Can Better Identifier Splitting Techniques Help Feature Location?

Bogdan Dit, Latifa Guerrouj, Denys Poshyvanyk, Giuliano Antoniol

SEMERU

SOCCER LAB
SOftware Cost-effective Change and Evolution Research Lab

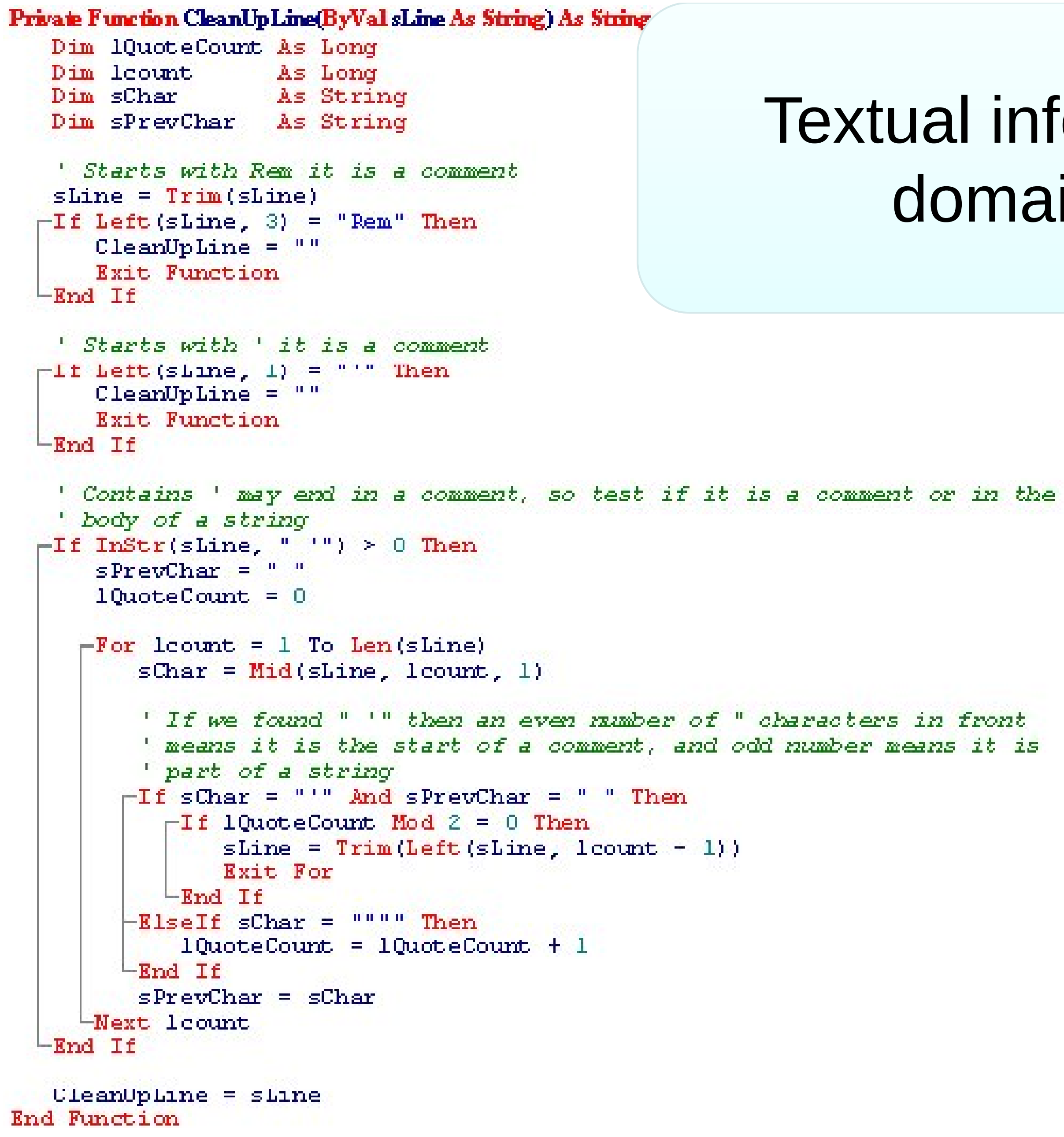
WILLIAM & MARY

ÉCOLE POLYTECHNIQUE MONTRÉAL

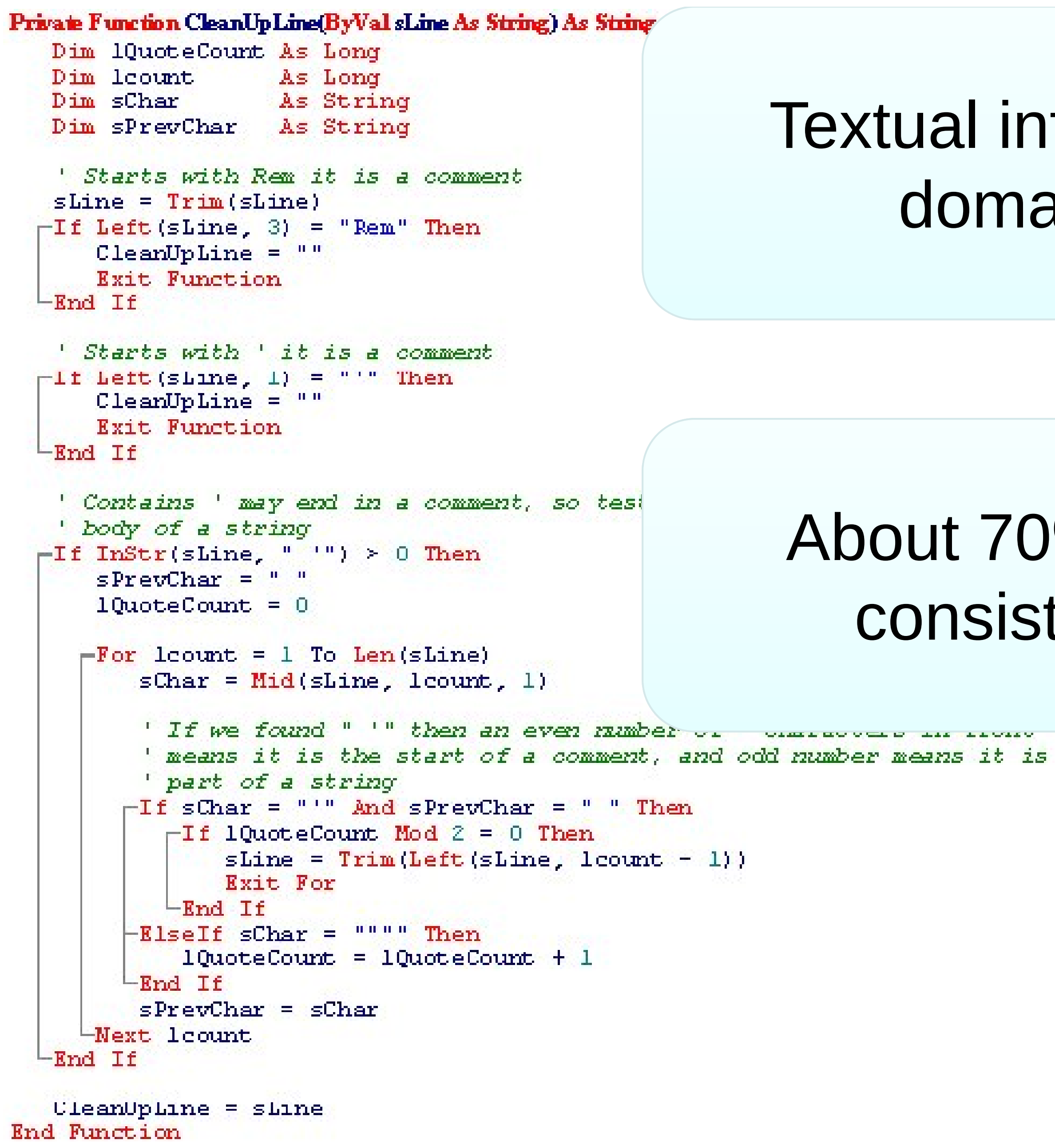
19th IEEE International Conference on Program Comprehension (ICPC'11) – Kingston, Ontario, Canada

```
Private Function CleanUpLine(ByVal sLine As String) As String
    Dim lQuoteCount As Long
    Dim lcount      As Long
    Dim sChar       As String
    Dim sPrevChar   As String

    ' Starts with Rem it is a comment
    sLine = Trim(sLine)
    If Left(sLine, 3) = "Rem" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Starts with ' it is a comment
    If Left(sLine, 1) = "'" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Contains ' may end in a comment, so test if it is a comment or in the
    ' body of a string
    If InStr(sLine, " '") > 0 Then
        sPrevChar = " "
        lQuoteCount = 0

        For lcount = 1 To Len(sLine)
            sChar = Mid(sLine, lcount, 1)

            ' If we found " '" then an even number of " characters in front
            ' means it is the start of a comment, and odd number means it is
            ' part of a string
            If sChar = "'" And sPrevChar = " " Then
                If lQuoteCount Mod 2 = 0 Then
                    sLine = Trim(Left(sLine, lcount - 1))
                    Exit For
                End If
            ElseIf sChar = """" Then
                lQuoteCount = lQuoteCount + 1
            End If
            sPrevChar = sChar
        Next lcount
    End If

    CleanUpLine = sLine
End Function
```

Textual information embeds domain knowledge



```
Private Function CleanUpLine(ByVal sLine As String) As String
    Dim lQuoteCount As Long
    Dim lcount      As Long
    Dim sChar       As String
    Dim sPrevChar   As String

    ' Starts with Rem it is a comment
    sLine = Trim(sLine)
    If Left(sLine, 3) = "Rem" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Starts with ' it is a comment
    If Left(sLine, 1) = "'" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Contains ' may end in a comment, so test if it is a comment or in the
    ' body of a string
    If InStr(sLine, " '") > 0 Then
        sPrevChar = " "
        lQuoteCount = 0

        For lcount = 1 To Len(sLine)
            sChar = Mid(sLine, lcount, 1)

            ' If we found " '" then an even number of " characters in front
            ' means it is the start of a comment, and odd number means it is
            ' part of a string
            If sChar = "'" And sPrevChar = " " Then
                If lQuoteCount Mod 2 = 0 Then
                    sLine = Trim(Left(sLine, lcount - 1))
                    Exit For
                End If
            ElseIf sChar = """" Then
                lQuoteCount = lQuoteCount + 1
            End If
            sPrevChar = sChar
        Next lcount
    End If

    CleanUpLine = sLine
End Function
```

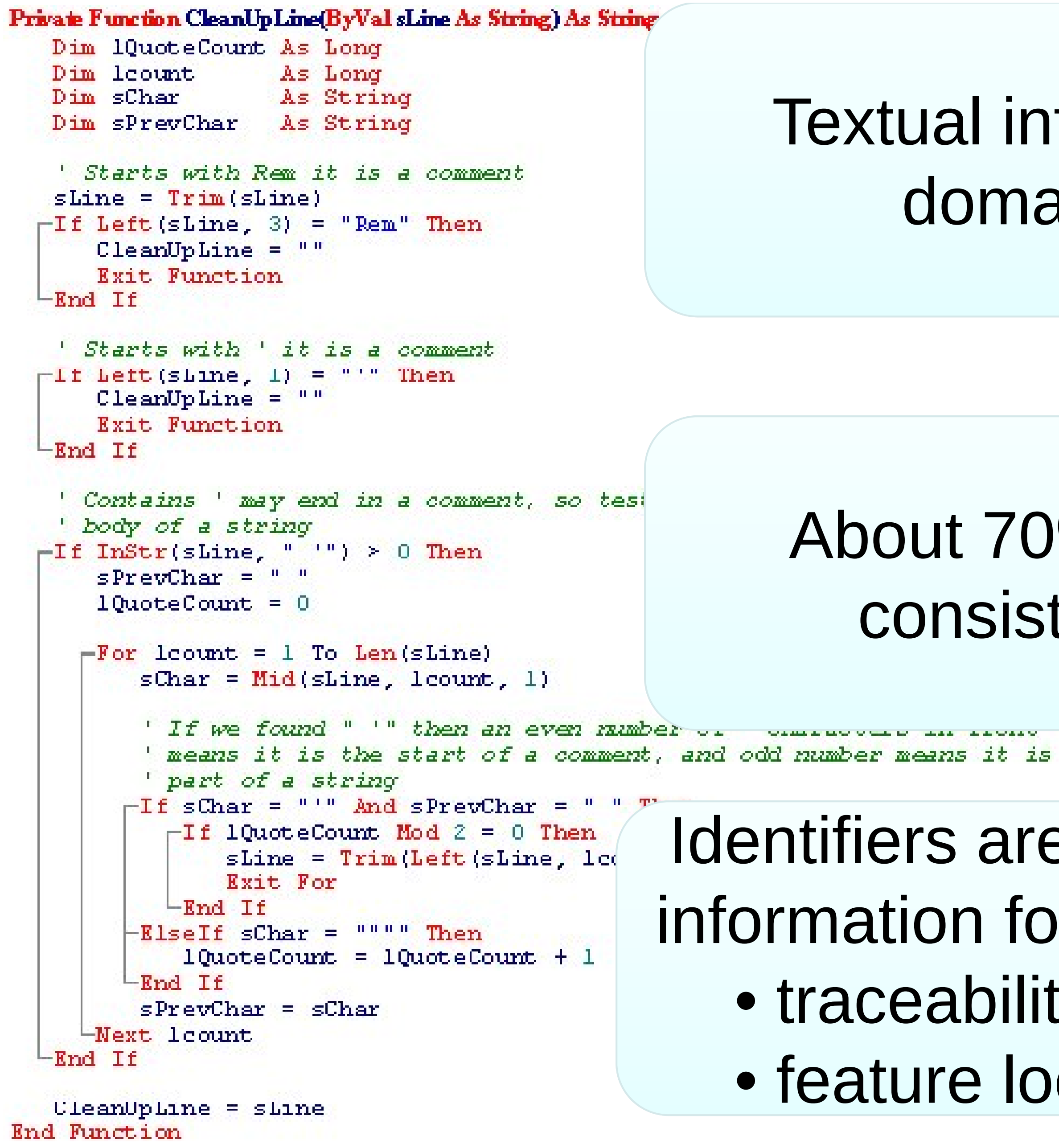


```
Private Function CleanUpLine(ByVal sLine As String) As String
    Dim lQuoteCount As Long
    Dim lcount      As Long
    Dim sChar       As String
    Dim sPrevChar   As String

    ' Starts with Rem it is a comment
    sLine = Trim(sLine)
    If Left(sLine, 3) = "Rem" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Starts with ' it is a comment
    If Left(sLine, 1) = "'" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Contains ' may end in a comment, so test
    ' body of a string
    If InStr(sLine, " '") > 0 Then
        sPrevChar = " "
        lQuoteCount = 0

        For lcount = 1 To Len(sLine)
            sChar = Mid(sLine, lcount, 1)

            ' If we found " '" then an even number of " characters in front
            ' means it is the start of a comment, and odd number means it is
            ' part of a string
            If sChar = "'" And sPrevChar = " " Then
                If lQuoteCount Mod 2 = 0 Then
                    sLine = Trim(Left(sLine, lcount - 1))
                    Exit For
                End If
            ElseIf sChar = """" Then
                lQuoteCount = lQuoteCount + 1
            End If
            sPrevChar = sChar
        Next lcount
    End If

    CleanUpLine = sLine
End Function
```

Textual information embeds domain knowledge

About 70% of source code consists of identifiers*

* Deissenboeck, F. and Pizka , M., "Concise and Consistent Naming", Software Quality Journal, vol. 14, no. 3, 2006, pp. 261-282



```
Private Function CleanUpLine(ByVal sLine As String) As String
    Dim lQuoteCount As Long
    Dim lcount      As Long
    Dim sChar       As String
    Dim sPrevChar   As String

    ' Starts with Rem it is a comment
    sLine = Trim(sLine)
    If Left(sLine, 3) = "Rem" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Starts with ' it is a comment
    If Left(sLine, 1) = "'" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Contains ' may end in a comment, so test
    ' body of a string
    If InStr(sLine, " '") > 0 Then
        sPrevChar = " "
        lQuoteCount = 0

        For lcount = 1 To Len(sLine)
            sChar = Mid(sLine, lcount, 1)

            ' If we found " '" then an even number
            ' means it is the start of a comment, and odd number means it is
            ' part of a string
            If sChar = "'" And sPrevChar = " "
                If lQuoteCount Mod 2 = 0 Then
                    sLine = Trim(Left(sLine, lc
                    Exit For
                End If
            ElseIf sChar = """" Then
                lQuoteCount = lQuoteCount + 1
            End If
            sPrevChar = sChar
        Next lcount
    End If

    CleanUpLine = sLine
End Function
```

