

Predicting Code Context Models for Software Development Tasks



Zhiyuan Wan



Gail Murphy



Xin Xia



MONASH University

Code Context for a Development Task



Bug [303431](#) - remove CommonColors.CONTEXT_ACTIVE

Status: RESOLVED FIXED

Alias: None

Product: Mylyn

Component: UI ([show other bugs](#))

Version: dev

Hardware: All All

Importance: P3 trivial ([vote](#))

Target Milestone: 3.4

Assignee: Frank Becker

QA Contact:

URL:

Whiteboard:

Keywords:

Depends on:

Blocks: [199345](#)

Show dependency [tree](#)

Reported: 2010-02-21 05:34 EST by Frank Becker ✓ ECA

Modified: 2010-02-25 21:51 EST ([History](#))

CC List: 3 users ([show](#))

See Also:

Code Context for a Development Task

eclipse-workspace - org.eclipse.mylyn.tasks.ui/src/org/eclipse/mylyn/tasks/ui/TaskElementLabelProvider.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- org.eclipse.mylyn.commons.ui [mylyn.commons 3ead864]
 - JRE System Library [J2SE-1.5]
 - icons
 - META-INF
 - src
 - about.html
 - build.properties
 - plugin.properties
- org.eclipse.mylyn.tasks.ui [mylyn.tasks 7845748]
 - JRE System Library [J2SE-1.5]
 - doc
 - icons
 - META-INF
 - schema
 - src
 - org
 - eclipse
 - mylyn
 - internal
 - tasks
 - ui
 - editors
 - wizards
 - AbstractRepositoryConnectorUi.java
 - AbstractTaskHyperlinkDetector.java
 - AbstractTaskRepositoryLinkProvider.java
 - IHighlightingHyperlink.java
 - ITasksUiConstants.java
 - ITasksUiFactory.java
 - LegendElement.java
 - TaskElementLabelProvider.java
 - TaskHyperlink.java
 - TaskHyperlinkPresenter.java
 - TaskRepositoryLocationUiFactory.java
 - TasksUi.java
 - TasksUiImages.java
 - TasksUiUtil.java
 - about.html
 - build.properties
 - plugin.properties
 - plugin.xml

CommonThemes.java TaskElementLabelProvider.java

```
2 * Copyright (c) 2004, 2009 Tasktop Technologies and others.
11
12 package org.eclipse.mylyn.tasks.ui;
13
14 import java.util.HashSet;
15
16 /**
17  * @author Mik Kersten
18  * @since 3.0
19  */
20 public class TaskElementLabelProvider extends LabelProvider implements IColorProvider {
21
22     private final IThemeManager themeManager = PlatformUI.getWorkbench().getThemeManager();
23
24     private boolean wideImages = false;
25
26     private class CompositeImageDescriptor {
27
28         ImageDescriptor icon;
29
30         ImageDescriptor overlayKind;
31
32     };
33
34     public TaskElementLabelProvider() {
35         this(false);
36     }
37
38     public TaskElementLabelProvider(boolean wideImages) {
39         super();
40         this.wideImages = wideImages;
41     }
42 }
```

Outline

- org.eclipse.mylyn.tasks.ui
 - TaskElementLabelProvider
 - themeManager : IThemeManager
 - wideImages : boolean
 - CompositeImageDescriptor
 - TaskElementLabelProvider()
 - getImage(Object) : Image
 - getImageDescriptor(Object) : Color
 - getText(Object) : String
 - getForeground(Object) : Color
 - getBackground(Object) : Color
 - getFont(Object) : Font
 - showHasActiveChild(ITaskContainer) : boolean
 - showHasActiveChildHelper(ITaskContainer) : boolean

Problems Javadoc Declaration Console

No consoles to display at this time.

Writable Smart Insert 2 : 1 : 82

Task: Fixing Bug 303431

Code Context for a Development Task

Repositories

mylyn.commons
3ead864

mylyn.tasks
7845748

Code Elements

- 1 `org.eclipse.mylyn.internal.provisional.commons.ui.CommonThemes^COLOR_TASK_ACTIVE`
- 2 `org.eclipse.mylyn.internal.provisional.commons.ui.CommonThemes`
- 3 `org.eclipse.mylyn.tasks.ui.TaskElementLabelProvider`
- 4 `org.eclipse.mylyn.tasks.ui.TaskElementLabelProvider [getForeground`
- 5 `org.eclipse.mylyn.tasks.ui.TaskElementLabelProvider ^themeManager`

```

11  2 * Copyright (c) 20
12  package org.eclipse
13
14  import java.util.Has
48
49  /**
50   * @author Mik Kers
51   * @since 3.0
52   */
53  public class TaskEl
54
55  private final I
56
57  private boolean
58
59  private class C
60
61  ImageDescri
62
63  ImageDescri
64
65  };
66
67  public TaskElementLabelProvider() {
68      this(false);
69  }
70
71  public TaskElementLabelProvider(boolean wideImages) {
72      super();
73      this.wideImages = wideImages;
74  }
    
```

Problems @ Javadoc Declaration Console
No consoles to display at this time.

Writable Smart Insert 2 : 1 : 82

Task: Fixing Bug 303431

Code Context Model

Repositories

mylyn.commons 3ead864

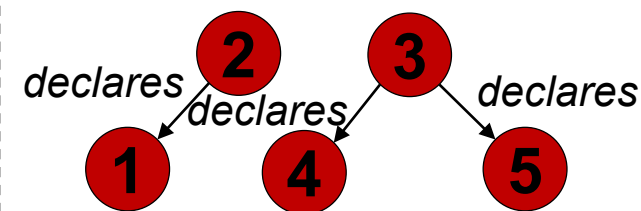
mylyn.tasks 7845748

Code Elements

- 1 org.eclipse.mylyn.internal.provisional.commons.ui.CommonThemes^COLOR_TASK_ACTIVE
- 2 org.eclipse.mylyn.internal.provisional.commons.ui.CommonThemes
- 3 org.eclipse.mylyn.tasks.ui.TaskElementLabelProvider
- 4 org.eclipse.mylyn.tasks.ui.TaskElementLabelProvider [getForeground
- 5 org.eclipse.mylyn.tasks.ui.TaskElementLabelProvider ^themeManager



Code Context Model



Task: Fixing Bug 303431

Related Work

Recommending elements in a code context model

1. Utilize Code Structure



Robillard [1] proposes *Suade* that leverages topological features of code structure.

2. Utilize Historical Information

Kersten and Gail [2], Robbes and Lanza [3], Zimmermann et al. [4].

[1] Martin P Robillard. 2008. *Topology analysis of software dependencies*. TOSEM.

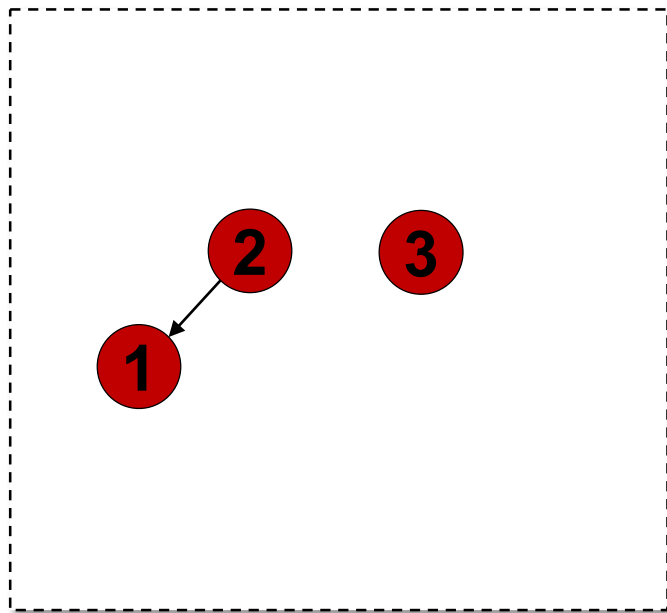
[2] Mik Kersten and Gail C. Murphy. 2006. *Using Task Context to Improve Programmer Productivity*. SIGSOFT '06/FSE-14.

