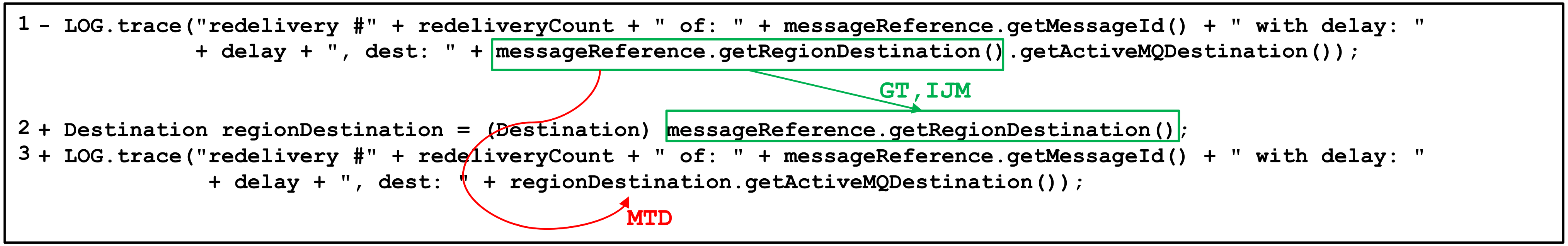
Supplementary Details for Our Paper

Outline

1. Illustrative examples for our false positive case
2. Illustrative examples for our false negative cases
3. Do different algorithms generate inaccurate mappings for the same sets of statements and file revisions?
4. Similarity measures that are most commonly used to determine inaccurate mappings for GT, MTD and IJM
5. Advantages and disadvantages of GT, MTD and IJM.

Our false positive case



```
1 - LOG.trace("redelivery #" + redeliveryCount + " of: " + messageReference.getMessageId() + " with delay: "
              + delay + ", dest: " + messageReference.getRegionDestination().getActiveMQDestination());

2 + Destination regionDestination = (Destination) messageReference.getRegionDestination();
3 + LOG.trace("redelivery #" + redeliveryCount + " of: " + messageReference.getMessageId() + " with delay: "
               + delay + ", dest: " + regionDestination.getActiveMQDestination());
```

GT , MTD and IJM represents the mappings inferred by GT, MTD and IJM.

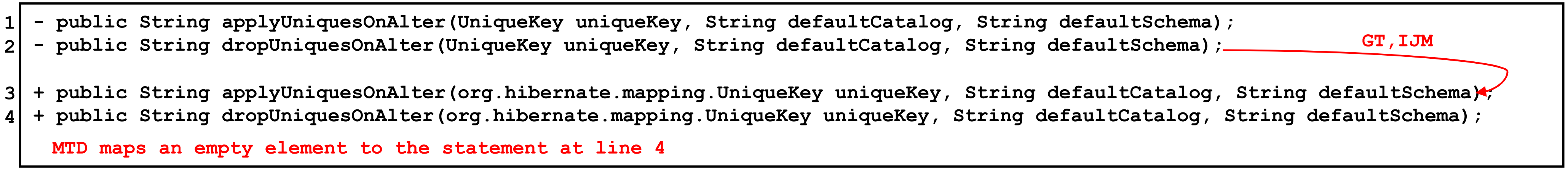
This is the only one false positive case in our evaluation dataset. The correct mapping is shown with a green arrow.

In this case, the developer is extracting the method invocation messageReference.getRegionDestination() as a new variable. GT and IJM accurately maps message.getRegionDestination() at line 1 to line 2. MTD inaccurately maps messageReference at line 1 to regionDestination at line 3.

The statement at line 1 should be mapped to the statement at line 3. In our paper, our approach considers that mapping two tokens in mapped statements is better than mapping two tokens from unmapped statements. However, when a developer performs refactoring changes, mapping tokens from unmapped statements may be better than mapping tokens in mapped statements.

https://github.com/apache/activemq/commit/9a8f6e415db43a4e43ad42a87b3617b3641aa07d#diff-12a98a6ac2236738502713b224a7b0c3e7e2e52c16e2e36461fed351334b8341

Our false negative case (1)



```
1 - public String applyUniquesOnAlter(UniqueKey uniqueKey, String defaultCatalog, String defaultSchema);
2 - public String dropUniquesOnAlter(UniqueKey uniqueKey, String defaultCatalog, String defaultSchema);

3 + public String applyUniquesOnAlter(org.hibernate.mapping.UniqueKey uniqueKey, String defaultCatalog, String defaultSchema);
4 + public String dropUniquesOnAlter(org.hibernate.mapping.UniqueKey uniqueKey, String defaultCatalog, String defaultSchema);
```