Textual information embeds domain knowledge

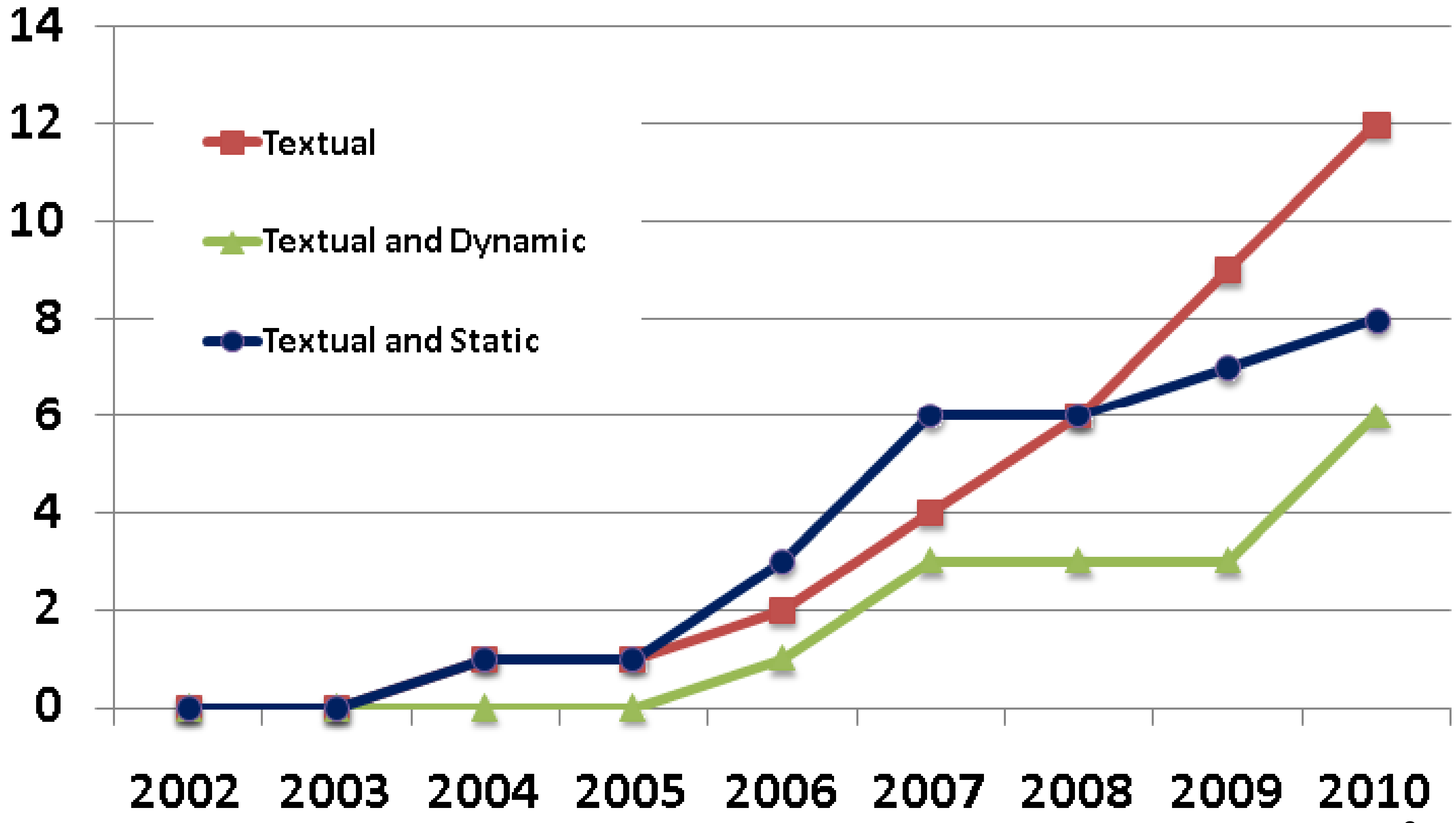
About 70% of source code consists of identifiers*

Identifiers are important source of information for maintenance tasks:

- traceability link recovery
- feature location

* Deissenboeck, F. and Pizka , M., "Concise and Consistent Naming", Software Quality Journal, vol. 14, no. 3, 2006, pp. 261-282

# of Cumulative Feature Location Papers based on Textual Information



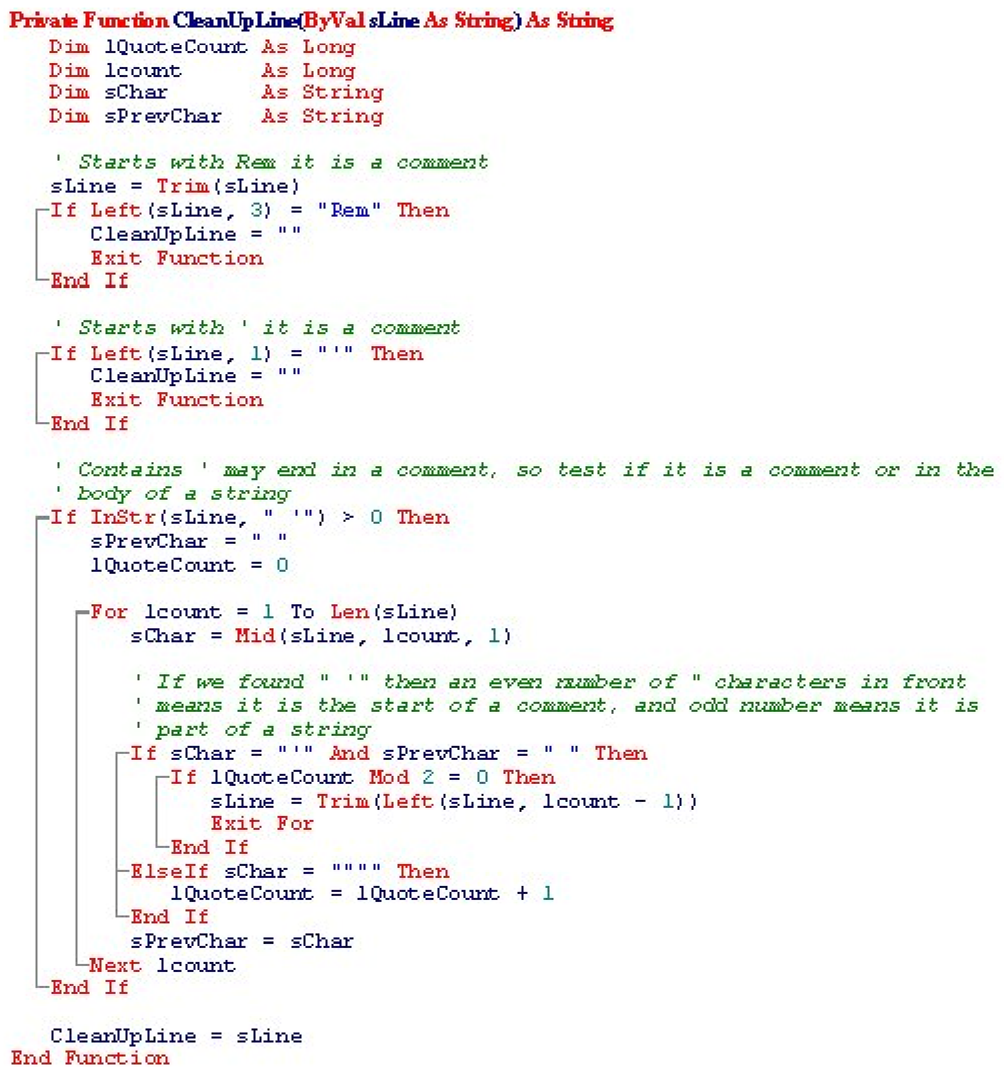
Related Work on Identifiers

- Takang et al. (JPL’96)
  - programs with full-word identifiers are more understandable than those with abbreviated ones
- Lawrie et al. (ICPC’06)
  - full words and recognizable abbreviations lead to better comprehension
- Binkley et al. (ICPC’09)
  - CamelCase style is easier to recognize than underscore

Related Work on Identifiers

- Enslen et al. (MSR'09)
  - Samurai: algorithm for splitting identifiers (using tables of identifier frequencies)
- Guerrouj et al. (JSME'11)
  - TIDIER: algorithm for splitting identifiers (using contextual information)
- Other related work
  - Deissenboeck and Pizka (SQJ'06), Antoniol et al. (ICSM'07), Haiduc and Marcus (ICPC'08), etc.

Splitting Identifiers Correctly is Challenging



```
Private Function CleanUpLine(ByVal sLine As String) As String
   Dim lQuoteCount As Long
   Dim lcount      As Long
   Dim sChar       As String
   Dim sPrevChar   As String

   ' Starts with Rem it is a comment
   sLine = Trim(sLine)
   If Left(sLine, 3) = "Rem" Then
      CleanUpLine = ""
      Exit Function
   End If

   ' Starts with ' it is a comment
   If Left(sLine, 1) = "'" Then
      CleanUpLine = ""
      Exit Function
   End If

   ' Contains ' may end in a comment, so test if it is a comment or in the
   ' body of a string
   If InStr(sLine, " '") > 0 Then
      sPrevChar = " "
      lQuoteCount = 0

      For lcount = 1 To Len(sLine)
         sChar = Mid(sLine, lcount, 1)

         ' If we found " '" then an even number of " characters in front
         ' means it is the start of a comment, and odd number means it is
         ' part of a string
         If sChar = "'" And sPrevChar = " " Then
            If lQuoteCount Mod 2 = 0 Then
               sLine = Trim(Left(sLine, lcount - 1))
               Exit For
            End If
         ElseIf sChar = """" Then
            lQuoteCount = lQuoteCount + 1
         End If
         sPrevChar = sChar
      Next lcount
   End If

   CleanUpLine = sLine
End Function
```

Identifier Splitting Algorithms

Original Identifier
userId
setGID
print_file2device
SSLCertificate
MINstring
USERID
currentsize
readadapterobject
tolocale
imitating
DEFMASKBit

Identifier Splitting Algorithms

Original Identifier	Camel Case
userId	user Id
setGID	set GID
print_file2device	print file 2 device
SSLCertificate	SSL Certificate
MINstring	MI Nstring
USERID	USERID
currentsize	currentsize
readadapterobject	readadapterobject
tolocale	tolocale
imitating	imitating
DEFMASKBit	DEFMASK Bit

Identifier Splitting Algorithms

Original Identifier	Camel Case
userId	user Id
setGID	set GID
print_file2device	print file 2 device
SSLCertificate	SSL Certificate
MINstring	MI Nstring
USERID	USERID
currentsize	currentsize
readadapterobject	readadapterobje
tolocale	tolocale
imitating	imitating
DEFMASKBit	DEFMASK Bit

Handles underscore and digits

Fails at mixed cases

Fails at same case identifiers

Identifier Splitting Algorithms

Original Identifier	Camel Case
userId	user Id
setGID	set GID
print_file2device	print file 2 device
SSLCertificate	SSL Certificate
MINstring	MI Nstring
USERID	USERID
currentsize	currentsize
readadapterobject	readadapterobject
tolocale	tolocale
imitating	imitating
DEFMASKBit	DEFMASK Bit

Identifier Splitting Algorithms

Original Identifier	Camel Case	Samurai
userId	user Id	user Id
setGID	set GID	set GID
print_file2device	print file 2 device	print file 2 device
SSLCertificate	SSL Certificate	SSL Certificate
MINstring	MI Nstring	MIN string
USERID	USERID	USER ID
currentsize	currentsize	current size
readadapterobject	readadapterobject	read adapter object
tolocale	tolocale	tol ocal e
imitating	imitating	imi ta ting
DEFMASKBit	DEFMASK Bit	DEF MASK Bit

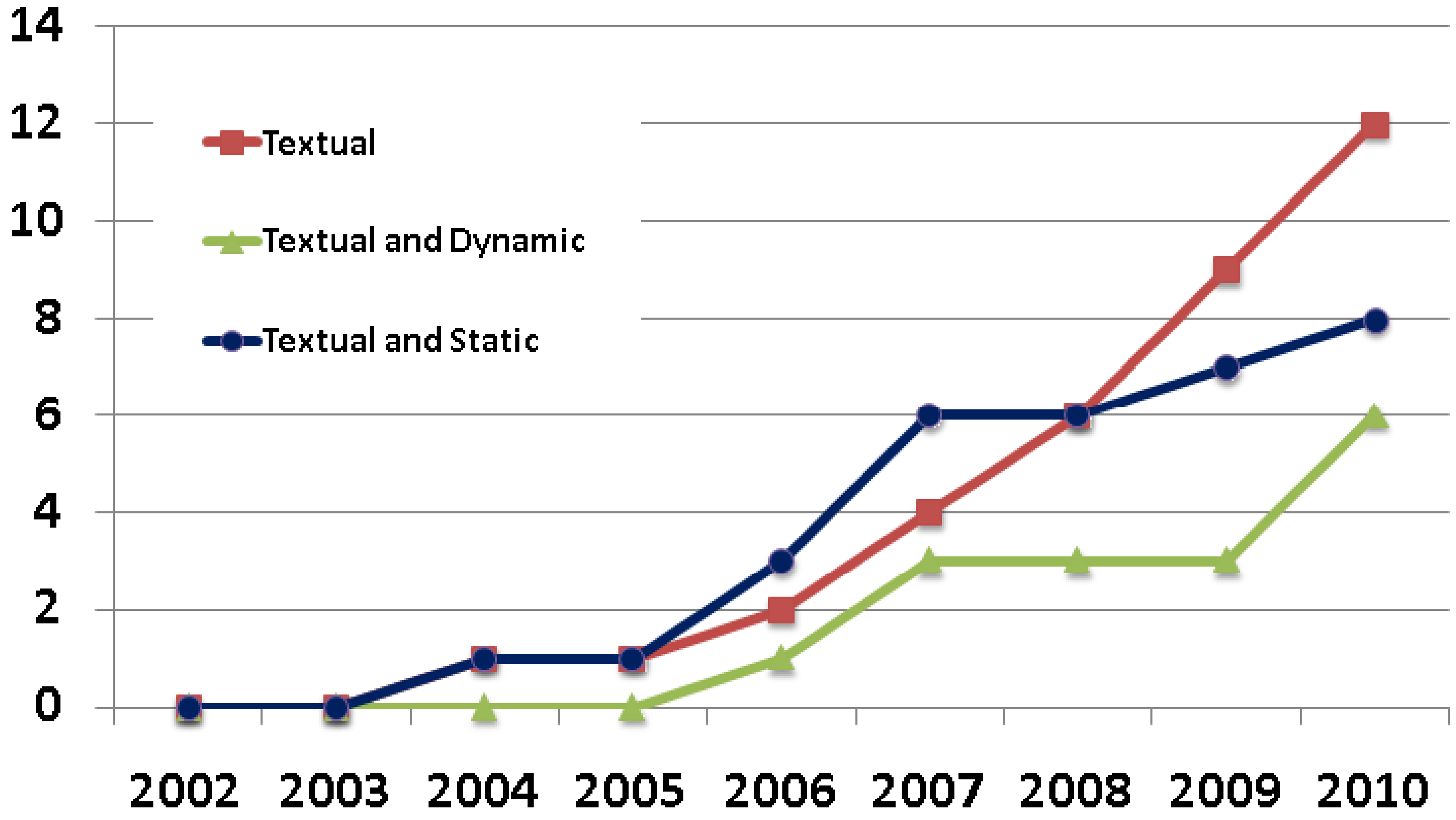
Identifier Splitting Algorithms

Original Identifier	Camel Case	Samurai
userId	[illegible]	user Id
setGID	[illegible]	set GID
print_file2device	[illegible]	print file 2 device
SSLCertificate	[illegible]	SSL Certificate
MINstring	MI Nstring	MIN string
USERID	USERID	USER ID
currentsize	currentsize	current size
readadapterobject	readadapterobject	read adapter object
tolocale	tolocale	tol ocal e
imitating	imitating	imi ta ting
DEFMASKBit	DEFMASK Bit	DEF MASK Bit

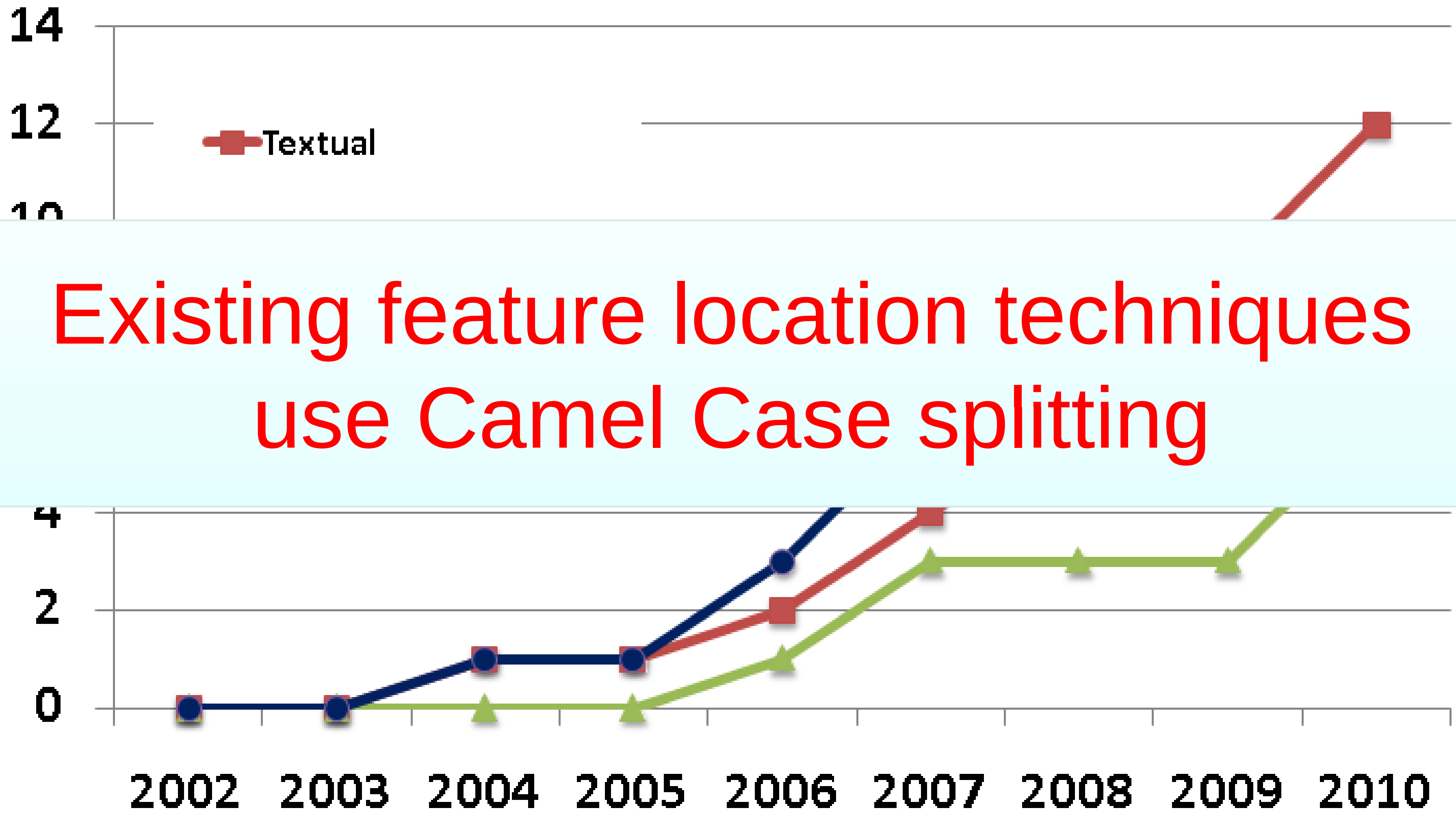
Splits some cases where CamelCase cannot