[3] Romain Robbes and Michele Lanza. 2010. *Improving code completion with program history*. ASE.

[4] Thomas Zimmermann et al.. 2005. *Mining version histories to guide software changes*. TSE.

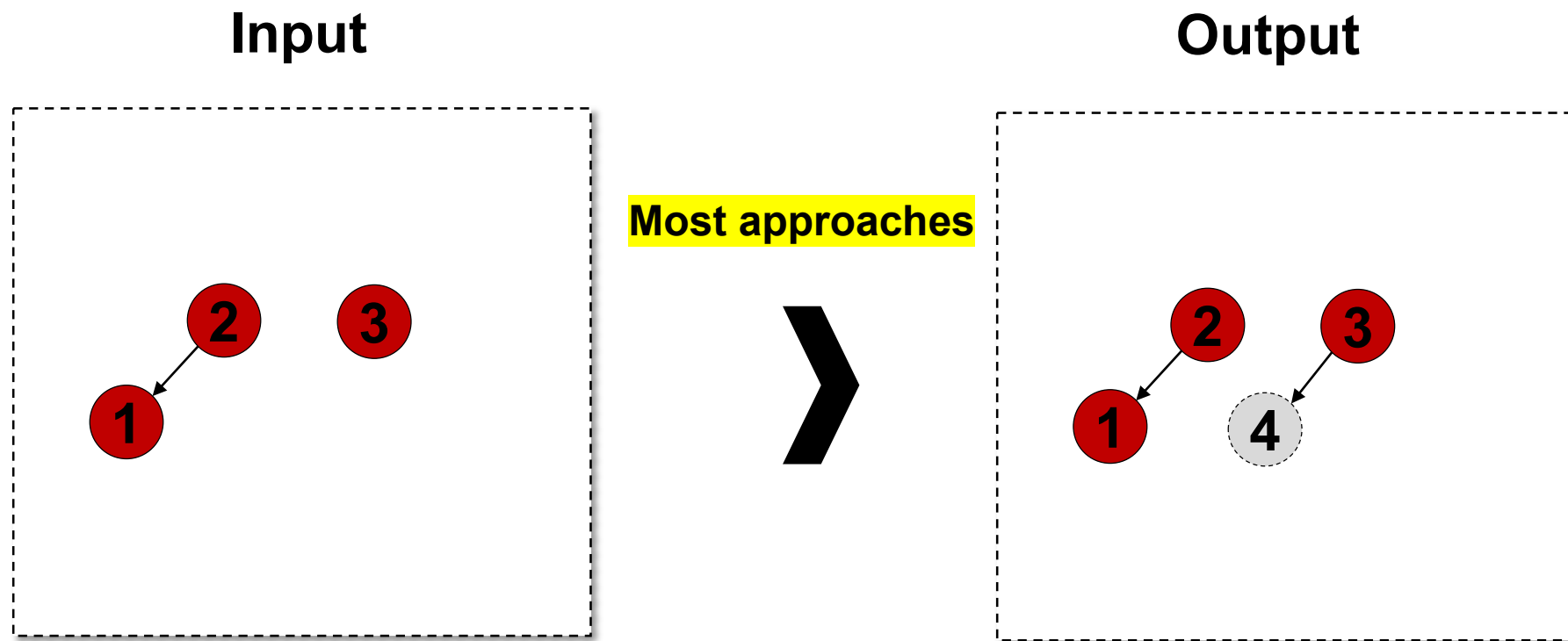
Recommendation of Code Elements

Input



In the Middle of the Task: Fixing **Bug 303431**

Recommendation of Code Elements

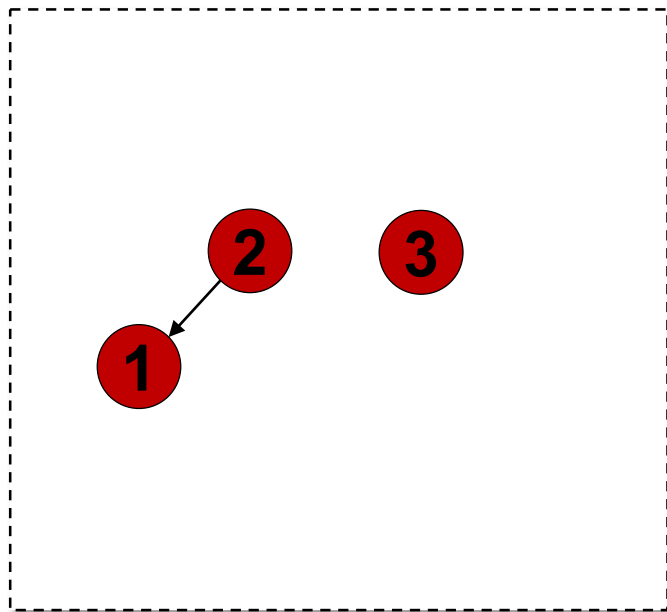


In the Middle of the Task: Fixing **Bug 303431**

Most approaches currently recommend **one element** at a time, this limits the support that can be conceived of for developers.

Recommendation of Code Elements

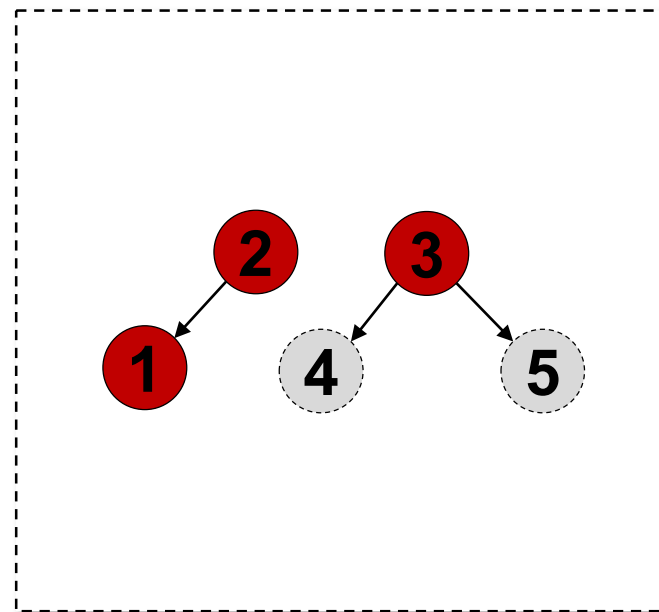
Input



Our approach



Output

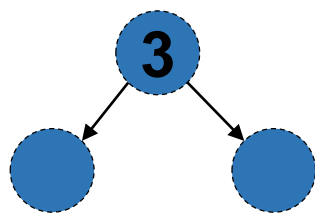


In the Middle of the Task: Fixing **Bug 303431**

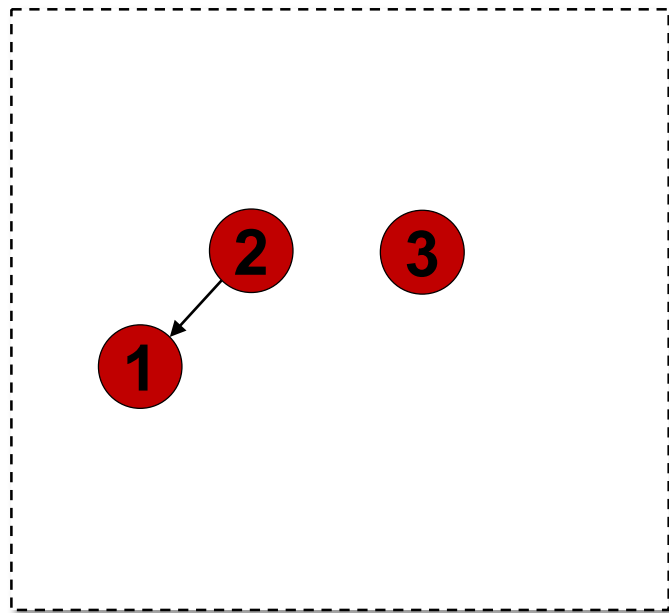
Our approach aims to recommend more than one element at a time.

