

**CODE COMBAT****Google**

E-SAN Thailand Coding & AI Academy

โครงการวิจัยโมเดลระบบนิเวศการเรียนรู้ที่บูรณาการ CODING & AI สำหรับเยาวชน

Model of Learning Ecosystem Platform integrate with Coding & AI for Youth

โครงการย่อยที่ 6

การพัฒนาเยาวชนเพื่อเข้าสู่วิชาชีพขั้นสูงด้าน Coding & AI
ร่วมกับ Coding Entrepreneur & Partnership: Personal AI

งานวิจัยที่ นำ ML ทางภาษา Code / Program ที่ใช้งาน ป้อนมาให้เครื่องเรียนรู้ → เครื่องเรียนรู้จะหา ผลลัพธ์

ตัวอย่าง ภาษา Python ภาษา Java และภาษา C++ ทางภาษาจะสามารถรับ input ของ code / program ที่ ผู้ใช้เข้าไป
โดยตรงแล้ว ทางเครื่องจะหาผลลัพธ์ตามที่ผู้ใช้ต้องการ ให้ได้มากที่สุด เช่นเดียวกัน

ข้อทั้งหมด Code Clone Detector - ระบบ AI สำหรับช่วยตรวจวัดความเหมือนของโค้ดและ

ตรวจจับการคัดลอกโค้ด

โค้ดภาษาต่างๆ ไม่ว่าจะเป็น Software → ฟังก์ชันต่างๆ ของ software ต้องใช้ภาษาเดียวกัน code / program ซึ่งต้องมีความคล้ายคลึงกันมาก มากถึง 90% ขึ้นไป

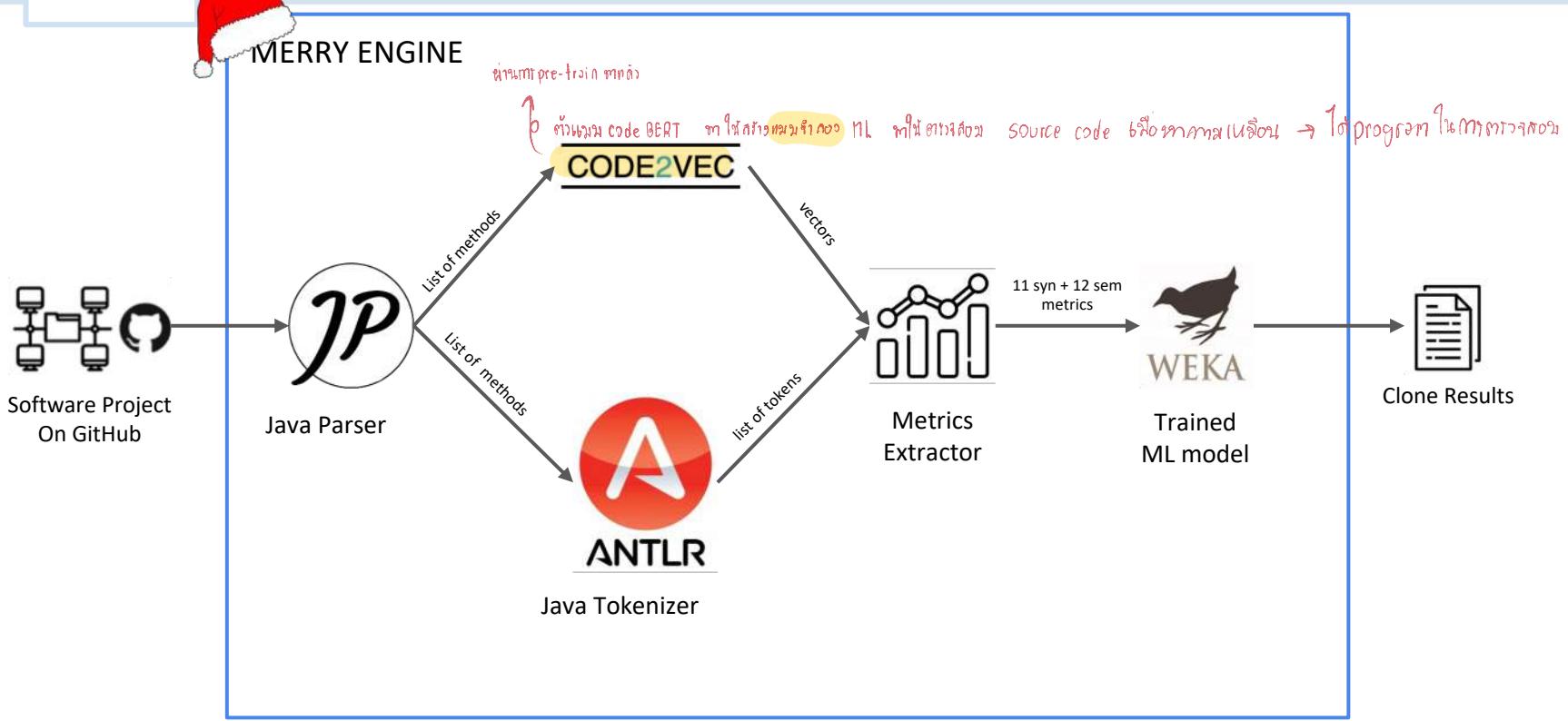
เช่น ก้า source code ภาษาต่างหาก ไม่ว่าจะเป็นภาษา C/C++ / Java / Python / JavaScript ฯลฯ ก็สามารถตรวจจับได้

แหล่งที่มา: [Plagiarism](#) → ลอก窃用 → Plagiarism

ดร.ชัยยงค์ รักขิตเวชสกุล

2

Using Merry Engine for Clone Detection





CODE COMBAT

Google



Let's Code
Thailand

DMAP
DIGITAL MAKERS
ASIA PACIFIC



E-SAN Thailand
Coding & AI Academy

Outline



การพัฒนาเยาวชนเพื่อเข้าสู่วิชาชีพชั้นสูงด้าน Coding & AI ร่วมกับ Coding Entrepreneur & Partnership:

Personal AI

โครงการวิจัยโมเดลระบบนิเวศการเรียนรู้ที่บูรณาการ CODING & AI สำหรับเยาวชน
Model of Learning Ecosystem Platform integrate with Coding & AI for Youth

1. โค้ดที่เหมือนกันคืออะไร? (What are code clones?) Source ก็
โค้ดต่างๆ ดี๊ดี

2. วิธีการสร้าง Machine Learning Model เพื่อตรวจจับโค้ดที่เหมือนกัน

3. วิธีการนำ Machine Learning Model ที่สร้างแล้ว มาตรวจจับโค้ดที่เหมือนกัน

4. การวัดประสิทธิภาพของ Machine Learning Model ที่สร้างขึ้น ในเชิงความแม่นยำ

5. การวัดประสิทธิภาพของ Machine Learning Model ที่สร้างขึ้น ในเชิงเครื่องมือ

6. บทสรุปและแนวทางการพัฒนาต่อในอนาคต

What are Code Clones ? code ที่เหมือนกัน คือ ?

ก็ที่มา source code 2 อย่างเดียวกัน เปลี่ยนภาษา / นิยามที่เก้าอี้

“ Two code fragments form a clone pair
if they are similar enough
according to a given definition of similarity. [1]”

ex. 2 code ล้วนๆ ไม่ต้องมีรูปแบบเดียวกัน ก็ถือ code เหมือนกัน 100%.

```
public int sum(int a, int b){  
    int sum;  
    sum = a + b;  
    return sum;  
}
```

```
public int sum(int num1, int num2){  
    int result;  
    result = num1 + num2;  
    return result;  
}
```

Clone Types: Syntactic Based [2]

คลาสร่องรอยภาษาจัดอันดับ

ว่า

Type 1

```
public int sum(int a, int b){  
    int sum;  
    sum = a + b;  
    return sum;  
}
```

```
public int sum(int a, int b){  
    int sum;  
    sum = a + b;  
    return sum;  
}
```

Identical code fragments
except for layout, white space, and
comments

Code คลาสเดียวกัน แต่ 100% ไม่เท่ากันใน layout, space, comment

Type 2

```
public int sum(int a, int b){  
    int sum;  
    sum = a + b;  
    return sum;  
}
```

```
public int sum(int num1, int num2){  
    int result;  
    result = num1 + num2;  
    return result;  
}
```

Identical code fragments
except for literals, identifiers, data types,
layout, white space, and comments.

Code คลาสเดียวกัน แต่ 100% ไม่เท่ากันใน identifier, data type, layout, space, comments

มาก

Type 3

```
public int sum(int a, int b){  
    int sum;  
    sum = a + b;  
    return sum;  
}
```

```
public int sum(int a, int b){  
    return a + b;  
}
```

Similar clone fragments
with added, changed, and removed
some statements.

อย่างเช่น ถ้ามี code 1 และ code 2 ให้เราลองนำ code 1 มาเปลี่ยน code 2 ให้เป็น

หากเราไม่ได้ตั้งใจ บางทีมันจะ

Clone Types: Functionality Based [2]

ນີ້ກ່າວເຖິງ \rightarrow ນີ້ແລ້ວໄດ້ກຳນົດ
def ດັວຍ extension ໂດຍກ່ອນ return ດັວຍອາຫາວຸນ filename

```
private static String getFormatByName(String name){  
    if(name != null){  
        final int j = name.lastIndexOf(".") + 1,  
        k = name.lastIndexOf("/") + 1;  
        if(j > k &&  
            j < name.length()){  
            return name.substring(j);  
        }  
    }  
    return null  
}
```

TYPE 4 ລາກສູດ

```
public static String getExtension(final String filename){  
    if(filename == null ||  
        filename.trim().length == 0 ||  
        !filename.contains(".")){  
        return null;  
    }  
    int pos = filename.lastIndexOf(".");  
    return filename.substring(pos+1);  
}
```

Code ແມ່ນສືບຕະຫຼາດ ແລ້ວ ຖໍ່ມີກຳນົດກຳນົດ ທີ່ໄດ້ຮັບກຳນົດກຳນົດ

Code fragments that have the same computation
but different in syntax or algorithm.

Why Code Clones need to be detect ?

ក្រុមពេទ្យ

- Source code plagiarism detection [3] ຜົນເປົ້າຂະໜາດ
 - Code Clones can be harmful in software maintenance [1]
ກ່ອນ source ຕາງໆ ISO : 079/2
ມາໃນ ກາດແລະ ນິນ ສະບົບຕາມ
ex. ດີວັນຈີນ. → ຂາຍຕົວຕະຫຼາດ
 - To improve code quality by reducing code redundancy
copy code (usually occur 7-23% in software) [2]
 - Some clones can be beneficial ລູ່ທ. → ເຄື່ອງກຳໄດ້ຕົກ
(e.g. software product line, well-written code) [4]
 - We need to detect clones so that we can manage them [5]

ຕົ້ນງາມ ຊາ ເຊີ່ມກໍ່ໃຈ ໄສ ອຸງລັບໄວ ໂພນ ຂວາງເຫັນ ເຊື່ອກໍ່ເກົ່າໃຈ ມະນີກາເສດ ດາວີຕ່ອນຈາກ ນ້າກາະ ສອງ

[1] T. Kamiya, S. Kusumoto and K. Inoue, "CCFinder: a multilingual token-based code clone detection system for large scale source code," July 2002.