Oversplits

# of Cumulative Feature Location Papers based on Textual Information



# of Cumulative Feature Location Papers based on Textual Information



Existing feature location techniques use Camel Case splitting

Information Retrieval FLT

- Generate corpus
- Preprocessing
  - Remove non-literals
  - Remove stop words
  - Split identifiers
  - Stemming
- Indexing
  - Term-by-document matrix
  - Singular Value Decomposition
- User formulate query
- Generate results
- Ranked list

```
synchronized void print(TestResult result,
long runTime) throws IOException{
    printHeader(runTime);
    printErrors(result);
    printFailures(result);
    printFooter(result);
}
```

```
synchronized void print TestResult result
long runTime throws IOException
printHeader runTime printErrors result
printFailures result printFooter result
```

```
print TestResult result runTime
IOException printHeader runTime
printErrors result printFailures result
printFooter result
```

Information Retrieval FLT

- Generate corpus
- Preprocessing
  - Remove non-literals
  - Remove stop words
  - **Split identifiers**
  - Stemming
- Indexing
  - Term-by-document matrix
  - Singular Value Decomposition
- User formulate query
- Generate results
- Ranked list

print Test Result result run Time IO Exception print Header run Time print Errors result print Failures result print Footer result

⇩

print test result result run time io exception print head run time print error result print fail result print foot result

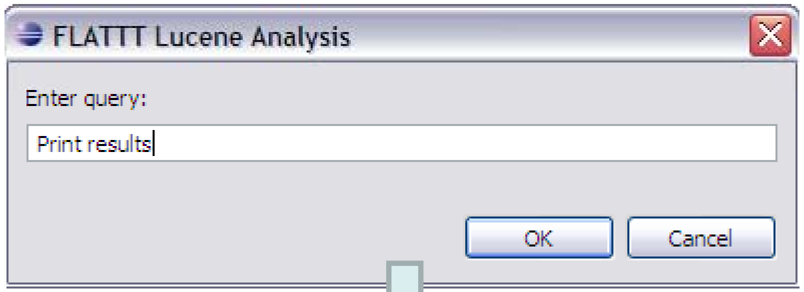
⇩

	print	test	result	...
m_1	5	1	3	...
m_2	...	...	...	...

Information Retrieval FLT

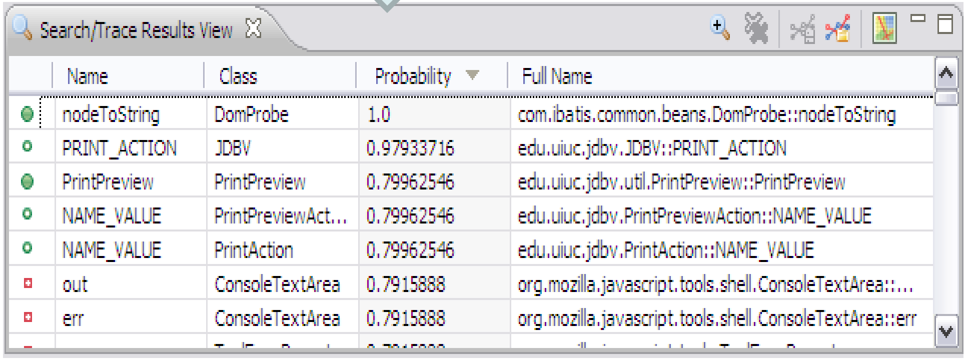
- Generate corpus
- Preprocessing
  - Remove non-literals
  - Remove stop words
  - Split identifiers
  - Stemming
- Indexing
  - Term-by-document matrix
  - Singular Value Decomposition
- User formulate query
- Generate results
- Ranked list

	print	test	result	...
m_1	5	1	3	...
m_2	...	...	...	...


FLATTT Lucene Analysis

Enter query:

Print results


Search/Trace Results View

	Name	Class	Probability	Full Name
●	nodeToString	DomProbe	1.0	com.ibatis.common.beans.DomProbe::nodeToString
○	PRINT_ACTION	JDBV	0.97933716	edu.uiuc.jdbv.JDBV::PRINT_ACTION
●	PrintPreview	PrintPreview	0.79962546	edu.uiuc.jdbv.util.PrintPreview::PrintPreview
○	NAME_VALUE	PrintPreviewAct...	0.79962546	edu.uiuc.jdbv.PrintPreviewAction::NAME_VALUE
○	NAME_VALUE	PrintAction	0.79962546	edu.uiuc.jdbv.PrintAction::NAME_VALUE
▫	out	ConsoleTextArea	0.7915888	org.mozilla.javascript.tools.shell.ConsoleTextArea::...
▫	err	ConsoleTextArea	0.7915888	org.mozilla.javascript.tools.shell.ConsoleTextArea::err

IR and Dynamic Information FLT

- Generate corpus
- Preprocessing
  - Remove non-literals
  - Remove stop words
  - Split identifiers
  - Stemming
- Indexing
  - Term-by-document matrix
  - Singular Value Decomposition
- User formulate query
- Generate results
- Ranked list **of executed methods**

Collect execution trace

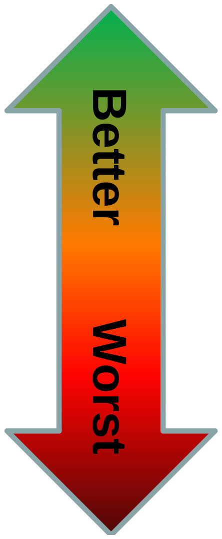
Research Goal

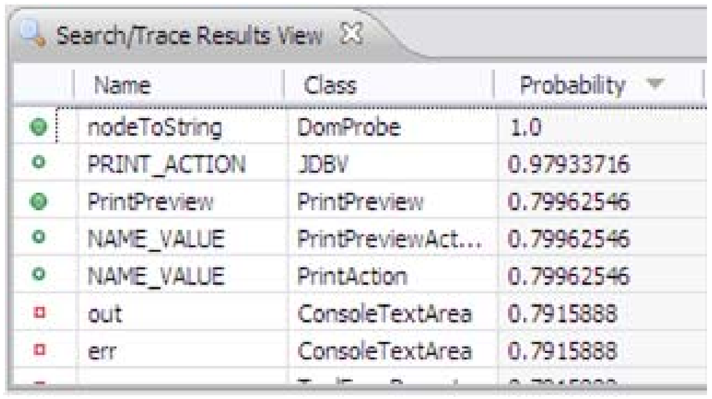
Evaluate how advanced splitting techniques impact the performance of feature location techniques

Information Retrieval FLT

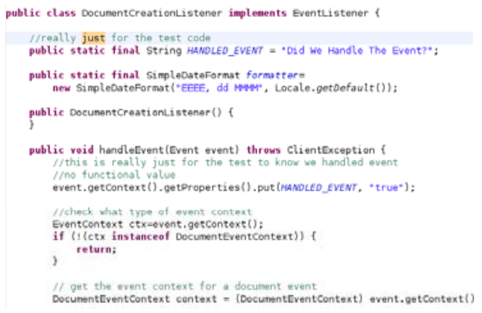
- Generate corpus
- Preprocessing
  - Remove non-literals
  - Remove stop words
  - **Split identifiers**
  - Stemming
- Indexing
  - Term-by-document matrix
  - Singular Value Decomposition
- User formulate query
- Generate results
- Ranked list

Replace Camel Case with :
•Samurai
•“Perfect” Splitting algorithm (Oracle)

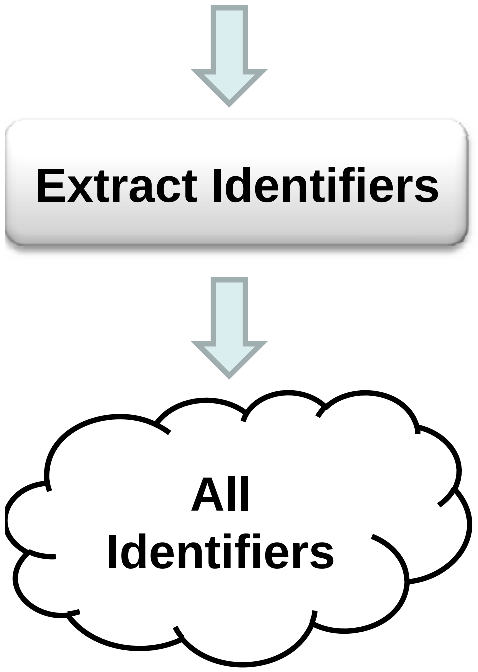




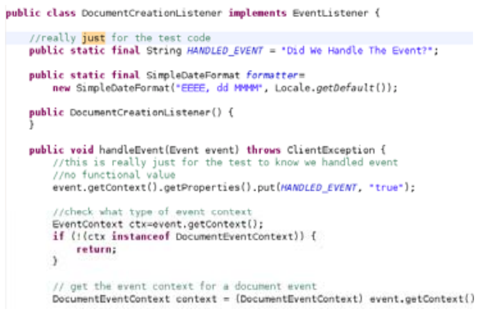
	Name	Class	Probability
●	nodeToString	DomProbe	1.0
○	PRINT_ACTION	JDBV	0.97933716
●	PrintPreview	PrintPreview	0.79962546
○	NAME_VALUE	PrintPreviewAct...	0.79962546
○	NAME_VALUE	PrintAction	0.79962546
▫	out	ConsoleTextArea	0.7915888
▫	err	ConsoleTextArea	0.7915888



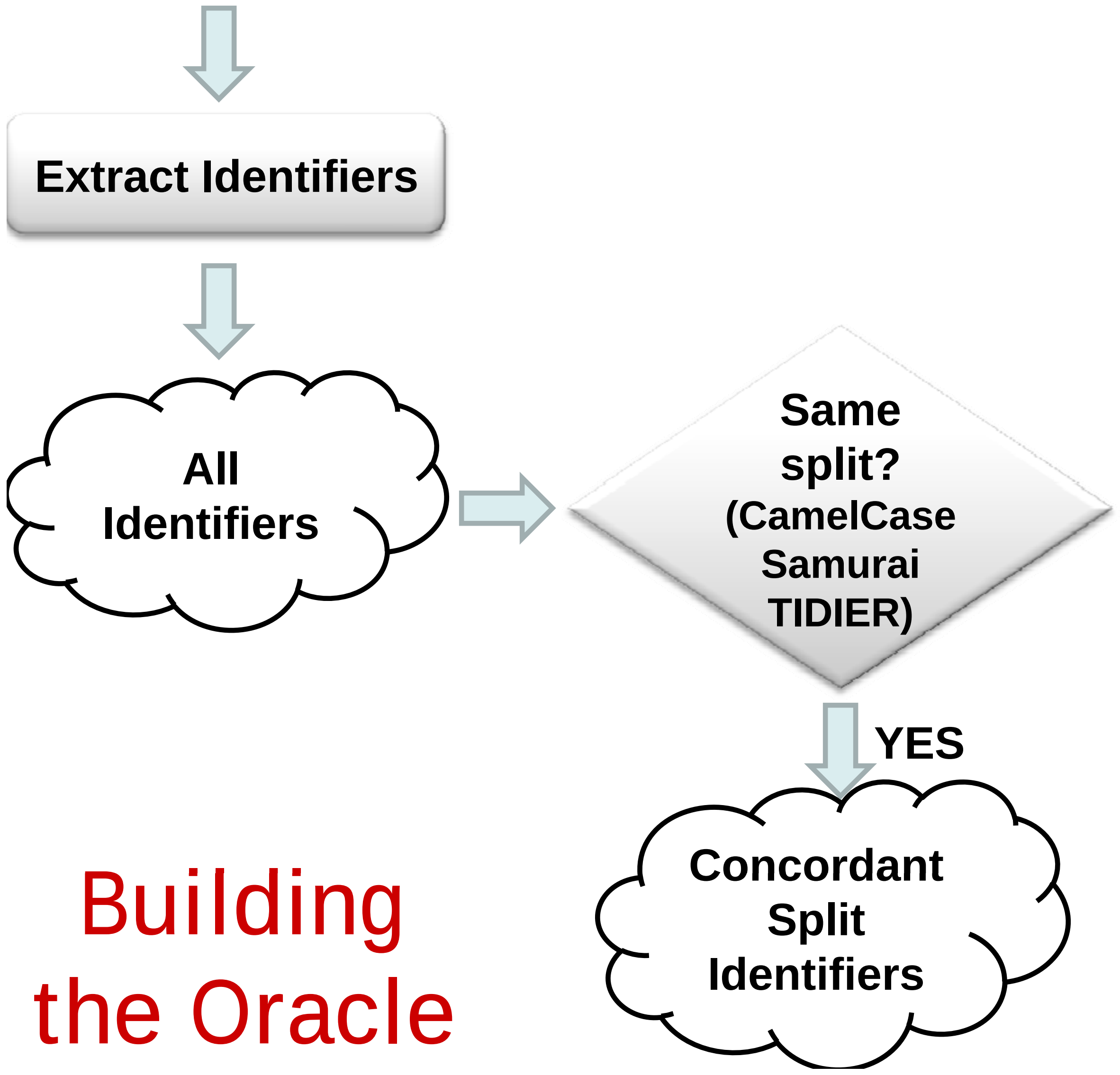
```
public class DocumentCreationListener implements EventListener {

    //really just for the test code
    public static final String HANDLED_EVENT = "Did We Handle The Event?";

    public static final SimpleDateFormat formatter=
        new SimpleDateFormat("EEEE, dd MMMM", Locale.getDefault());

    public DocumentCreationListener() {
    }

    public void handleEvent(Event event) throws ClientException {
        //this is really just for the test to know we handled event
        //no functional value
        event.getContext().getProperties().put(HANDLED_EVENT, "true");

        //check what type of event context
        EventContext ctx=event.getContext();
        if (!(ctx instanceof DocumentEventContext)) {
            return;
        }

        // get the event context for a document event
        DocumentEventContext context = (DocumentEventContext) event.getContext()
```



Building the Oracle

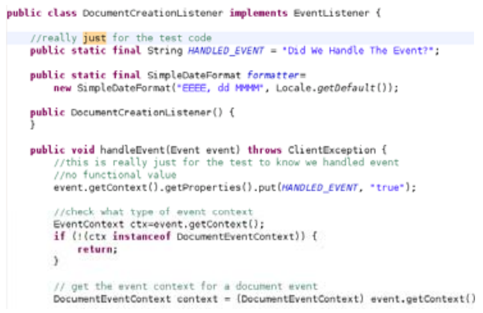


```
public class DocumentCreationListener implements EventListener {

    //really just for the test code
    public static final String HANDLED_EVENT = "Did We Handle The Event?";

    public static final SimpleDateFormat formatter=
        new SimpleDateFormat("EEEE, dd MMMM", Locale.getDefault());

    public DocumentCreationListener() {
    }

    public void handleEvent(Event event) throws ClientException {
        //this is really just for the test to know we handled event
        //no functional value
        event.getContext().getProperties().put(HANDLED_EVENT, "true");

        //check what type of event context
        EventContext ctx=event.getContext();
        if (!(ctx instanceof DocumentEventContext)) {
            return;
        }

        // get the event context for a document event
        DocumentEventContext context = (DocumentEventContext) event.getContext()
```