Overview of Our Approach

1. Learn structural patterns from historical code context models.



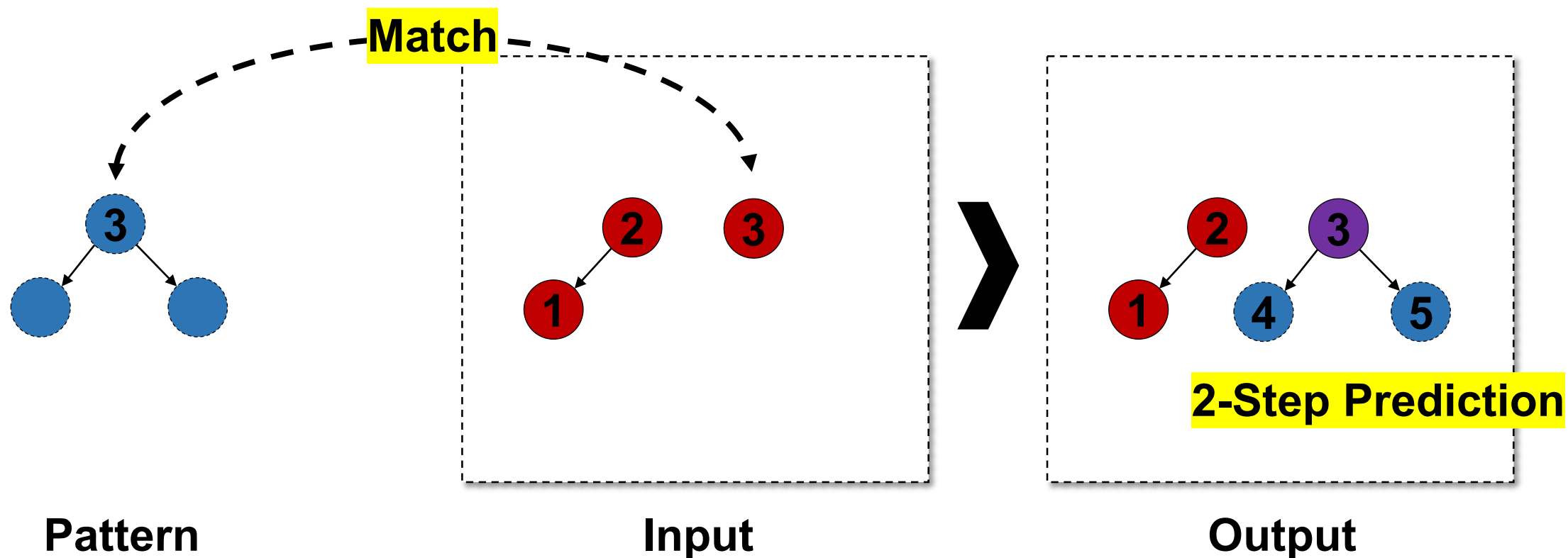
Pattern



Input

Overview of Our Approach

1. Learn structural patterns from historical code context models.
2. Use structural patterns to recommend multiple code elements at a time.



Research Questions

RQ1 What kinds of **patterns** can be learned from interaction histories?

RQ2 How does the **performance** of our approach differ over various values of **number of steps** of prediction?

RQ3 How does our approach compare to **the state of the art**?

Dataset: Mylyn Project



**Collect Fixed
Bug Report**

**Extract Final
Interaction
History**

**Extract Code
Elements**

**Resolve
Repository
Snapshots**

**Form Code
Context Model**

Interaction histories for over 15 years and over 117 developers.

Dataset: Mylyn Project

Collect Fixed
Bug Report

Extract Final
Interaction
History

Extract Code
Elements

Resolve
Repository
Snapshots

Form Code
Context Model



Bug_303431 - remove CommonColors.CONTEXT_ACTIVE

Status: RESOLVED FIXED

Alias: None

Product: Mylyn


Component: UI ([show other bugs](#))

Version: dev 

Hardware: All All

Importance: P3 trivial ([vote](#))

Target Milestone: 3.4 

Assignee: Frank Becker 

QA Contact:

URL:


Whiteboard:

Keywords:

Depends on:

Blocks: [199345](#)

[Show dependency tree](#)

Reported: 2010-02-21 05:34 EST by Frank Becker 

Modified: 2010-02-25 21:51 EST ([History](#))

CC List: 3 users ([show](#))

See Also:

5,208 FIXED bug reports (2004 - 2019)

Dataset: Mylyn Project

Collect Fixed Bug Report

Extract Final Interaction History

Extract Code Elements

Resolve Repository Snapshots

Form Code Context Model

Bug 303431 - remove CommonColors.CONTEXT_ACTIVE

Status: RESOLVED FIXED

Reported: 2010-02-21 05:34 EST by Frank Becker

Modified: 2010-02-25 21:51 EST (History)

CC List: 3 users (show)

See Also:

Alias: None

Product: Mylyn

Component: UI (show other bugs)

Version: dev

Hardware: All All

Importance: P3 trivial (vote)

Target Milestone: 3.4

Assignee: Frank Becker

QA Contact:

URL:

Whiteboard:

Keywords:

Depends on:

Blocks: 199345 Show dependency tree

1,246 FIXED bug reports with interaction histories.

Attachments		
committed patch (3.65 KB, patch) 2010-02-21 05:44 EST, Frank Becker	no flags	Details Diff
mylyn/context/zip (1.55 KB, application/octet-stream) 2010-02-21 05:44 EST, Frank Becker	no flags	Details
commitet patch (3.03 KB, patch) 2010-02-21 09:58 EST, Frank Becker	no flags	Details Diff
mylyn/context/zip (1.74 KB, application/octet-stream) 2010-02-21 09:58 EST, Frank Becker	no flags	Details
Add an attachment (proposed patch, testcase, etc.)		View All



Interaction History 159708

Dataset: Mylyn Project

Collect Fixed
Bug Report

Extract Final
Interaction
History

Extract Code
Elements

Resolve
Repository
Snapshots

Form Code
Context Model

Interaction History 159708

```
<InteractionHistory Version="1" Id="https://bugs.eclipse.org/bugs-303431">
...
<InteractionEvent CreationCount="9" NumEvents="4" StructureKind="java"
StructureHandle="=org.eclipse.mylyn.commons.ui/src&lt;org.eclipse.mylyn.internal.
provisional.commons.ui{CommonThemes.java[CommonThemes^COLOR_TASK_A
CTIVE" StartDate="2010-02-21 11:36:05.55 CET"
OriginId="org.eclipse.jdt.ui.PackageExplorer" Navigation="null" Kind="selection"
Interest="4.0" EndDate="2010-02-21 15:21:48.606 CET" Delta="null"/>
...
</InteractionHistory>
```

Event Interval > 2 hours



Working Period

```
<InteractionEvent />
...
```



Code Elements

- 1 org.eclipse.mylyn.internal.provisional.commons.ui.CommonThemes^COLOR_TASK_ACTIVE
- 2 org.eclipse.mylyn.internal.provisional.commons.ui.CommonThemes
- 3 org.eclipse.mylyn.tasks.ui.TaskElementLabelProvider
- 4 org.eclipse.mylyn.tasks.ui.TaskElementLabelProvider[getForeground
- 5 org.eclipse.mylyn.tasks.ui.TaskElementLabelProvider^themeManager