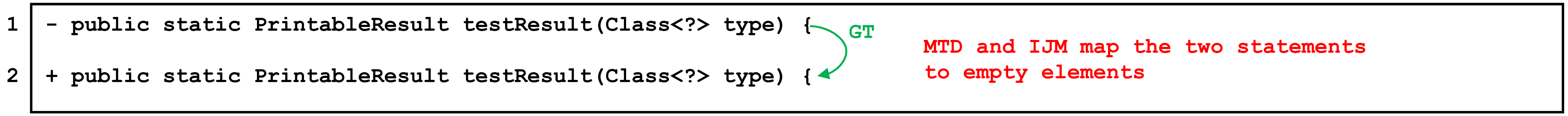
Correct mappings are: `(line 1 -> line 3) (line 2 -> line 4)`

All of the studies algorithms cannot generate inaccurate mapping for the statement at line 2.

Both GT and IJM maps the statement at line 2 to the statement at line 3. Thus, by comparing the two algorithms, we cannot find the inaccurate mapping.

https://github.com/hibernate/hibernate-orm/commit/7b05f4aed8845d4ccefce71eea438b81be10610e#diff-040ff7552977d9d5efea3002e4a2cfc2b4ca61186e239ad445cf58d68f149e05

Our false negative case (2)



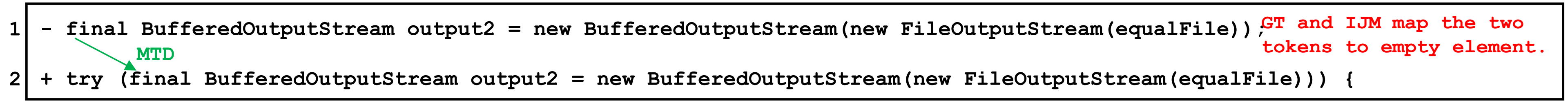
```
- public static PrintableResult testResult(Class<?> type) {
+ public static PrintableResult testResult(Class<?> type) {
```

Correct mapping is shown with a green arrow.

When an algorithm maps two statements and another algorithm maps them to empty elements, our method is not able to determine if the two statements should be mapped. Thus, our method cannot determine which algorithm generates the inaccurate mapping.

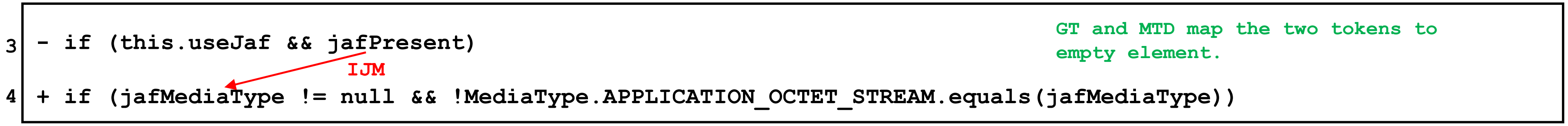
https://github.com/junit-team/junit4/commit/409a8e06c9f2ec5aa0d9db8a3d413f394c290f6d#diff-c40e1815088500323bf382f34b4869a70ae146cf28b72c7a2505bba965ce26f0

Our false negative case (3)



```
- final BufferedOutputStream output2 = new BufferedOutputStream(new FileOutputStream(equalFile));
+ try (final BufferedOutputStream output2 = new BufferedOutputStream(new FileOutputStream(equalFile))) {
```

GT and IJM map the two tokens to empty element.



```
- if (this.useJaf && jafPresent)
+ if (jafMediaType != null && !MediaType.APPLICATION_OCTET_STREAM.equals(jafMediaType))
```

GT and MTD map the two tokens to empty element.

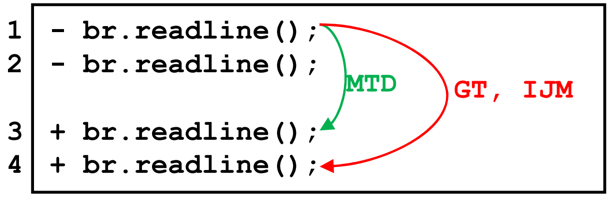
Correct mapping in the first figure is shown with the green arrow. Correct mapping in the second figure is shown with the green text.

