A DATASET FOR DYNAMIC DISCOVERY OF SEMANTIC CHANGES IN VERSION CONTROLLED SOFTWARE HISTORIES

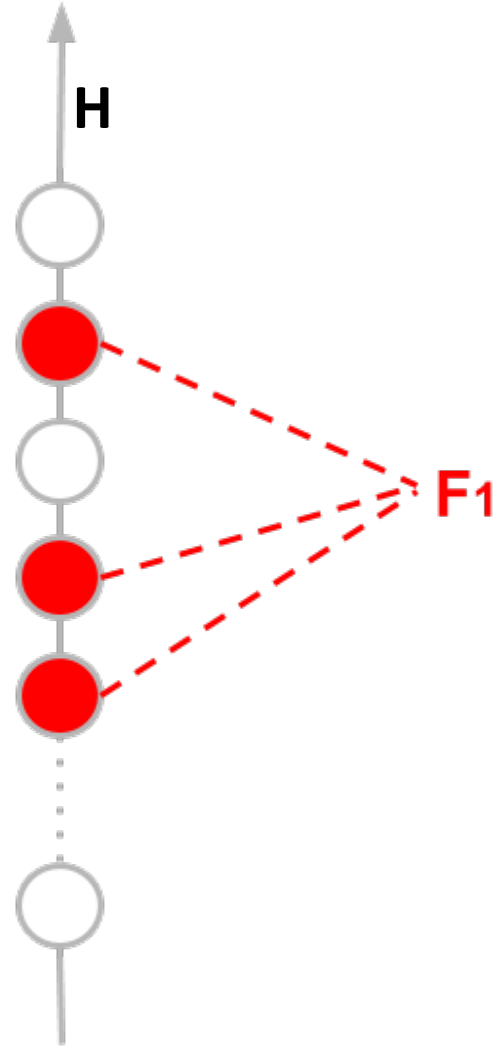
MSR
MAY 21, 2017


UNIVERSITY OF TORONTO

Chenguang Zhu
Yi Li
Julia Rubin
Marsha Chechik


THE UNIVERSITY OF BRITISH COLUMBIA

SEMANTIC HISTORY SLICING



Goal: Given a version-controlled software history,
identify code changes related to a particular high-level functionality.

Uses:

- Assist in transferring functionalities across branches
- Produce focused pull-requests
- Locate features [SPLC'17]
- Etc.

Semantic History Slice: A (minimal) sub-sequence of a change history that preserves the functionality of interest as *defined by a set of test cases.*

[ASE'15, ASE'16, TSE'17]

CHALLENGES IN EVALUATING SEMANTIC HISTORY SLICING

- Need a significant number of well-documented functionalities
- Functionalities should be accompanied by test cases
- Need ground truth

} Difficult to obtain!

Time consuming

Contribution of this work:

- 98 items of semantic change data, collected from 10 open-source Java projects.

DATASET CREATION

- Chose well-documenting projects
  - 10 Apache Java projects, using JIRA for issue tracking
- Chose projects with test cases
  - Functionalities committed together with a test suite
- Obtained ground truth
  - Ran a delta debugging-style partitioning algorithm, to produce 1-minimal slices
    - 1-minimal – removing a single commit makes the result invalid

DATA SCHEMA

Meta-data: YAML

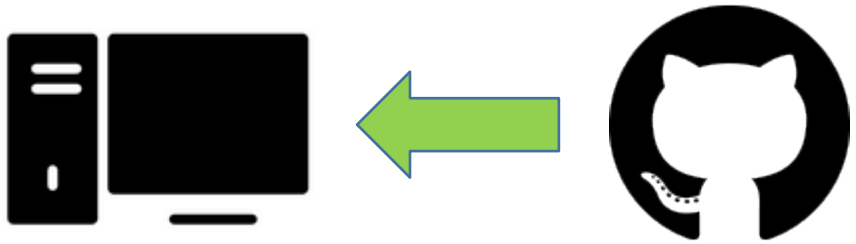
Example: CALCITE-1168

```yaml
id: CALCITE-1168
description: Add DESCRIBE SCHEMA/DATABASE/TABLE/query
project:
  name: Calcite
  project url: https://github.com/MSR-2017/calcite
issue url: https://issues.apache.org/jira/browse/CALCITE-1168
history start: 8eebfc6d
history end: aeb6bf14
test suite:
  - "SqlParserTest.testDescribeSchema"
  - "SqlParserTest.testDescribeTable"
  - "SqlParserTest.testDescribeStatement"
history slice:
  - "a065200a"
  - "da875a67"
developer labeled commits:
  - "a065200a"
  - "da875a67"
```