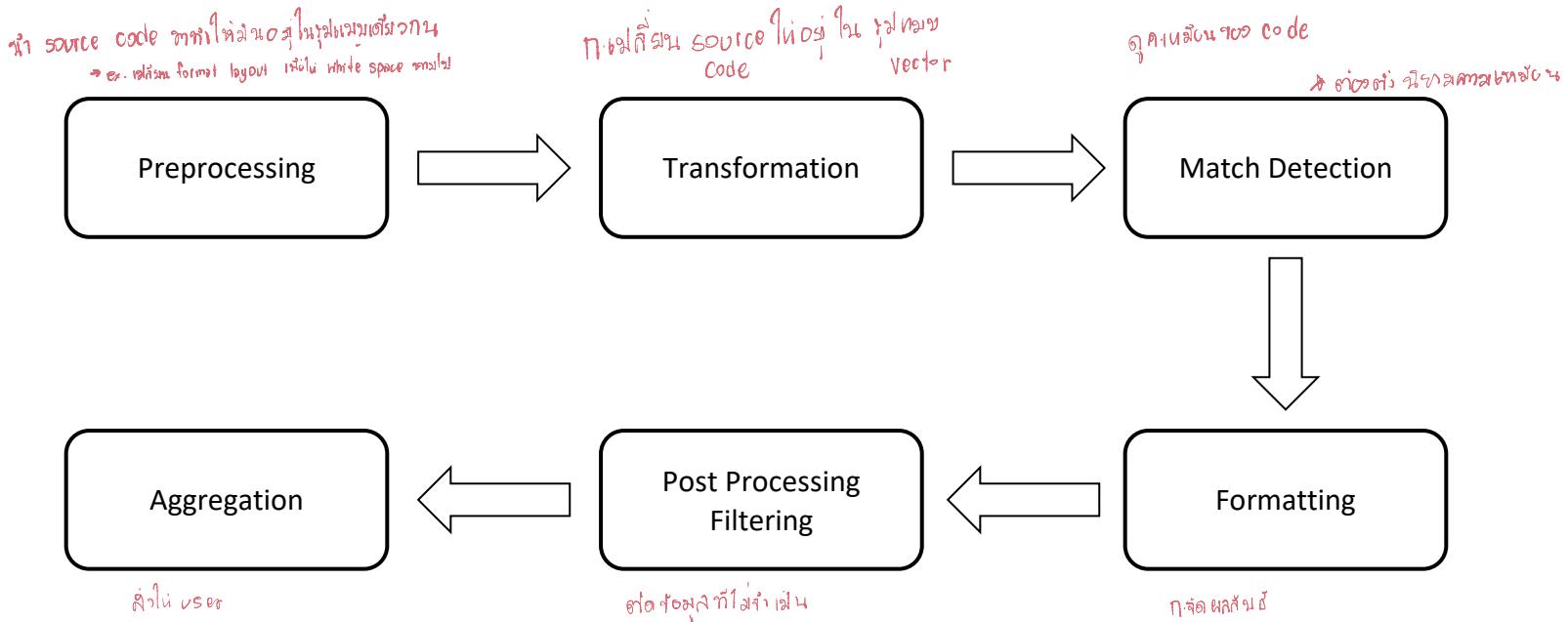
[2] Roy et al. Comparison and evaluation of code clone detection techniques and tools: A qualitative approach, Science of Computer Programming (2009)

[3] Prechelt, L., Malpohl, G., & Philippsen, M. (2002). Finding Plagiarisms among a Set of Programs with JPlag. *Journal Of Universal Computer Science*, 8(11), 1016–1038.

[4] Kapser, C. J., & Godfrey, M. W. (2008). Cloning considered harmful considered harmful: patterns of cloning in software. *Empirical Software Engineering*, 13(6), 645–692.

[5] D. Chatterji, J. C. Carver, and N. A. Kraft. Code clones and developer behavior: results of two surveys of the clone research community. *Empirical Software Engineering*, 21(4):1476–1508, Aug 2016.

Code Clone Detection Process



Problem Statements

មុននេះ tool ទាំងអស់យើង ត្រូវបានរកចំណែកលើ source code ក៏ដែលពេញលេញ គឺជាបញ្ហាដែលត្រូវបានដោះស្រាយ

- The existing techniques and tools are still facing challenges when detecting clones with several modifications (e.g., added/deleted/modified statements).
- Existing clone detection and plagiarism detection tools are difficult to use because it is command line based tool.



Objectives

ກົດໆ

1. To create a code clone detection tool using machine learning techniques and study its effectiveness. ຮຶກສິນເປົ້າຫວັງ

2. To enhance the user experience of code clone detection tools

- providing code clone detection as a web application to users
- providing visualization of clone results

Merry: Web-Based Code Clone Detection System Using Machine Learning

Following source code
Features

11 Syntactic
Code Metrics
+
12 Semantic
Code Metrics
(code2vec) model

Machine Learning

Decision tree, Random
forest, SVM,
SVM + SMO

data set

BigCloneBench

Largest
and
Credible clone data

Web-based Tool
with
User Interface

GitHub Integration



Visualizations and
reports

วิธีการสร้าง
Machine Learning Model เพื่อ^{เพื่อ}
ตรวจจับโค้ดที่เหมือนกัน

Merry Engine Implementation

1

Building Merry Engine → ສໍາງ code clone detection model

2

Using Merry Engine for Clone Detection ດູໃຈນຳຕາຫວັນ

1

Building Merry Engine

1.1

Data collection and preparation

សរុបទូទៅពេលវេលា

1.2

ការបញ្ចូនតម្លៃកិច្ច

Syntactic
Semantic

1.3

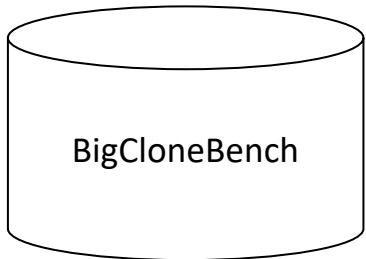
ន.កិច្ចអនុវត្ត ML model

Machine learning models

1.1

Data collection and preparation

BigCloneBench [7] => ចំណេកកំពូល

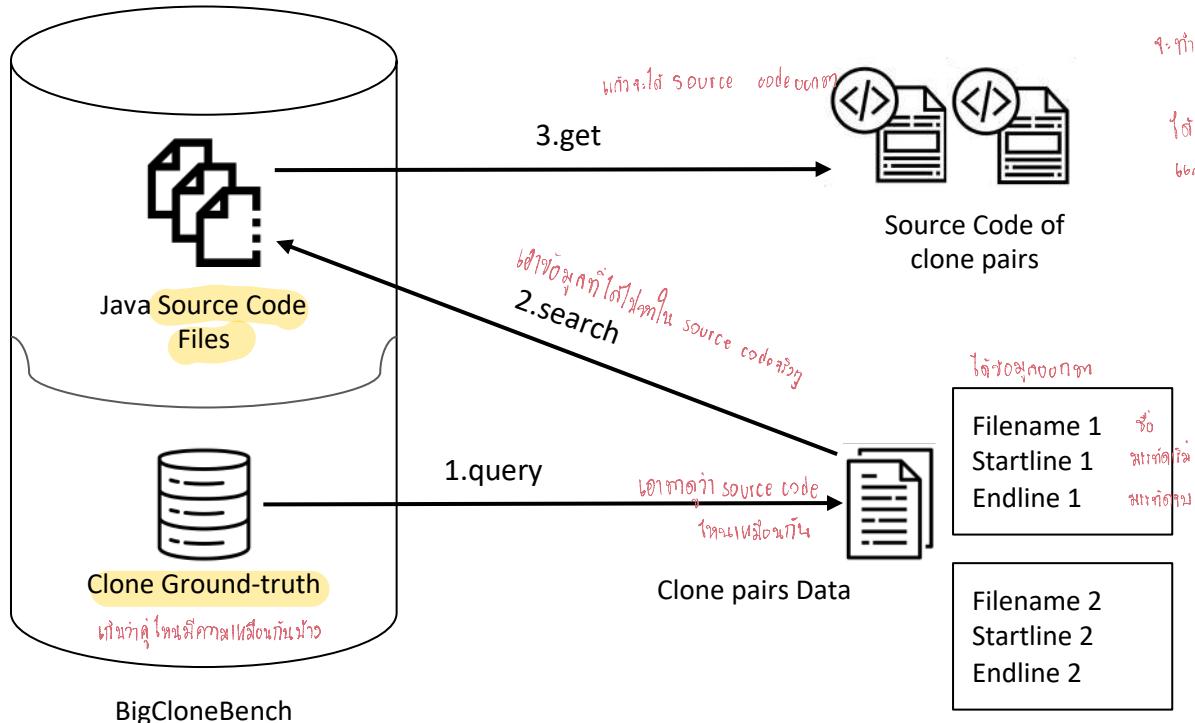


- Largest and reliable clone data
- Well-known in code clone detection field គឺជាជំនួយដែលមានការអនុវត្តន៍យ៉ាងខ្ពស់នៅក្នុងការអភិវឌ្ឍន៍បញ្ហាសម្រាប់កូដ
- Big data of inter-project repository IJaDataset 2.0
- Created from 25,000 Java projects ត្រូវបានរកឃើញពីកូដ Java ទូទៅ
- Contains tagged true and false clone pairs មានការបោះពីរថ្មី ដើម្បីពិនិត្យថាគូដណានៗនេះជូនិតិយកូដឬមិនជាផ្លូវការ។

1.1

How do we get clone pairs from BCB?

ກ. လົງຈຳຄຸນຈາກ BCB



ກ. ຖົມນີ້ກັ່ງ ກໍໄບສິບນີ້ ກ. ອົງເລື່ອນກັນ



ເສີ່ມວິທີກົດຕະຫຼອງ source code ອົງ
ໂດຍແລ້ວອື່ນກັນ
ໄດ້ຮັບອຸປະກອດ

1.1

Training Set and Testing Set Splitting

Training Data [8]

True Clone Pairs (22,663 pairs) ແກ້ໄຂຕົວເລີມທີ່ຈະ				ແຕ່ງ False Clone Pairs
Type 1	Type 2	Very Strong Type 3	Strongly Type 3	
13,750	3,104	1,207	4,602	22,663

Testing Data

True Clone Pairs (4,724 pairs - stratified sampling)				False Clone Pairs
Type 1	Type 2	Very Strong Type 3	Strongly Type 3	
2,383	557	307	1,477	18,893

1.2

Code Metrics Extraction

ទូរការ Syntax → លោកឡើងការកិច្ចការ code / វិធានការអនុវត្តន៍យ៉ាង code

Syntactic metrics

លោកឡើង code គារបែងចាយ និងការសម្រេច
=

Metrics extracted from syntactic characteristics of the source code fragments

ទូរការអនុវត្តន៍ , func, Method និងឈាម និងការណា !!