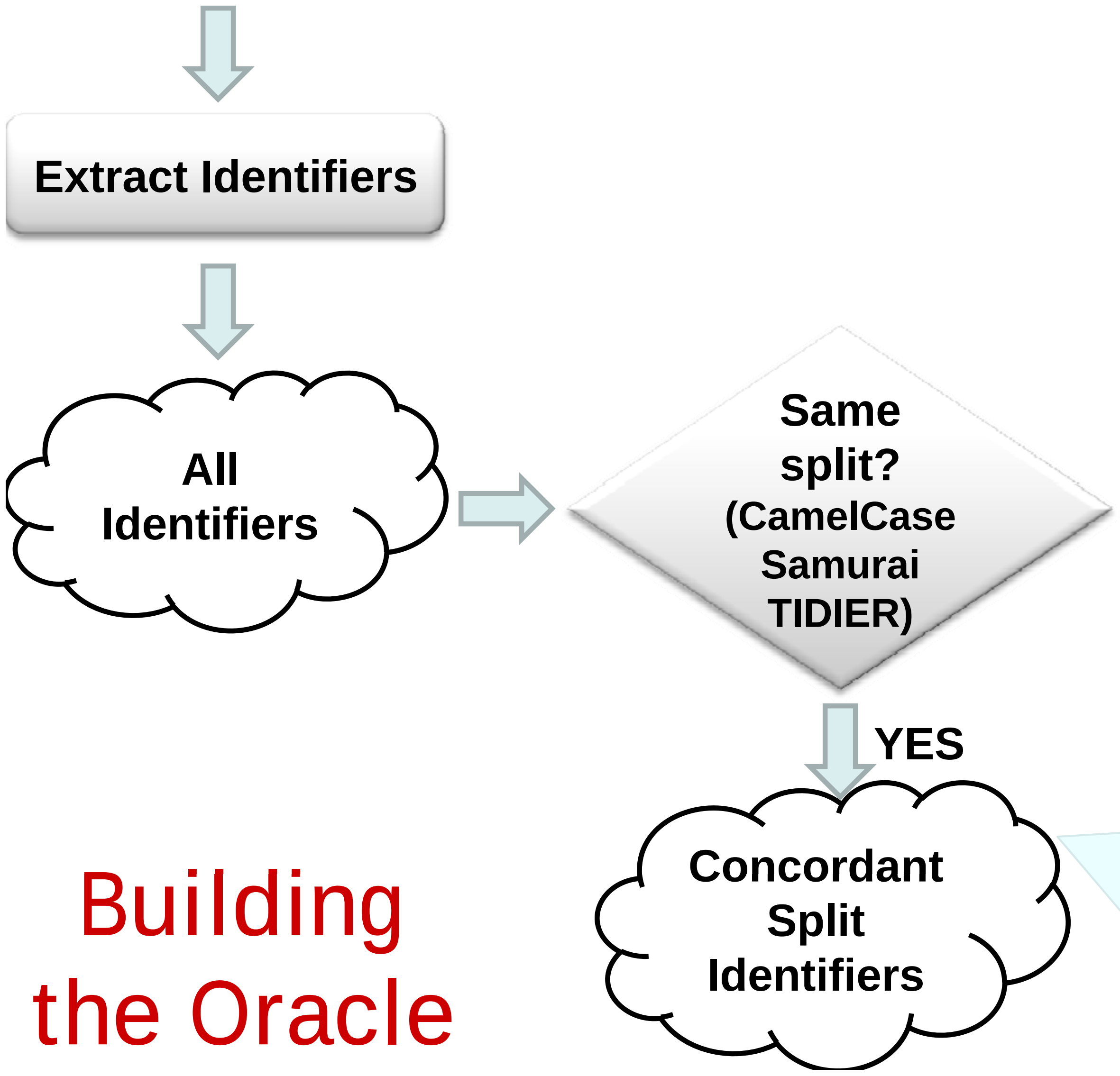

Extract Identifiers
All Identifiers
Same split? (CamelCase Samurai TIDIER)
YES
Concordant Split Identifiers

Building the Oracle

Building the Oracle



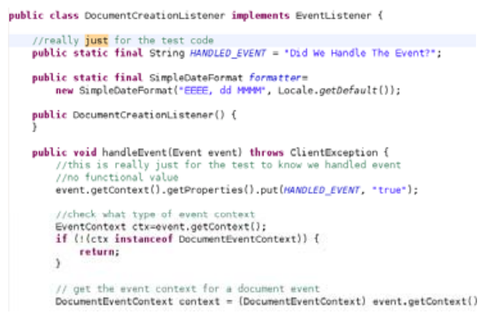
```java
public class DocumentCreationListener implements EventListener {

    //really just for the test code
    public static final String HANDLED_EVENT = "Did We Handle The Event?";

    public static final SimpleDateFormat formatter=
        new SimpleDateFormat("EEEE, dd MMMM", Locale.getDefault());

    public DocumentCreationListener() {
    }

    public void handleEvent(Event event) throws ClientException {
        //this is really just for the test to know we handled event
        //no functional value
        event.getContext().getProperties().put(HANDLED_EVENT, "true");

        //check what type of event context
        EventContext ctx=event.getContext();
        if (!(ctx instanceof DocumentEventContext)) {
            return;
        }

        // get the event context for a document event
        DocumentEventContext context = (DocumentEventContext) event.getContext()
```



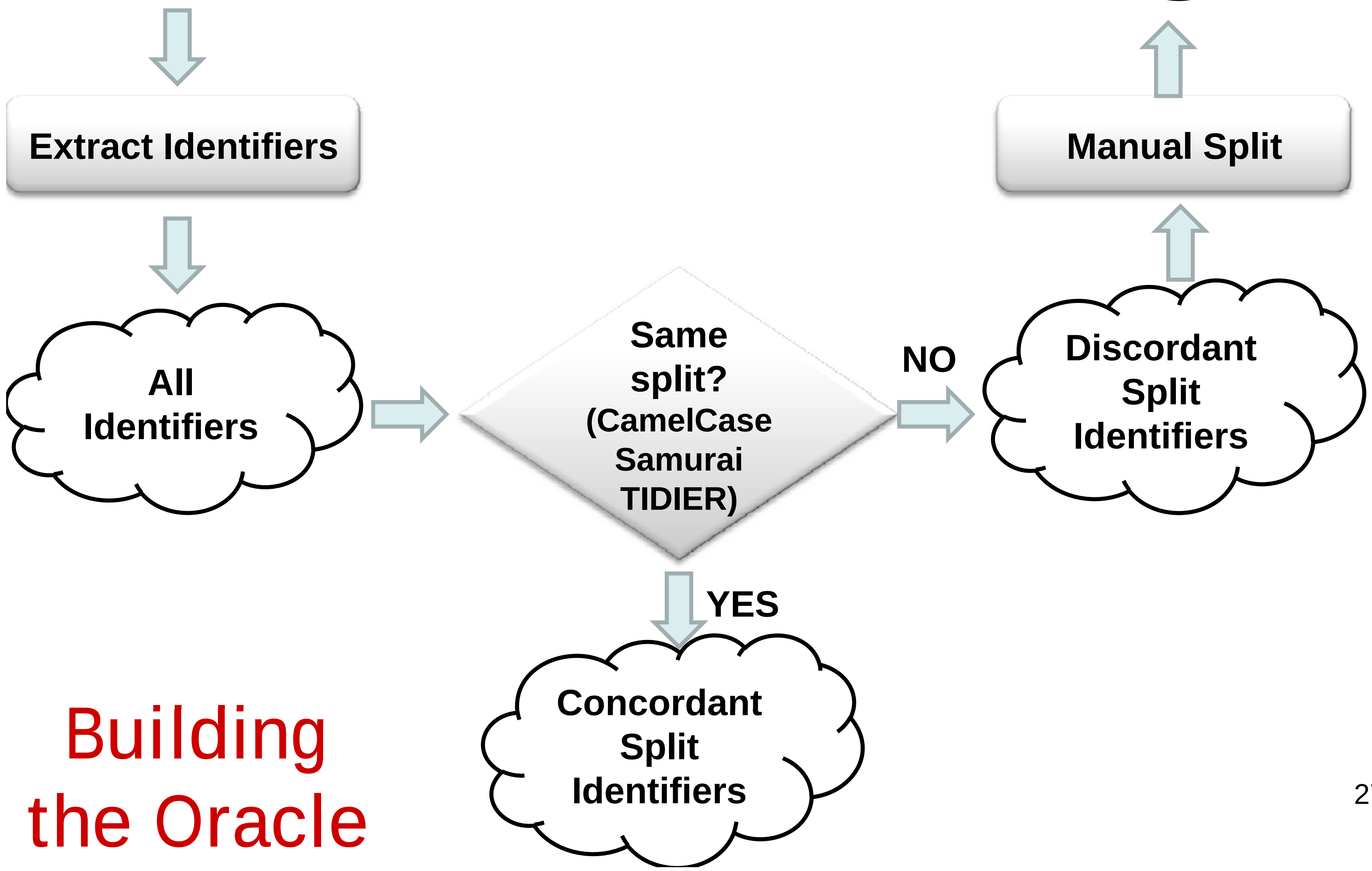
Extract Identifiers
All Identifiers
Same split? (CamelCase Samurai TIDIER)
YES
Concordant Split Identifiers

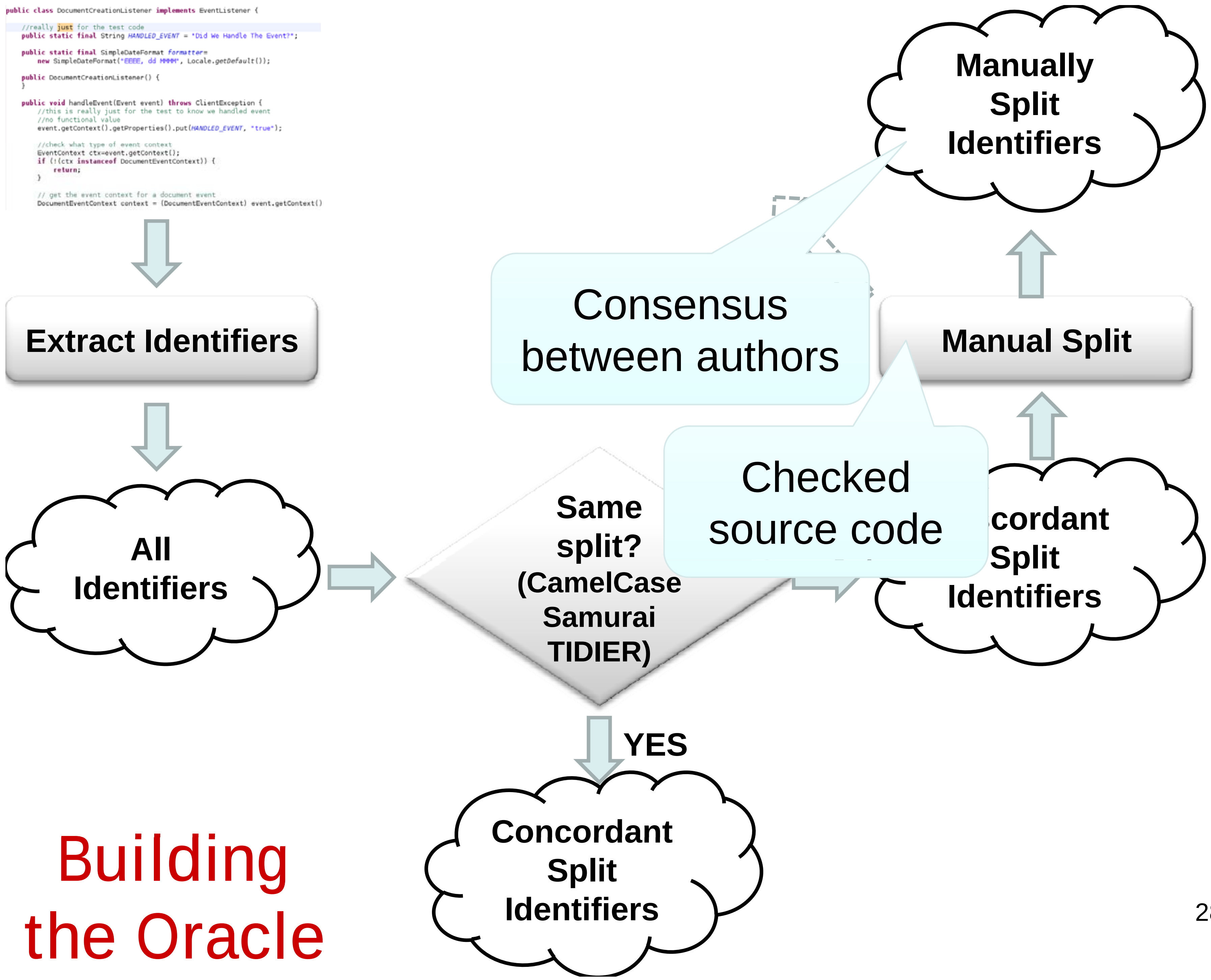
- Assume they are correct
- Manually verified a sample
- Threat to validity

Building the Oracle



```
public class DocumentCreationListener implements EventListener {

    //really just for the test code
    public static final String HANDLED_EVENT = "Did We Handle The Event?";

    public static final SimpleDateFormat formatter=
        new SimpleDateFormat("EEEE, dd MMMM", Locale.getDefault());

    public DocumentCreationListener() {
    }

    public void handleEvent(Event event) throws ClientException {
        //this is really just for the test to know we handled event
        //no functional value
        event.getContext().getProperties().put(HANDLED_EVENT, "true");

        //check what type of event context
        EventContext ctx=event.getContext();
        if (!(ctx instanceof DocumentEventContext)) {
            return;
        }

        // get the event context for a document event
        DocumentEventContext context = (DocumentEventContext) event.getContext()
```

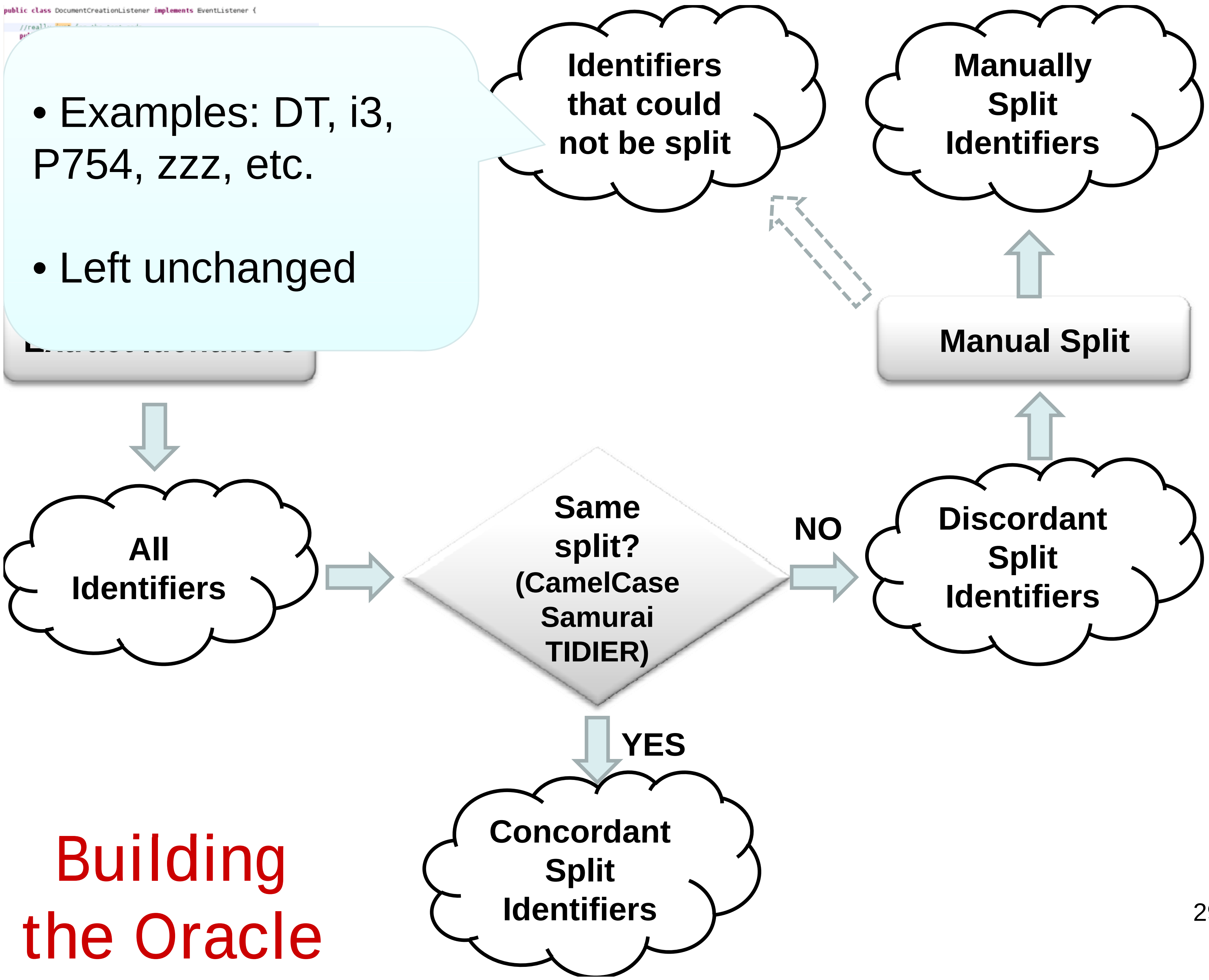

Extract Identifiers
All Identifiers
Same split? (CamelCase Samurai TIDIER)
NO
Discordant Split Identifiers
Manual Split
Manually Split Identifiers
YES
Concordant Split Identifiers



```
public class DocumentCreationListener implements EventListener {

    //really just for the test code
    public static final String HANDLED_EVENT = "Did We Handle The Event?";

    public static final SimpleDateFormat formatter=
        new SimpleDateFormat("EEEE, dd MMMM", Locale.getDefault());

    public DocumentCreationListener() {
    }

    public void handleEvent(Event event) throws ClientException {
        //this is really just for the test to know we handled event
        //no functional value
        event.getContext().getProperties().put(HANDLED_EVENT, "true");

        //check what type of event context
        EventContext ctx=event.getContext();
        if (!(ctx instanceof DocumentEventContext)) {
            return;
        }

        // get the event context for a document event
        DocumentEventContext context = (DocumentEventContext) event.getContext()
```

Extract Identifiers
All Identifiers
Same split? (CamelCase Samurai TIDIER)
YES
Concordant Split Identifiers
cordant Split Identifiers
Manual Split
Manually Split Identifiers
Consensus between authors
Checked source code

Building the Oracle



Identifiers that could not be split
Manually Split Identifiers
Manual Split
All Identifiers
Same split? (CamelCase Samurai TIDIER)
NO
Discordant Split Identifiers
YES
Concordant Split Identifiers

- Examples: DT, i3, P754, zzz, etc.