HOW TO USE THE DATASET

1. Pick a functionality (e.g., CALCITE-1168), view the meta-data.

2. Clone the project repository to the local machine.



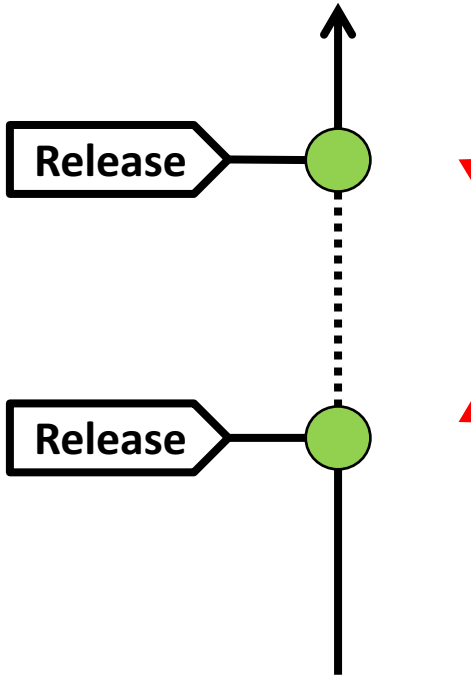
3. Extract all test cases defining the functionality.

F_1 { TestA, TestB.testf(), TestC.testg(), ... }

```
id: CALCITE-1168
description: Add DESCRIBE SCHEMA/DATABASE/TABLE/query
project:
  name: Calcite
  project url: https://github.com/MSR-2017/calcite
issue url: https://issues.apache.org/jira/browse/CALCITE-1168
history start: 8eebfc6d
history end: aeb6bf14
test suite:
  - "SqlParserTest.testDescribeSchema"
  - "SqlParserTest.testDescribeTable"
  - "SqlParserTest.testDescribeStatement"
history slice:
  - "a065200a"
  - "da875a67"
developer labeled commits:
  - "a065200a"
  - "da875a67"
```

HOW TO USE THE DATASET

4. Extract the starting point and the ending point of the history segment (SHA-1-expressed commit number)



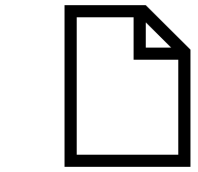

```
id: CALCITE-1168
description: Add DESCRIBE SCHEMA/DATABASE/TABLE/query
project:
  name: Calcite
  project url: https://github.com/MSR-2017/calcite
issue url: https://issues.apache.org/jira/browse/CALCITE-1168
history start: 8eebfc6d
history end: aeb6bf14
test suite:
  - "SqlParserTest.testDescribeSchema"
  - "SqlParserTest.testDescribeTable"
  - "SqlParserTest.testDescribeStatement"
history slice:
  - "a065200a"
  - "da875a67"
developer labeled commits:
  - "a065200a"
  - "da875a67"
```

5. Using the selected test cases (Step 3) and history segment (Step 4) as input, run the history slicing/feature location tool to be evaluated

HOW TO USE THE DATASET

6. Compare the obtained result with ground truth we provide

1) Semantic History Slicing:

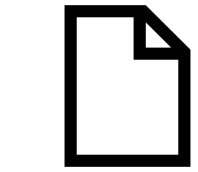


User's result

Compare

```
id: CALCITE-1168
description: Add DESCRIBE SCHEMA/DATABASE/TABLE/query
project:
  name: Calcite
  project url: https://github.com/MSR-2017/calcite
issue url: https://issues.apache.org/jira/browse/CALCITE-1168
history start: 8eebfc6d
history end: aeb6bf14
test suite:
  - "SqlParserTest.testDescribeSchema"
  - "SqlParserTest.testDescribeTable"
  - "SqlParserTest.testDescribeStatement"
history slice:
  - "a065200a"
  - "da875a67"
developer labeled commits:
  - "a065200a"
  - "da875a67"
```

2) Feature location:



User's result

Compare

DATASET OVERVIEW

Project	#F	#R	Avg. Commits	Avg. Files	Avg. LOC	Avg. Tests	Avg. Slice	Avg. Reduce (%)
commons-lang	20	4	334.25	191.75	17423.95	5.55	43.1	87.11
calcite	18	7	89.83	332.67	31150.78	3.39	6.61	92.64
maven	11	6	82.09	183.09	7153.27	2.27	8.18	89.24
commons-compress	9	2	155	156.33	7172.67	5	17.33	88.82
flume	9	3	104.11	299.33	21355.56	4	20.22	79.82
pdfbox	5	3	203	188.4	10184	6.2	2	98.7
commons-configuration	3	2	117.33	254	54576	6	20.67	65.61
commons-net	3	2	205	188.33	7202.33	6.67	29	87.05
commons-csv	4	1	79	28	2353	3.75	42.5	46.2
commons-io	16	2	138.25	158.38	8047.44	9.5	24.06	82.59
Overall	98	32	163.74	212.79	16521.01	5.24	21.66	86.77

STATUS

URL: https://github.com/Chenguang-Zhu/DoSC

Contents

- Meta-data template
- 98 pieces of data
- Our tool for obtaining 1-minimal history slice