Dataset: Mylyn Project

Collect Fixed
Bug Report

Extract Final
Interaction
History

Extract Code
Elements

Resolve
Repository
Snapshots

Form Code
Context Model

Interaction History 159708

```
<InteractionHistory Version="1" Id="https://bugs.eclipse.org/bugs-303431">
...
<InteractionEvent CreationCount="9" NumEvents="4" StructureKind="java"
StructureHandle="org.eclipse.mylyn.commons.ui/src&lt;org.eclipse.mylyn.internal.
provisional.commons.ui{CommonThemes.java[CommonThemes^COLOR_TASK_A
CTIVE" StartDate="2010-02-21 11:36:05.55 CET"
OriginId="org.eclipse.jdt.ui.PackageExplorer" Navigation="null" Kind="selection"
Interest="4.0" EndDate="2010-02-21 15:21:48.606 CET" Delta="null"/>
...
</InteractionHistory>
```



Event Timestamps

First event 2010-02-21 11:35:53
Last event 2010-02-21 15:23:27



Repository Snapshots

mylyn.commons @Comit 3ead864c 2010-02-21 10:43:34
mylyn.tasks @Commit 78457484 2010-02-21 10:43:32

Dataset: Mylyn Project

Collect Fixed
Bug Report

Extract Final
Interaction
History

Extract Code
Elements

Resolve
Repository
Snapshots

Form Code
Context Model

Repository Snapshots

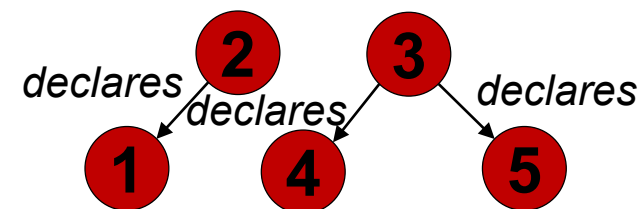
mylyn.common @Comit 3ead864c 2010-02-21 10:43:34
mylyn.tasks @Commit 78457484 2010-02-21 10:43:32

Run *Doxygen*

Code Elements

- 1 org.eclipse.mylyn.internal.provisional.common.ui.
CommonThemes^COLOR_TASK_ACTIVE
- 2 org.eclipse.mylyn.internal.provisional.common.ui.
CommonThemes
- 3 org.eclipse.mylyn.tasks.ui.TaskElementLabelProvider
- 4 org.eclipse.mylyn.tasks.ui.TaskElementLabelProvider
[getForeground]
- 5 org.eclipse.mylyn.tasks.ui.TaskElementLabelProvider
^themeManager

Code Context Model



Dataset: Mylyn Project

Collect Fixed

Repository Snapshots

1,887 code context models.

On average, each context model contains:

12 nodes, **4.6** connected components

Diameter of connected components **0.9**

Repository
Snapshots

Form Code
Context Model

2 org.eclipse.mylyn.internal.provisional.common.ui.
CommonThemes
3 org.eclipse.mylyn.tasks.ui.TaskElementLabelProvider
4 org.eclipse.mylyn.tasks.ui.TaskElementLabelProvider
[getForeground]



Observation in Dataset



Class

4.8 times



Method

1.9 times



Field

1.2 times

**Low rates of access to concrete code elements.
We must abstract from the specific code elements accessed.**

Approach: Stereotype

Stereotype: the role an code element plays in the system.

Method Elements: 17 stereotypes, **4** categories

Structural Accessor, Structural Mutator (e.g., Command), Creational, Collaborational

Class Elements: 13 stereotypes

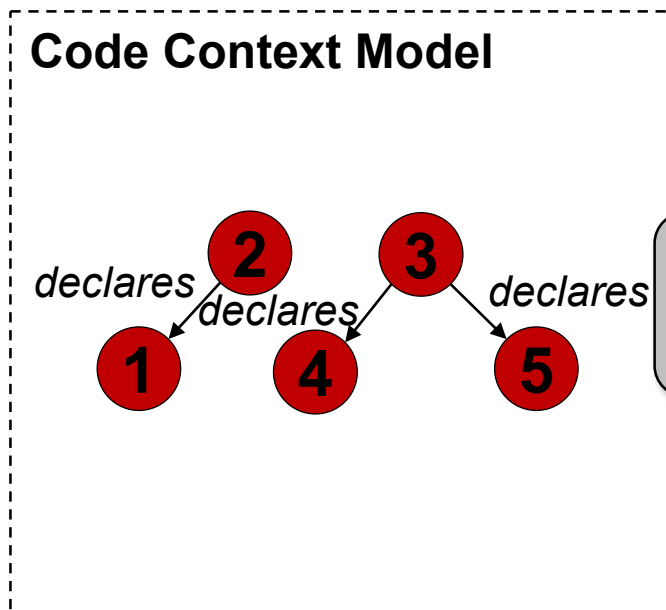
e.g., Data Provider, Pure Controller

Laura Moreno and Andrian Marcus. "*JStereoCode: Automatically Identifying Method and Class Stereotypes in Java Code*". **ASE 2012**.

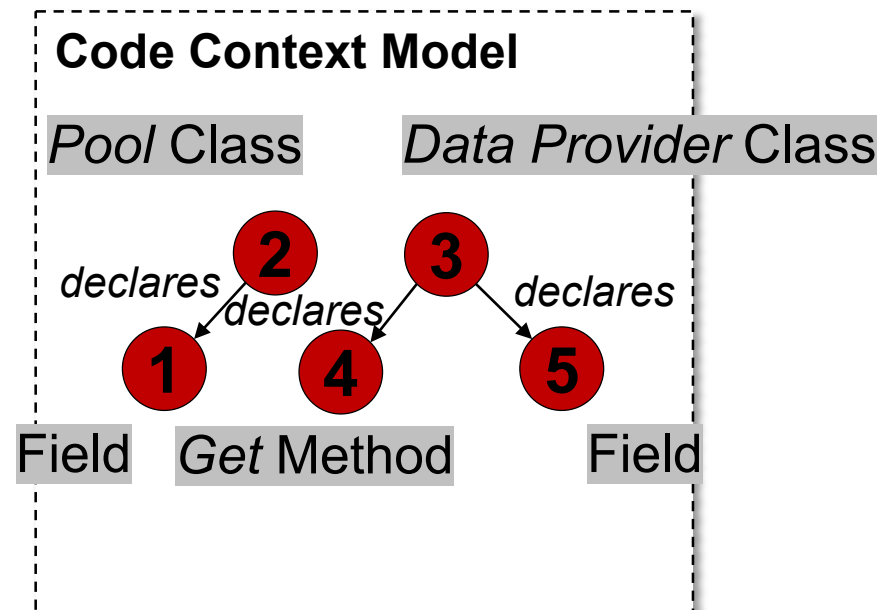
Approach: Stereotype Assignment

Code Elements

```
1 org.eclipse.mylyn.internal.provisional.commons.ui.CommonThemes^COLOR_TASK_ACTIVE
2 org.eclipse.mylyn.internal.provisional.commons.ui.CommonThemes
3 org.eclipse.mylyn.tasks.ui.TaskElementLabelProvider
4 org.eclipse.mylyn.tasks.ui.TaskElementLabelProvider[getForeground
5 org.eclipse.mylyn.tasks.ui.TaskElementLabelProvider^themeManager
```