- Left unchanged

Building the Oracle

Design of the Case Study

Design of the Case Study

- RQ: Does a **FLT with an advanced splitting** algorithm produce better results than the same **FLT using the CamelCase splitting** algorithm?

How to Compare two FLTs?

How to Compare two FLTs?

- Effectiveness measure for each feature

IR

Method	LSI score
M_{121}	0.92
M_{64}	0.89
M_{15}	0.86
M_{29}	0.80
M_{7}	0.74
M_{152}	0.65
M_{234}	0.56
M_{12}	0.54
M_{78}	0.52

Gold set method

Effectiveness = 5

How to Compare two FLTs?

- Effectiveness measure for each feature

IR

Method	LSI score
M_{121}	0.92
M_{64}	0.89
M_{15}	0.86
M_{29}	0.80
M_{7}	0.74
M_{152}	0.65
M_{234}	0.56
M_{12}	0.54
M_{78}	0.52

Gold set method

Effectiveness = 2

IRDyn

Method	LSI score
M_{15}	0.86
M_{7}	0.74
M_{234}	0.56
M_{12}	0.54

Method

Executed method (from trace)

Which FLTs are we Comparing?

$$\begin{Bmatrix} IR \\ IRDyn \end{Bmatrix} \times \begin{Bmatrix} CamelCase \\ Samurai \\ Oracle \end{Bmatrix} \rightarrow \begin{Bmatrix} IR_{CamelCase} & IR_{CamelCase}Dyn \\ IR_{Samurai} & IR_{Samurai}Dyn \\ IR_{Oracle} & IR_{Oracle}Dyn \end{Bmatrix}$$

Software Systems

- Rhino 1.6R5
- 138 classes, 1,870 methods, 32K LOC
- Eaddy et al.'s data*
- 2 datasets

Dataset	Size	Queries	Gold Sets	Execution Information
$\text{Rhino}_{\text{Features}}$	241	Sections of ECMAScript documentation	Eaddy et al.*	Full Execution Traces (from unit tests)
$\text{Rhino}_{\text{Bugs}}$	143	Bug title and description	Eaddy et al.* (CVS)	N/A

* http://www.cs.columbia.edu/~eaddy/concerntagger/

Software Systems

- jEdit 4.3
- 483 classes, 6.4K methods, 109K LOC
- 2 datasets

Dataset	Size	Queries	Gold Sets	Execution Information
jEdit$_{Features}$	64	Feature (or Patch) title and description	SVN	Marked Execution Traces
jEdit$_{Bugs}$	86	Bug title and description	SVN	Marked Execution Traces

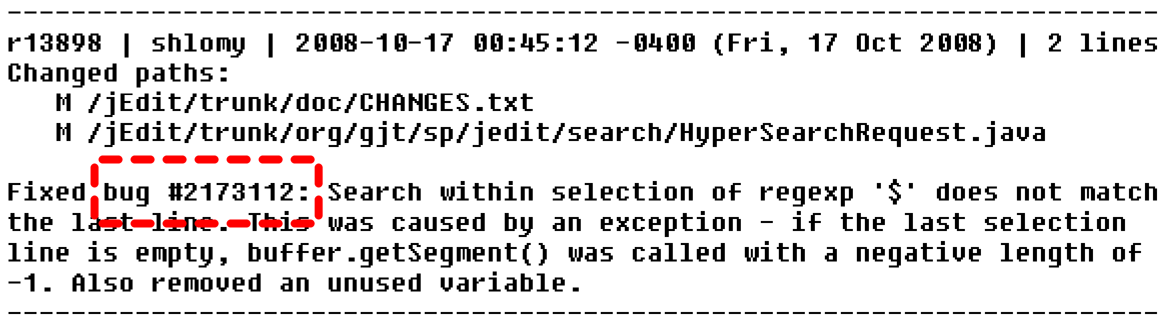
Datasets available at:
http://www.cs.wm.edu/semeru/data/icpc11-identifier-splitting/

Generating the jEdit Datasets

SVN Commits between v4.2-v4.3

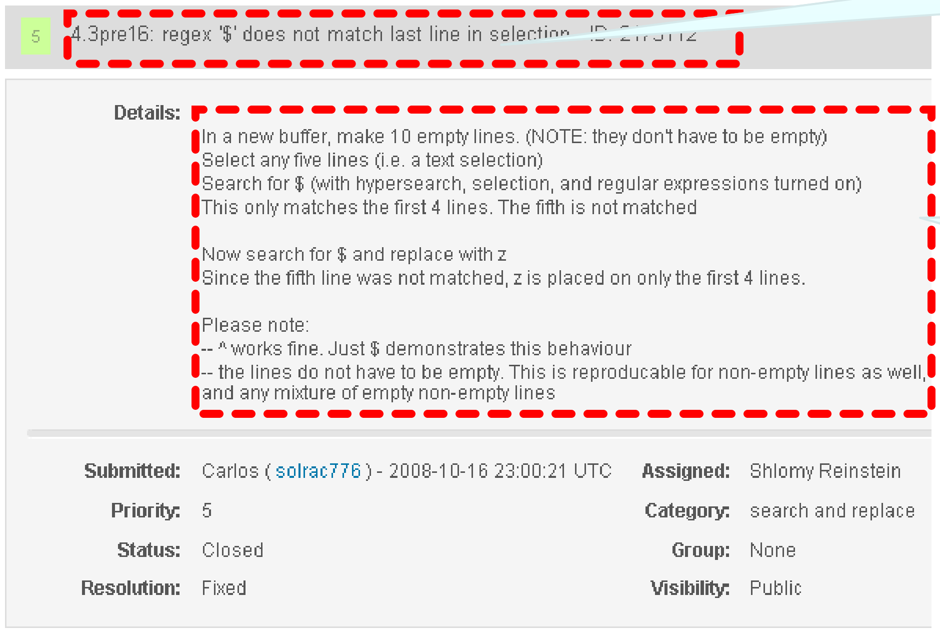
```
------------------------------------------------------------------------
r13898 | shlomy | 2008-10-17 00:45:12 -0400 (Fri, 17 Oct 2008) | 2 lines
Changed paths:
   M /jEdit/trunk/doc/CHANGES.txt
   M /jEdit/trunk/org/gjt/sp/jedit/search/HyperSearchRequest.java

Fixed bug #2173112: Search within selection of regexp '$' does not match
the last line. This was caused by an exception - if the last selection
line is empty, buffer.getSegment() was called with a negative length of
-1. Also removed an unused variable.
------------------------------------------------------------------------
```

Generating the jEdit Datasets



```
------------------------------------------------------------------------
r13898 | shlomy | 2008-10-17 00:45:12 -0400 (Fri, 17 Oct 2008) | 2 lines
Changed paths:
   M /jEdit/trunk/doc/CHANGES.txt
   M /jEdit/trunk/org/gjt/sp/jedit/search/HyperSearchRequest.java

Fixed bug #2173112: Search within selection of regexp '$' does not match
the last line. This was caused by an exception - if the last selection
line is empty, buffer.getSegment() was called with a negative length of
-1. Also removed an unused variable.
------------------------------------------------------------------------
```

SVN commit message



5 4.3pre16: regex '$' does not match last line in selection - ID: 2173112

Details: In a new buffer, make 10 empty lines. (NOTE: they don't have to be empty)
Select any five lines (i.e. a text selection)
Search for $ (with hypersearch, selection, and regular expressions turned on)
This only matches the first 4 lines. The fifth is not matched

Now search for $ and replace with z
Since the fifth line was not matched, z is placed on only the first 4 lines.

Please note:
-- ^ works fine. Just $ demonstrates this behaviour
-- the lines do not have to be empty. This is reproducable for non-empty lines as well,
and any mixture of empty non-empty lines

Submitted:	Carlos (solrac776) - 2008-10-16 23:00:21 UTC	Assigned:	Shlomy Reinstein
Priority:	5	Category:	search and replace
Status:	Closed	Group:	None
Resolution:	Fixed	Visibility:	Public

Title

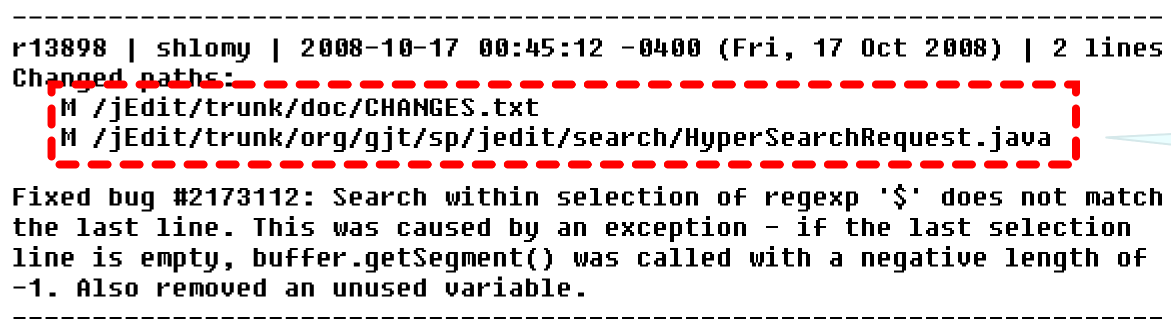
+

Description

=

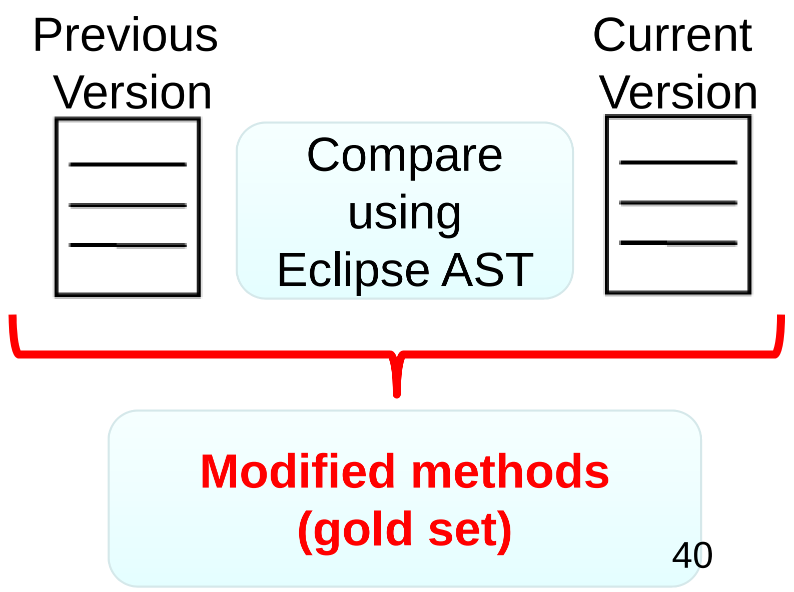
Query

Generating the jEdit Datasets



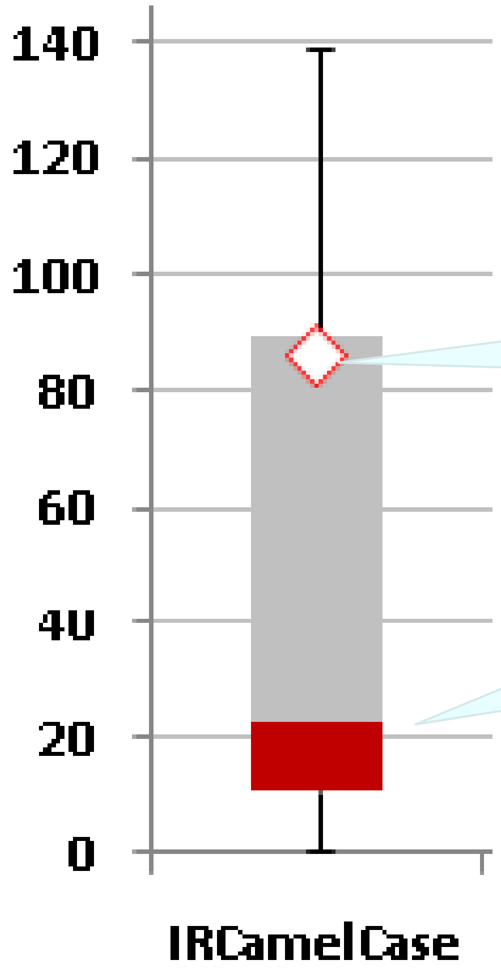
```
------------------------------------------------------------------------
r13898 | shlomy | 2008-10-17 00:45:12 -0400 (Fri, 17 Oct 2008) | 2 lines
Changed paths:
   M /jEdit/trunk/doc/CHANGES.txt
   M /jEdit/trunk/org/gjt/sp/jedit/search/HyperSearchRequest.java

Fixed bug #2173112: Search within selection of regexp '$' does not match
the last line. This was caused by an exception - if the last selection
line is empty, buffer.getSegment() was called with a negative length of
-1. Also removed an unused variable.
------------------------------------------------------------------------
```

Changed files



Presenting the Results

Presenting the Results

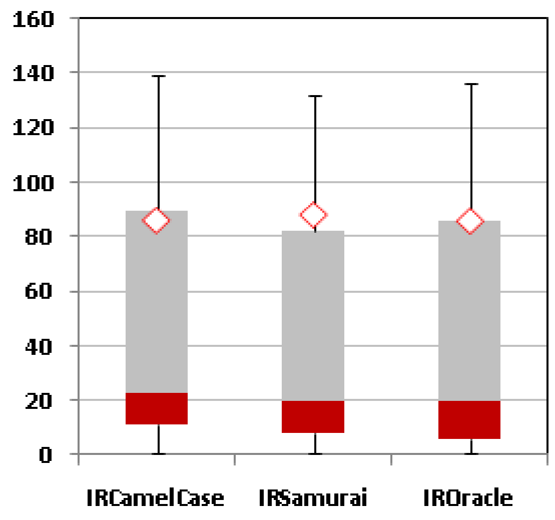


Box plot of all effectiveness measure in datasets
(e.g., 241 datapoints for Rhino$_{Features}$)