Semantic metrics

=

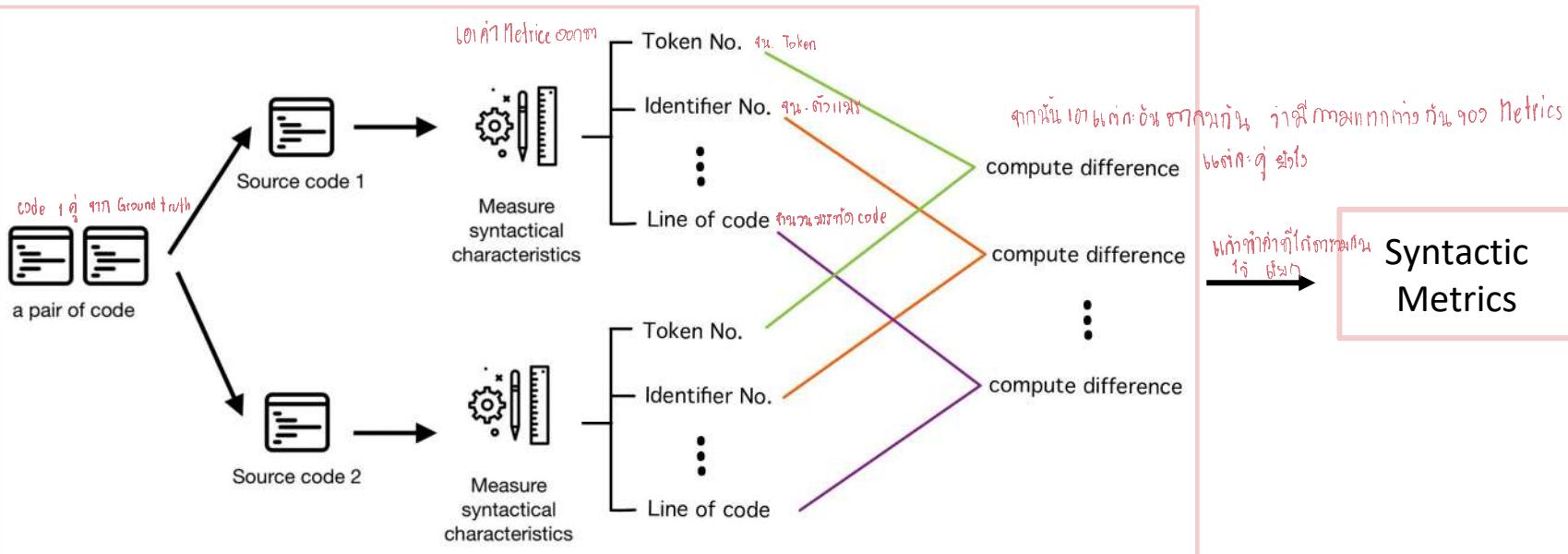
Metrics that capture the behaviour of the source code fragments



1.2

Syntactic Code Metrics

AI Metrics កំពើទីនាក់ក្នុងការបង្កើតរឹង នៃ source code នៅក្បាល់រួម
អាជីវកម្មនៃការងារជូន ចាប់ផ្តើមជា vector ចាប់អាពិល



1.2

Syntactic Code Metrics

No.	Metric	Description
1	Token No [9]	Difference of number of tokens กົດເລກອຳນວຍໃນ source code
2	Unique Token No [10]	Difference of number of unique tokens
3	Identifier No [10]	Difference of number of Identifiers ຈຸນສ້າງໄວ້
4	Unique Identifier No [10]	Difference of number of unique Identifiers
5	Operator No [10]	Difference of number of operators + - x ÷
6	Unique Operator No [10]	Difference of number of unique operators
7	Token Types Diversity [9]	Difference of number of unique token types ດັບເລກອຳນວຍຈົດ data type ex. ຂະໜິຕ່ຕັ້ງຂູ້ ທີ່ມີຄຳຄົວກຳ
8	Diff File Name Score	File names difference score ຊົວໄຟລ໌
9	Diff Method Name Score	Method names difference score ຊົວວິທີ
10	Similar Return Type	Same return type or not ຂອງໃຈ ຖໍ່ ອອກປົວ
11	DiffLOC	Difference of lines of code

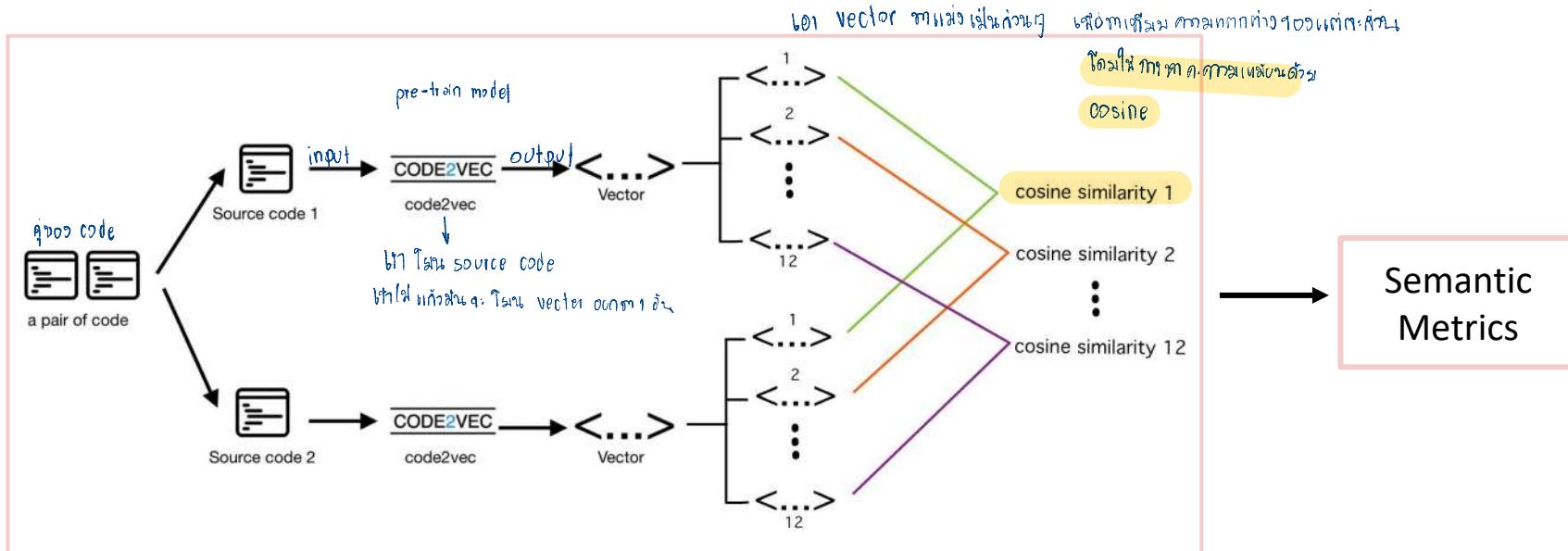
[9] Koschke R., Bazrafshan S., "Software-clonerasinopen-sourceprogramswritten in C or C++", In: 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER). vol. 3. IEEE; 2016. p. 1–7.

[10] Vara A., Rainer K., Chaiyong R., Morakot C., Thanwadee S., "Improving CloneDetection Precision using Machine Learning Techniques", In: Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. IWESEP; 2019.

1.2

Semantic Code Metrics

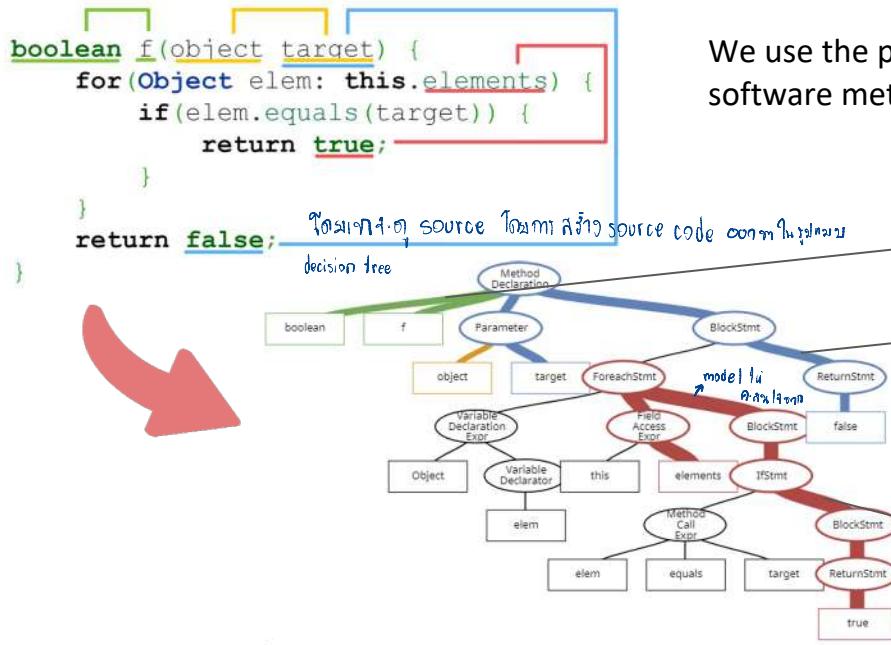
ក. ការបង្កើតមេរី កំណត់ការងារ



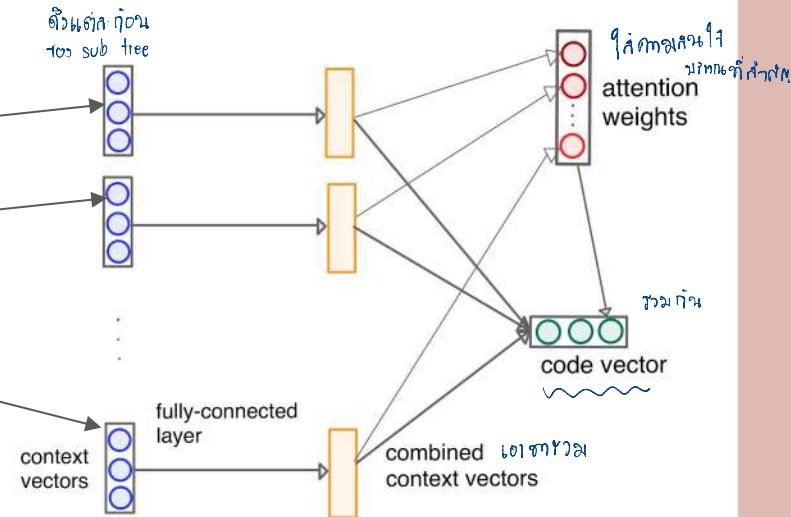
code2vec in Detail

ເນື່ອໃນ neural network ທີ່ກໍາໄທໃຫ້ code snippet ເຊີ່ມີຂະໜາດໃຈ່ນ vector ກໍ່ສິ້ງຄວາມຄວາມຄົງສັ່ງ
ກໍາຕຳກຳການໂລດ source code ຂອງຕາມວິທີ vector → ຕ່າງໆມີການແນ່ໃນກົນ ອະນຸຍາດ Vector + ດັກລົງທຶນ

code2vec is neural model that represent code snippet as fixed-length vectors (size = 384) that capture semantic of the source code [12].