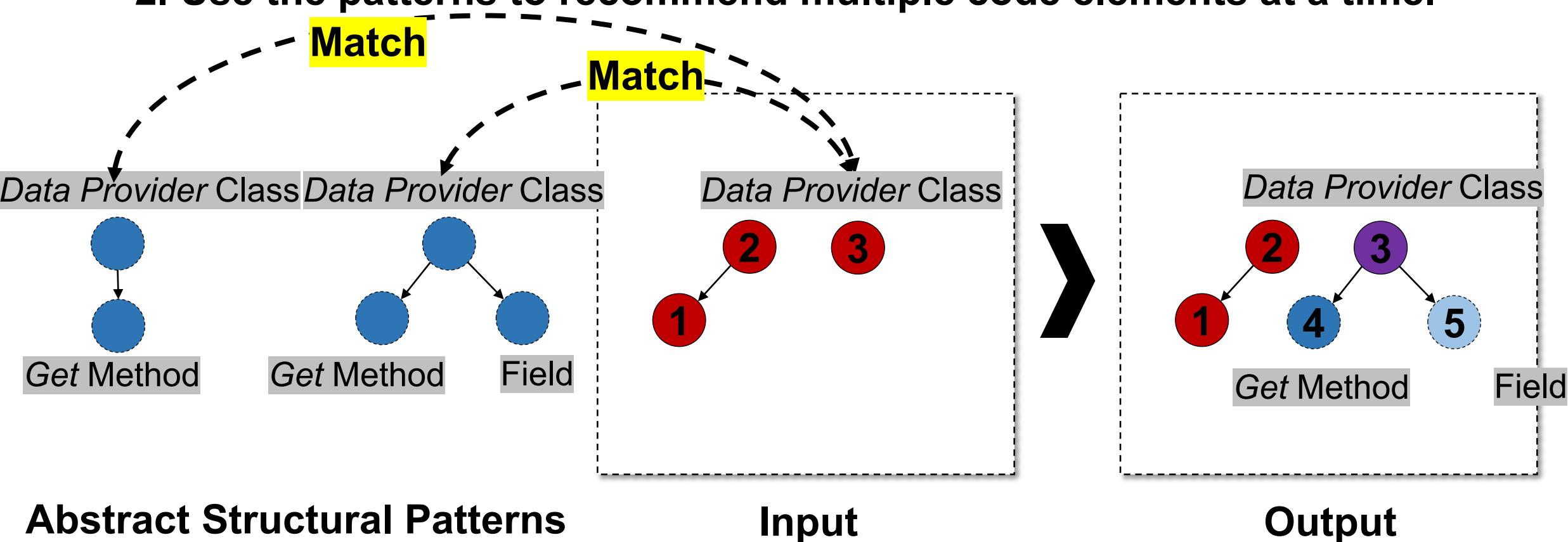


Run
JStereoCode



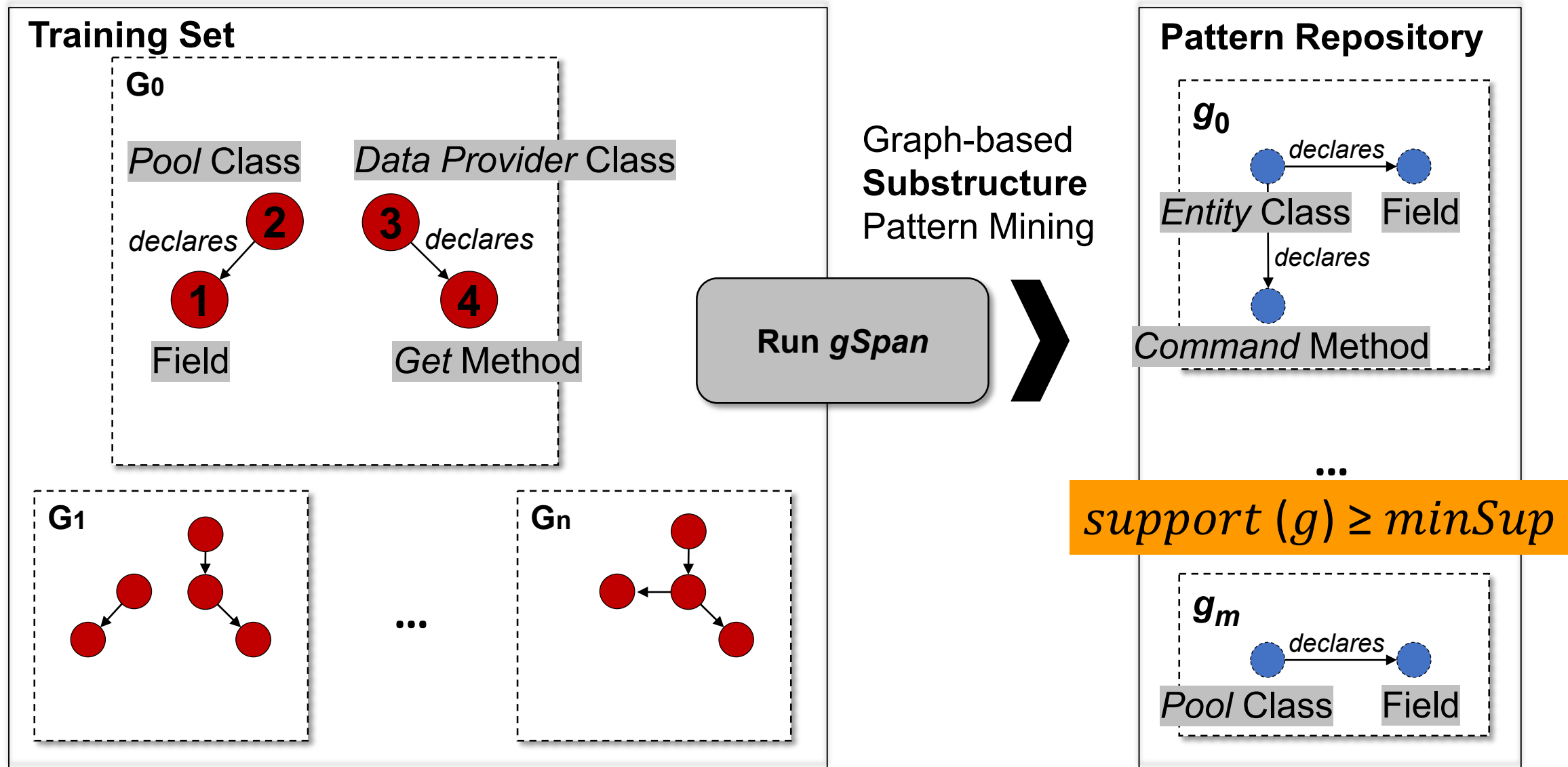
Overview of Our Approach

1. Learn **abstract** structural patterns from historical code context models.
2. Use the patterns to recommend multiple code elements at a time.

