Class 3: AI for detecting code plagiarism

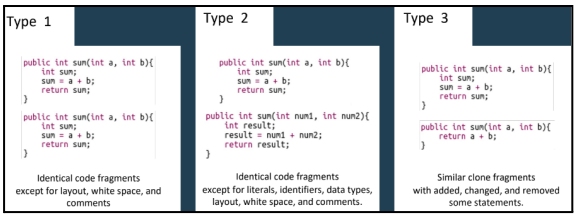
What are the code clones?

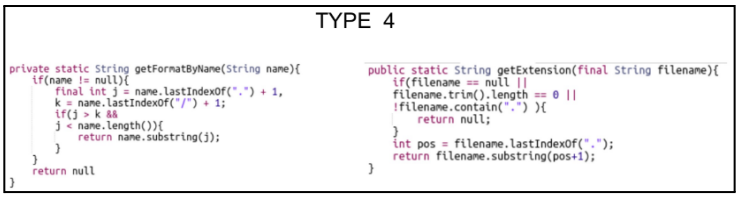
⇒ โค้ดที่เหมือนกัน (code clone) คือการที่เรามี source code 2 อัน (code fragment) เราจะสร้างคู่ของ source code ที่มีความหมือนกัน เรียกว่า “clone pair” ถ้า code มีความเหมือนกันมากพอ ก็ถือว่าเป็น code clone

```
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;
}
```

- Syntactic Based
  - ☐ Clone type 1: เหมือนกันทั้งหมด เหมือนเป๊ะๆ 100% ไม่สนใจ layout, space, comment
  - ☐ Clone type 2: เหมือนกันเกือบหมด ต่างกันแค่ชื่อตัวแปร หรือ string, data type, layout, space
  - ☐ Clone type 3: มีการ added, changed, removed some statements
- Functionality Based
  - ☐ Clone type 4: have the same computation but different in syntax or algorithm.





→ plagiarism : ทำผิดทางจริยธรรม
→ can b harmful in software maintenance: ถ้าต้นฉบับ bugs เราก็ต้องไปตามหา fixed bugs
→ usually occur 7-23% in software
→ some can be beneficial

- ☐ Code Clone Process
  1. Preprocessing : การนำซอสโค้ดมาทำให้อยู่ในรูปแบบเดียวกัน เช่น format layout
  2. Transformation : การทำให้โค้ดอยู่ในรุปแบบของ vector โดยใช้ ML
  3. Match Detection : การดูว่าซอสโค้ดมันเหมือนกันไหม ส่วนที่เราจะตั้งนิยามคำว่า “ความเหมือน”
  4. Formatting : เอามาจัดให้สวยงาม
  5. Post Processing Filtering : ตัดข้อมูลที่ไม่จำเป็นออก
  6. Aggregation : รวมผลลัพธ์ส่งให้ผู้ดูแลระบบ ให้ตรวจสอบ
- ☐ Problem Statement

  ⇒ ใช้วิธีการทาง ML เพื่อตรวจสอบ source code ที่มีการแก้ไขจำนวนมากๆ เช่น added, deleted, modified statement

  ⇒ เครื่องมือที่มีอยู่ใช้งานค่อนข้างยาก cmd-based
- ☐ Objectives

  ⇒ สร้างเครื่องมือ “Code clone detection tools” using ML

  ⇒ ตรวจสอบว่าผู้ใช้งานชอบไหม ผลลัพธ์เข้าใจง่ายดีไหม
- ☐ Merry (Web-based) code clone Detection system using ML

  → ML algorithms: DT, RF, SVM, SVM+SMO

  → Features: 11 synthetic code metrics + 12 Semantic Code metrics (code2vec)