We use the pretrain model that trained from 12 M. of real software methods.



1.2

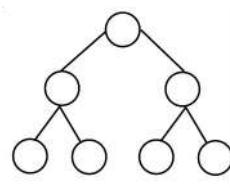
Example of Code Metrics

	source code 1	source code 2	
sum1.java	<pre>public int sum(int a, int b){ int sum; sum = a + b; return sum; }</pre>	<pre>public int sum(int a, int b){ return a + b; }</pre>	
			<u>Semantic metrics</u> 384 → ต้องใช้ 12 ค่า cosine sim
			↓
	<u>Syntactic metrics</u>		
Diff. No. Tokens	:	vector#1 similarity	: 0.5206784273085
Diff. No. Unique Token	:	vector#2 similarity	: 0.457455059252847
Diff. No. Identifier	:	vector#3 similarity	: 0.588965399230431
Diff. No. Unique Identifier	:	vector#4 similarity	: 0.652574760995595
Diff. No. Operator	:	vector#5 similarity	: 0.529570896554088
Diff. No. Unique Operator No	:	vector#6 similarity	: 0.490625929242188
Diff. Token Types Diversity	:	vector#7 similarity	: 0.792832125905209
Diff. File Name Score	:	vector#8 similarity	: 0.562600679903085
Diff. Method Name Score	:	vector#9 similarity	: 0.799728586131479
Similar Return Type	:	vector#10 similarity	: 0.642114908324134
Diff. No. LOC	:	vector#11 similarity	: 0.540922173109001
		vector#12 similarity	: 0.729365945085146

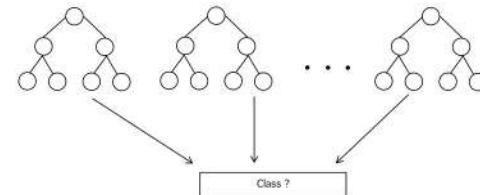
1.3

机器学习

Machine Learning Models



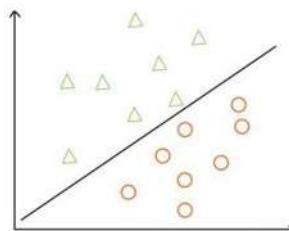
Decision Tree



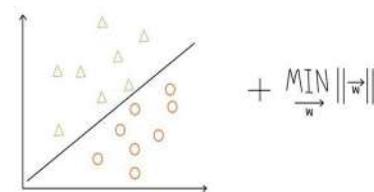
Random Forest



WEKA (library)



Support Vector Machine

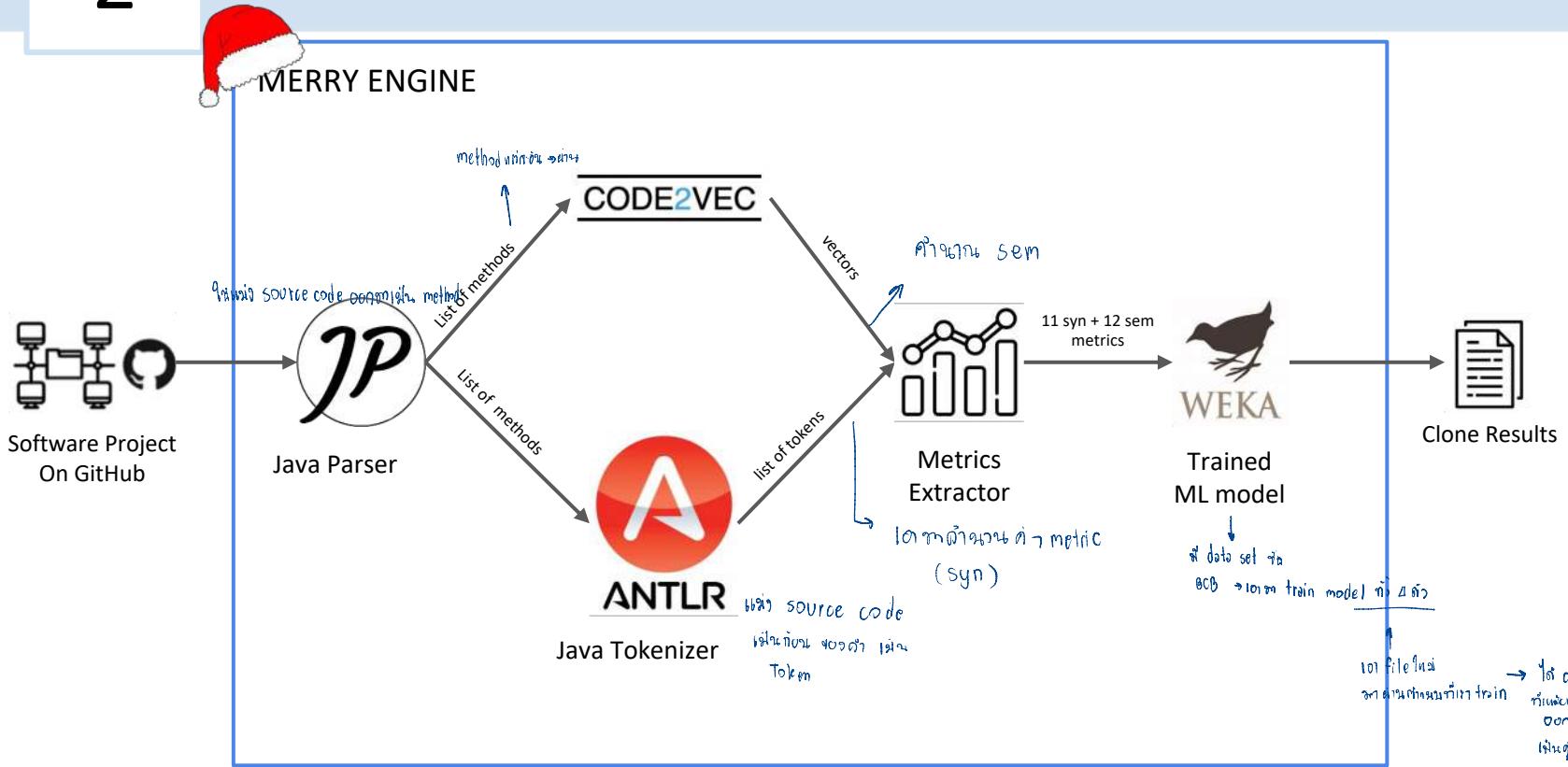


SVM using SMO optimization

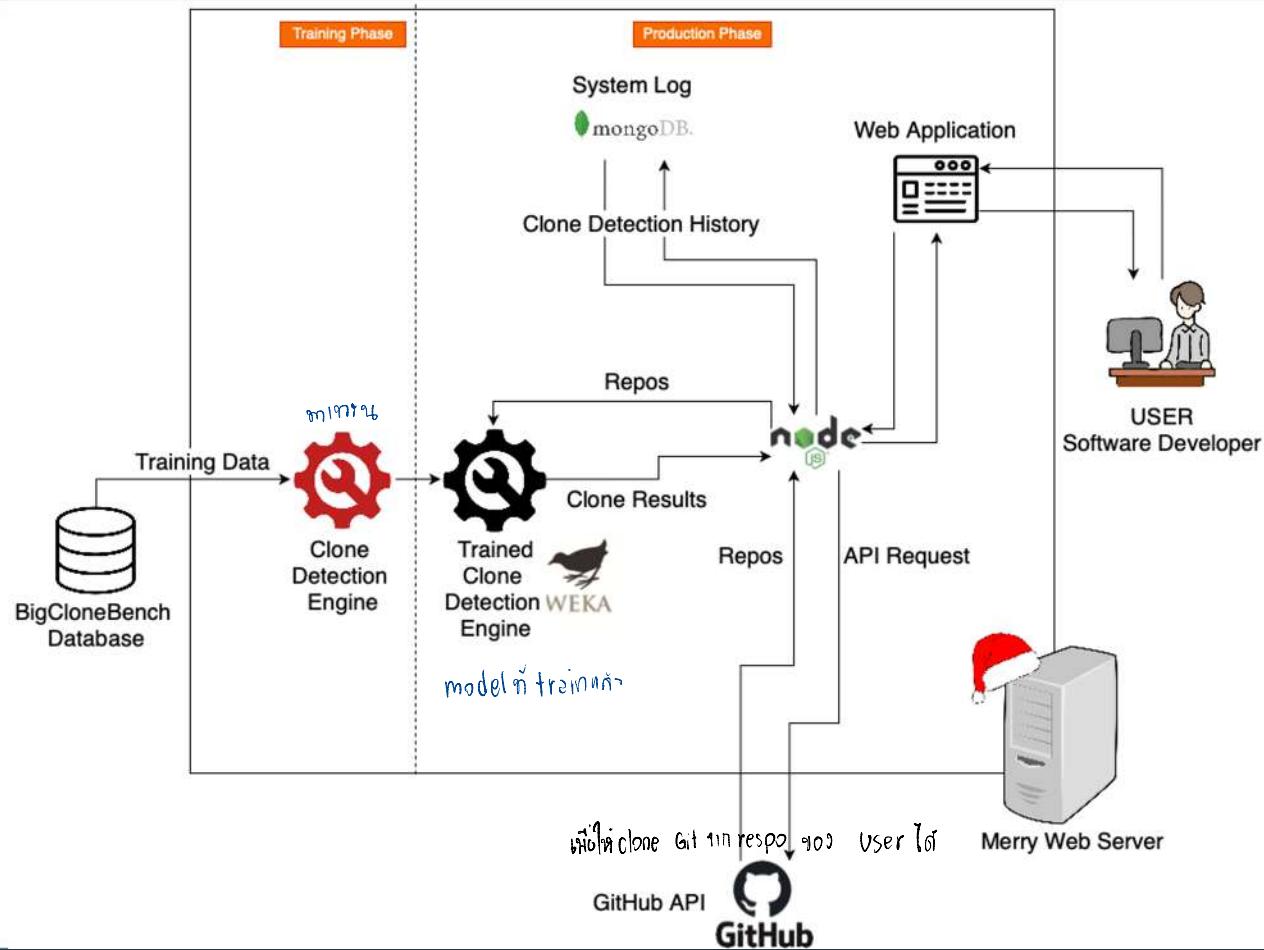
วิธีการนำ Machine Learning Model ที่สร้างแล้ว มาตรวจจับโค้ดที่ เหมือนกัน