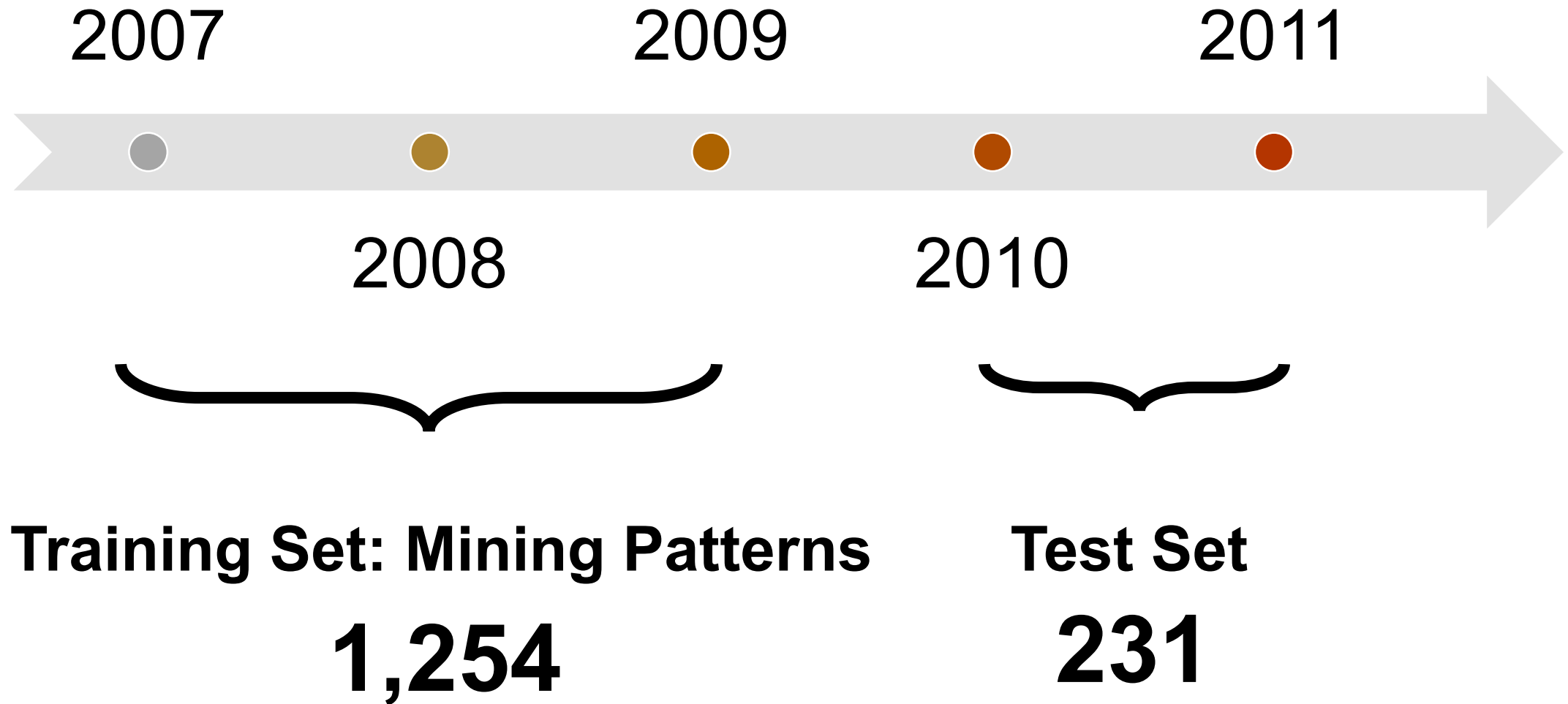


Predicted code elements “more frequently matched, higher confidence”

