

Locus: Locating Bugs from Software Changes



Ming Wen



Rongxin Wu



Shing-Chi Cheung

{mwena, wurongxin, scc}@cse.ust.hk

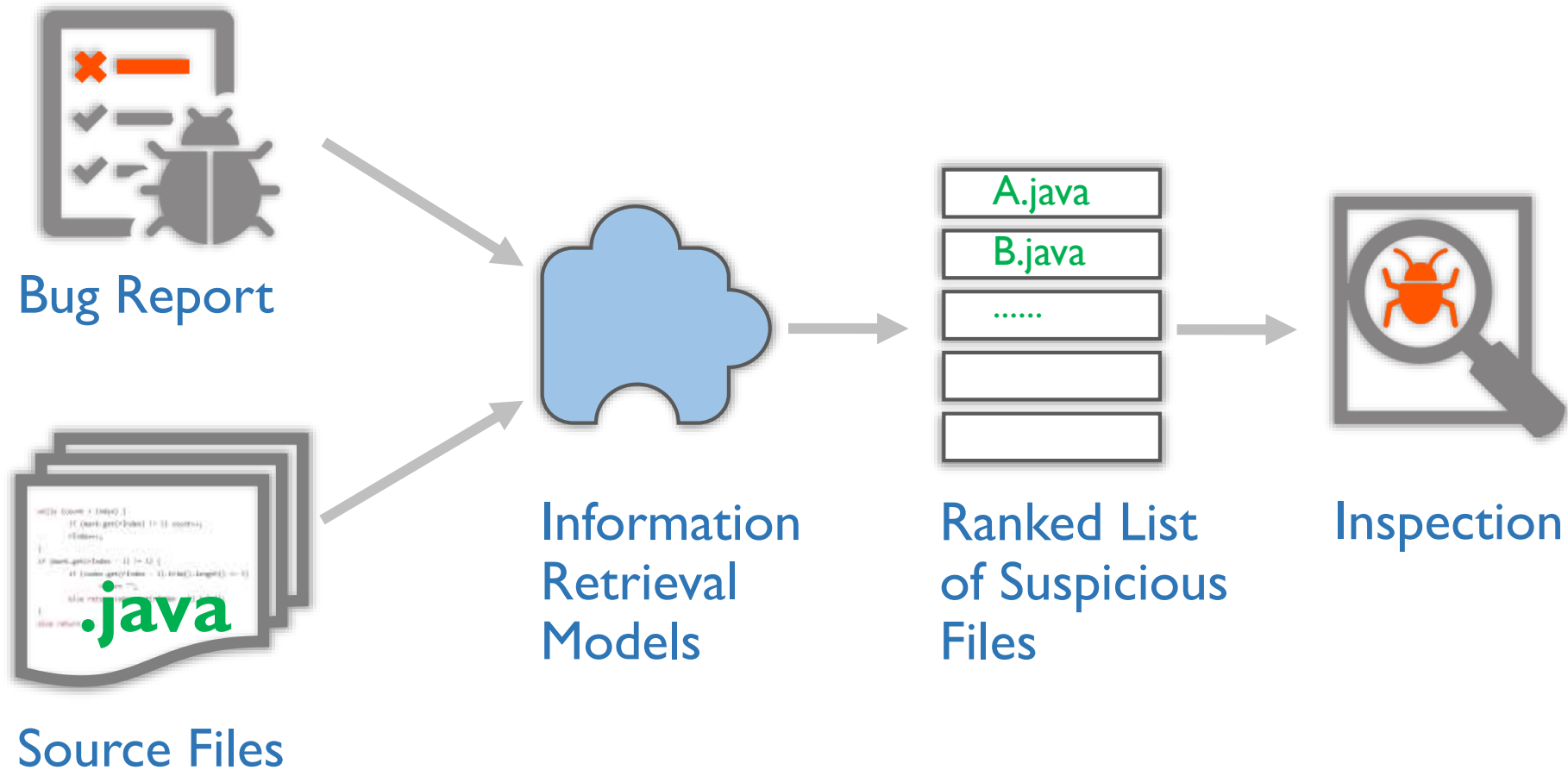


ASE 2016, 6th Sept, Singapore

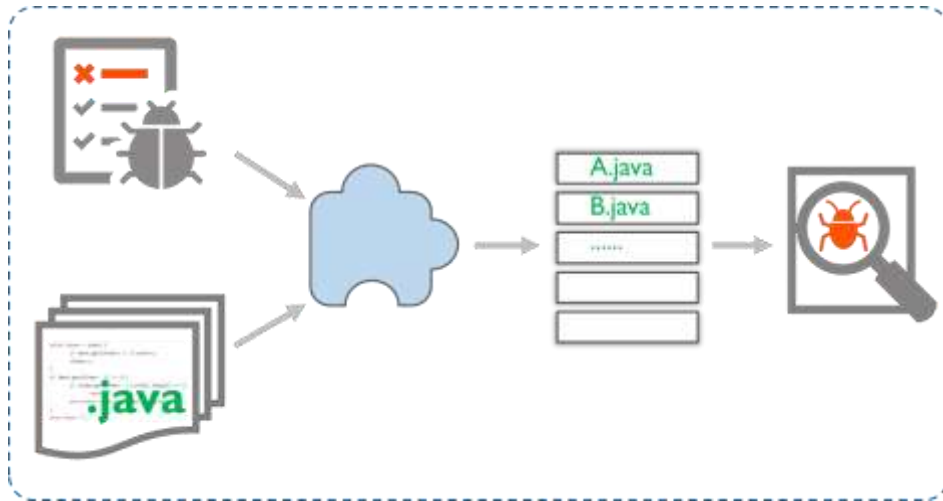
Debugging is Painful!!



IR-Based Fault Localization



Limitations



➤ **Limit#1 Granularity**
Source file results are coarse-grained [Wang et al. ISSTA'15].

➤ **Limit#2 Context**
Lack of contextual clues [Parnin et al. ISSTA'11].

Comments of Bug Report

Konstantin Kolinko 2014-09-01 09:15:22 UTC

[Comment 1](#)

Created [attachment 31958](#) [\[details\]](#)
2014-09-01_tc8_56905_v1.patch

Patch (v1) for current trunk at [r1621698](#)
I have not tested it yet.

Mark Thomas 2014-09-03 13:41:40 UTC

[Comment 2](#)

Patch looks sensible to me. Applied to 8.0.x for 8.0.13 onwards and to 7.0.x got 7.0.56 onwards.

Réda Housni Alaoui 2016-05-03 20:55:14 UTC

[Comment 3](#)

Hello,

Running on tomcat 8.0.33, we are having this issue.

Here is the trace:

```
03-May-2016 22:09:13.405 WARNING [http-nio-8002-exec-150]
org.apache.tomcat.websocket.server.WsServerContainer.destroy Unable to destroy
WebSocket thread group [WebSocketServer-localhost-/0
*****] as [1] threads were still running when the web application was stopped.
The thread group will be destroyed once the threads terminate.
```

The application that we tried to stop does not restart.
I will add the thread dump in attachment asap.

Mark Thomas 2016-05-04 13:07:28 UTC

[Comment 4](#)

(In reply to Réda Housni Alaoui from [comment #3](#))

Comments of Bug Report

Reverting to
the change
that induced
the bug.

“This regression was caused by Bugzilla 257440.
That **patch** was **reverted** and this is now fixed. ¹”

“Ouch, that's my fault. . . .We should **revert**
revision 484642 ASAP. ²”

¹https://bugs.eclipse.org/bugs/show_bug.cgi?id=262975

²https://bz.apache.org/bugzilla/show_bug.cgi?id=41551

Developers' Feedback



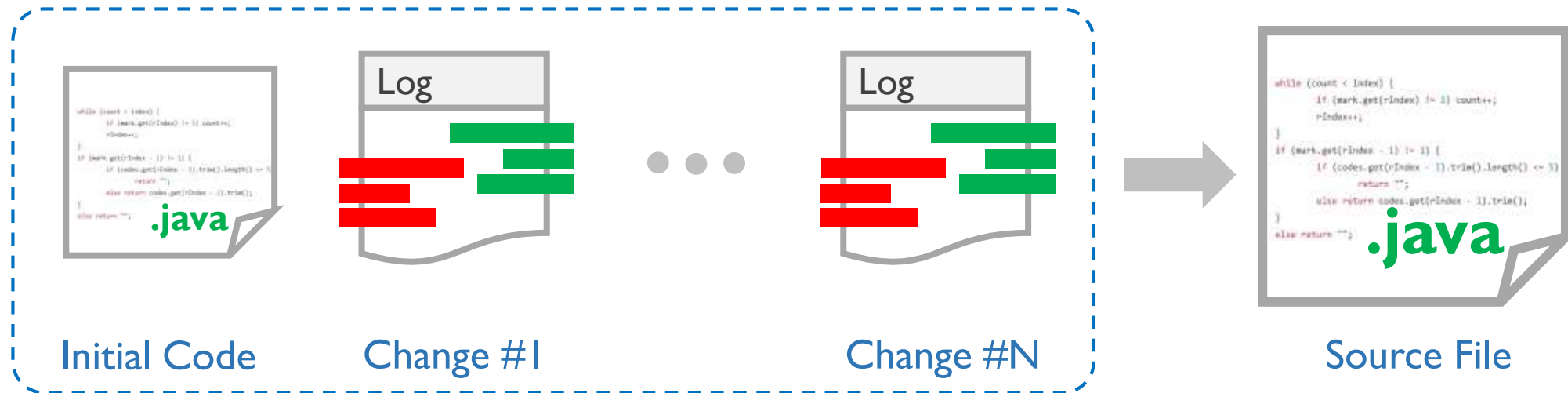
Is the information of inducing changes **useful**?

“It can be a eureka moment for the developer, where they see the inducing change and say ‘I know exactly **why this is happening**’, thus resulting in a fix typically in a matter of days, even hours. It really is a **critical piece of the puzzle**.”

What **actions** will be **taken** if inducing change is available?

“The action taken is usually getting the responsible developer to either **A) back out the change** or **B) code and land a follow-up fix** as soon as possible.”

Bug Inducing Change



- Changes are committed to fix bugs, introduce new features or refactor.
- Changes can induce new bugs, and those changes are regarded as **bug-inducing changes**.