2

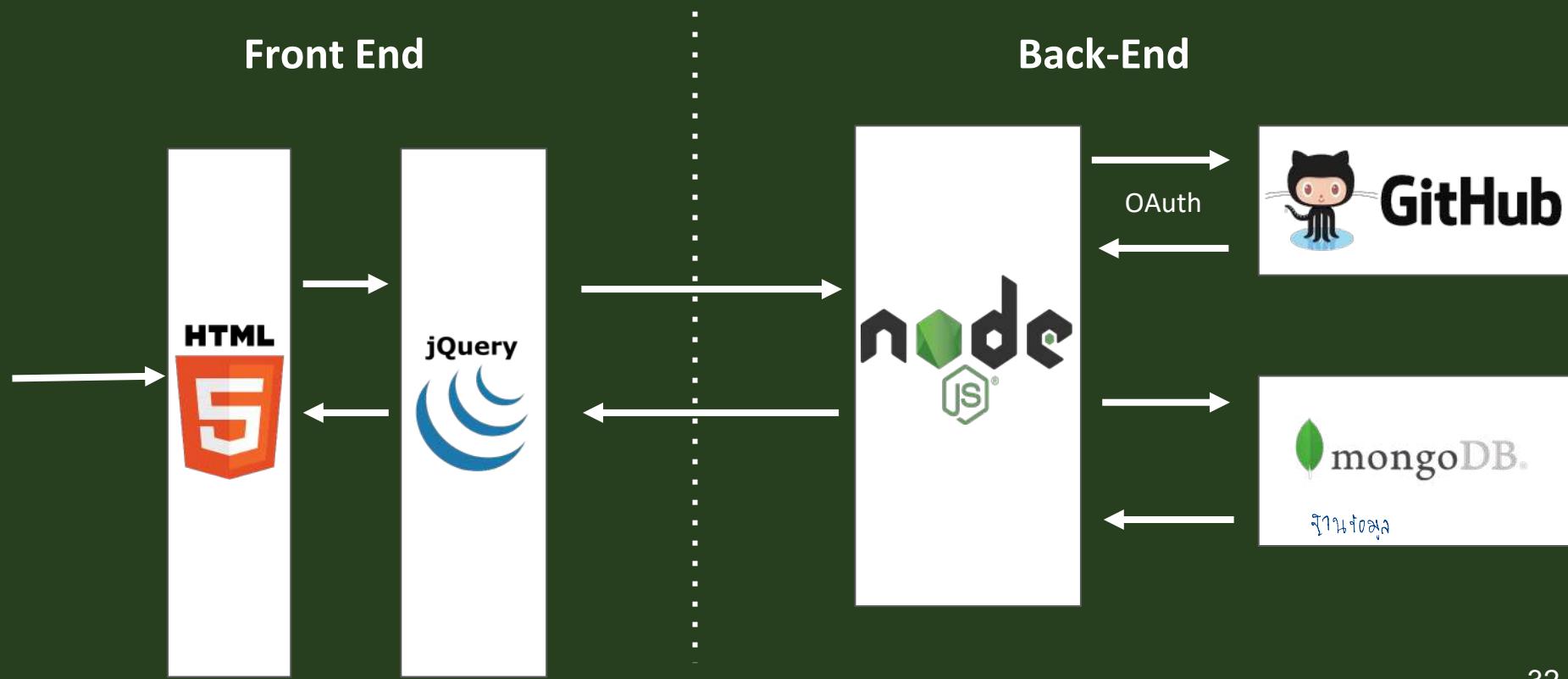
Using Merry Engine for Clone Detection



System Architecture



Frameworks and Tools



GitHub Integration



การวัดประสิทธิภาพของ Machine Learning Model ที่สร้าง ขึ้น ในเชิงความแม่นยำ

Merry : ອັນ source code ກົດສາມາເຫຼືອນຳໃນ (code clone)

Evaluation

ໂຄຣເນັດນັກໃຫ້ Data ອັນ BCB ທີ່ເທົກກວດ ດັ່ງນີ້ກໍ່ເພີ້ມອາຄຸກວ່າມາສີການເຫຼືອນຳໄດ້ໃກ້ຈະ ອັນມີການ code clone

RQ1 How accurate is Merry code clone detection on BigCloneBench?

RQ2 How accurate is Merry code clone detection on real software projects?

RQ3 How likely are command line based tool and our Merry tool adopted by programmers? ອັນຊັງໃຊ້ເປົ້າໃຫຍ່ເຊົາໃຫຍ່

↑ > ამა რანგში არის კოდის კლონების მიზანის 99% ანალიზის

Evaluation of Merry ML Clone Detection Engine using BCB

RQ1: How accurate is Merry code clone detection on BigCloneBench?

Evaluation of Merry ML Clone Detection Engine using BCB

កំណត់ អចការ្យកុកា និងការពិនិត្យ

$$precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

បង្ហាញរបាយខ្លួន និងការសម្រេច វីដែអ៊ូលីមីន

គ្មានទូទៅក្នុង អនុញ្ញាតកុកា និងការពិនិត្យ

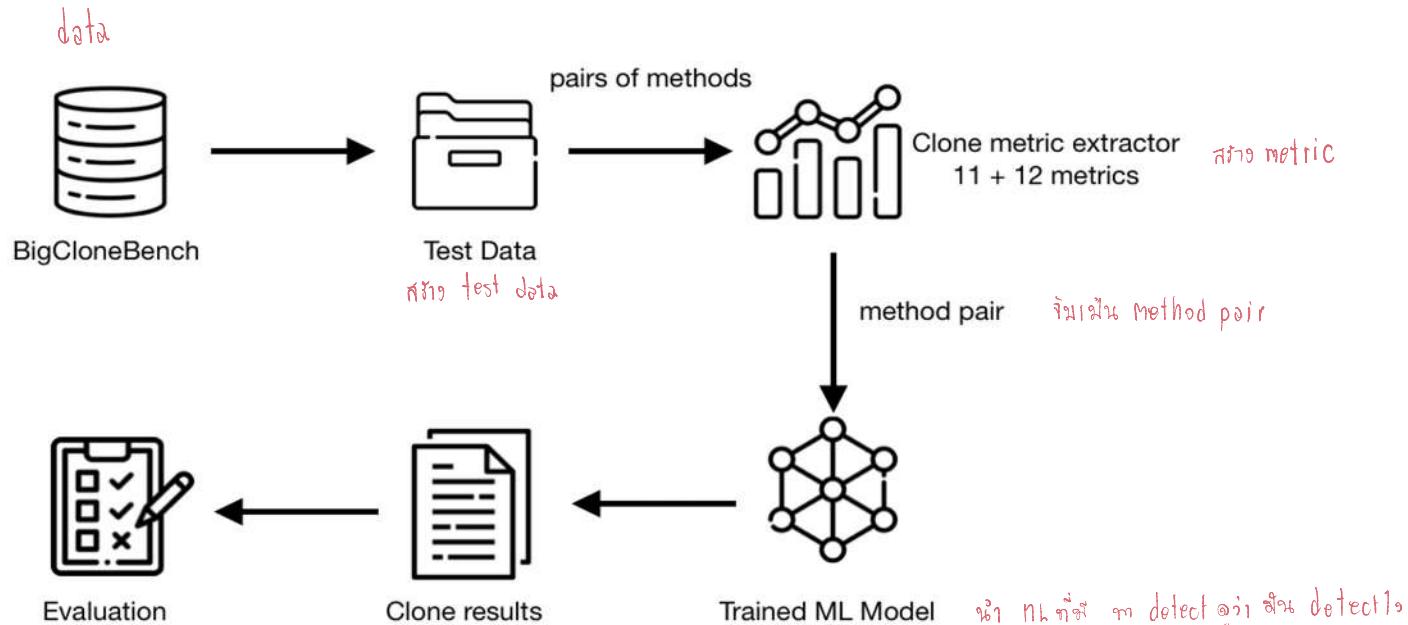
$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

ពិនិត្យ ការសម្រេច និងការពិនិត្យ

អាមេរិក

$$F1 - score = 2 \cdot \frac{Precision - Recall}{Precision + Recall}$$

Evaluation of Merry ML Clone Detection Engine using BCB (cont.)



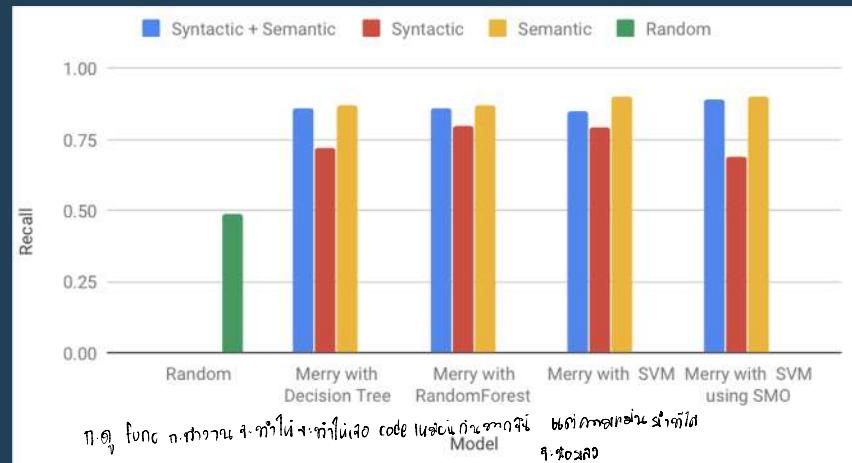
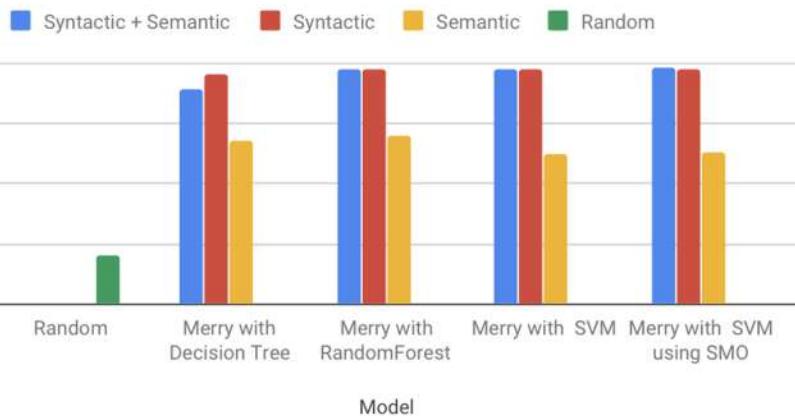
Evaluation of Merry ML Clone Detection Engine using BCB (cont.)