When an algorithm maps two tokens and another algorithm maps the two tokens to empty elements, our similarity measures cannot determine which algorithm generates the inaccurate mapping.

https://github.com/apache/commons-io/commit/79b4df582d0035e196d4dc10894778fae58311ce#diff-d7f0d0432bfbd4488035ea3c9db78425b68e83cca9893fc876b3400b7d74d440

https://github.com/spring-projects/spring-framework/commit/83c83d4d152ff6d8bffe79e9eece31ea0fc89c0e#diff-d30fa6bec53e504481aacb52d762606d3980515b7c2400456011e026954a243f

Our false negative case (4)



```
1 - br.readline();
2 - br.readline();

3 + br.readline();
4 + br.readline();
```

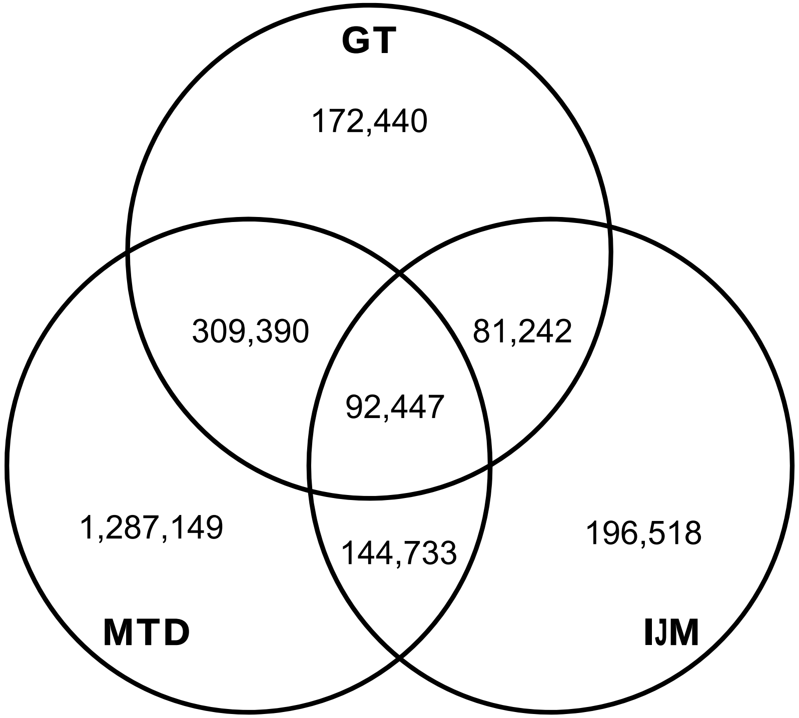
Correct mapping is shown with the green arrow.

From the figure, we find that the four statements are identical. The statement at line 1 should be mapped to the statement at line 3. And the statement at line 2 should be mapped to the statement at line 4.

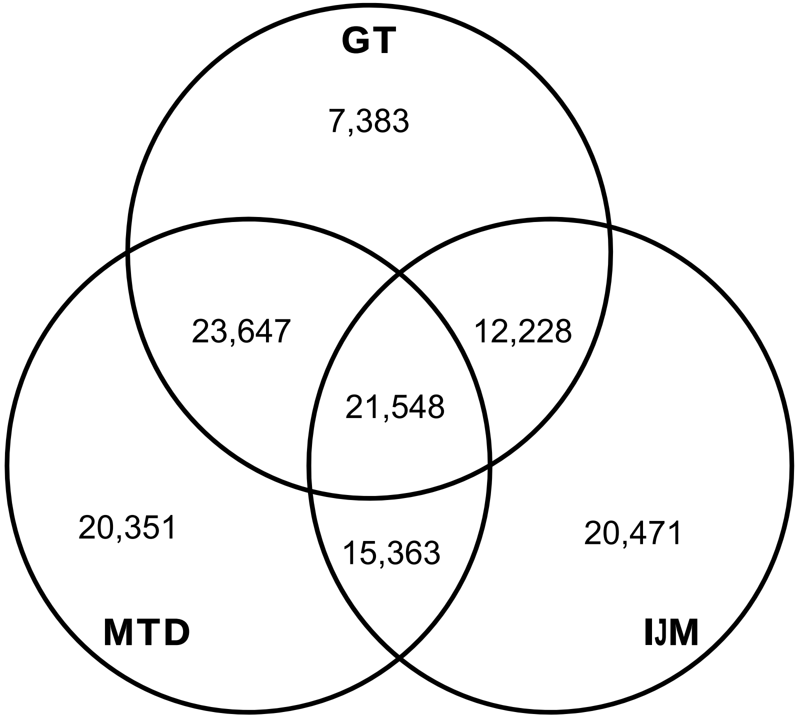
Currently, our similarity measure does not consider the order of mapped statements. Hence, we cannot determine the accuracy of the mappings as generated by the three algorithms.

https://github.com/apache/commons-io/commit/6b57d2a14089735cf1c653a2717d05023a3be441#diff-43bf96e3930e668aa2f391ea621d0397d5ca12e08b34075db235b5327dcec5f4

Different algorithms can generate inaccurate mappings in the same sets of statements and file revisions.