Change A(#2653cea) of Tomcat WsServerContainer.java

Commit #2653cea

Author : Mark Emlyn David Thomas <markt@apache.org>

Date : Tue Apr 22 08:31:56 2014 +0000

Log: Refactor server container shutdown into the destroy method. Destroy the thread group on shutdown. Log a warning if the thread group can't be destroyed

```
@@ -270,6 +275,21 @@ public void addEndpoint(Class<?> pojo)
+     shutdownExecutor();
+     super.destroy();
+     try {
+         threadGroup.destroy();
+     } catch (IllegalThreadStateException itse) {
...
    boolean areEndpointsRegistered() {
        return endpointsRegistered;
    }

@@ -550,11 +563,18 @@ public void toString()
```

Change #2653cea of Tomcat

WsServerContainer.java

Commit #2653cea

Author : Mark Emlyn David Thomas <markt@apache.org>

Date : Tue Apr 22 08:31:56 2014 +0000

Log: Refactor server container shutdown into the destroy method. Destroy the thread group on shutdown. Log a warning if the thread group can't be destroyed

```
@@ -270,6 +275,21 @@ public void addEndpoint(Class<?> pojo)
+     shutdownExecutor();
+     super.destroy();
+     try {
+         threadGroup.destroy();
+     } catch (IllegalThreadStateException itse) {
+ ...
    boolean areEndpointsRegistered() {
        return endpointsRegistered;
    }

@@ -550,11 +563,18 @@ public void toString()
```

Developer Log Message

Change #2653cea of Tomcat

WsServerContainer.java

```
Commit #2653cea
Author : Mark Emlyn David Thomas <markt@apache.org>
Date : Tue Apr 22 08:31:56 2014 +0000
```

Log: Refactor server container shutdown into the destroy method. Destroy the thread group on shutdown. Log a warning if the thread group can't be destroyed

```
@@ -270,6 +275,21 @@ public void addEndpoint(Class<?> pojo)
+     shutdownExecutor();
+     super.destroy();
+     try {
+         threadGroup.destroy();
+     } catch (IllegalThreadStateException itse) {
...
    boolean areEndpointsRegistered() {
        return endpointsRegistered;
    }
}
```

```
@@ -550,11 +563,18 @@ public void toString()
```

Hunk

A group of continuous lines that are changed along with contextual unchanged lines. A change may contain multiple hunks.

Bug #56905

Summary: Unable to destroy WebSocket thread group when reloading webapp generally there might be threads that are still running, ..., threadGroup.enumerate() have not returned

Commit #2653cea (inducing change)

Author : **Mark Emlyn David Thomas**

Date : Tue Apr 22 08:31:56 2014 +0000

Log: Refactor server container shutdown into the destroy method. Destroy the thread group on shutdown. Log a warning if the thread group can't be destroyed

```
@@ -270,6 +275,21 @@ public void addEndpoint(Class<?> pojo)
+     shutdownExecutor();
+     super.destroy();
+     try {
+         threadGroup.destroy();
+     } catch (IllegalThreadStateException e) {
+         log.warn("Thread group cannot be destroyed: " + e);
+     }
+ }
...
boolean areEndpointsReady() {
    return endpointsReady;
}
```

Bug-Inducing Change

```
@@ -550,11 +563,18 @@ public void toString()
```

Commit #a027afd (fixing change)

Author : **Mark Emlyn David Thomas**

Date : Wed Sep 03 13:36:43 2014 +0000

Log: Fix https://issues.apache.org/bugzilla/show_bug.cgi?id=56905

```
@@ -273,14 +273,42 @@ public void addEndpoint(Class<?> pojo)
+     int threadCount = threadGroup.activeCount();
+     boolean success = false;
+     try {
-         threadGroup.destroy();
-     } catch (IllegalThreadStateException itse) {
+         log.warn("Thread group cannot be destroyed: " + itse);
+     }
+     while (true) {
+         int oldThreadCount = threadCount;
+         synchronized (threadGroup) {
```

Bug Report #56905 of Tomcat

Change A (#2653cea) WsServerContainer.java

Refactoring and
adding new features

**Bug #56905 is
reported**

Change B (#a027afd) WsServerContainer.java Fixing Change

by the **same developer**
of the inducing change

Changes are Useful

- Reverting Bug Inducing Change

Limit#2 Context

- Facilitating Bug Triaging

Limit#2 Context

- Fine Granularity

Limit#1 Granularity



Inspected many bug reports.



Changes are Useful

- Reverting Bug Inducing Change

Limit#2 Context

- Facilitating Bug Triaging

Limit#2 Context

- Fine Granularity

Limit#1 Granularity

Triaging bugs to the committer of the inducing change.

Subject	#Dev	#Bugs	#Same	Ratio (%)
SWT 3.1	86	97	65	67.00%
JDT 4.5	95	94	67	71.30%
Tomcat 8.0	29	193	167	86.50%

Ratio of Bug Reports Whose Fixing Developer is the Same as the Developer of the Bug Inducing Change

77.86% of the bugs are fixed by the committer of the inducing change.

Changes are Useful

- Reverting Bug Inducing Change

Limit#2 Context

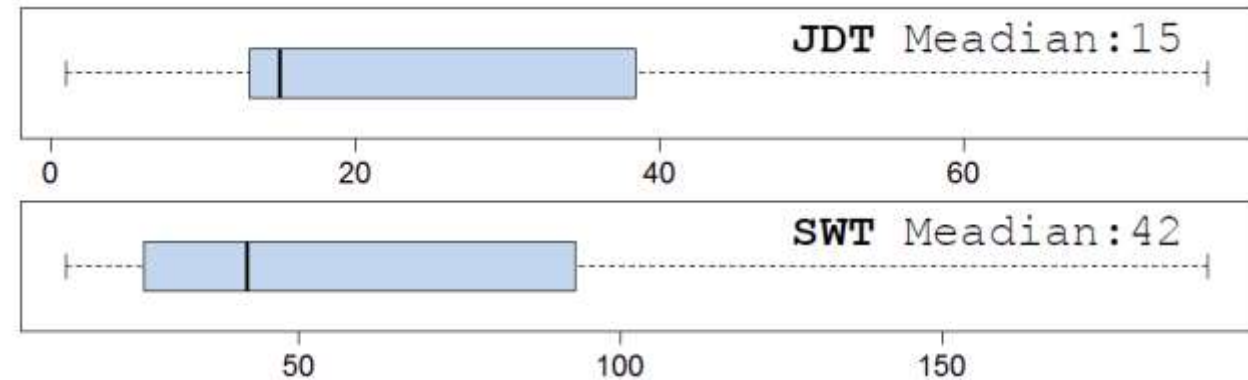
- Facilitating Bug Triaging

Limit#2 Context

- Fine Granularity

Limit#1 Granularity

Triaging bugs to the committer of the inducing change.



Bug Triage Time for JDT and SWT (days)

Many manual efforts are required to triage bugs.

Changes are Useful

- Reverting Bug Inducing Change

Limit#2 Context

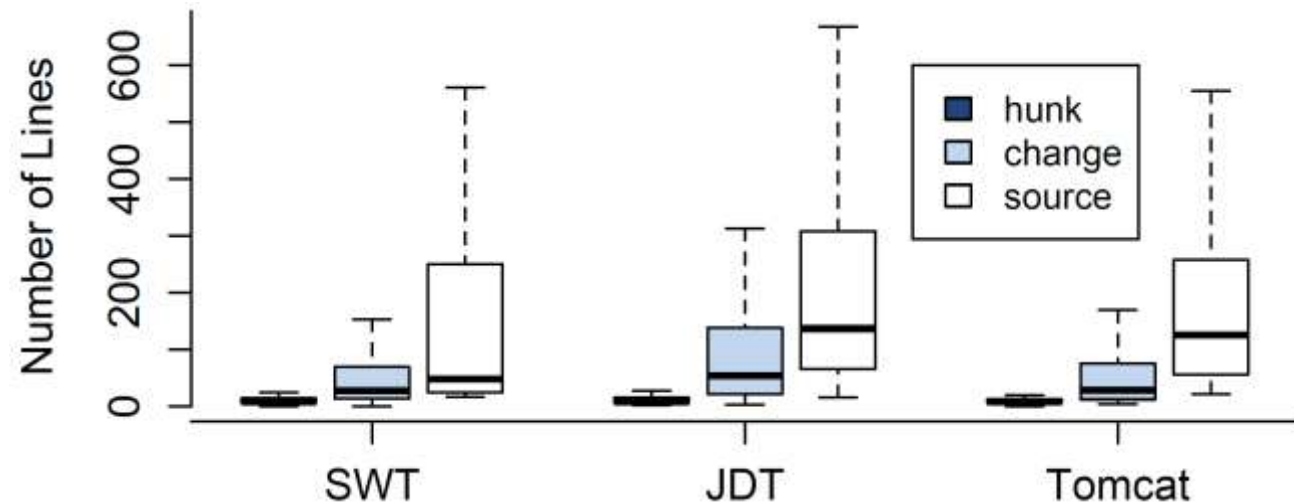
- Facilitating Bug Triaging

Limit#2 Context

- Fine Granularity

Limit#1 Granularity

Debugging at the change level can significantly save effort when compared **with source level** [Kamei et al.TSE'13].



Comparison of Lines of Codes among Hunks, Changes and Source Files

Using Software Changes in IR-Based Fault Localization

```
while (start < end) {  
    if (arrk.get(index) > 1) count++;  
    index++;  
}  
  
if (arrk.get(index - 1) < 1) {  
    if (index.get(index - 1, true).length() > 5)  
        return "";  
    else return codek.get(index - 1, true);  
}  
else return "";
```

.java

Source File



Bug Report



Information Retrieval Models

Using Software Changes in IR-Based Fault Localization



Source File



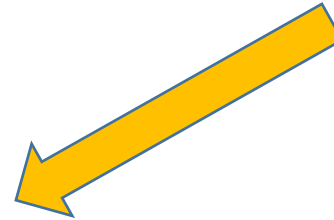
Software Changes



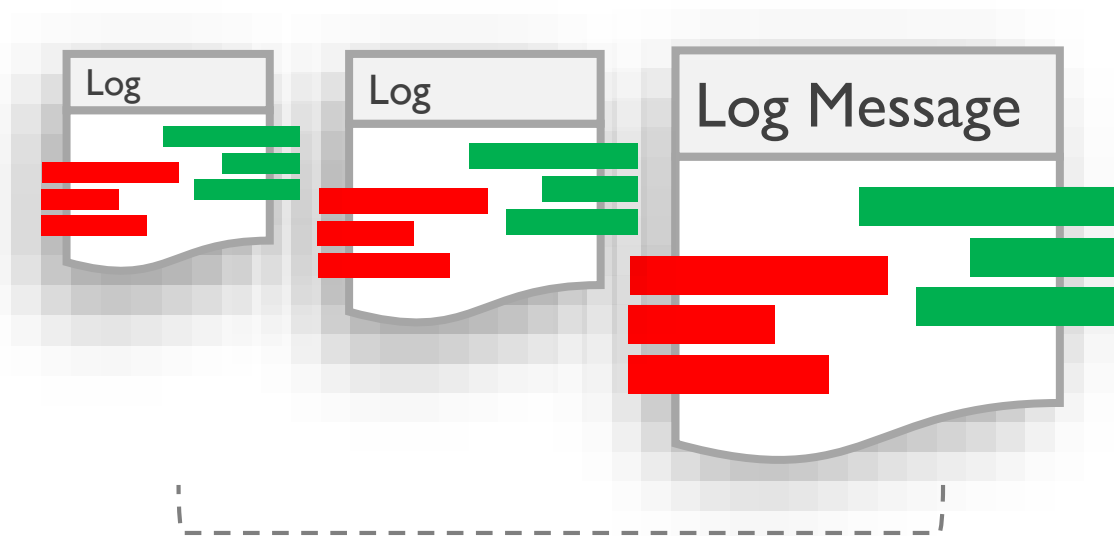
Bug Report



Information Retrieval Models



Benefits of Using Software Changes



3. Change Histories

1. Informative Change Logs
2. Highly Correlated Contents

Bug #56905

Summary: Unable to destroy WebSocket thread group when reloading webapp generally there might be threads that are still running, ..., threadGroup.enumerate() have not returned