RQ1: How accurate is Merry code clone detection on BigCloneBench?

ពាណិជ្ជកម្ម

Model	Metrics	Precision	Recall	F1-Score
Randomization (baseline)	ព័ត៌មានអង្គភាព និងសម្រាប់បញ្ជាផល ML	0.20	0.49	0.28
Decision Tree	Syntactic + Semantic	0.89	0.86	0.87
	Syntactic	0.95	0.72	0.86
	Semantic	0.68	0.87	0.76
Random Forest	Syntactic + Semantic	0.97	0.86	0.91
	Syntactic	0.97	0.80	0.87
	Semantic	0.70	0.87	0.78
SVM	Syntactic + Semantic	0.97	0.85	0.91
	Syntactic	0.97	0.79	0.87
	Semantic	0.62	0.90	0.73
SVM using SMO	Syntactic + Semantic	0.98	0.89	0.93
	Syntactic	0.97	0.69	0.81
	Semantic	0.63	0.90	0.74

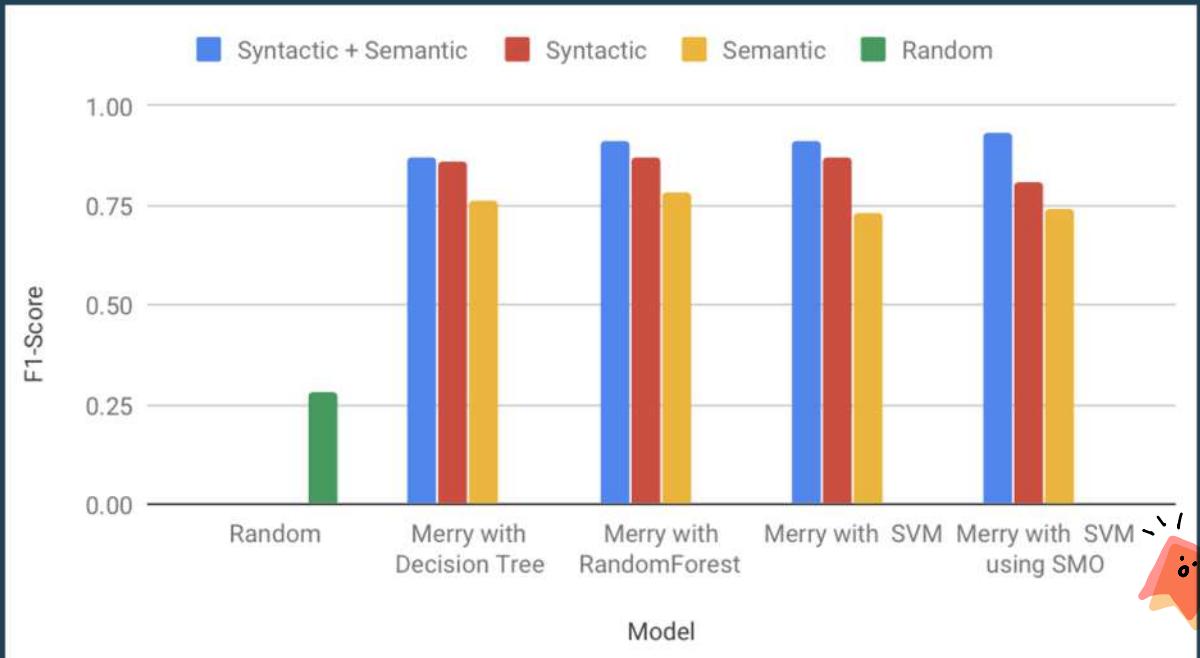
Evaluation of Merry ML Clone Detection Engine using BCB (cont.)



Precision

Recall

Evaluation of Merry ML Clone Detection Engine using BCB (cont.)

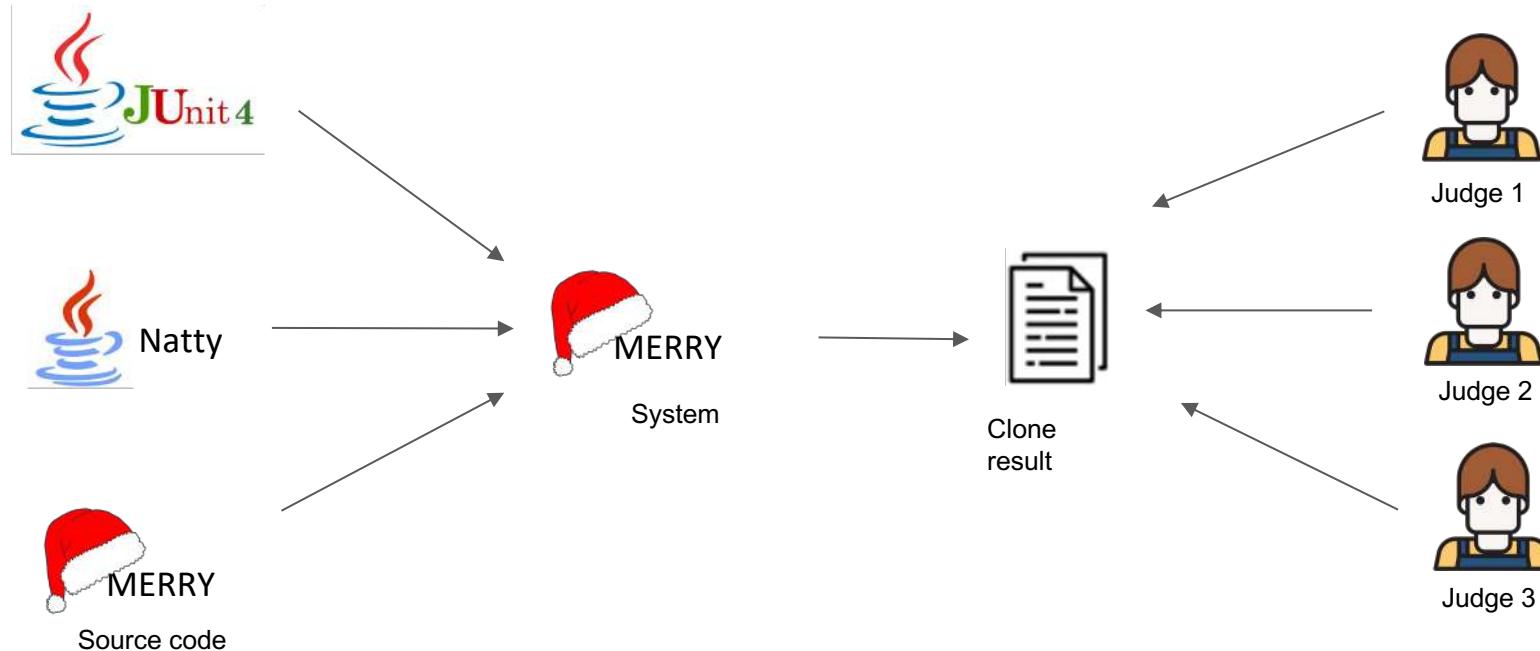


F1-score

Evaluation of Merry Clone Detection Engine on Real Software Projects

RQ2: How accurate is Merry code clone detection on real software projects?

Evaluation of Merry Clone Detection Engine on Real Software Projects



Percent of overall agreement = 69.34%
Fleiss' kappa = 0.39

Evaluation of Merry Clone Detection Engine on Real Software Projects

RQ2: How accurate is Merry code clone detection on real software projects?

line of code

Project	LOC	Methods	Clone pairs	TP	FP	Precision	Runtime (sec)
JUnit4	7550	1529	187	77	110	0.41	552
Natty	1761	146	9	7	2	0.77	99
Merry	931	102	8	6	2	0.75	115

The result of evaluation on real software project

Example of Detected Type-1 Clone Pair

Project : JUnit4

ເຊື້ອງກັນ 100%.

ເຊື້ອງກັນ 100%.

```
1. private void runBefores() throws FailedBefore {
2.     try {
3.         try {
4.             List<Method> beforees =
testMethod.getBefores();
5.             for (Method before : beforees) {
6.                 before.invoke(test);
7.             }
8.         } catch (InvocationTargetException e) {
9.             throw e.getTargetException();
10.        }
11.    } catch (AssumptionViolatedException e) {
12.        throw new FailedBefore();
13.    } catch (Throwable e) {
14.        addFailure(e);
15.        throw new FailedBefore();
16.    }
17. }
```

```
1. private void runBefores() throws FailedBefore {
2.     try {
3.         try {
4.             List<Method> beforees =
testMethod.getBefores();
5.             for (Method before : beforees) {
6.                 before.invoke(test);
7.             }
8.         } catch (InvocationTargetException e) {
9.             throw e.getTargetException();
10.        }
11.    } catch (AssumptionViolatedException e) {
12.        throw new FailedBefore();
13.    } catch (Throwable e) {
14.        addFailure(e);
15.        throw new FailedBefore();
16.    }
17. }
```

Filename: ClassRoadie.java

Startline: 51

Endline: 67

Filename: MethodRoadie.java

Startline: 128

Endline: 144

Example of Detected Type-2 Clone Pair ໂດຍສັນເກືດຕັ້ງໆ

Project : JUnit4

```
1. @Override
2. protected Collection<FrameworkMethod>
   getSingleDataPointMethods (ParameterSignature sig) {
3.   Collection<FrameworkMethod> methods =
   super.getSingleDataPointMethods(sig);
4.   String requestedName =
   sig.getAnnotation(FromDataPoints.class).value();
5.   List<FrameworkMethod> methodsWithMatchingNames = new
   ArrayList<FrameworkMethod> ();
6.   for (FrameworkMethod method : methods) {
7.     String[] methodNames =
   method.getAnnotation(DataPoint.class).value();
8.     if
   (Arrays.asList(methodNames).contains(requestedName)) {
9.       methodsWithMatchingNames.add(method);
10.    }
11.  }
12.  return methodsWithMatchingNames;
13. }
```

```
1. @Override
2. protected Collection<Field>
   getDataPointsFields (ParameterSignature sig) {
3.   Collection<Field> fields =
   super.getDataPointsFields(sig);
4.   String requestedName =
   sig.getAnnotation(FromDataPoints.class).value();
5.   List<Field> fieldsWithMatchingNames = new
   ArrayList<Field> ();
6.   for (Field field : fields) {
7.     String[] fieldNames =
   field.getAnnotation(DataPoints.class).value();
8.     if
   (Arrays.asList(fieldNames).contains(requestedName)) {
9.       fieldsWithMatchingNames.add(field);
10.    }
11.  }
12.  return fieldsWithMatchingNames;
13. }
```