  → BigCloneBench: Largest and Credible clone data

  → Web-based tool with User interface

  → GitHub Integration

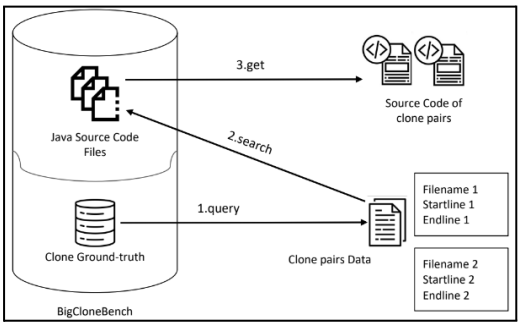
  → Visualization and Report

Modeling

⇒ วิธีการสร้าง ML model เพื่อตรวจจับ code clone

- ☐ Building Merry engine
- Data collection and Preparation

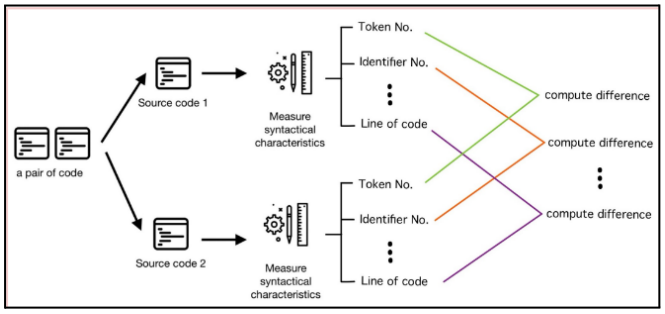
ข้อมูลจาก BigCloneBench เป็นฐานข้อมูลขนาดใหญ่ มีข้อมูล clone จำนวนมาก และถูกใช้บ่อย เป็น source code จริง ภาษา java > 25k projects และมีการlabel



Training Data [8]				
True Clone Pairs (22,663 pairs)				**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

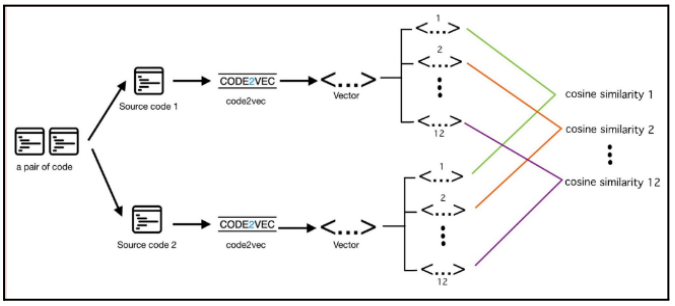
- Code metrics extraction

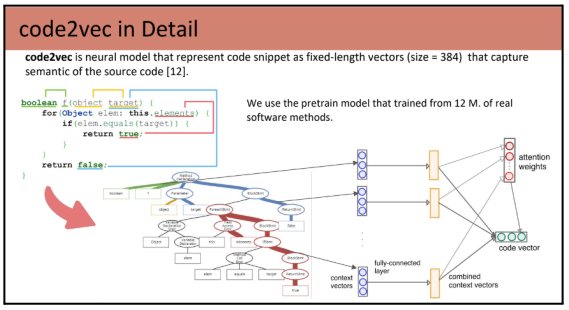
⇒ Synthetics Metrics : ดูจาก syntax โครงสร้าง วิธีการเขียนโค้ด



No.	Metric	Description
1	Token No [9]	Difference of number of tokens
2	Unique Token No [10]	Difference of number of unique tokens
3	Identifier No [10]	Difference of number of Identifiers
4	Unique Identifier No [10]	Difference of number of unique Identifiers
5	Operator No [10]	Difference of number of operators
6	Unique Operator No [10]	Difference of number of unique operators
7	Token Types Diversity [9]	Difference of number of unique token types
8	Diff File Name Score	File names difference score
9	Diff Method Name Score	Method names difference score
10	Similar Return Type	Same return type or not
11	DiffLOC	Difference of lines of code

⇒ Semantic Metrics : ดูจากวิธีการทำงานของโค้ด





source code 1

sum1.java

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

source code 2

sum2.java

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

Syntactic metrics

Diff. No. Tokens	:	7
Diff. No. Unique Token	:	1
Diff. No. Identifier	:	3
Diff. No. Unique Identifier	:	0
Diff. No. Operator	:	1
Diff. No. Unique Operator No	:	1
Diff. Token Types Diversity	:	0
Diff. File Name Score	:	0.25
Diff. Method Name Score	:	0.0
Similar Return Type	:	TRUE
Diff. No. LOC	:	2