Easy to extend

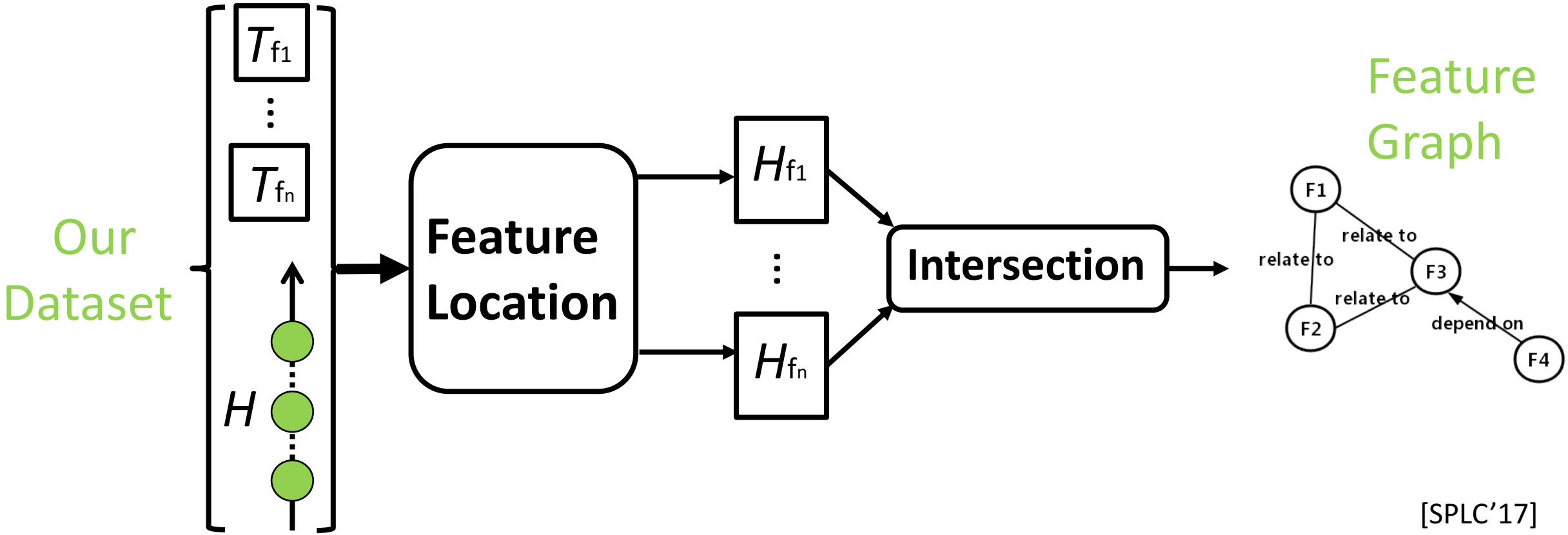
- … e.g., by including more repositories
- … or by adding histories containing bugs and failed test cases – for fault localization

WE LOOK FORWARD TO YOUR USES AND EXTENSIONS!!!!!

Thank you!

A RECENT USE CASE

FHistorian – A feature relationship analysis tool



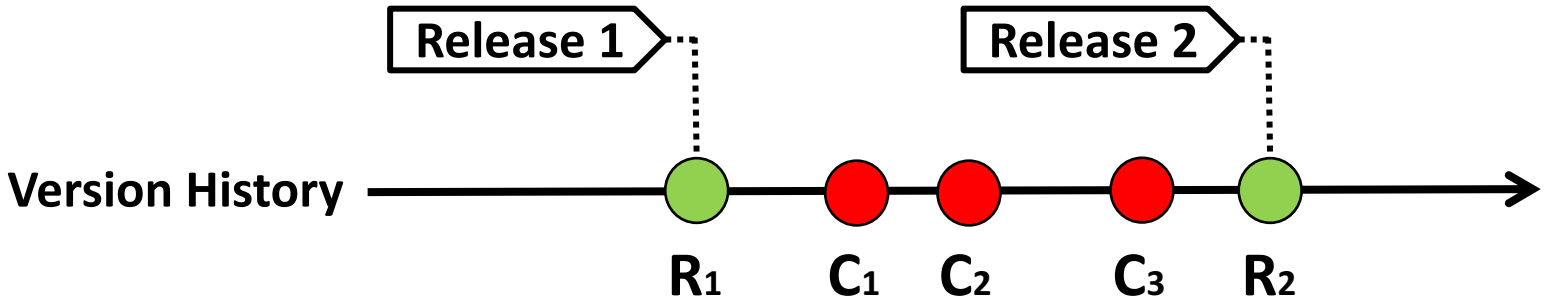
[SPLC'17]

Thank you !

Try it here!  https://github.com/Chenguang-Zhu/DoSC

DATASET CREATION

- Project selection: 10 Apache Java projects, using JIRA for issue tracking.
- Functionality selection: Functionalities accompanied by a test suite, committed together.
- History range selection: Between two release versions. (R_1 – R_2)



- Obtaining ground truth: Run a delta debugging-style partitioning algorithm, to produce the 1-minimal slice.