Filename: SpecificDataPointsSupplier.java

Startline: 56

Endline: 71

Filename: SpecificDataPointsSupplier.java

Startline: 39

Endline: 54

Example of True Type 3 Clone Pair ໜີ້ / ນອກຈຸນກາ

Project : JUnit4

```
1. @Override
2. public void evaluate() throws Throwable {
3.     before();
4.     List<Throwable> errors = new
5.         ArrayList<Throwable>();
6.     try {
7.         base.evaluate();
8.     } catch (Throwable t) {
9.         errors.add(t);
10.    } finally {
11.        try {
12.            after();
13.        } catch (Throwable t) {
14.            errors.add(t);
15.        }
16.    }
17.    MultipleFailureException.assertEmpty(errors);
}
```

Filename: ExternalResource.java

Startline: 48

Endline: 65

```
1. @Override
2. public void evaluate() throws Throwable {
3.     List<Throwable> errors = new
4.         ArrayList<Throwable>();
5.     try {
6.         next.evaluate();
7.     } catch (Throwable e) {
8.         errors.add(e);
9.     } finally {
10.        for (FrameworkMethod each : afters) {
11.            try {
12.                invokeMethod(each);
13.            } catch (Throwable e) {
14.                errors.add(e);
15.            }
16.        }
17.    }
18.    MultipleFailureException.assertEmpty(errors);
}
```

Filename: RunAfters.java

Startline: 23

Endline: 40

Example of False Clone Pair

Project : JUnit4

error definition

```
1. protected void runChild(final FrameworkMethod
   method, RunNotifier notifier) {
2.     Description description =
   describeChild(method);
3.     if (isIgnored(method)) {
4.         notifier.fireTestIgnored(description);
5.     } else {
6.         Statement statement = new Statement() {
7.
8.             @Override
9.             public void evaluate() throws Throwable
10.            {
11.                methodBlock(method).evaluate();
12.            }
13.            runLeaf(statement, description, notifier);
14.        }
15.    }
```

Filename: BlockJUnit4ClassRunner.java

Startline: 91

Endline: 105

```
1. public Result run(Runner runner) {
2.     Result result = new Result();
3.     RunListener listener = result.createListener();
4.     notifier.addFirstListener(listener);
5.     try {
6.
7.         notifier.fireTestRunStarted(runner.getDescription()
   );
8.         runner.run(notifier);
9.         notifier.fireTestRunFinished(result);
10.    } finally {
11.        removeListener(listener);
12.    }
13. }
```

Filename: BlockJUnit4ClassRunner.java

Startline: 131

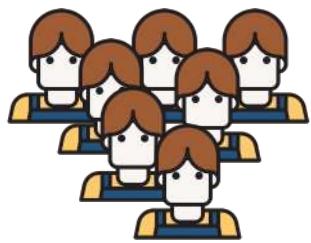
Endline: 143

การวัดประสิทธิภาพของ Machine Learning Model ที่สร้างขึ้น ในเชิงเครื่องมือ

ผู้ให้คะแนน ชอนันดา

Evaluation of Merry Web Application by Users

RQ3: How likely are command line based tool and our Merry tool adopted
by programmers?



Computer Science
students

Target users



Programmers,
Developers

User study methodology

ແຜ່ນກຳລົມ

→ ກອນຫະກໍາຕ້າງໆໃຫຍ່ 2 ເລື່ອງ BiSS

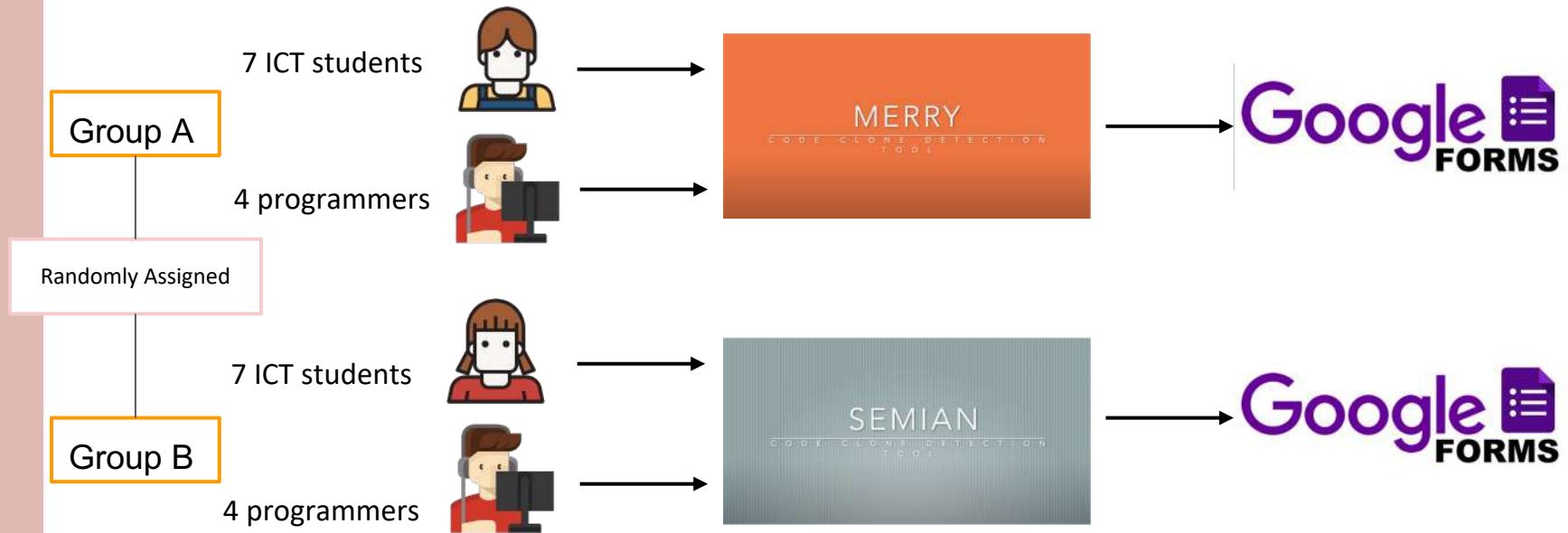
Between-Subjects study design



Reduce the impact of transfer across conditions

Evaluation of Merry Web Application by Users

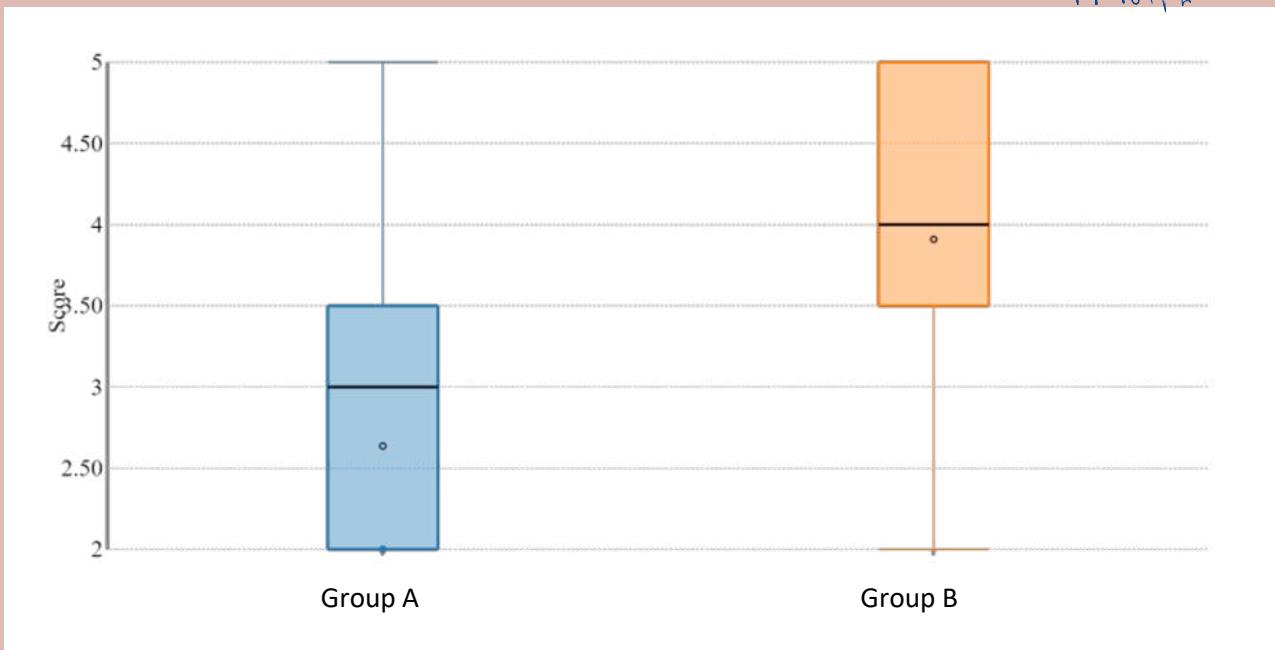
RQ3: How likely are command line based tool and our Merry tool adopted by programmers?



Likeliness of using the tool

If this tool will be applied to one of your software project,
how likely will you use it?

0.96198

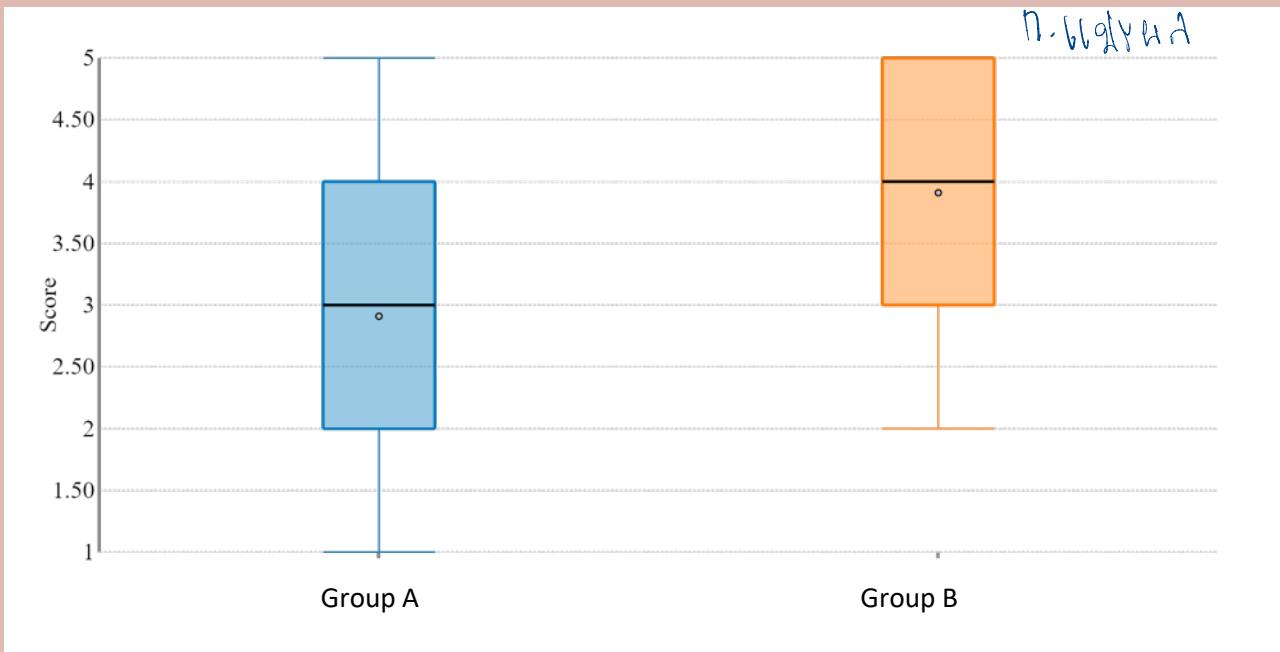


Group A = Simian command line based tool

Group B = Merry web based clone detection tool

Ease of Understanding

Did you find the tool easy to understand
(how to run, how to interpret results)?