Commit #2653cea **(inducing change)**

Author : Mark Emlyn David Thomas <markt@apache.org>

Date : Tue Apr 22 08:31:56 2014 +0000

Log: Refactor server container shutdown into the destroy method. Destroy the thread group on shutdown. Log a warning if the thread group can't be destroyed

```
@@ -270,6 +275,21 @@ public void addEndpoint(Class<?> pojo)
+     shutdownExecutor();
+     super.destroy();
+     try {
+         threadGroup.destroy();
+     } catch (IllegalThreadStateException itse) {
...
    boolean areEndpointsRegistered() {
        return endpointsRegistered;
    }
```

```
@@ -550,11 +563,18 @@ public void toString()
```

Commit #a027afd **(fixing change)**

Author : Mark Emlyn David Thomas <markt@apache.org>

Date : Wed Sep 03 13:36:43 2014 +0000

Log: Fix https://issues.apache.org/bugzilla/show_bug.cgi?id=56905

```
@@ -273,14 +273,42 @@ public void addEndpoint(Class<?> pojo)
+     int threadCount = threadGroup.activeCount();
+     boolean success = false;
+     try {
-         threadGroup.destroy();
-     } catch (IllegalThreadStateException itse) {
+
+     while (true) {
+         int oldThreadCount = threadCount;
+         synchronized (threadGroup) {
```

Bug Report #56905 of Tomcat

Inducing Change
Refactoring and adding
new features
Inducing bug #56905

Fixing Change
by the same developer
of the inducing change

Bug #56905

Summary: Unable to **destroy** WebSocket **thread group** when reloading webapp generally there might be **threads** that are still running,..., **threadGroup.enumerate()** have not

Commit #2653cea **(inducing change)**

Author : Mark Emlyn David Thomas <markt@apache.org>

Date : Tue Apr 22 08:31:56 2014 +0000

Log: Refactor server container shutdown into the **destroy** method. **Destroy** the **thread group** on shutdown. Log a warning if the **thread group** can't be **destroyed**

```
@@ -270,6 +275,21 @@ public void addEndpoint(Class<?> pojo)
+     shutdownExecutor();
+     super.destroy();
+     try {
+         threadGroup.destroy();
+     } catch (IllegalThreadStateException itse) {
+ ...
+     boolean areEndpointsRegistered() {
+         return endpointsRegistered;
+     }
+ }
```

Commit #a027afd **(fixing change)**

Author : Mark Emlyn David Thomas <markt@apache.org>

Date : Wed Sep 03 13:36:43 2014 +0000

Log: Fix https://issues.apache.org/bugzilla/show_bug.cgi?id=56905

```
@@ -273,14 +273,42 @@ public void addEndpoint(Class<?> pojo)
+     int threadCount = threadGroup.activeCount();
+     boolean success = false;
+     try {
+         threadGroup.destroy();
+     } catch (IllegalThreadStateException itse) {
+
+         while (true) {
+             int oldThreadCount = threadCount;
```

Bug Report #56905 of Tomcat

Inducing Change

Refactoring and adding new features

Lots of common tokens are shared between the **bug report** and the **log messages** of the change.

Fixing Change

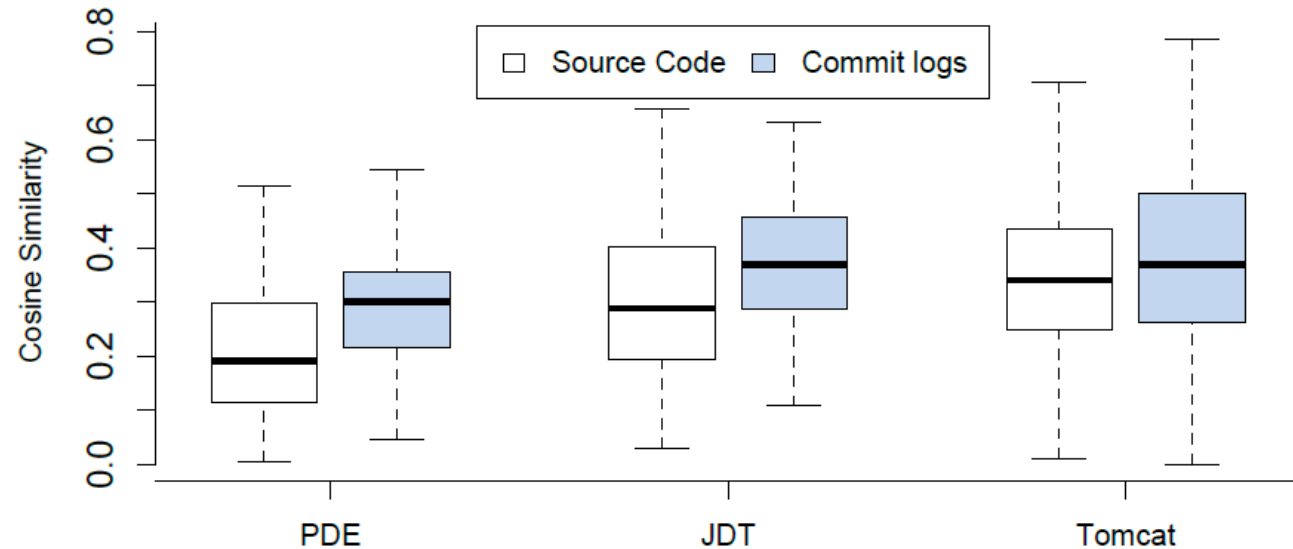
by the same developer of the inducing change

Benefits of Using Software Changes

➤ Informative
Change Logs

➤ Highly Correlated
Contents

➤ Change History



Text Similarities between Bug Reports and the Buggy Files as well as the Change Logs

Change logs share a **substantial** number of common tokens with bug reports.

Benefits of Using Software Changes

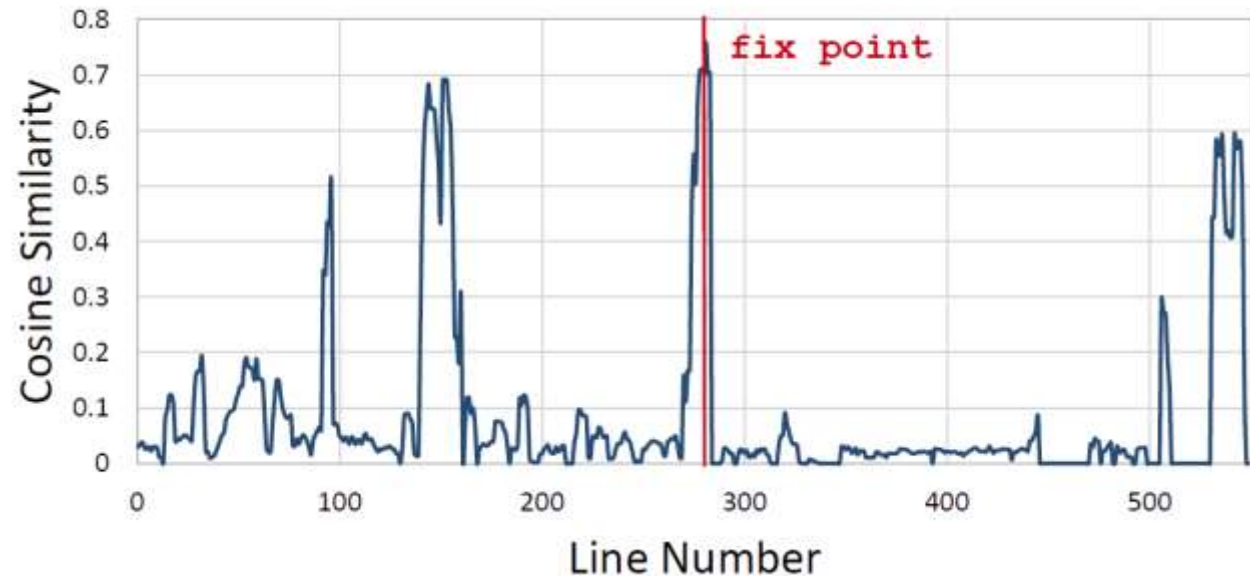
➤ Informative
Change Logs

➤ Highly Correlated
Contents

➤ Change History

- Large source files are susceptible to noise due to the fuzziness arising from information retrieval [Wong et al. ICSME'14, Ye et al. FSE'14].

Bug Report #56905 and its Cosine Similarity of
Source File *WsServerContainer.java*



Benefits of Using Software Changes

➤ Informative
Change Logs

➤ **Highly Correlated
Contents**

➤ Change History

- Segmenting source files into equal-sized segments [[Wong et al. ICSME'14](#)].
- Segmenting source files into methods [[Ye et al. FSE'14](#)].

Benefits of Using Software Changes

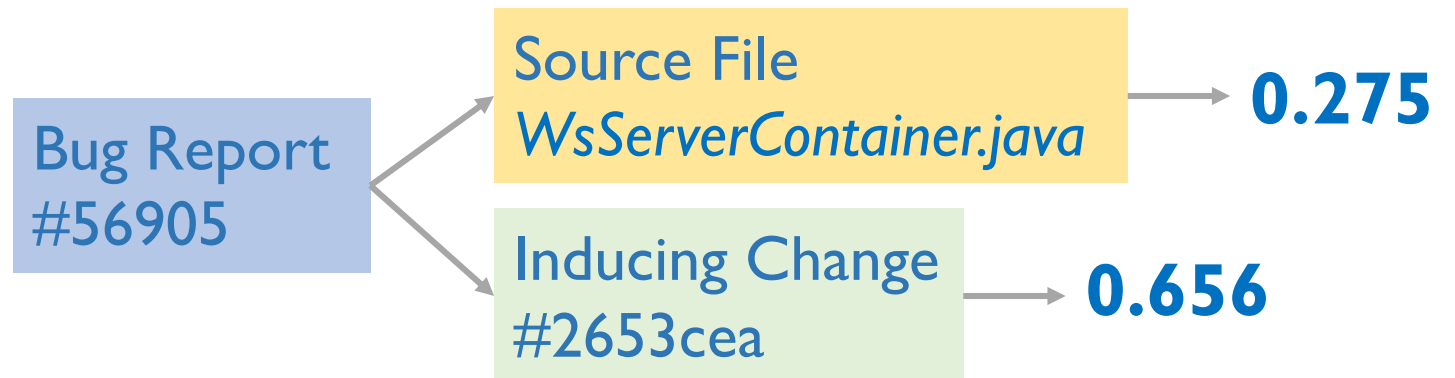
➤ Informative
Change Logs

➤ **Highly Correlated
Contents**

➤ Change History

Highly correlated and small pieces of code are desired in retrieval models.