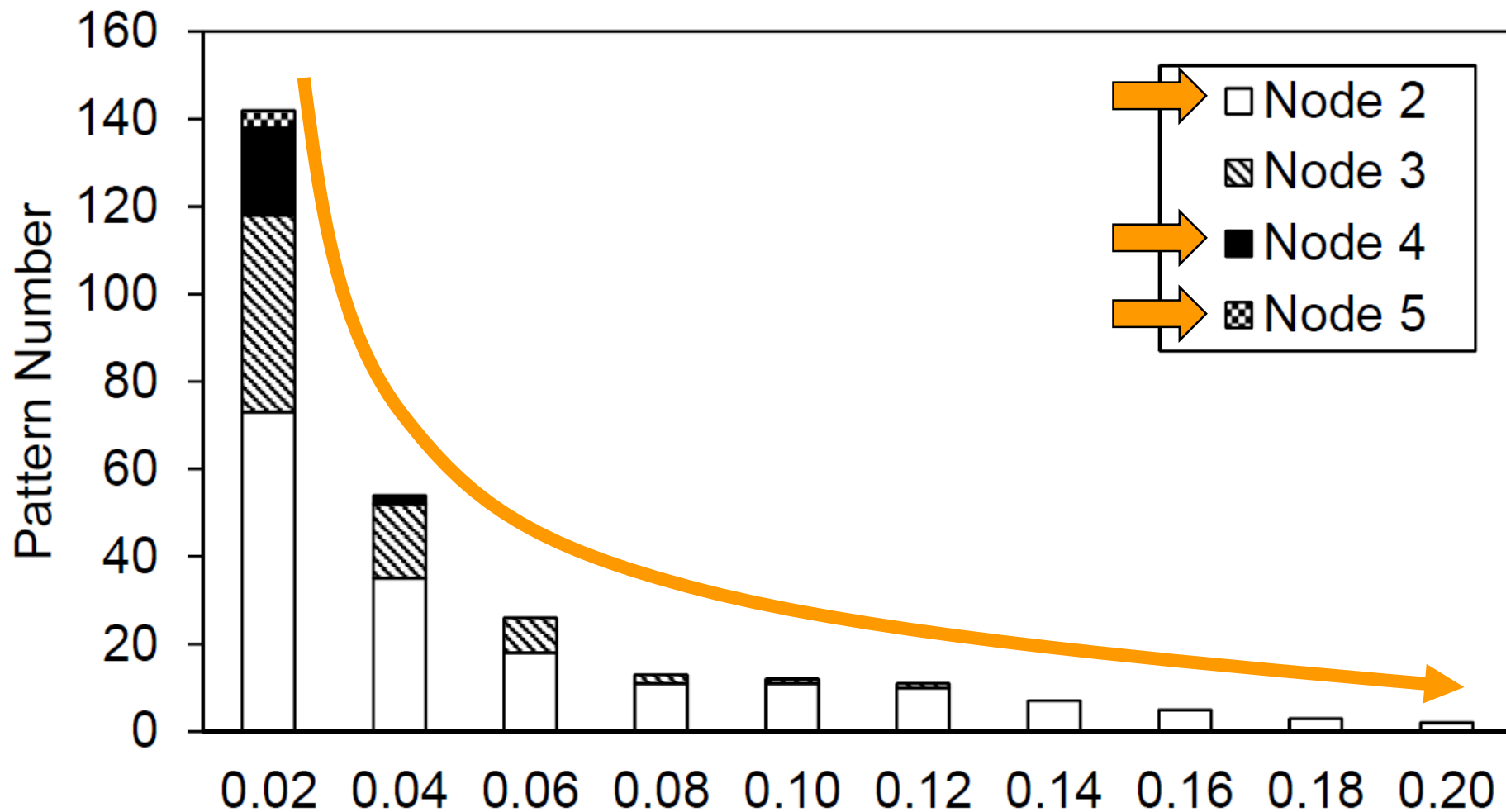
Learn Abstract Structural Patterns



Experimental Setup



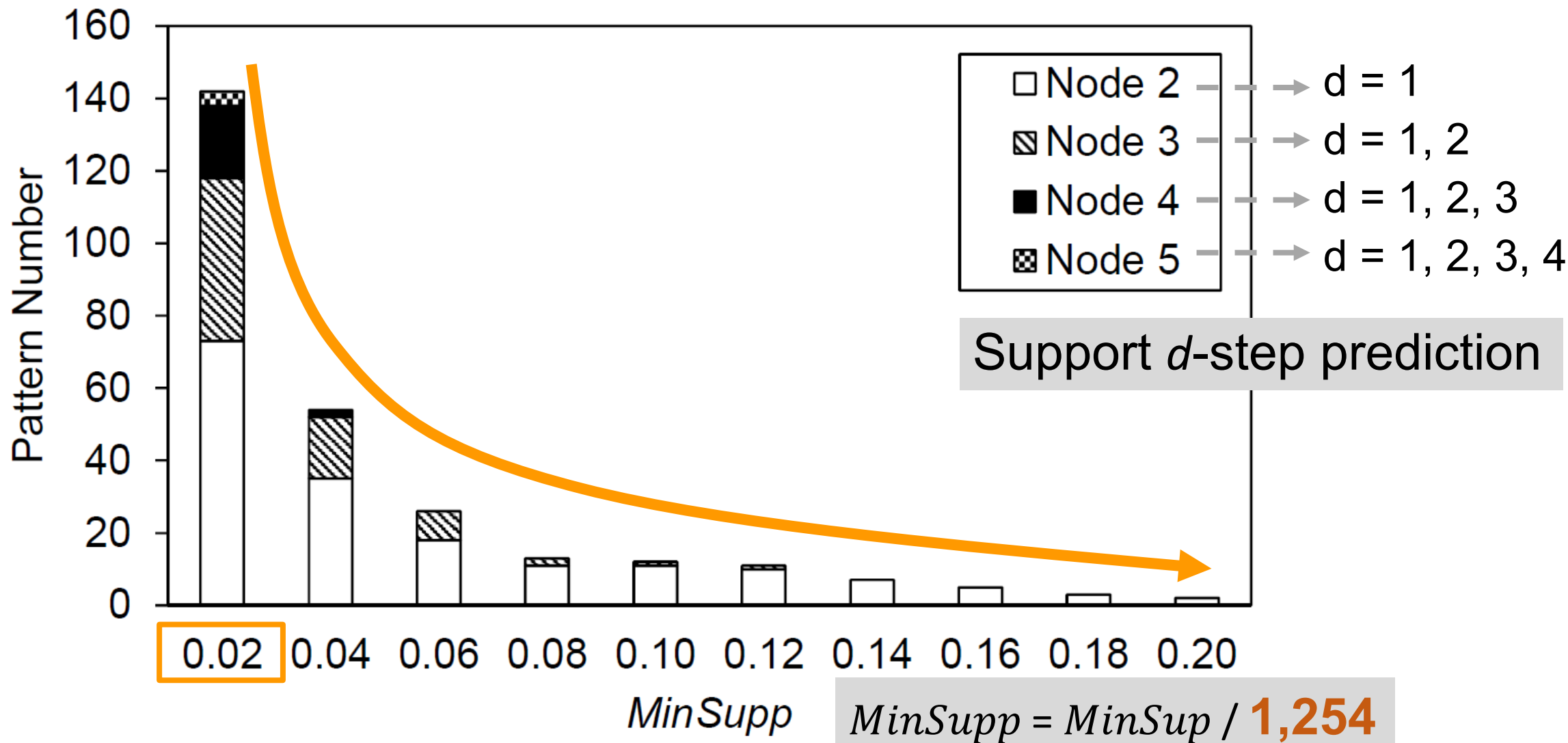
RQ1 Results: Abstract Topological Patterns



MinSupp

MinSupp = *MinSup* / 1,254

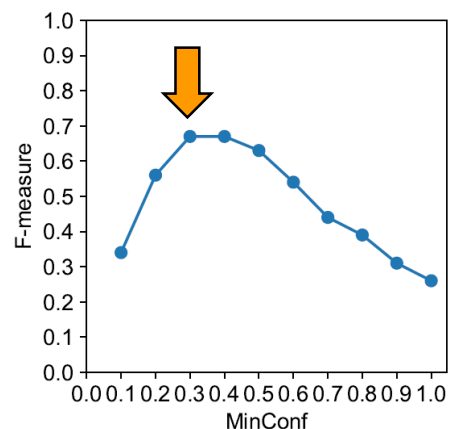
RQ1 Results: Abstract Topological Patterns



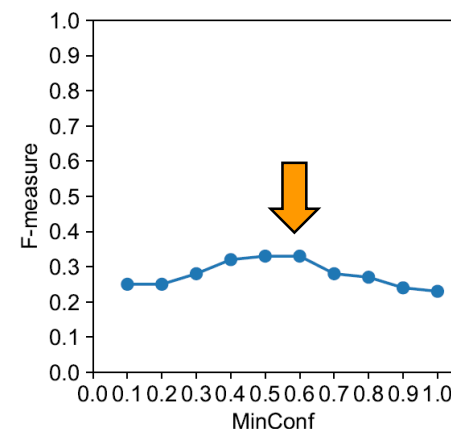
RQ2 Results: Prediction Performance

F-measure

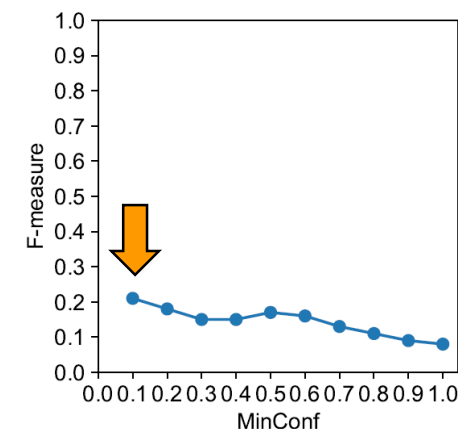
1-step prediction significantly outperforms 2-step and 3-step predictions.



(a) 1-step prediction



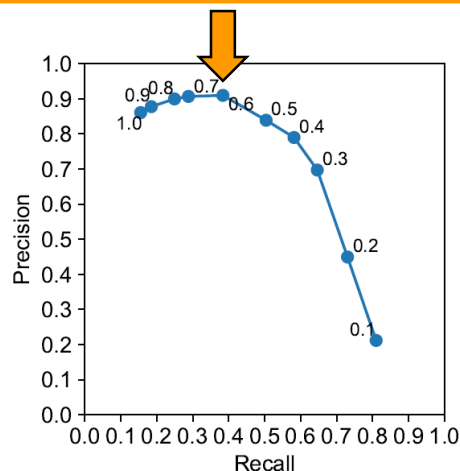
(b) 2-step prediction



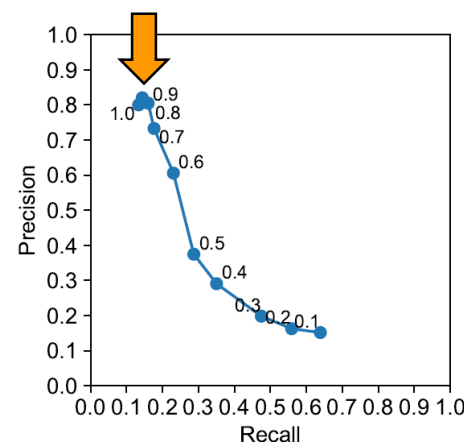
(c) 3-step prediction

Precision and Recall Graph

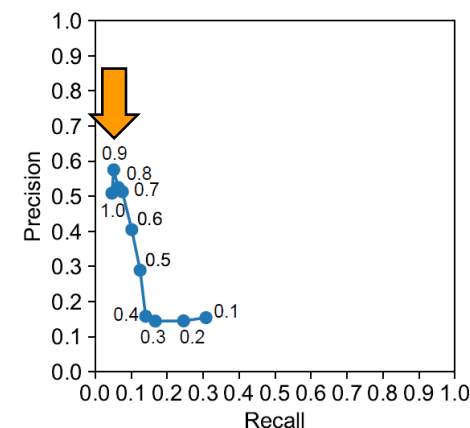
MinConf can be used to make a tradeoff between precision and recall for the predictions.