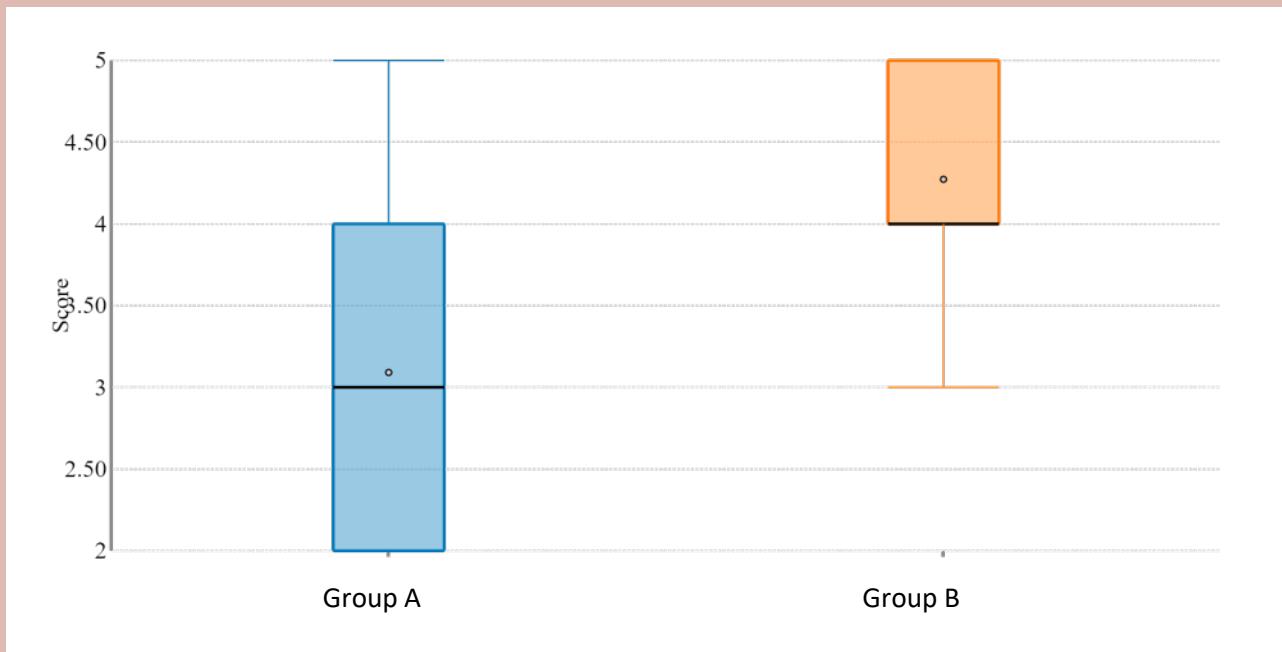


Group A = Simian command line based tool

Group B = Merry web based clone detection tool

Ease of Use

Did you find the tool easy to use
(environment needed to run to the tool and see the results)?



Group A = Simian command line based tool

Group B = Merry web based clone detection tool

บทสรุปและแนวทางการพัฒนา ต่อในอนาคต

រាយការកម្មវិធី, លេខា git10

នៅទីនេះ ដឹងពីរបៀប

Conclusion

Conclusion

- We create a web-based code clone detection tool that use machine learning techniques, called Merry.
- The aim is to accurately detect clones and improve user experience.

Merry Engine

- The clone detection ML engine is based on 4 machine learning models.
- Evaluation of the performance of Merry by using BigCloneBench shows high precision and recall.
- Evaluation on the real software projects has varied performance.

Merry Web Application

- The web application is integrated with GitHub.
- It offers a user friendly and convenient code clone detection function.

Problems and Limitations

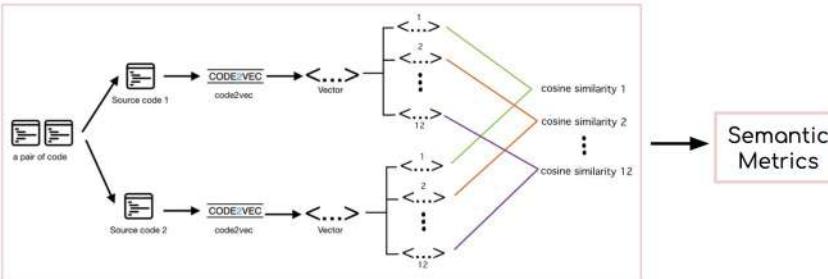
- Supports only Java language
- code2vec poor run-time performance
- The evaluated precision and recall might not reflect on real software projects.
- Current MongoDB configuration has an issue where there is large amount.
- The user study results may be biased due to the sample size is small.

Future Work

- Expand the tool to detect clones in other languages ยกระดับเครื่องมือให้สามารถตรวจจับคลอนในภาษาอื่นๆ ได้
- Improve the code2vec run-time / ปรับปรุงเวลาการทำงานของ code2vec
- Create dedicated machine learning model per clone type
- Solve the MongoDB limitation by query a part of MongoDB document at a time
- Expand the number of participants in the user study.

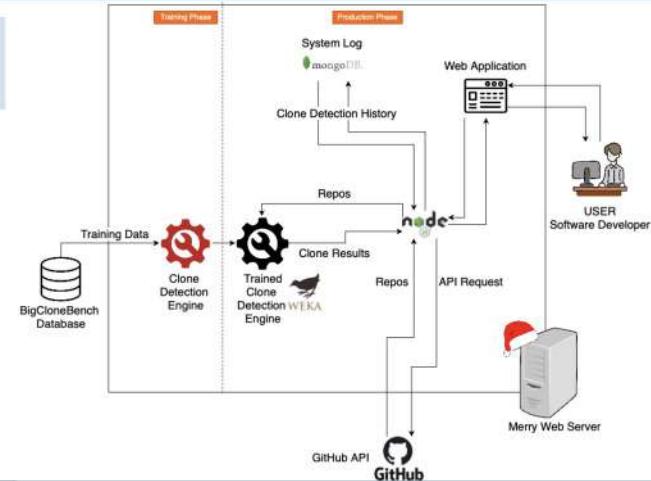
1.2

Semantic Code Metrics



20

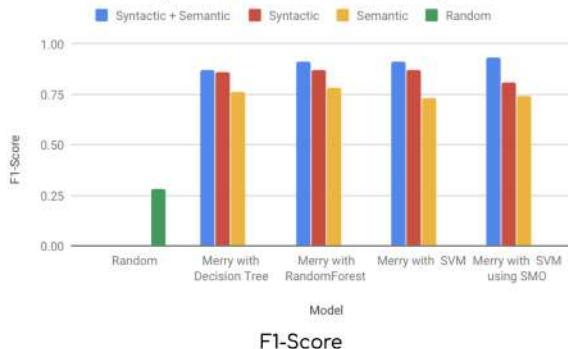
System Architecture



27

Evaluation of Merry ML Clone Detection Engine using BCB (cont.)

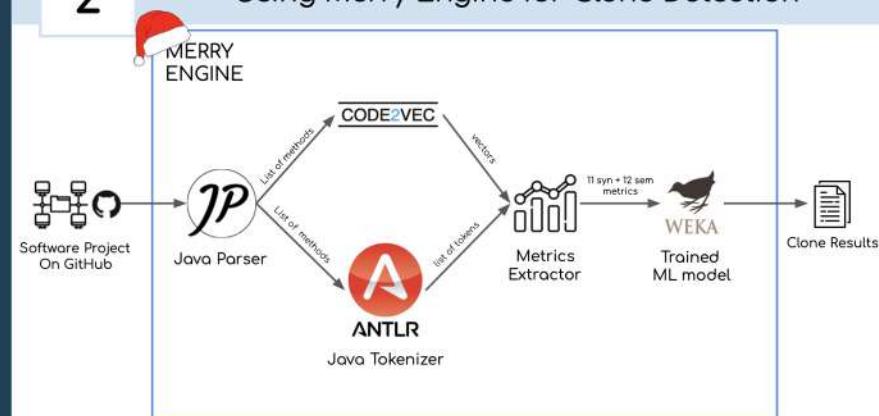
RQ1: How accurate is Merry code clone detection on BigCloneBench?



37

2

Using Merry Engine for Clone Detection



24



CODE COMBAT

Google

The Asia Foundation

Let's Code Thailand

DMAP
DIGITAL MAKERS ASIA PACIFIC

THAI
PROGRAMMER



E-SAN Thailand Coding & AI Academy

โครงการวิจัยโมเดลระบบนิเวศการเรียนรู้ที่บูรณาการ CODING & AI สำหรับเยาวชน

Model of Learning Ecosystem Platform integrate with Coding & AI for Youth

โครงการย่อยที่ 6

การพัฒนาเยาวชนเพื่อเข้าสู่วิชาชีพขั้นสูงด้าน Coding & AI

ร่วมกับ Coding Entrepreneur & Partnership: Personal AI

ชื่อหัวข้อ Code Clone Detector - ระบบ AI สำหรับช่วยตรวจวัดความเหมือนของโค้ดและ
ตรวจจับการคัดลอกโค้ด

ดร. ชัยยงค์ รักขิตเวชสกุล





Coding Workshop

- ดาวน์โหลดแบบฝึกหัดได้ที่ <https://colab.research.google.com/drive/1qZdJLYtSeLt3D2czSyFSch824-9AnJvu?usp=sharing> หรือ Scan QR Code ด้านล่าง