Change hunks are intrinsically small in size and correlated in contents. [Alali et al. ICPC'08].



Benefits of Using Software Changes

- Informative Change Logs
- Highly Correlated Contents
- Change History

Software changes capture the history of source files (e.g. ownership, fixing history), which indicate the **proneness** of source files to **contain faults**.

- Defect prediction [Moser ICSE'08, Rahman FSE'11]
- Fault Localization [Ye. FSE'14, Wang ICPC'14]
- Google¹

[1] <http://google-engtools.blogspot.hk/2011/12/bug-prediction-at-google.html>. Accessed: 2015-03-22.

Locus

Bug #56905

Summary: Unable to **destroy** WebSocket **thread group** when reloading webapp generally there might be **threads** that are still running, ..., `threadGroup.enumerate()` have not returned

Bug Report: Summary + Description

Commit #2653cea

Log: Refactor server container shutdown into the **destroy** method. **Destroy** the thread **group** on shutdown. Log a warning if the thread group can't be **destroyed**

```
@@ -270,6 +275,21 @@ public void addEndpoint(Class<?> pojo)
+     shutdownExecutor();
+     super.destroy();
+     try {
+         threadGroup.destroy();
+     } catch (IllegalThreadStateException itse) {
+ ...
    boolean areEndpointsRegistered() {
        return endpointsRegistered;
    }
```

Hunk: Log Message + Changed Lines + Contextual Lines

➤ Index **N**atural **L**anguage Tokens (**NL**)

e.g. destroy, web, socket, group, thread,...

Locus

Bug #56905

Summary: Unable to destroy **WebSocket** thread group when reloading webapp generally there might be threads that are still running,..., **threadGroup.enumerate()** have not returned

Bug Report: Summary + Description

Commit #2653cea

Log: Refactor server container shutdown into the destroy method. Destroy the thread group on shutdown. Log a warning if the thread group can't be destroyed

```
@@ -270,6 +275,21 @@ public void addEndpoint(Class<?>
pojo)
+     shutdownExecutor();
+     super.destroy();
+     try {
+         threadGroup.destroy();
+     } catch (IllegalThreadStateException itse)
{
...
    boolean areEndpointsRegistered() {
        return endpointsRegistered;
    }
}
```

Hunk: Log Message + Changed Lines + Contextual Lines

➤ Index **Natural Language Tokens (NL)**

e.g. destroy, web, socket, group, thread,...

➤ Index **Code Entity Names (CE)**

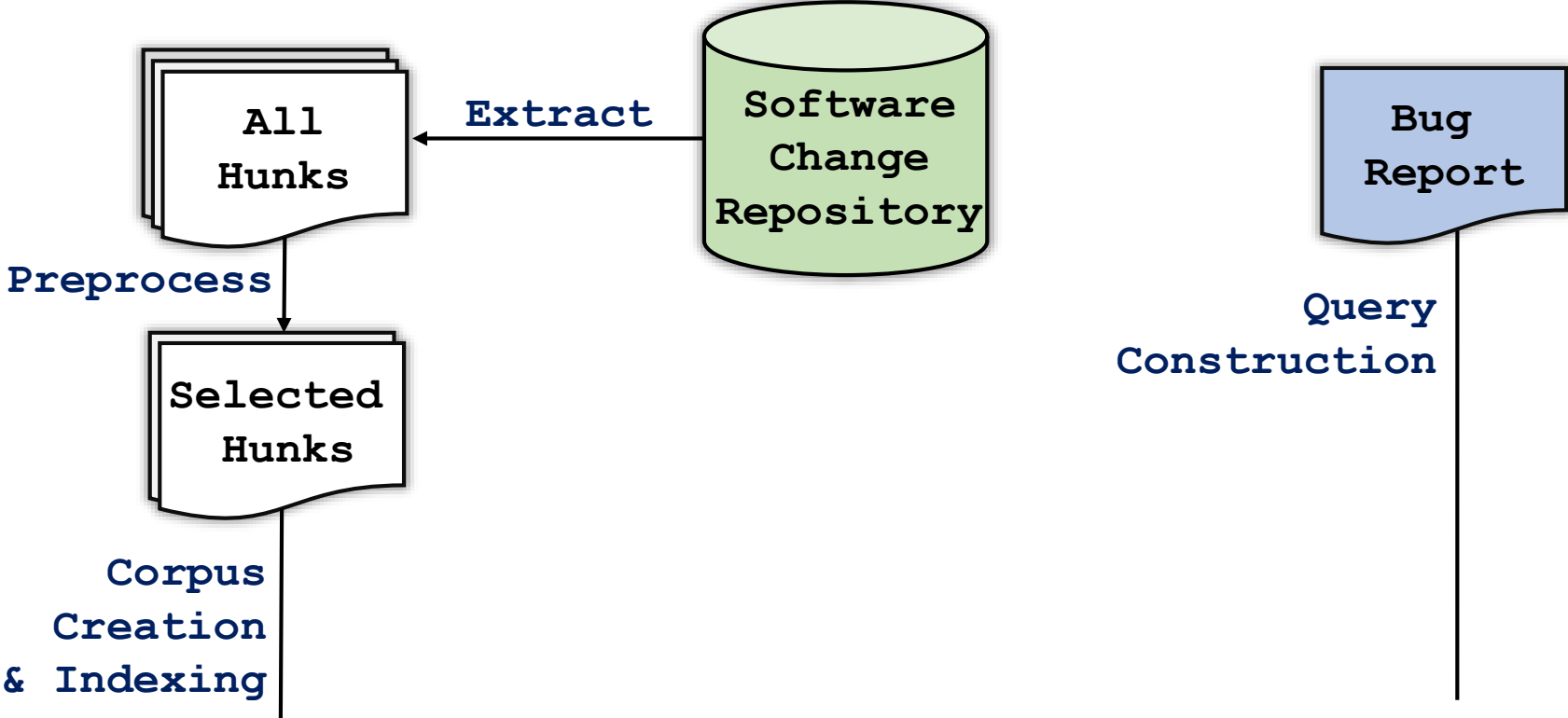
e.g. addEndpoint, threadGroup, ...

➤ Leverage Change History (**Boosting**)