(a) 1-step prediction



(b) 2-step prediction

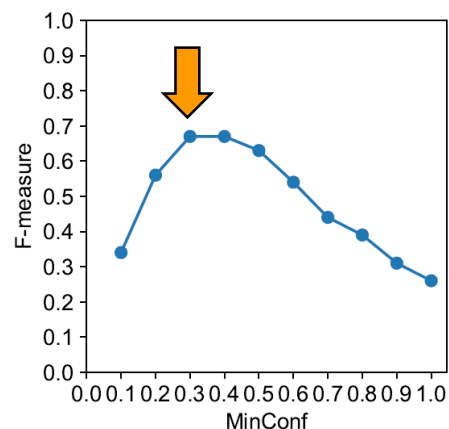


(c) 3-step prediction

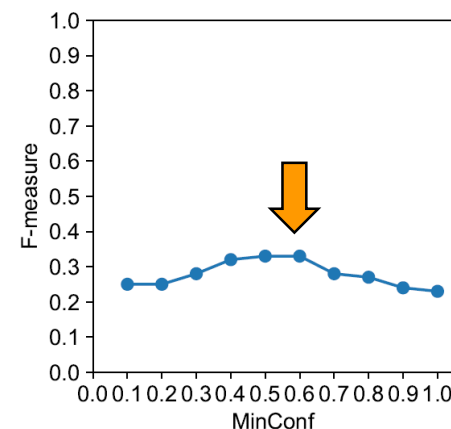
RQ2 Results: Prediction Performance

F-measure

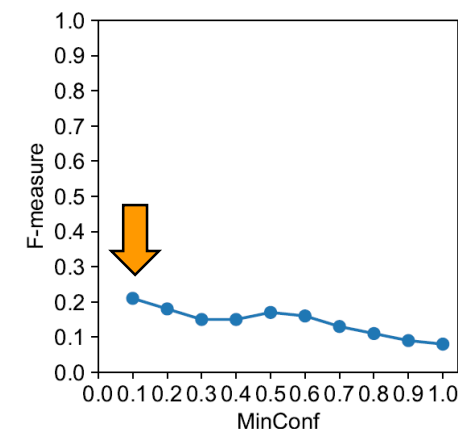
1-step prediction significantly outperforms 2-step and 3-step predictions.



(a) 1-step prediction



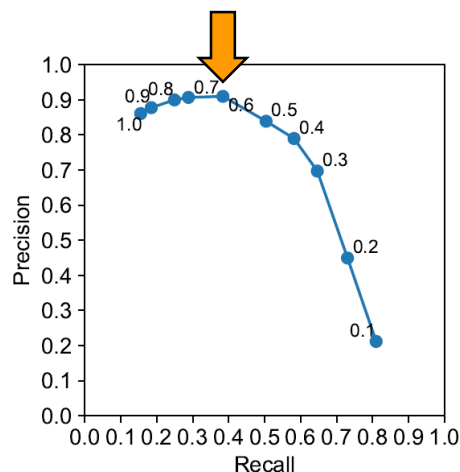
(b) 2-step prediction



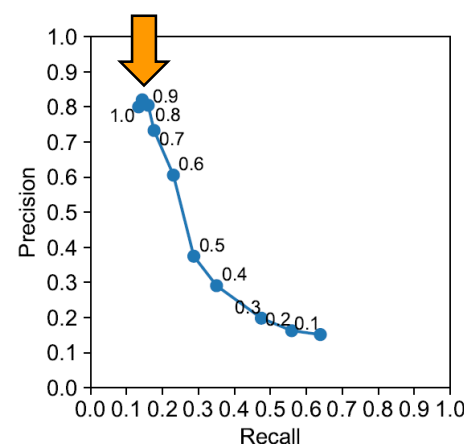
(c) 3-step prediction

Precision and Recall Graph

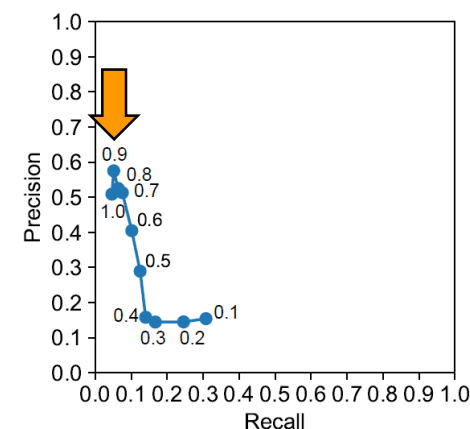
MinConf can be used to make a tradeoff between precision and recall for the predictions.



(a) 1-step prediction



(b) 2-step prediction



(c) 3-step prediction

RQ3 Results: Our Approach vs. *Suade*

	Our Approach			Suade		
	P	R	F	P	R	F
$k = 1$	0.86	0.53	0.65	0.23	0.17	0.20
$k = 3$	0.50	0.69	0.58	0.20	0.32	0.25
$k = 5$	0.36	0.74	0.49	0.18	0.42	0.25

k : window size

F-measure 0.57 over 0.23 on average

Predicting Code Context Models for Software Development Tasks



Zhiyuan Wan



Gail Murphy

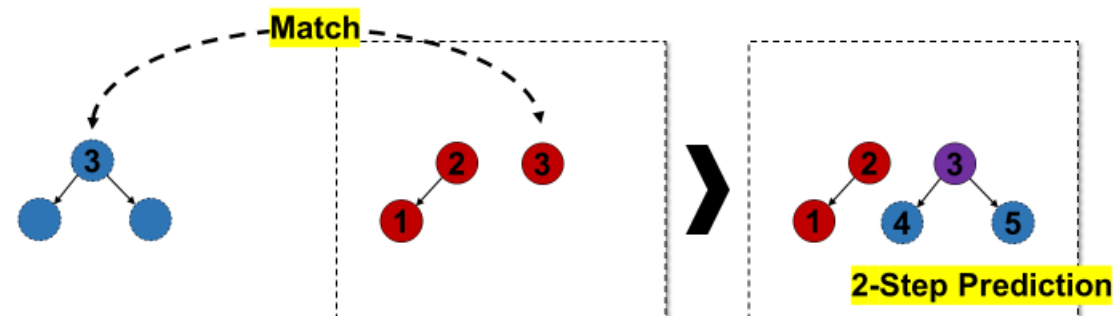


Xin Xia



Overview of Our Approach

1. Learn structural patterns from historical code context models.
2. Use structural patterns to recommend multiple code elements at a time.



Thank you!

For more details, please refer to our preprint at: <https://zhiyuan-wan.github.io>

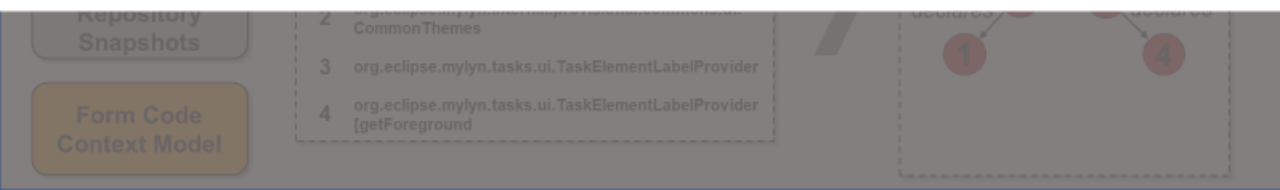
Contact: wanzhiyuan@zju.edu.cn

1,887 code context models.

On average, each context model contains:

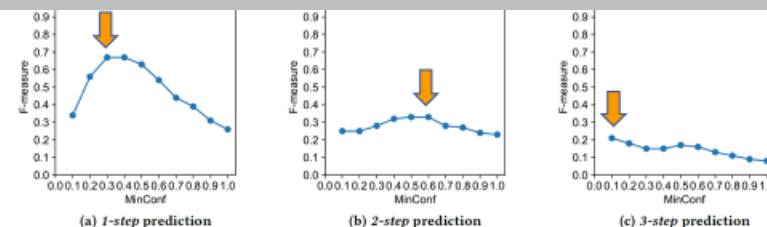
12 nodes, **4.6** connected components

Diameter of connected components **0.9**



F-measures

1-step prediction significantly outperforms 2-step and 3-step predictions.



Precision and Recall Graph

MinConf can be used to make a tradeoff between precision and recall for the predictions.