Average

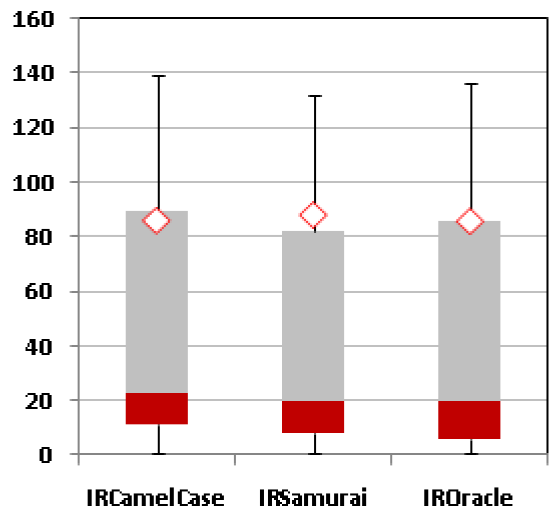
Median

IR FLTs



$\text{Rhino}_{\text{Features}}$

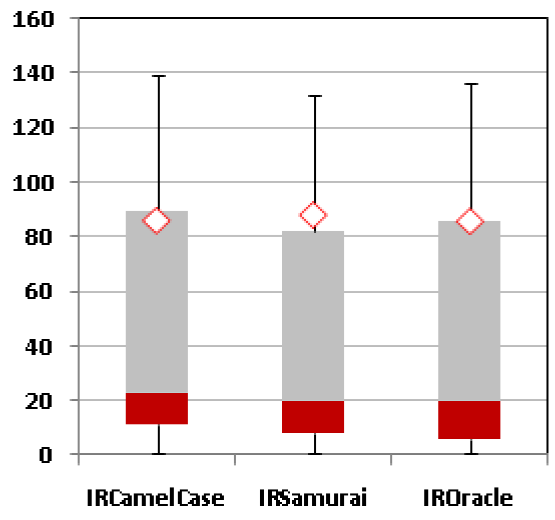
IR FLTs



$\text{Rhino}_{\text{Features}}$

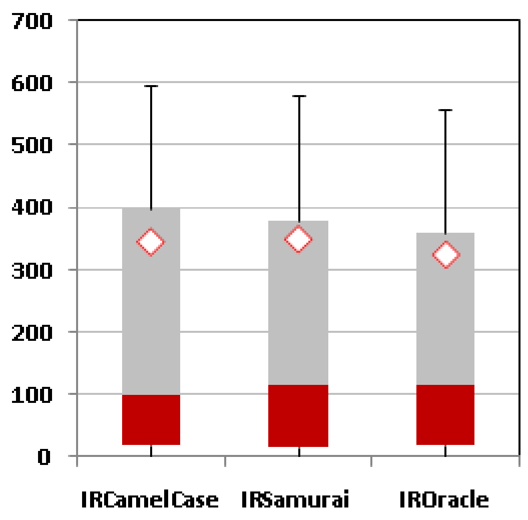
Similar median and average

IR FLTs

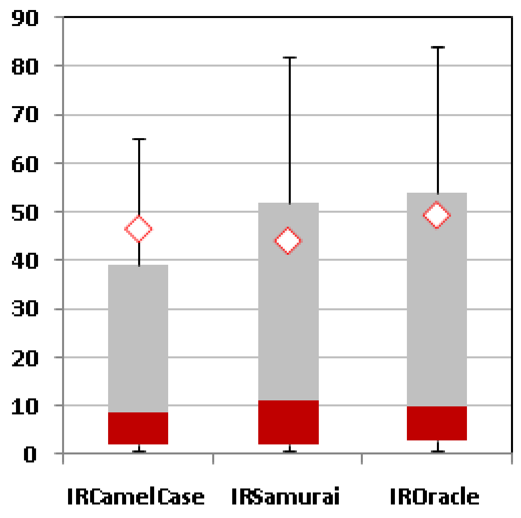


Rhino$_{Features}$

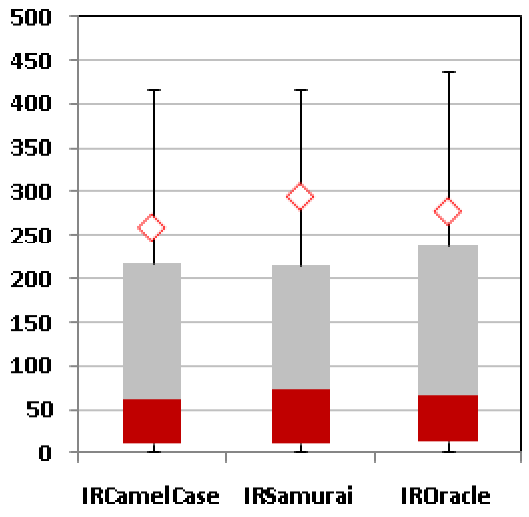
Similar median and average



Rhino$_{Bugs}$

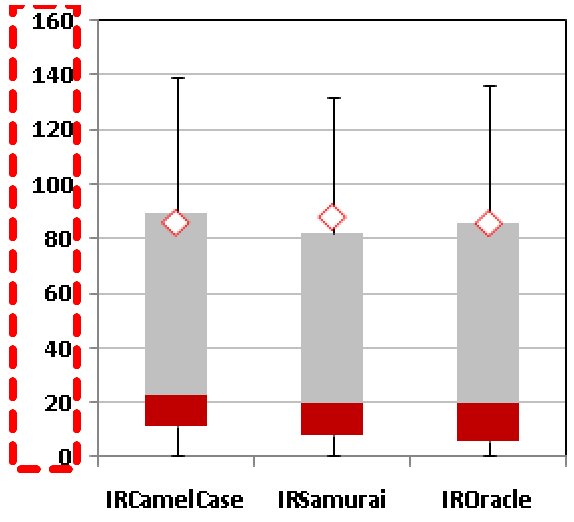


jEdit$_{Features}$



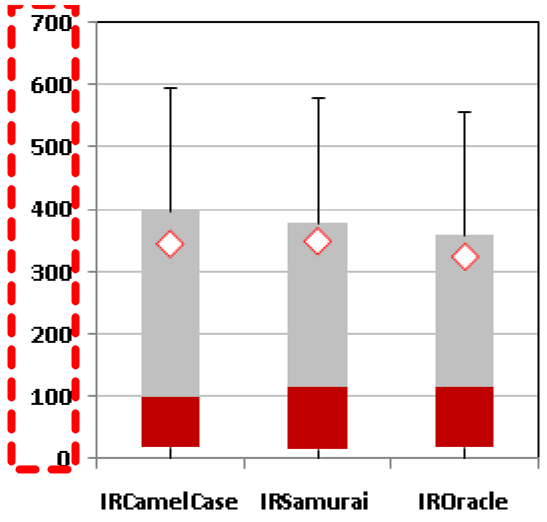
jEdit$_{Bugs}$

IR FLTs

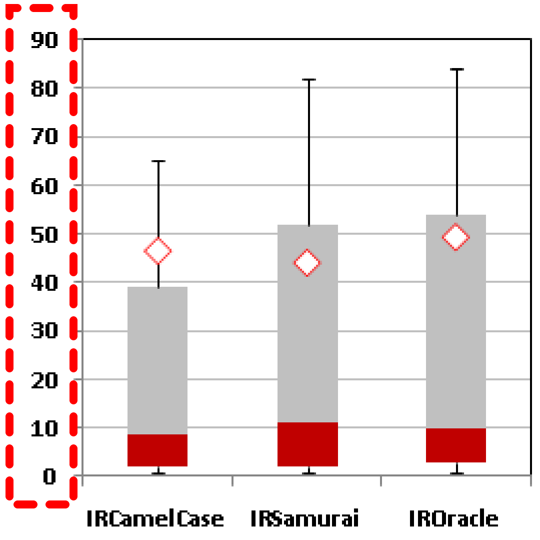


Rhino$_{\text{Features}}$

Similar median and average

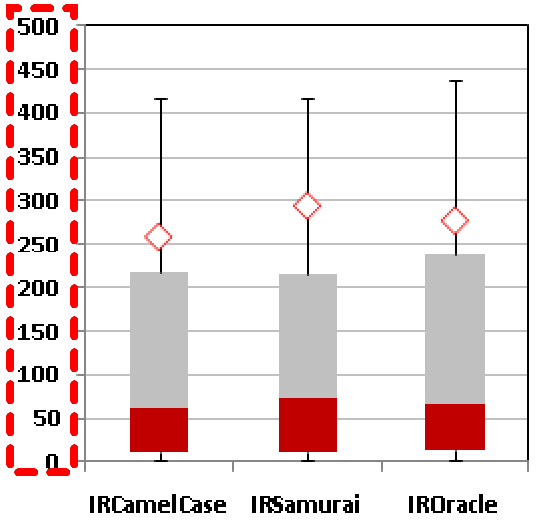


Rhino$_{\text{Bugs}}$



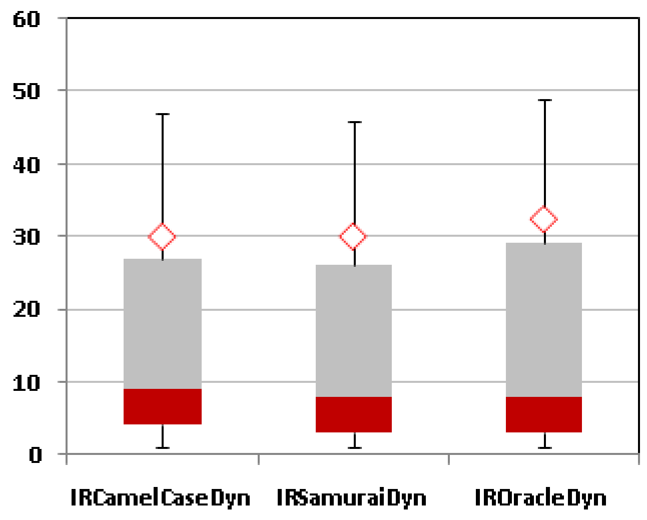
jEdit$_{\text{Features}}$

Datasets with features have better results than datasets with bugs



jEdit$_{\text{Bugs}}$

IRDyn FLTs

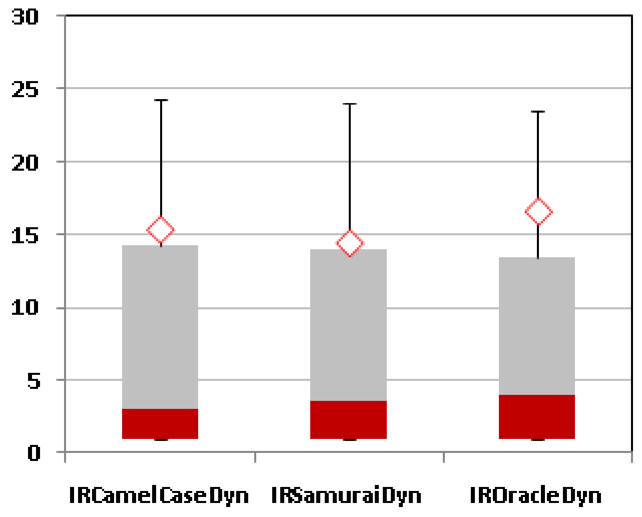


Rhino$_{\text{Features}}$

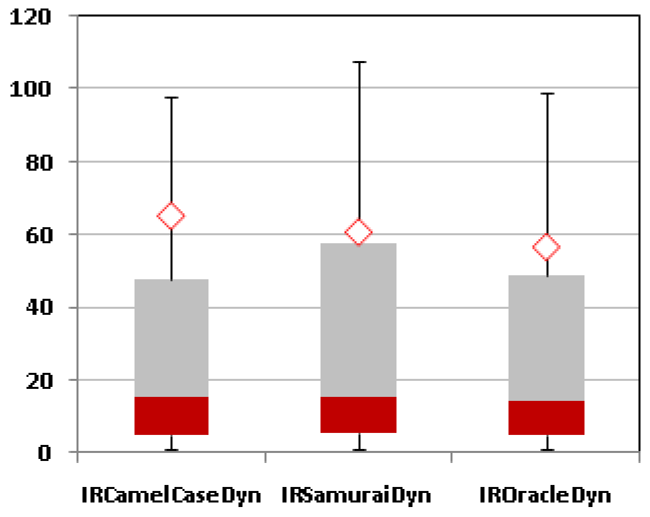
Similar median and average

N/A

Rhino$_{\text{Bugs}}$

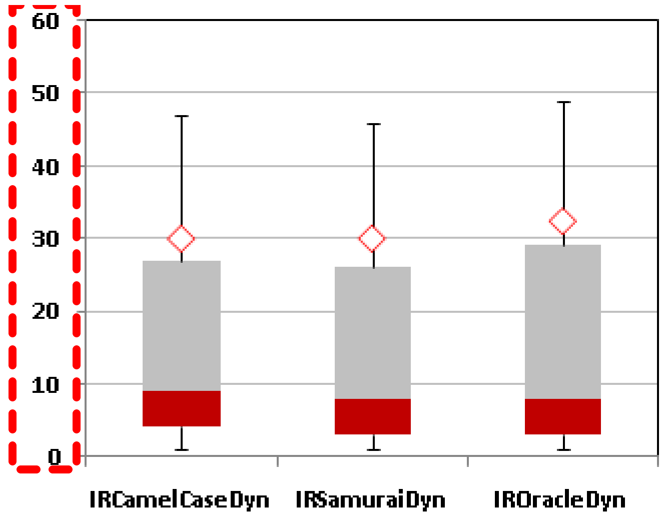


jEdit$_{\text{Features}}$



jEdit$_{\text{Bugs}}$

IRDyn FLTs

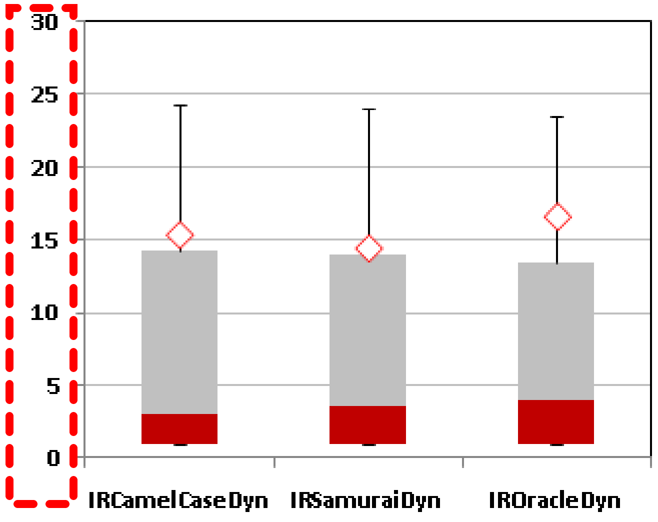


Rhino$_{\text{Features}}$

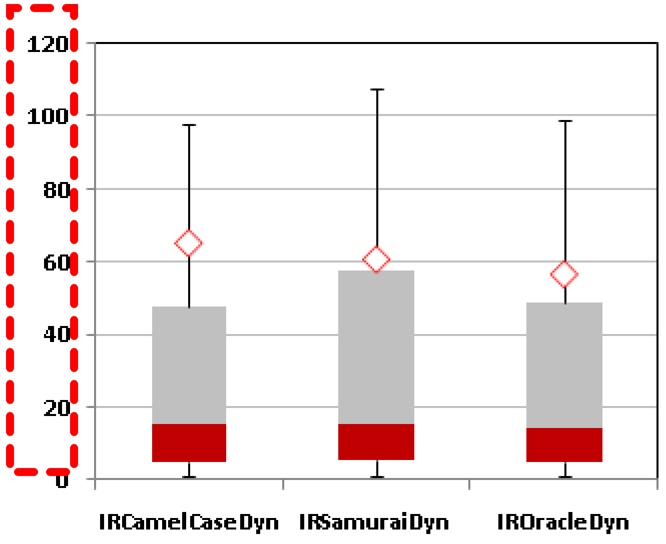
Similar median and average

N/A

Rhino$_{\text{Bugs}}$



Datasets with features have better results than datasets with bugs



jEdit$_{\text{Features}}$

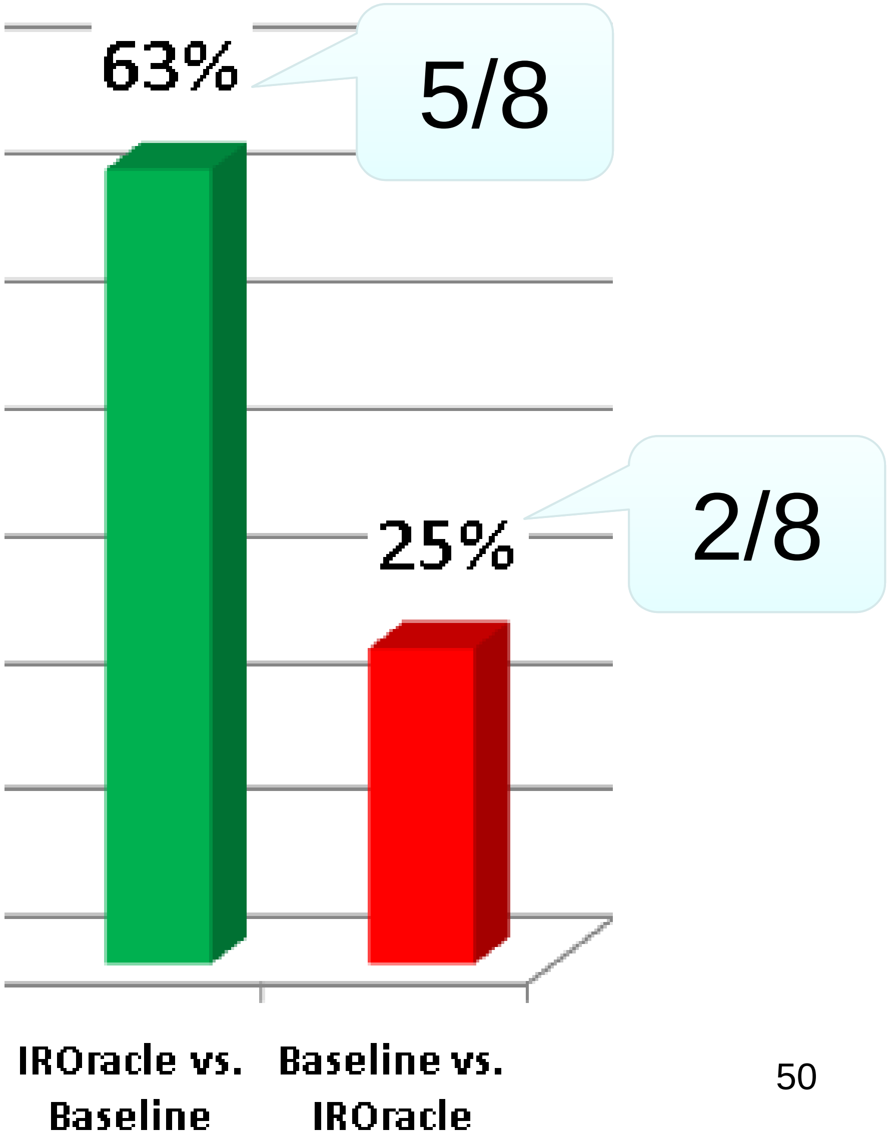
jEdit$_{\text{Bugs}}$

Compare FLTs by Percentages

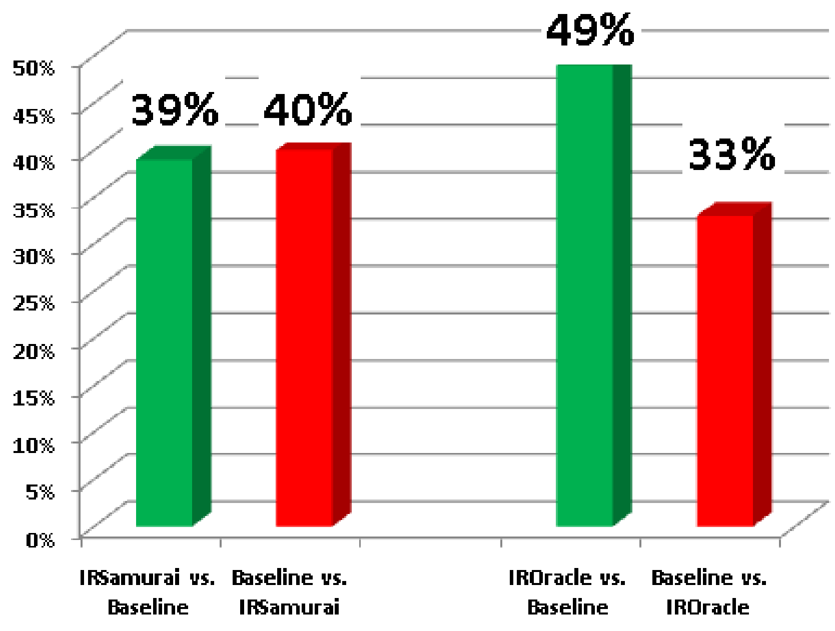
IR_{Oracle}	IRCamelCase (Baseline)
10	17
20	**15**
18	18
5	9
4	16
19	**7**
12	28
14	15