e.g. fixing history, change time

Locus Overview

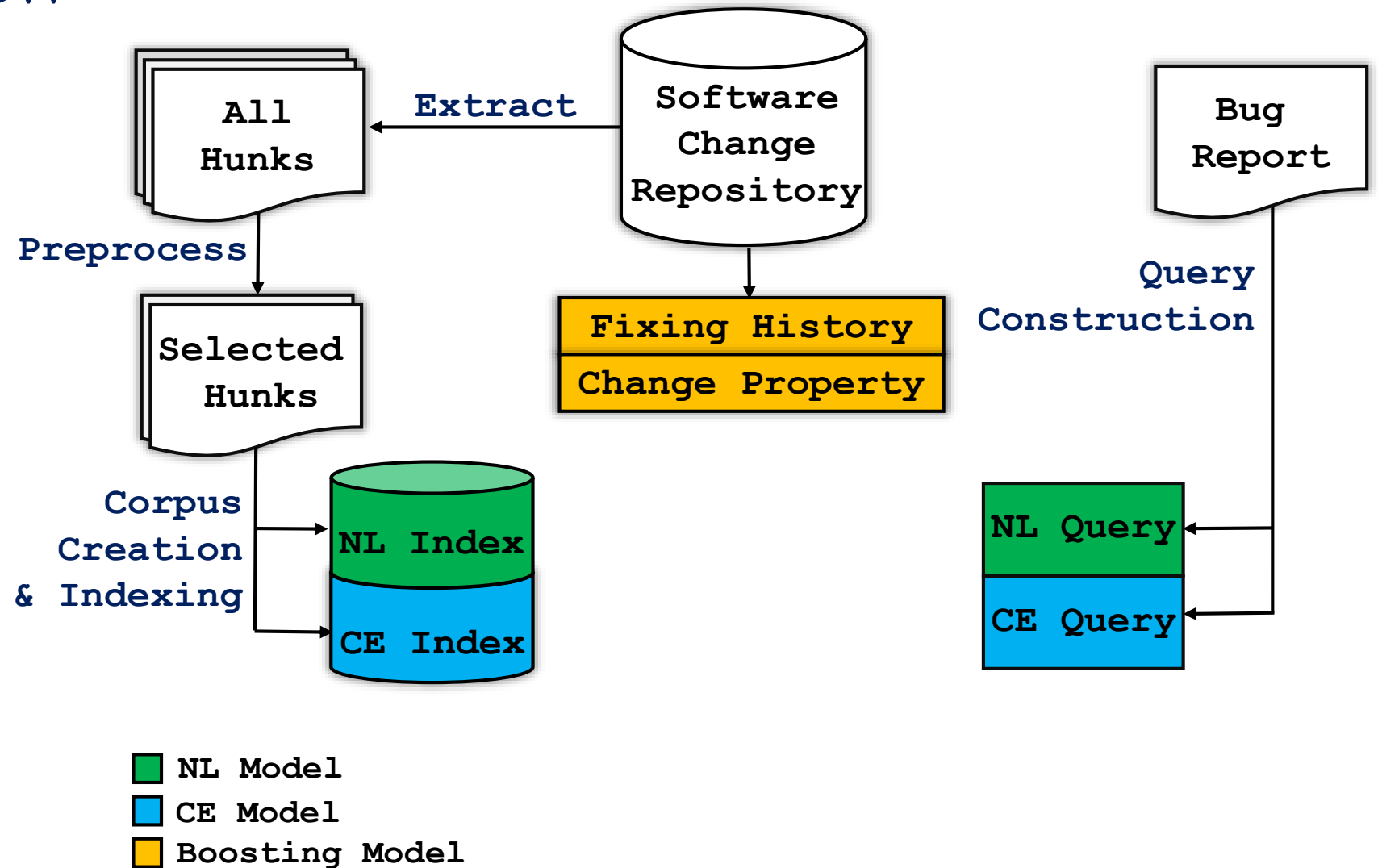
➤ Vector Space Model



Locus Overview

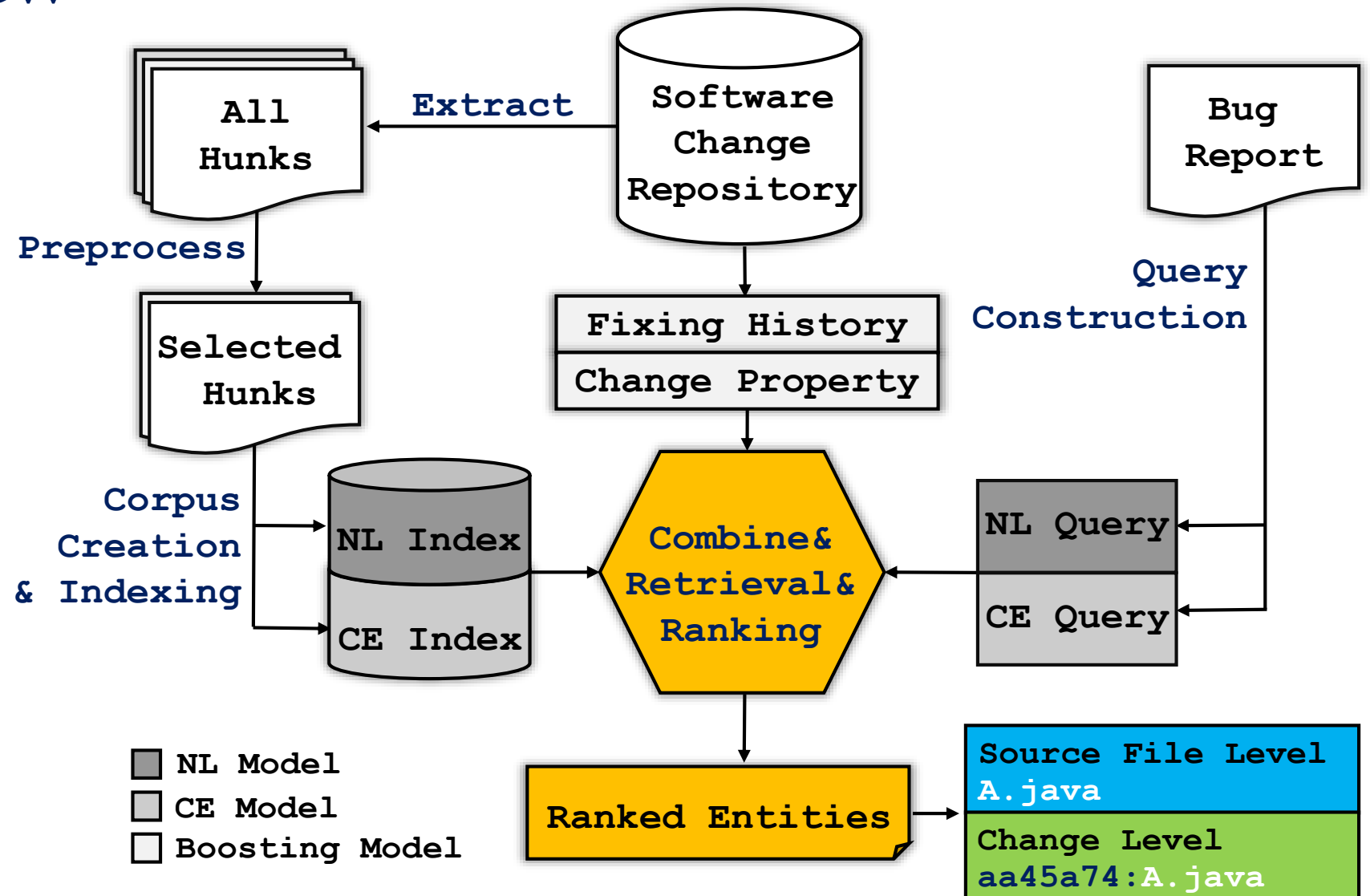
➤ Vector Space Model

➤ Three Models
NL / CE / Boosting



Locus Overview

- Vector Space Model
- Three Models
NL / CE /
Boosting
- Combing Results
Source Level
Change Level



Experiment Setup

➤ Dataset

Subject	Num Bugs	Num Files	K Loc	K Changes
ZXing	20	391	49.6	3.14
SWT 3.1	98	484	141.9	11.9
AspectJ	244	6,485	511.9	7.7
PDE 4.4	60	5,273	565.2	11.3
JDT 4.5	94	6,775	1,675.30	21.7
Tomcat 8.0	193	2,042	485.7	16.1

Benchmark dataset
from BugLocator

Collected by us
All the bugs with valid links
to fix changes



Tomcat



SWT, PDE, JDT, AspectJ



ZXing

Experiment Setup

➤ Evaluation Metrics

➤ Top@N (N = 1, 5, 10...)

The percentage of bugs whose relevant files can be listed in the top N of the ranked list.

➤ (MRR) Mean Reciprocal Rank

How well the first relevant files are ranked, the higher the better.

➤ (MAP) Mean Average Precision

How well all relevant files are ranked, the higher the better

Research Questions

❖ [RQ1] Software Changes VS. Source Files

Can the text tokens extracted from software changes effectively locate bugs in **IR-based techniques**?

❖ [RQ2] Locus VS. Baselines

How is the performance of **Locus** compared with state-of-the-art approaches.

❖ [RQ3] Contributions of Each Model

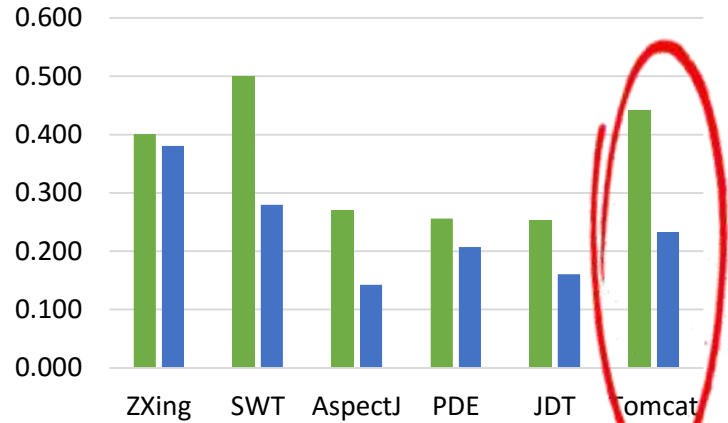
Can each model we proposed improves the final performance?

[RQ I] Software Changes VS. Source Files

- Compare the localization results using the text tokens extracted from **software changes** with those from **source files**.
- Keeping only the natural language tokens.
- Keeping only the tokens of code entity names.
- Using both of them.

[RQ I] Software Changes VS. Source Files

NL MAP

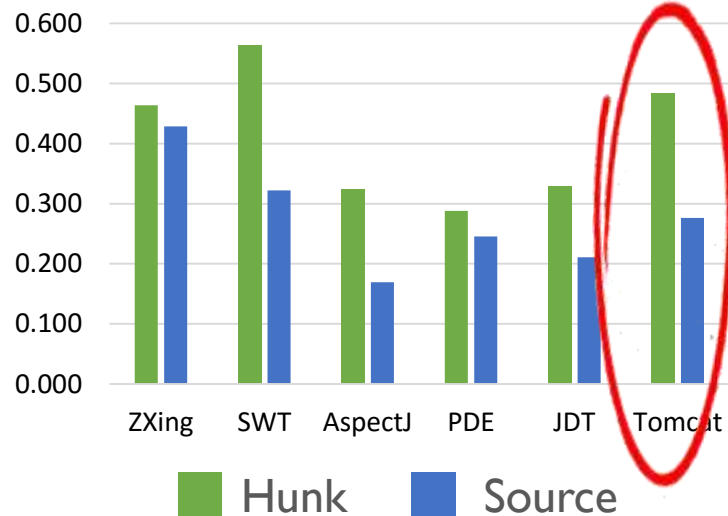


Tomcat

The **MAP** has been improved by **89.43%**.

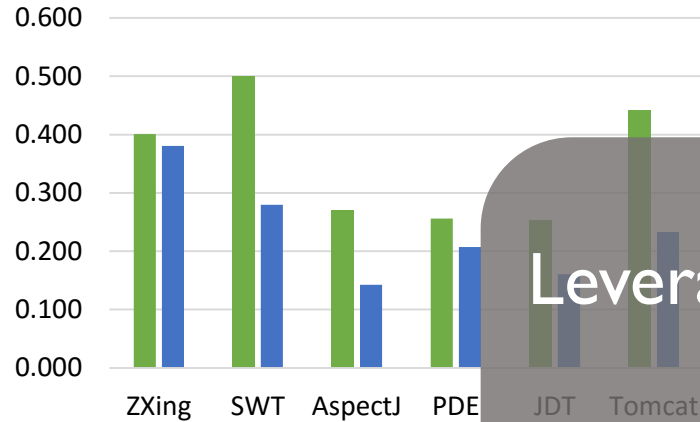
The **MRR** has been improved by **75.20%**.