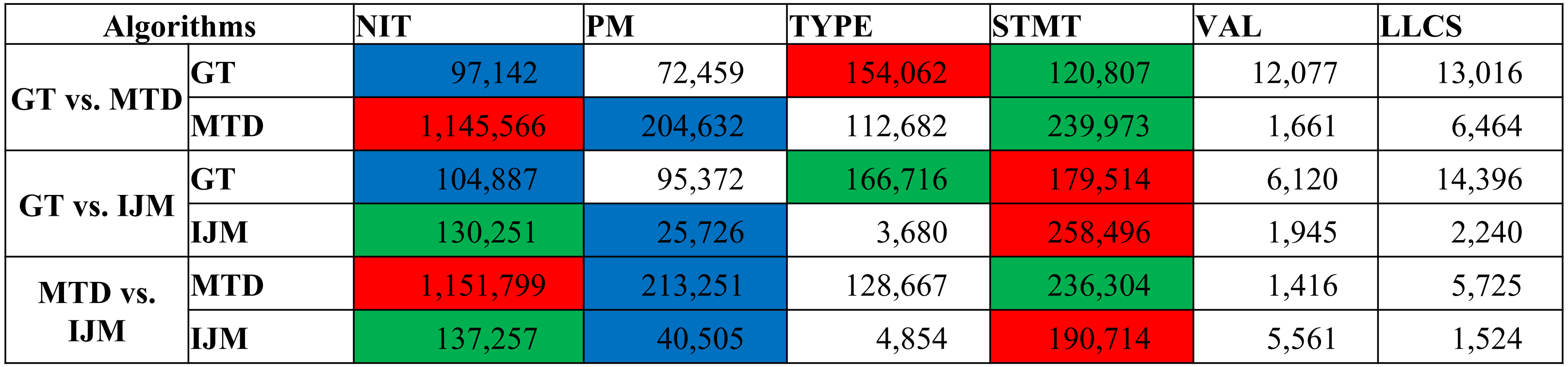
A Venn diagram of the statements with inaccurate mappings for comparing GT, MTD and IJM



A Venn diagram of the file revisions with inaccurate mappings for comparing GT, MTD and IJM

Similarity measures used to determine the accuracy of generated mappings by different algorithms

Table 2. Number of statements with inaccurate mappings that are determined by the similarity measures when comparing each pair of algorithms.



Algorithms		NIT	PM	TYPE	STMT	VAL	LLCS
GT vs. MTD	GT	97,142	72,459	154,062	120,807	12,077	13,016
	MTD	1,145,566	204,632	112,682	239,973	1,661	6,464
GT vs. IJM	GT	104,887	95,372	166,716	179,514	6,120	14,396
	IJM	130,251	25,726	3,680	258,496	1,945	2,240
MTD vs. IJM	MTD	1,151,799	213,251	128,667	236,304	1,416	5,725
	IJM	137,257	40,505	4,854	190,714	5,561	1,524

🟥 Rank1

🟩 Rank2

🟦 Rank3

We highlight the top 3 commonly used measures to detect statements with inaccurate mappings for each algorithm (i.e., for each row).

Findings:

1. **NIT, PM and STMT are the most commonly used measures to determine inaccurate mappings for different algorithms**
2. **For GT, TYPE is also an important measure for detecting the inaccurate mappings**

Advantages and Disadvantages of GT, MTD and IJM

Table 2. Number of statements with inaccurate mappings that are determined by the similarity measures when comparing each pair of algorithms.

Algorithms		NIT	PM	TYPE	STMT	VAL	LLCS
GT vs. MTD	**GT**	97,142	72,459	154,062	120,807	12,077	13,016
	MTD	1,145,566	204,632	112,682	239,973	1,661	6,464
GT vs. IJM	**GT**	104,887	95,372	166,716	179,514	6,120	14,396
	IJM	130,251	25,726	3,680	258,496	1,945	2,240
MTD vs. IJM	**MTD**	1,151,799	213,251	128,667	236,304	1,416	5,725
	IJM	137,257	40,505	4,854	190,714	5,561	1,524

We highlight the algorithm that is detected to generate the most and least number of inaccurate mappings for each measure (i.e., for each row).

Advantages:

GT:

1. GT is less likely to map dissimilar statements with less identical tokens

MTD:

1. MTD is less likely to map tokens with different values.

IJM:

1. IJM is less likely to map dissimilar statements without parent nodes mapped.
2. IJM is less likely to map tokens with different types.
3. IJM is more likely to sequentially map the tokens in mapped statements.

Disadvantages:

GT:

1. GT is more likely to map tokens with different types
2. GT is more likely to map tokens with different values
3. GT is more likely to map tokens out-of-order in mapped statements

MTD:

1. MTD is more likely to map dissimilar statements, for which we can find better mapped statements with a larger number of identical tokens or parent nodes mapped.

IJM:

1. IJM is more likely to generate inaccurate mappings of tokens because tokens in the mapped statements are not mapped.