Compare FLTs by Percentages

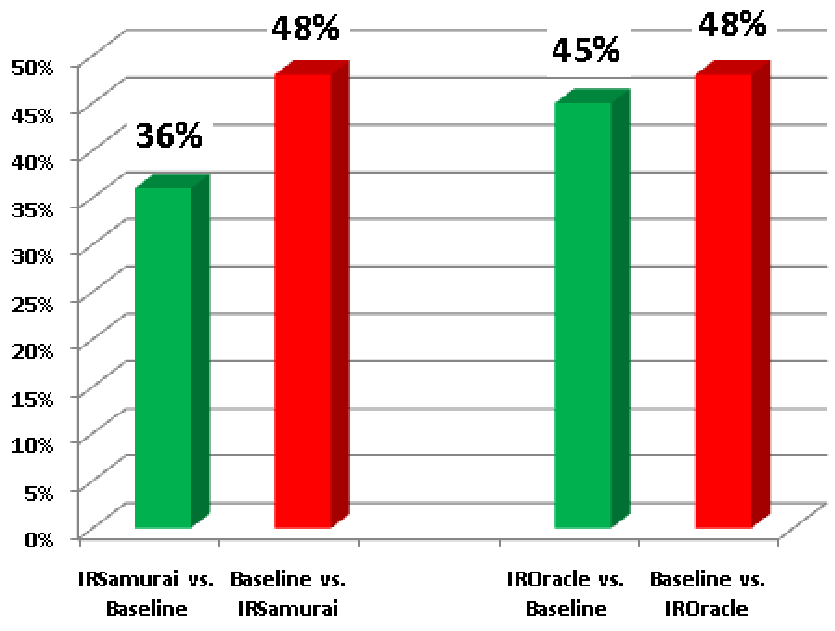
IR_{Oracle}	IRCamelCase (Baseline)
10	17
20	**15**
18	18
5	9
4	16
19	**7**
12	28
14	15