NL MRR

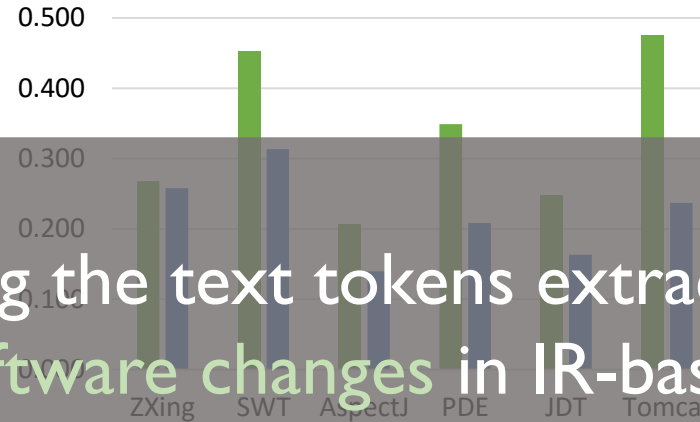


[RQ I] Software Changes VS. Source Files

NL MAP



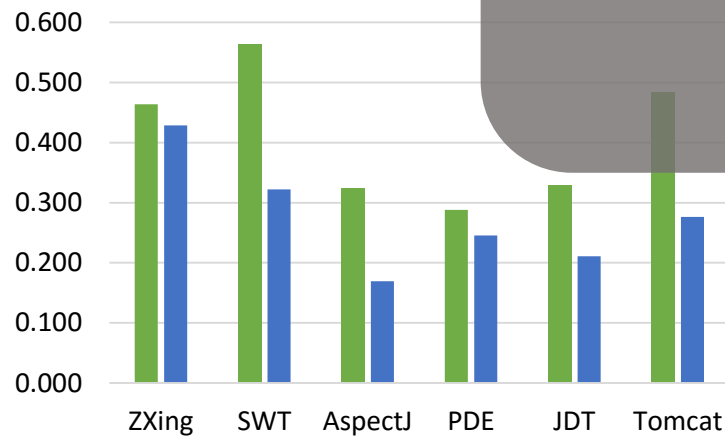
CE MAP



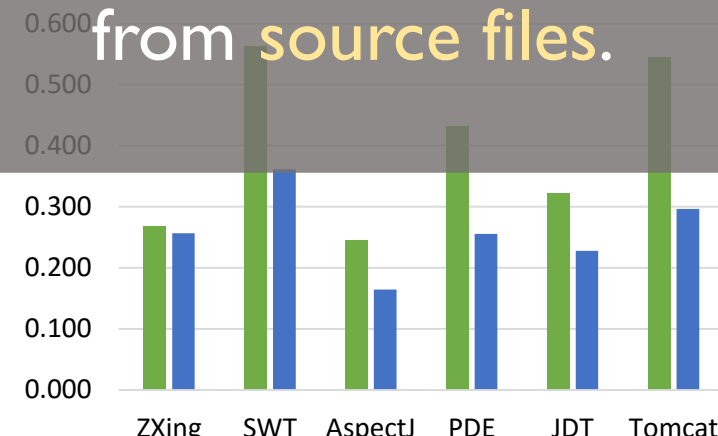
NL+CE MAP



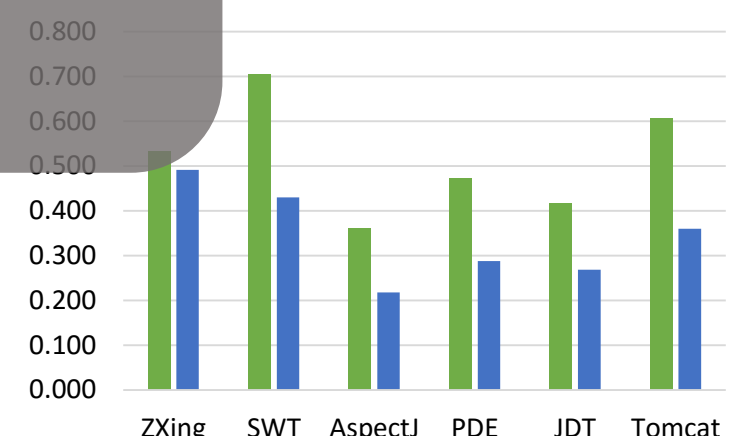
NL MRR



CE MRR



NL+CE MRR



■ Hunk ■ Source

■ Hunk ■ Source

■ Hunk ■ Source

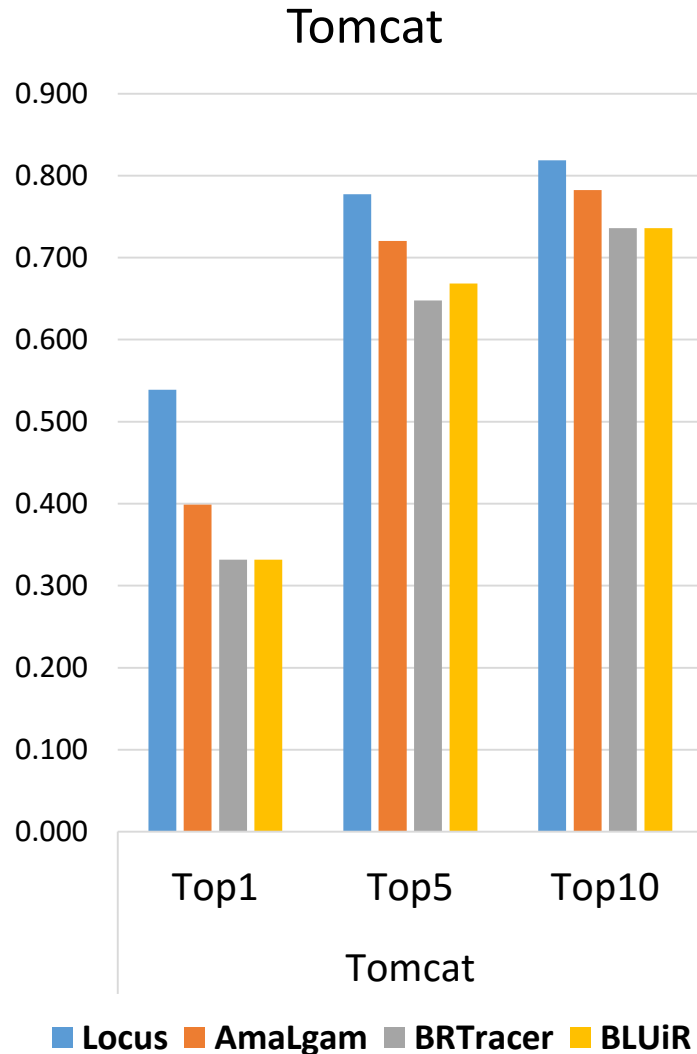
Leveraging the text tokens extracted from software changes in IR-based models achieves better results than that from source files.

[RQ2] Locus VS. Baselines

Three state-of-the-art **IR-based** approaches:

- BRTracer [[Wong et al. ICSME'2014](#)]
Leveraging stack traces and segmenting source files
- BLUiR [[Saha et al. ASE'2013](#)]
Leveraging code structures information
- AmaLgam [[Wang et al. ICPC'2014](#)]
Combining similar bugs, code structures and fixing histories together.

[RQ2] Locus VS. Baselines



Tomcat

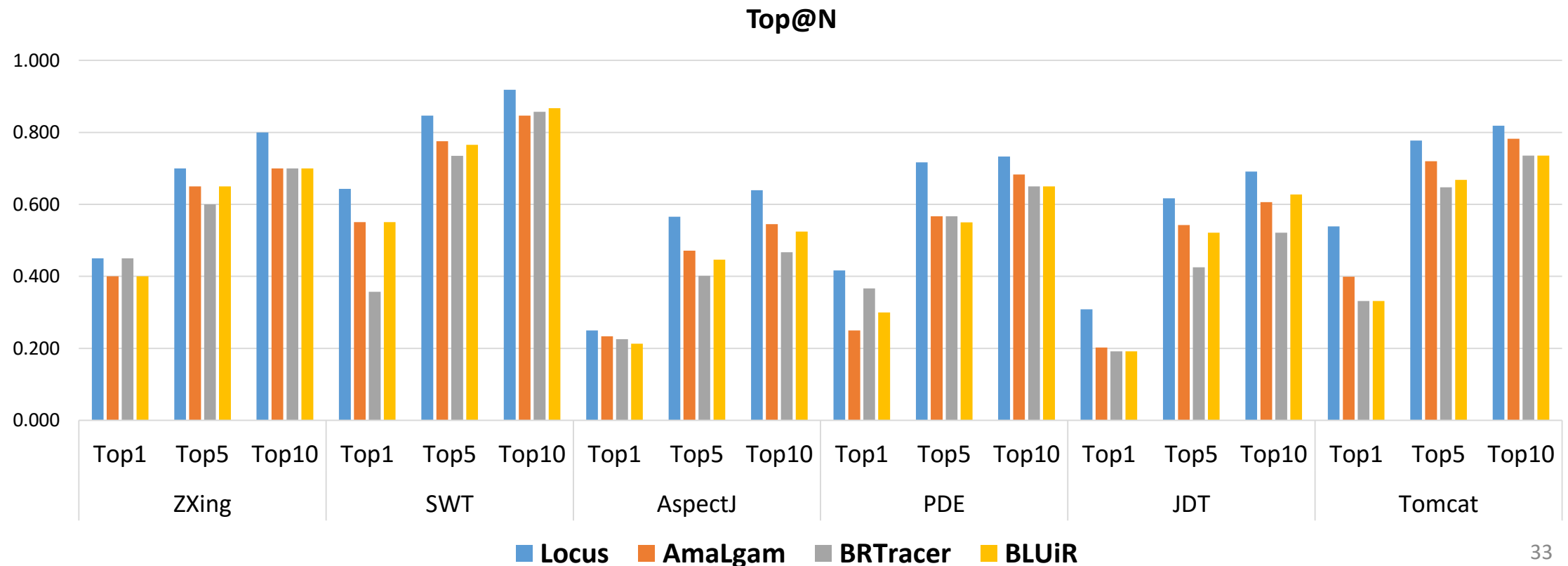
Locates **56.6%** of the bugs and rank them at **Top 1**. **27 more bugs**

Locates **77.7%** of the bugs and rank them at **Top 5**. **11 more bugs**

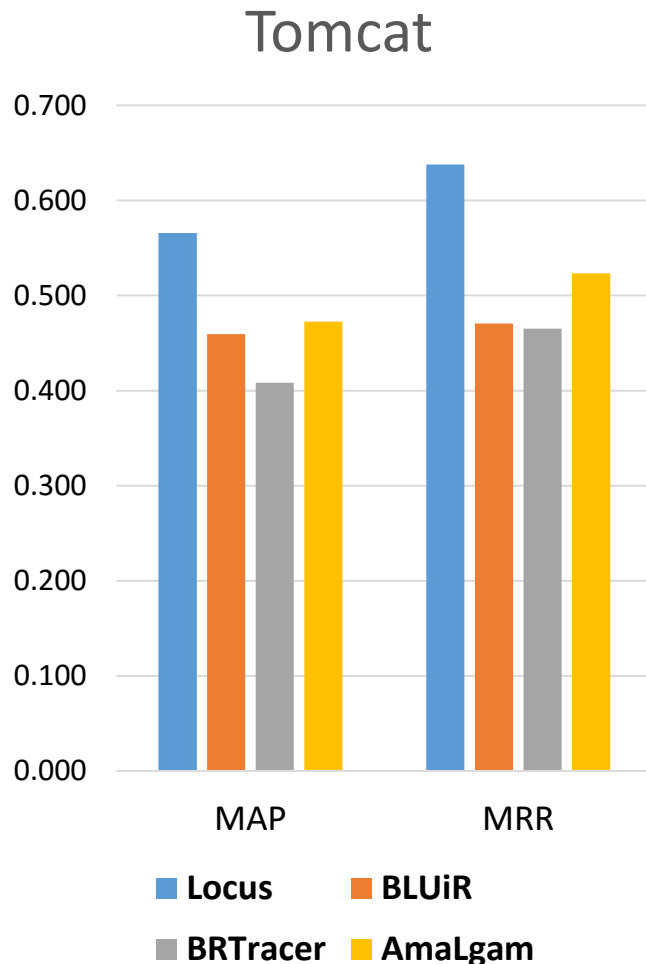
Locates **81.9%** of the bugs and rank them at **Top 10**. **7 more bugs**

[RQ2] Locus VS. Baselines

- Locus locates the buggy files and rank them as **top 1** for **41.0%** of the bugs.
- Locus **outperforms** the three baselines works for all subjects under all Top@N metrics.



