Log4j 1.0 ->1.1	Lucene 2.0 ->2.2	Lucene 2.2 ->2.4	POI 1.5 ->2.5	POI 2.5 ->3.0	Xalan 2.4 ->2.5	**Avg.**
StrucTexVec	34.2	43.4	**46.8**	**50.2**	**59.5**	**64.7**	62.7	63.9	65.2	77.8	71.4	**47.5**	**57.3**
Word2vec	35.5	44.9	43.3	47.0	52.0	60.5	**65.7**	62.6	**66.3**	64.8	**72.4**	41.7	54.7
StrucTexVec^{-text}	**38.1**	**50.2**	46.4	45.2	57.4	58.9	62.9	**66.9**	63.1	**81.2**	71.8	43.6	57.1

南京航空航天大学
Nanjing University of Aeronautics and Astronautics

2022

感谢您的欣赏

THNAK YOU

GRADUATION DEFENSE