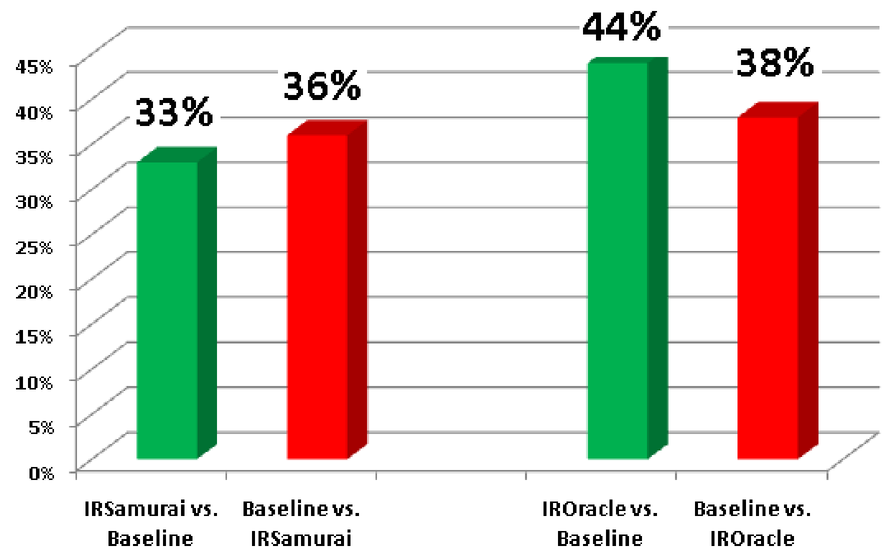
IR



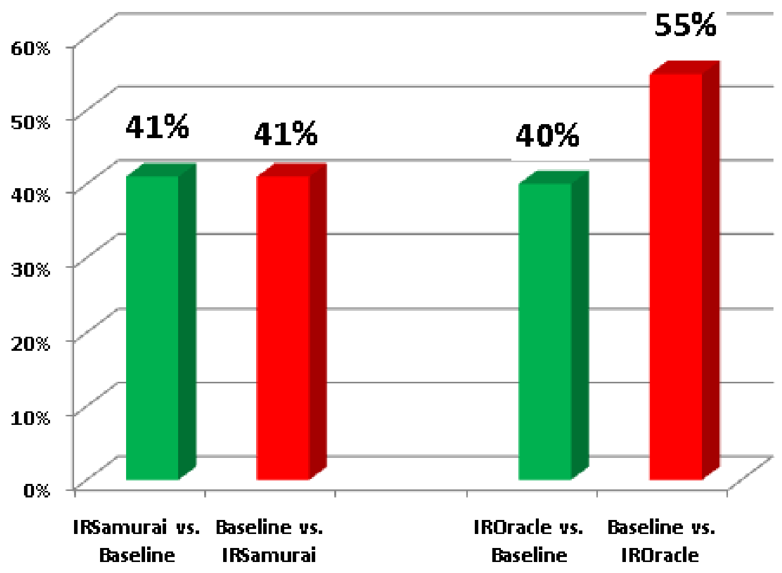
Rhino$_{Features}$



Rhino$_{Bugs}$

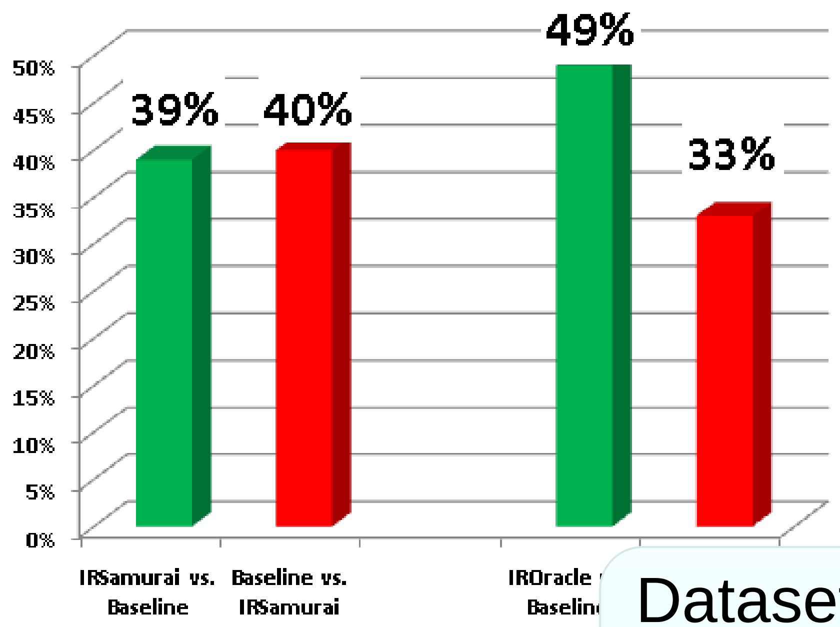


jEdit$_{Features}$

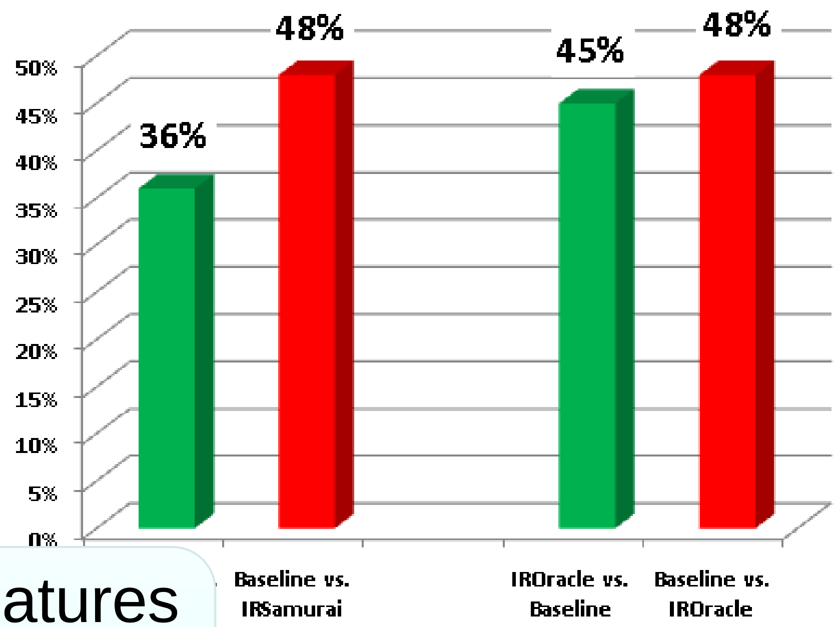


jEdit$_{Bugs}$

IR

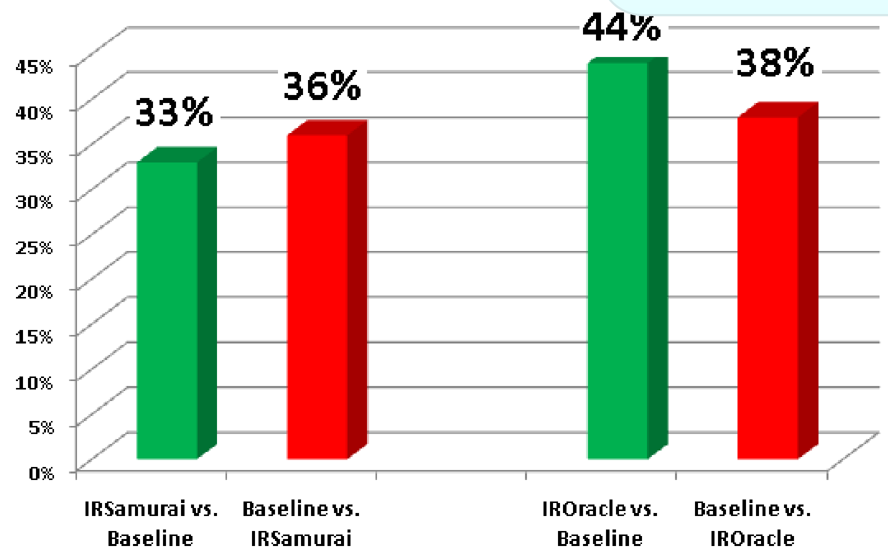


Rhino$_{\text{Features}}$

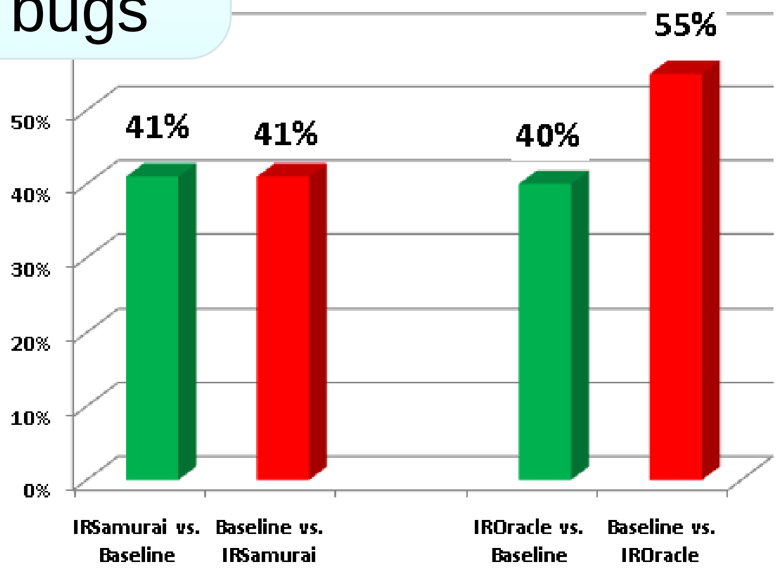


Rhino$_{\text{Bugs}}$

Datasets with features
vs.
Datasets with bugs

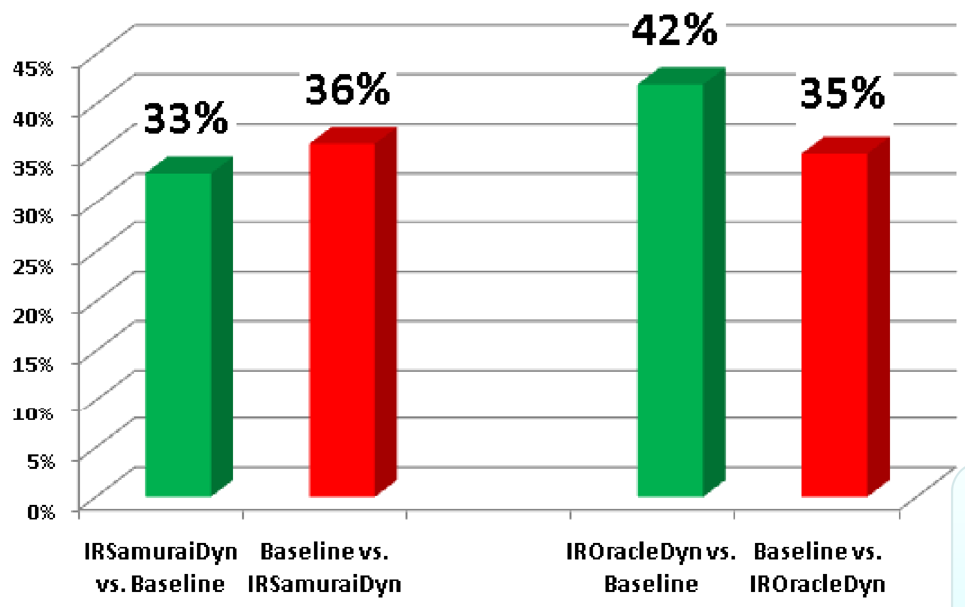


jEdit$_{\text{Features}}$



jEdit$_{\text{Bugs}}$

IRDyn

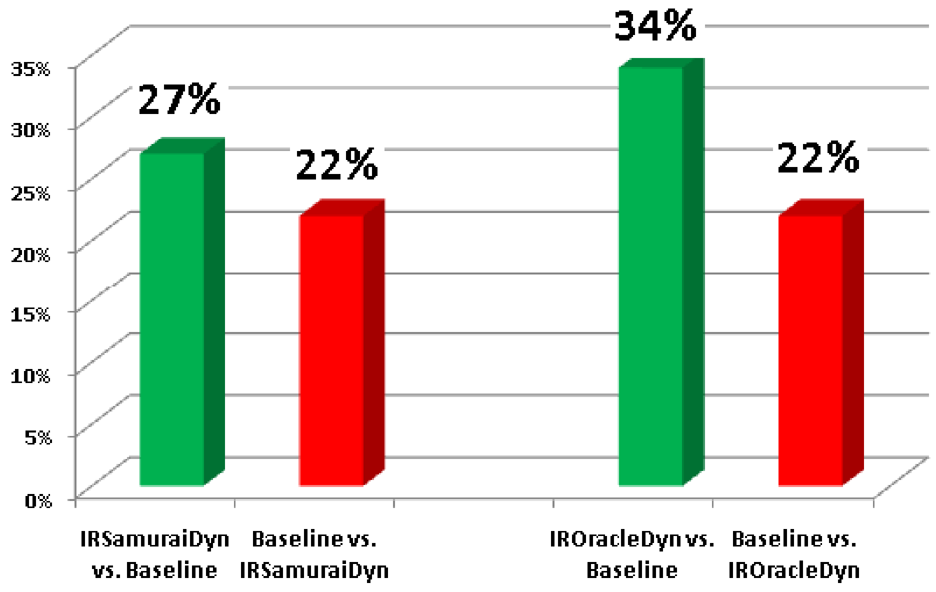


$\text{Rhino}_{\text{Features}}$

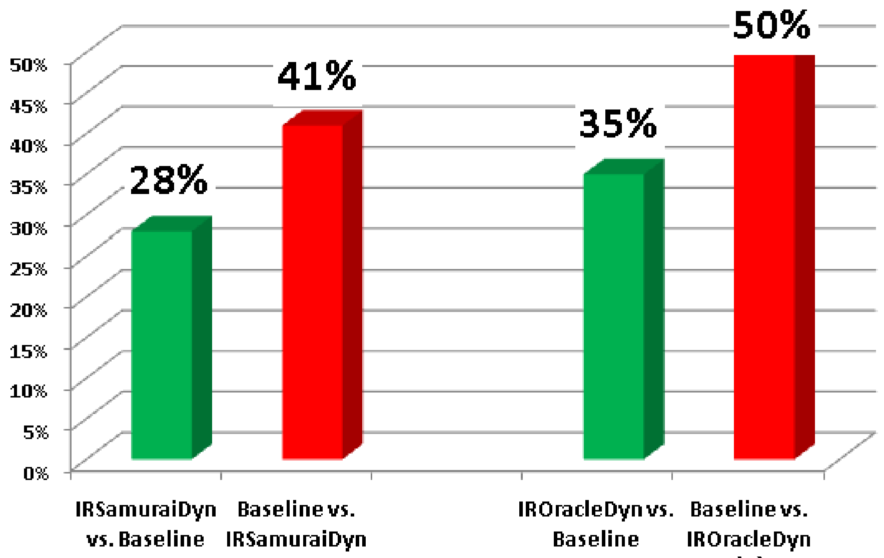
N/A

Similar trend

$\text{Rhino}_{\text{Bugs}}$



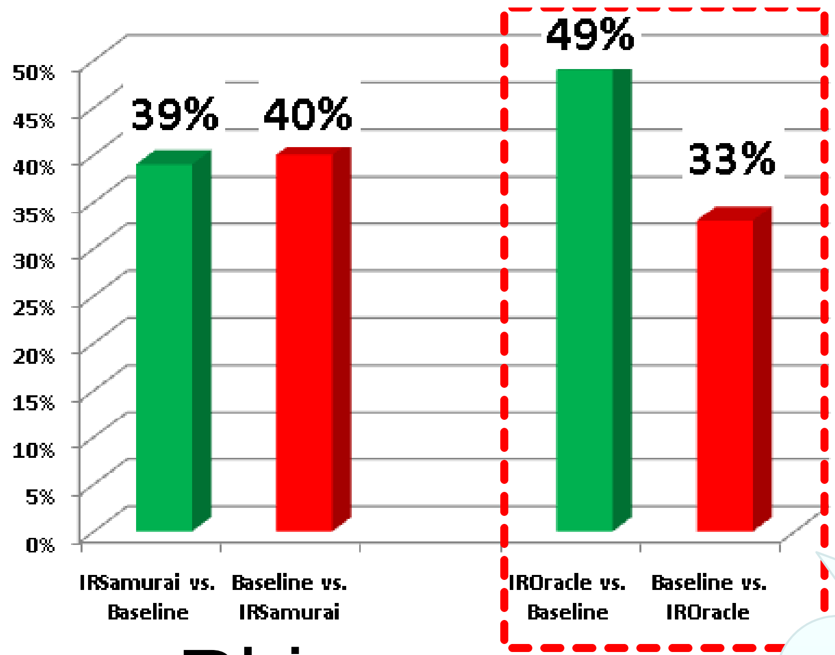
$\text{jEdit}_{\text{Features}}$



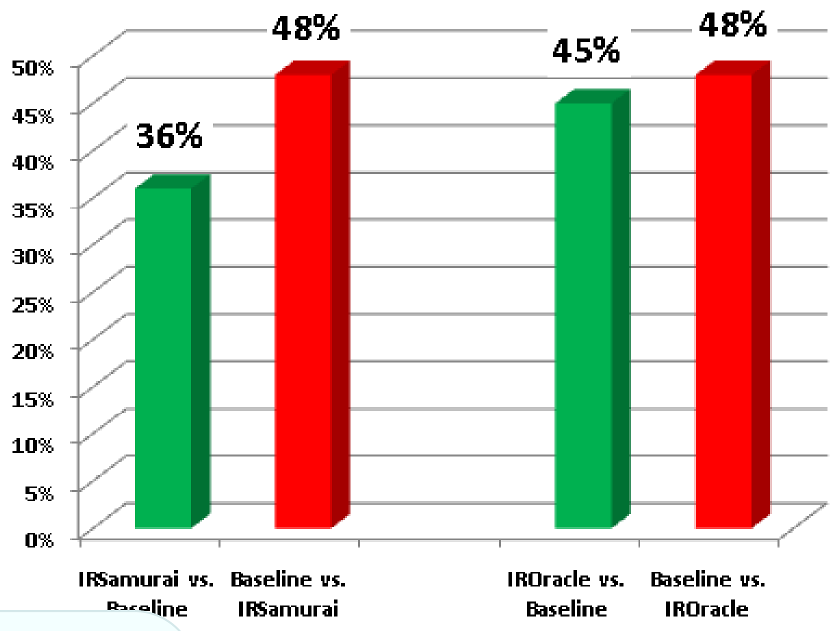
$\text{jEdit}_{\text{Bugs}}$

Statistical Results

- Wilcoxon signed-rank test
- Null hypothesis
  - There is no statistical significance difference in terms of effectiveness between $IR_{Samurai}/IR_{Oracle}$ and $IR_{CamelCase}$
- Alternative hypothesis
  - $IR_{Samurai}/IR_{Oracle}$ has statistically significantly higher effectiveness than $IR_{CamelCase}$
- alpha = 0.05



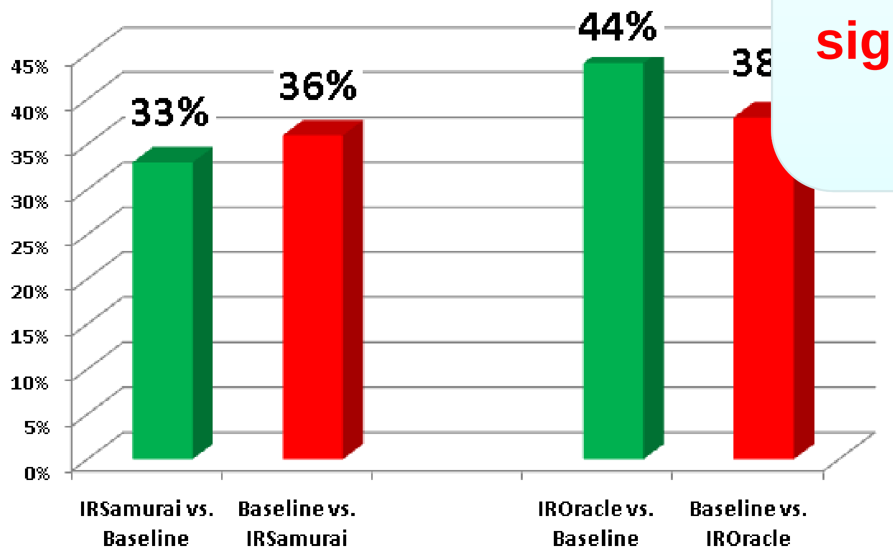
IR

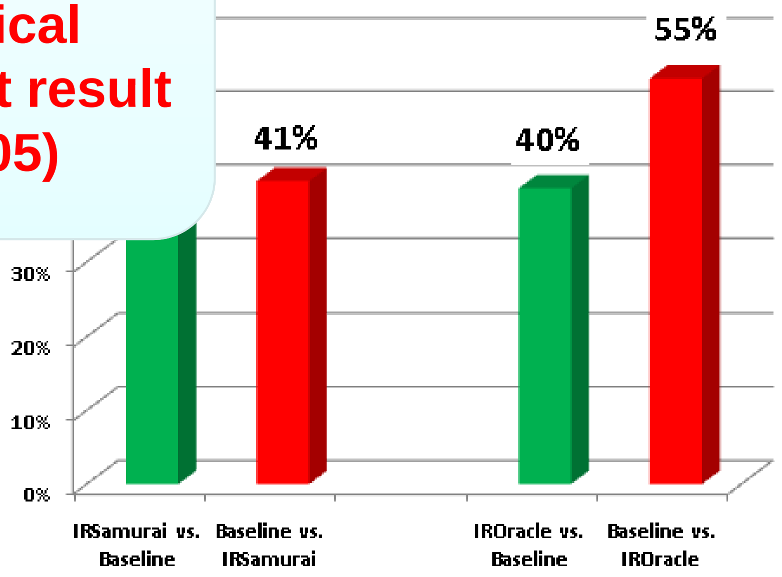


Rhino$_{\text{Features}}$

Rhino$_{\text{Bugs}}$

The only statistical significant result (p=0.05)





jEdit$_{\text{Features}}$

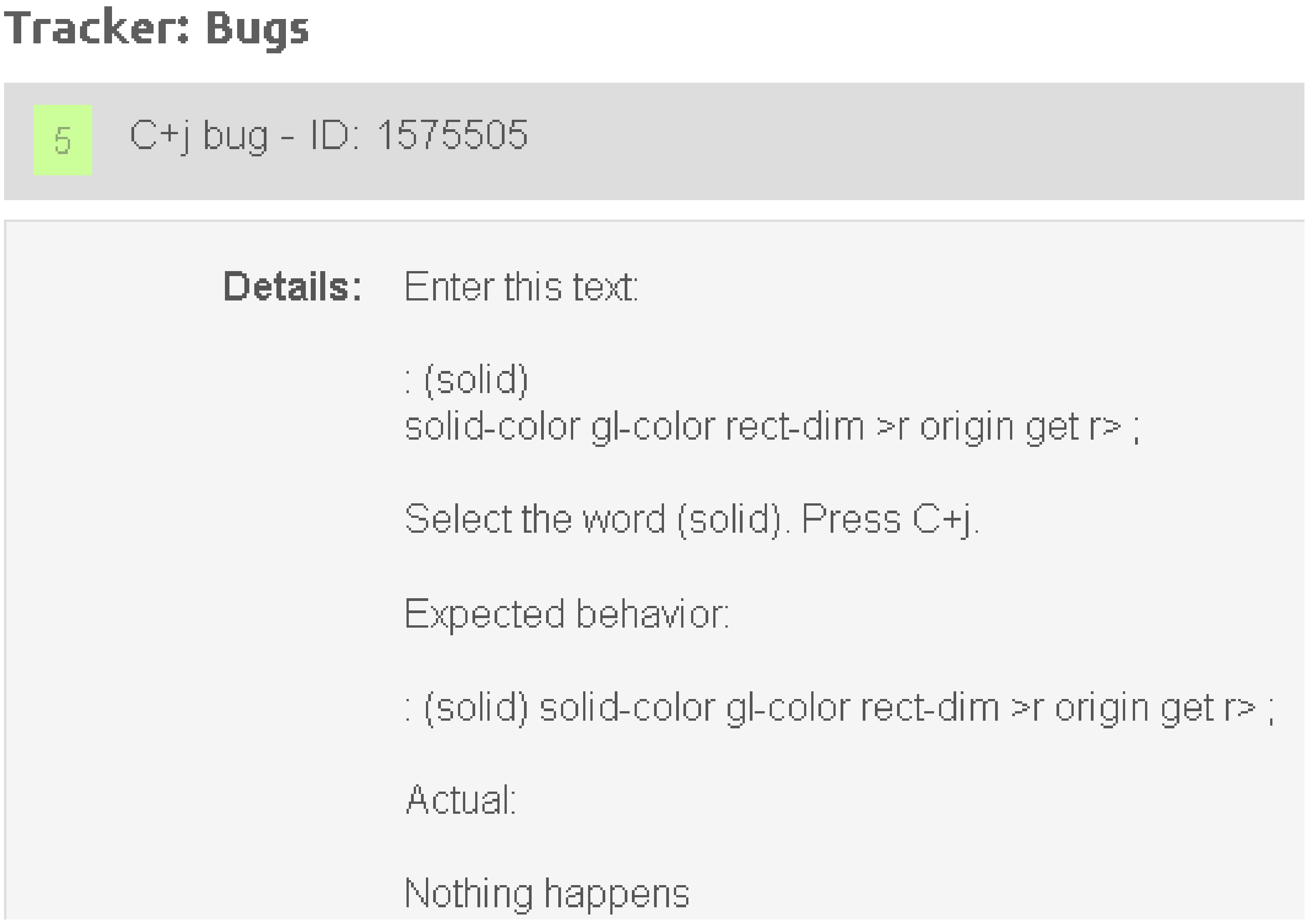
jEdit

Qualitative Results

- Vocabulary mismatch between queries and code:
  - Name of developers (e.g., Slava, Carlos)
  - Identifiers specific to communication (e.g., thanks, greetings, annoying)

Qualitative Results

- Features are more "descriptive" than bugs



Tracker: Bugs

5 C+j bug - ID: 1575505

Details: Enter this text:

: (solid)
solid-color gl-color rect-dim >r origin get r> ;

Select the word (solid). Press C+j.

Expected behavior:

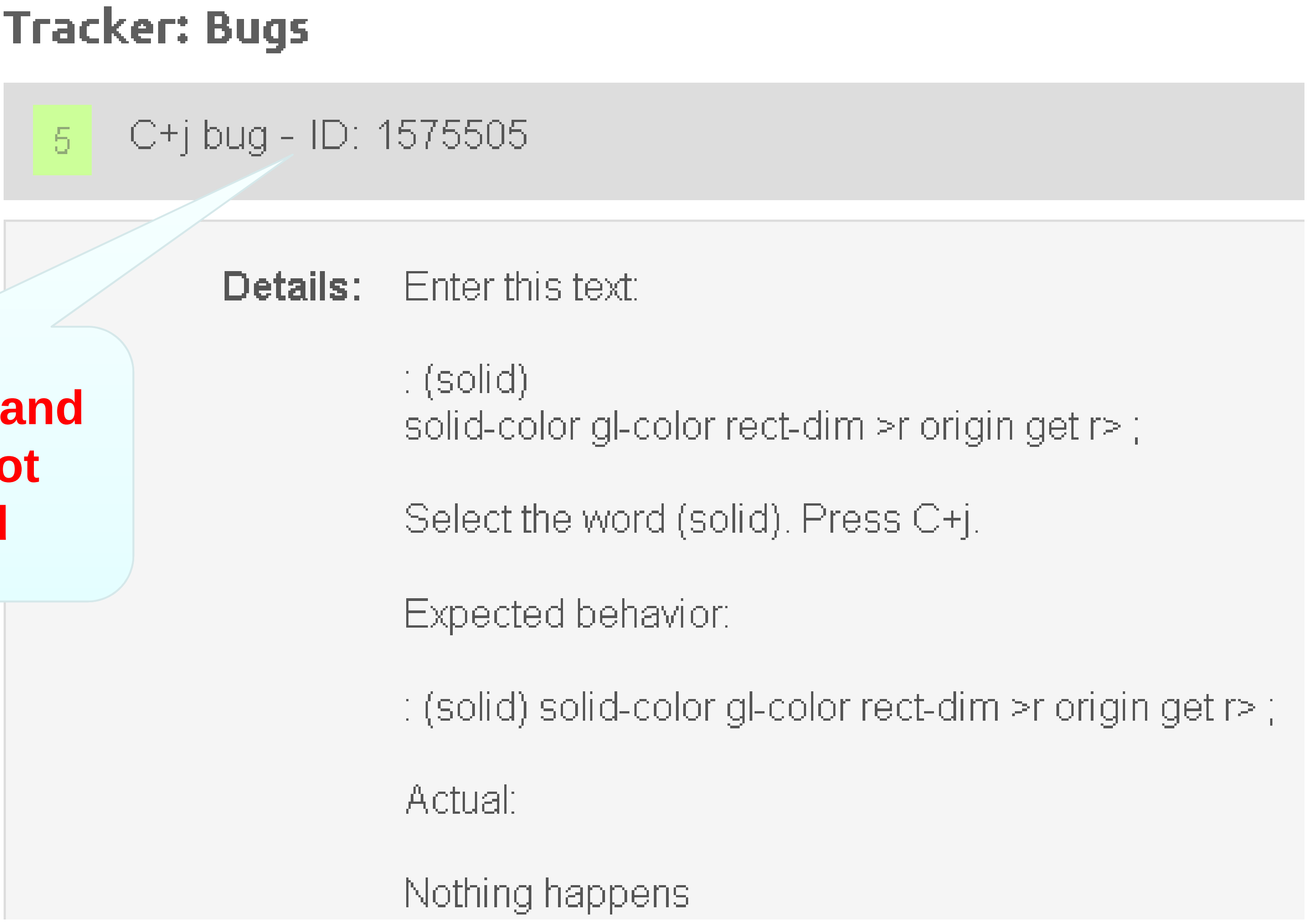
: (solid) solid-color gl-color rect-dim >r origin get r> ;

Actual:

Nothing happens

Qualitative Results

- Features are more “descriptive” than bugs



Tracker: Bugs

5 C+j bug - ID: 1575505

Details: Enter this text:

: (solid)
solid-color gl-color rect-dim >r origin get r> ;

Select the word (solid). Press C+j.

Expected behavior:

: (solid) solid-color gl-color rect-dim >r origin get r> ;

Actual:

Nothing happens

Words “join” and “line” are not mentioned

Threats to Validity

- External
  - 2 Java applications (different domains)
  - More systems needed
- Construct
  - Errors may be present in Oracle and gold sets
  - We used data produced by other researchers
- Internal
  - Subjectivity and bias in building the Oracle
- Conclusion
  - Non-parametric test: Wilcoxon signed-rank

Research Questions

- RQ1 Does $IR_{Samurai}$ outperform $IR_{CamelCase}$ in terms of effectiveness? NO
- RQ2 Does $IR_{Samurai}Dyn$ outperform $IR_{CamelCase}Dyn$ in terms of effectiveness? NO
- RQ3 Does IR_{Oracle} outperform $IR_{CamelCase}$ in terms of effectiveness? In some cases (Rhino)
- RQ4 Does $IR_{Oracle}Dyn$ outperform $IR_{CamelCase}Dyn$ in terms of effectiveness? NO

Future Work

- More systems and datasets
- Different maintenance tasks
  - Traceability link recovery
- Consider other splitting algorithms

Conclusions

- Advanced splitting technique could improve FLTs
  - We found some empirical evidence
- Splitting has more impact on IR FLT
- If execution information is available, it is not necessary to use an advance splitting technique

Thank you! Questions?

SEMERU @ William and Mary

http://www.cs.wm.edu/semeru/

bdit@cs.wm.edu

References

- Takang et al. (1996) Takang, A., Grubb, P., and Macredie, R., "The Effects of Comments and Identifier Names on Program Comprehensibility: An Experimental Investigation", Journal of Programming Languages, vol. 4, no. 3, 1996, pp. 143-167
- Lawrie et al. (2006) Lawrie, D., Morrell, C., Feild, H., and Binkley, D., "What's in a Name? A Study of Identifiers", in Proc. of IEEE ICPC'06, June 14-16 2006, pp. 3-12
- Binkley et al. (2009) Binkley, D., Davis, M., Lawrie, D., and Morrell, C., "To CamelCase or Under_score", in Proc. of IEEE ICPC'09, May 17-19 2009, pp. 158-167
- Enslen et al. (2009) Enslen, E., Hill, E., Pollock, L., and Vijay-Shanker, K., "Mining Source Code to Automatically Split Identifiers for Software Analysis", in Proc. of IEEE MSR'09, May 16-17 2009, pp. 71-80
- Guerrouj et al. (2011) Guerrouj, L., Di Penta, M., Antoniol, G., and Guéhéneuc, Y.-G., "TIDIER: An Identifier Splitting Approach using Speech Recognition Techniques", JSME, vol. to appear, 2011