[RQ2] Locus VS. Baselines

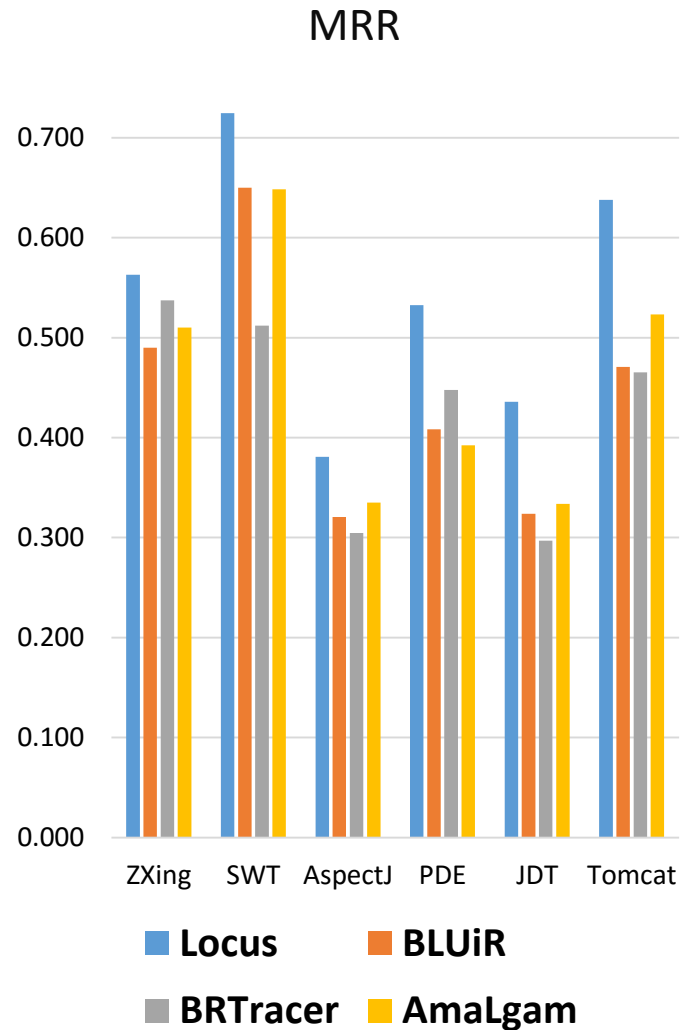
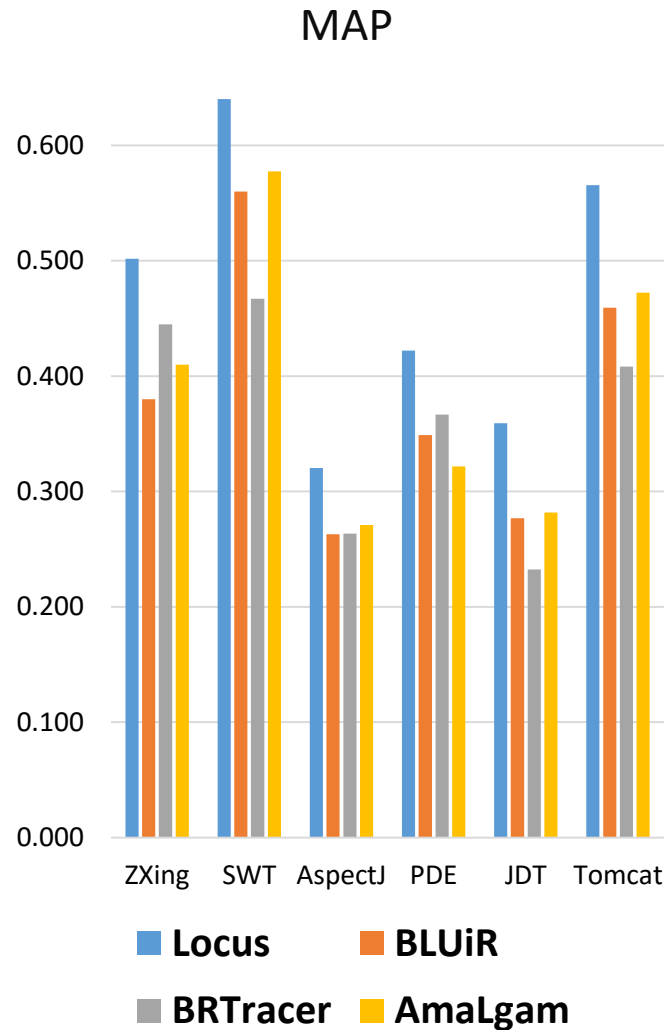


Tomcat

Achieves an MAP and MRR of **0.566** and **0.638**.

The MAP and MRR have outperform the best baseline for **19.7%** and **21.9%** respectively.

[RQ2] Locus VS. Baselines



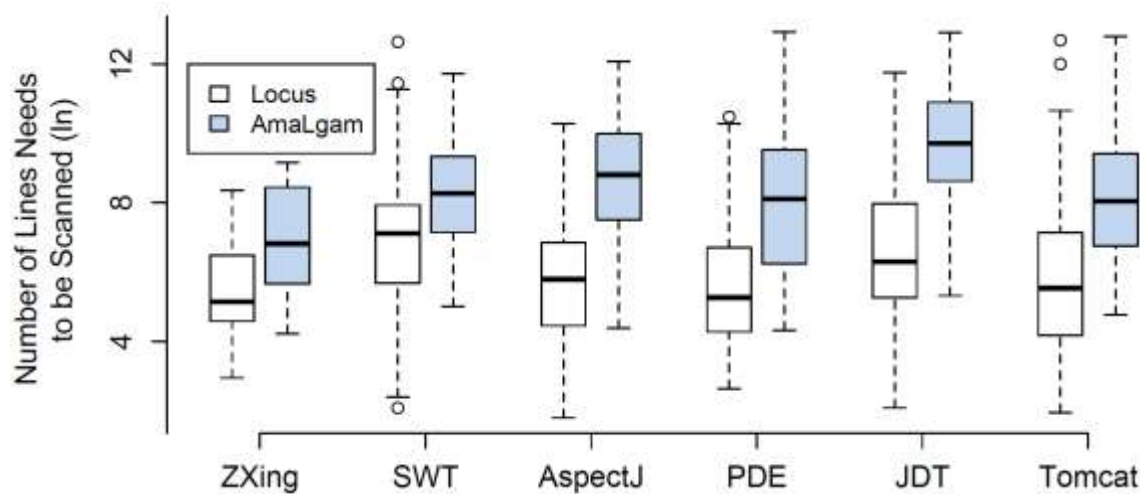
- Locus achieves an average MAP of **0.468** and MRR of **0.546**.
- The average improvements

	MAP	MRR
AmaLgam	20.1%	20.5%
BLUiR	22.4%	25.3%
BRTracer	31.9%	32.4%

[RQ2] Change Level Results

Results of MAP, MRR and Top@N at the Change Level

Subject	MAP	MRR	Top@1	Top@5	Top@10	Top@20
ZXing	0.262	0.333	0.200	0.400	0.500	0.900
SWT	0.14	0.224	0.141	0.308	0.436	0.551
AspectJ	0.217	0.315	0.228	0.406	0.506	0.628
PDE	0.219	0.33	0.208	0.479	0.604	0.667
JDT	0.103	0.223	0.162	0.275	0.385	0.474
Tomcat	0.268	0.390	0.276	0.511	0.598	0.701



Comparison between Locus and Amalgam, in terms of the Effort-Based Evaluation.

- Locus achieves an average MAP of **0.205** and MRR of **0.256**.
- Locus locates the inducing changes and rank them within top5 for **41.0%** of the bugs.
- The lines of codes needing to be inspected has been reduced by an order of magnitude.
- The debugging efforts can be **significantly** saved.

[RQ2] Change Level Results – Case Study

Bug #56905 of Tomcat

Locus : ranked the inducing change at **1st**
58 lines including contextual lines

AmaLgam : ranked the buggy file as **21st**

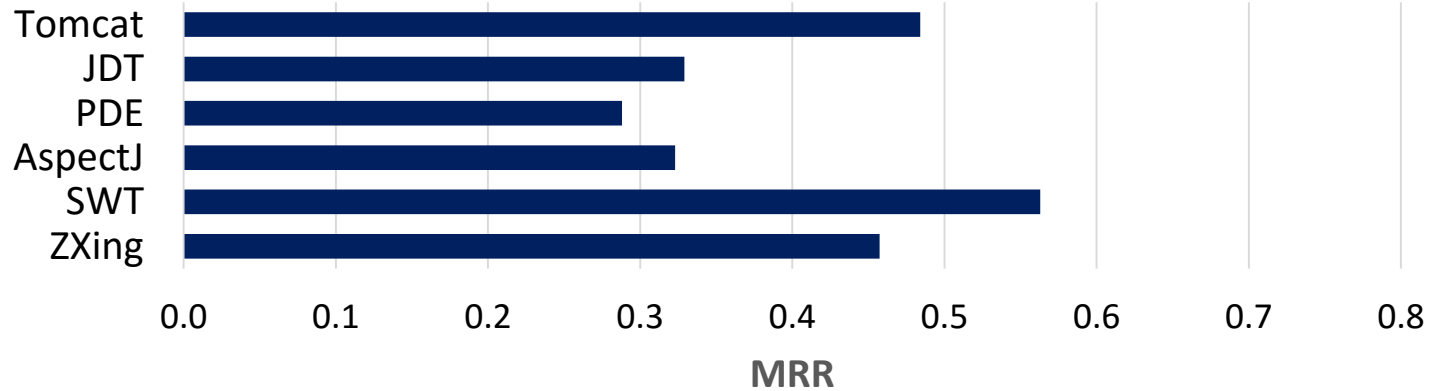
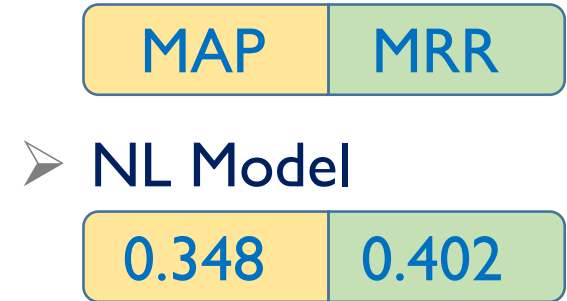
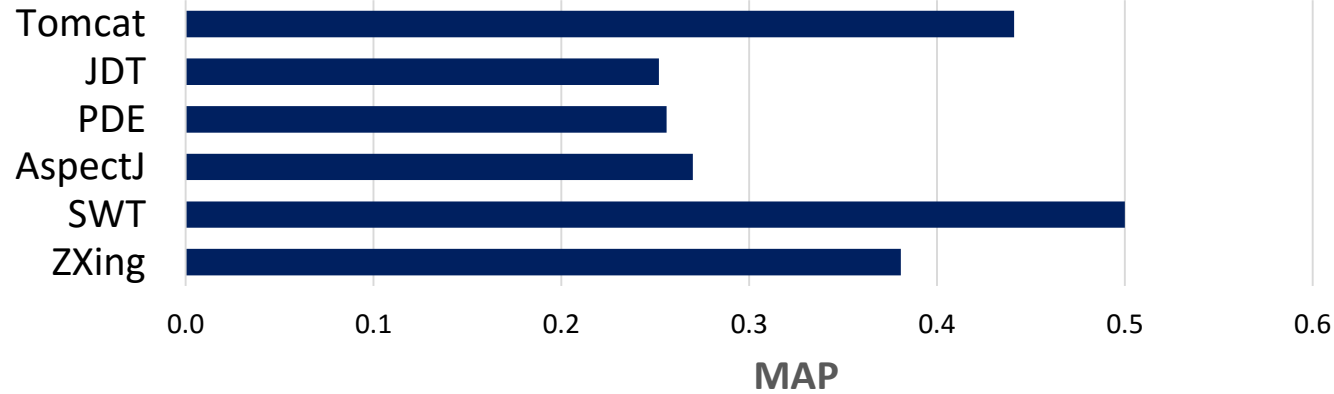
Bug #56199 of Tomcat

```
119 124 this.isTagFile = isTagFile;
120 125 this.directivesOnly = directivesOnly;
121 126 this.isTop = true;
127 +
128 + String blockExternalString = ctxt.getServletContext().getInitParameter(
129 + Constants.XML_BLOCK_EXTERNAL_INIT_PARAM);
130 + boolean blockExternal;
131 + if (blockExternalString == null) {
132 + blockExternal = Constants.IS_SECURITY_ENABLED;
133 + } else {
134 + blockExternal = Boolean.parseBoolean(blockExternalString);
135 + }
136 +
137 + this.entityResolver = new LocalResolver(
138 + DigestFactory.SERVLET_API_PUBLIC_IDS,
139 + DigestFactory.SERVLET_API_SYSTEM_IDS,
```