Semantic metrics

vector#1 similarity	:	0.5206784273085
vector#2 similarity	:	0.457455059252847
vector#3 similarity	:	0.588965399230431
vector#4 similarity	:	0.652574760995595
vector#5 similarity	:	0.529570896554088
vector#6 similarity	:	0.490625929242188
vector#7 similarity	:	0.792832125905209
vector#8 similarity	:	0.562600679903085
vector#9 similarity	:	0.799728586131479
vector#10 similarity	:	0.642114908324134
vector#11 similarity	:	0.540922173109001
vector#12 similarity	:	0.729365945085146

- ML Model
- ☐ Using Merry engine for Clone Detection

⇒ Software Project on GH

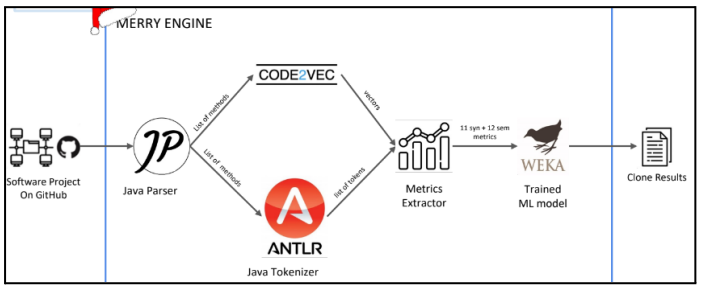
⇒ Java Parser : แบ่ง source code ออกเป็น method

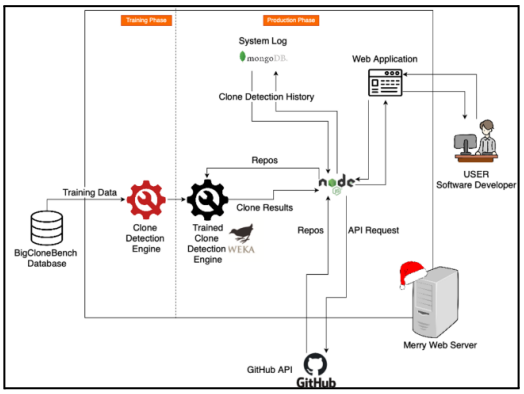
เอา method แต่ละอันผ่าน ⇒ Code2Vec

⇒ Java Tokenizer : แบ่งตัว source code ออกเป็น token

⇒ แล้วเอา vector หรือ tokens ที่ได้ มาคำนวณเมทริกซ์ ⇒ Metrics Extractor จำนวน variable, identifier, line

⇒ ข้อมูลจาก BigCloneBench มา Trained ML model ผ่าน Weka ได้โมเดลแล้ว เอามาใช้กับ metrics ⇒ Clone Result



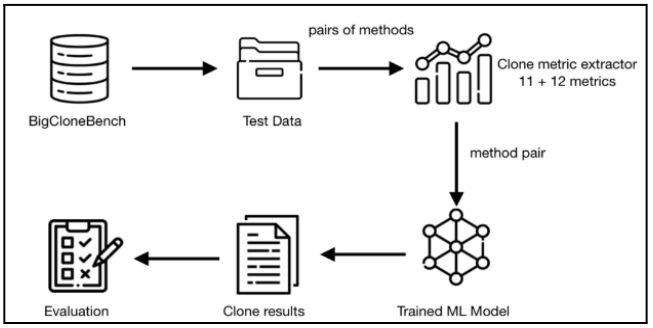


Evaluation : เชิงความแม่นยำ

– How ACC is Merry code clone detection on BigCloneBench?

- Precision, recall, F1-score

Model	Metrics	Precision	Recall	F1-Score
Randomization (baseline)		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



– How ACC is Merry code clone detection on real Software Project?

– How likely are CMD-based tools and Merry adopted by programmers?

Evaluation : เชิงความเครื่องมือ Merry vs Semian

⇒ วิธีการ Between Subjects study designed แบ่งกลุ่ม 2 กลุ่ม ใช้คนละเครื่องมือ Reduce the impact of transfer across conditions ⇒ Merry คนอยากใช้ / เครื่องมือเข้าใจง่ายกว่า / easy to use