Both **Locus** and **AmaLgam** ranked as Top 1

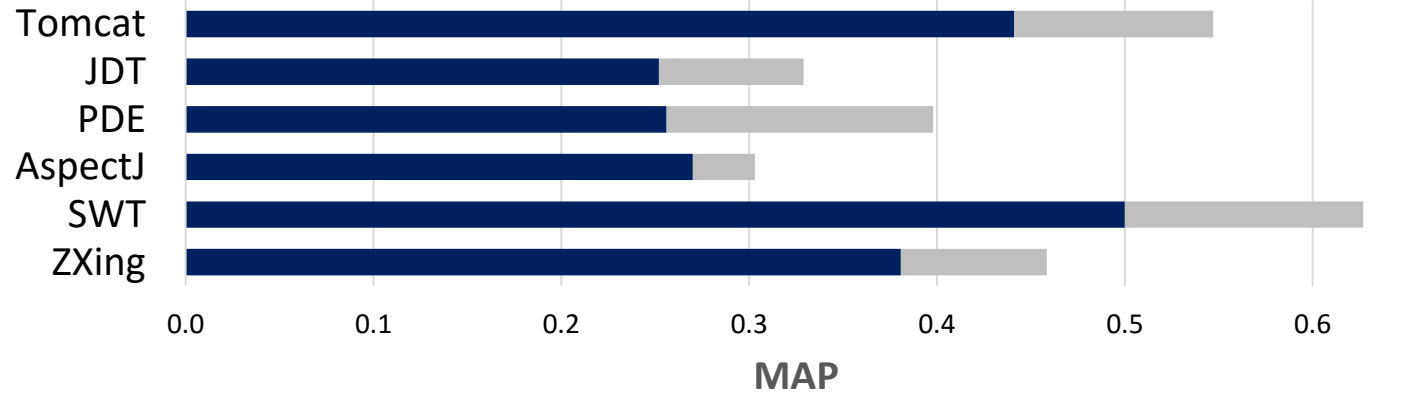
The inducing change contains **32** lines
of code while the whole file contains
864 lines.

[RQ3] Contributions of Each Model



Contributions of Each Model at Source Level

[RQ3] Contributions of Each Model



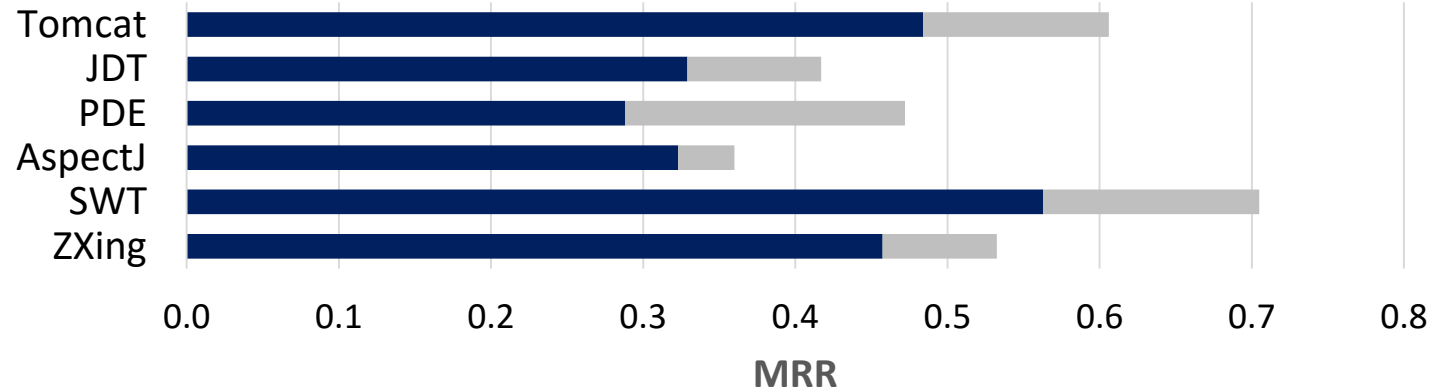
MAP MRR

➤ NL Model

0.348 0.402

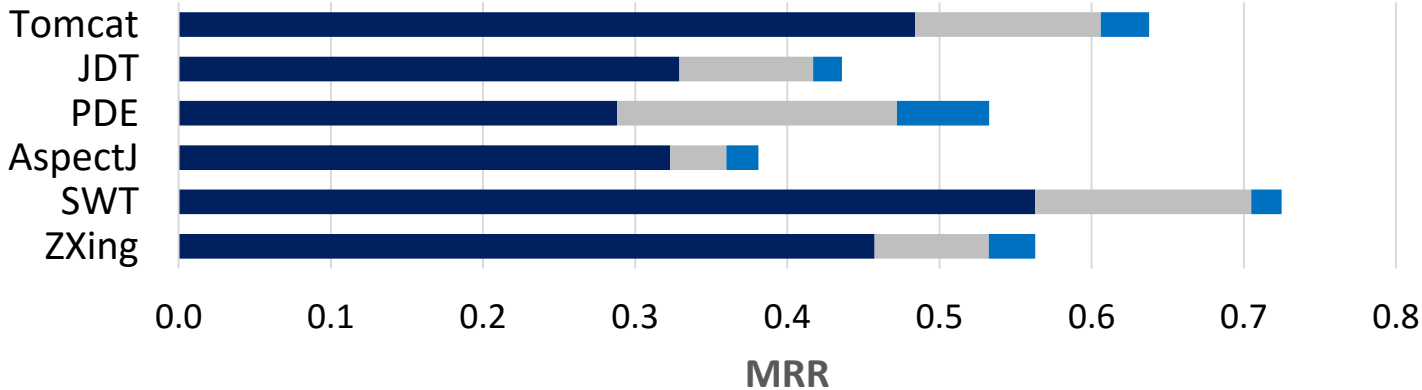
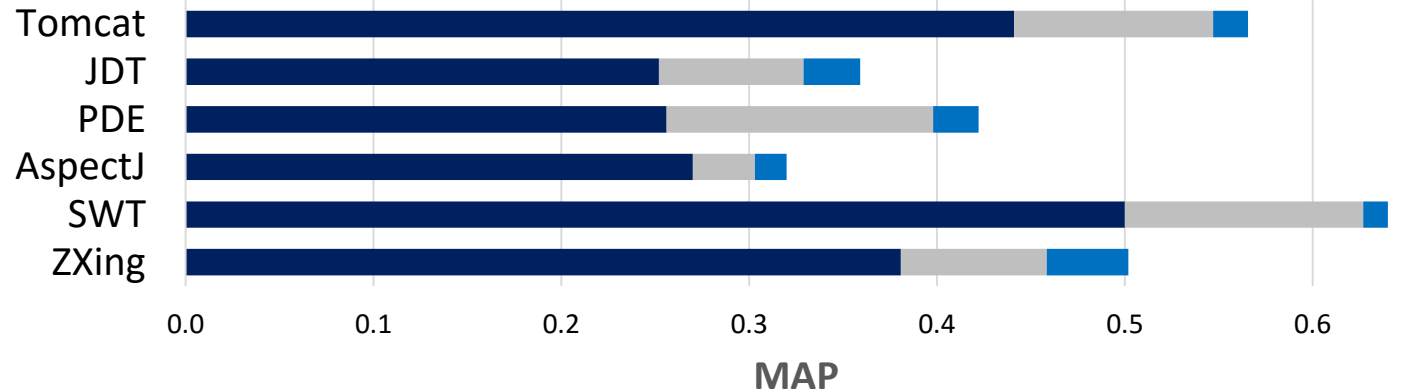
➤ NL + CE

↑23.6% ↑23.7%



Contributions of Each Model at Source Level

[RQ3] Contributions of Each Model



Contributions of Each Model at Source Level

MAP MRR

➤ NL Model

0.348 0.402

➤ NL + CE

↑23.6% ↑23.7%

➤ NL + CE + Boosting

↑5.1% ↑5.7%

Conclusions

- Bug inducing changes can help developers in debugging.
- Software changes can benefit IR-based bug localization techniques.
- We propose **Locus** based on our observations, the evaluation results show that Locus outperforms the state-of-the-art approaches.

Future Work

- We plan to leverage more **properties** of software changes (e.g. ownership, change patterns) to improve the performance at the change level.
- We plan to conduct **user studies** to evaluate the practical usefulness.

Q & A

