An Introduction to Bug Report Management Techniques

Xin Xia (夏鑫)

xxia@zju.edu.cn

August 5, 2016

The First Recorded Bug!

9/9

0800 Arctan started
1000 " stopped - arctan ✓ { 1.2700 9.037 847 025
9.037 846 795 correct
13 u/c (032) MP - MC ~~2.130476915 (033)~~ 4.615925059(-2)
(033) PRO 2 2.130476415
correct 2.130676415
Relays 6-2 in 033 failed special speed test
in relay " 11.00 test.
Relays changed
1100 Started Cosine Tape (Sine check)
1525 Started Multy Adder Test.
1545 
Relay #70 Panel F
(moth) in relay.
First actual case of bug being found.
1630 Arctangent started.
1700 closed down.

Bugs are inevitable!

The Impact of Software Bugs!

- 2002, software errors cost U.S. economy $59.5 billion (NIST)
- Cost of debugging in a software system consumes 50% - 80% of the development and maintenance cost

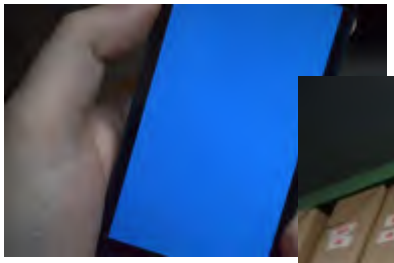
The number of bugs are increased!

- Eclipse: more than 400,000 bug reports.
- In 2005, Eclipse would receive 200 new bug reports every day.
- In the

We need some automated techniques to manage these bug reports!

Why need manage bug reports?

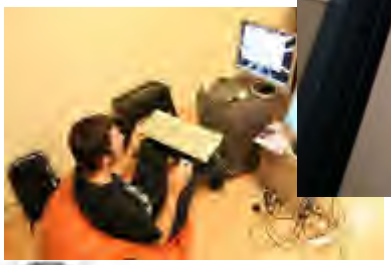
Product Users



Bug Tracking System!

duct Developers

Product Teste



Why textual analysis?

- Three types of analyses in Software Engineering:
  - Static Analysis: leverage structural information such as control or data flow dependencies.
  - Dynamic Analysis: leverage runtime behaviours
  - Textual Analysis: leverage text information in documents.
- Bug report is provided by human.
- Analysing these textual information helps developers comprehend bugs and improves bug reports management process.

Background

- An example of a bug report
- Lifecycle of a bug report
- Research problems

Sample Bug Report

Bug 822256 - Robocop: Add test for 'System Pages' feature

Title/Summary

Attachments

SystemPages test V9 (6.58 KB, patch) gbrown: review+ Details | Diff | Review
2013-02-01 05:03 PST, Paul Feher

Add an attachment (proposed patch, testcase, etc.) Show Obsolete (10) View All

screenshot or patch

Summon comment box

Paul Feher 2012-12-17 07:09:47 PST Description [reply] [-]

This bug tracks the Robocop tests part of System Pages feature on Firefox for Android.
The tests check first the loading of system pages from the awesome bar and then from Firefox menu.

about:rights https://moztrap.mozilla.org/manage/case/1029/
Opening a local page (about:) in another local page should open in the same tab:
https://moztrap.mozilla.org/manage/case/1030/
The search input is not in focus https://moztrap.mozilla.org/manage/case/1033/
about:home and about:firefox: https://moztrap.mozilla.org/manage/case/1027/

Description

Paul Feher 2012-12-18 04:46:42 PST Comment 1 [reply] [-]

Created ~~attachment 693333 [details] [diff]~~ [review]
SystemPages test V1

Modified the selectMenuItem method in BaseTest.java.in to cover the changes of the menu layout (the tools submenu) on ICS+ devices.
Created tests to cover the tests done on system pages.

Comments

Description Paul Feher 2012-12-17 07:09:47 PST

This bug tracks the Robocop tests part of System Pages feature on Firefox for Android.

The tests check first the loading of system pages from the awesome bar and then from Firefox menu.
about:rights https://moztrap.mozilla.org/manage/case/1029/
Opening a local page (about:) in another local page should open in the same tab: https://moztrap.mozilla.org/manage/case/1030/

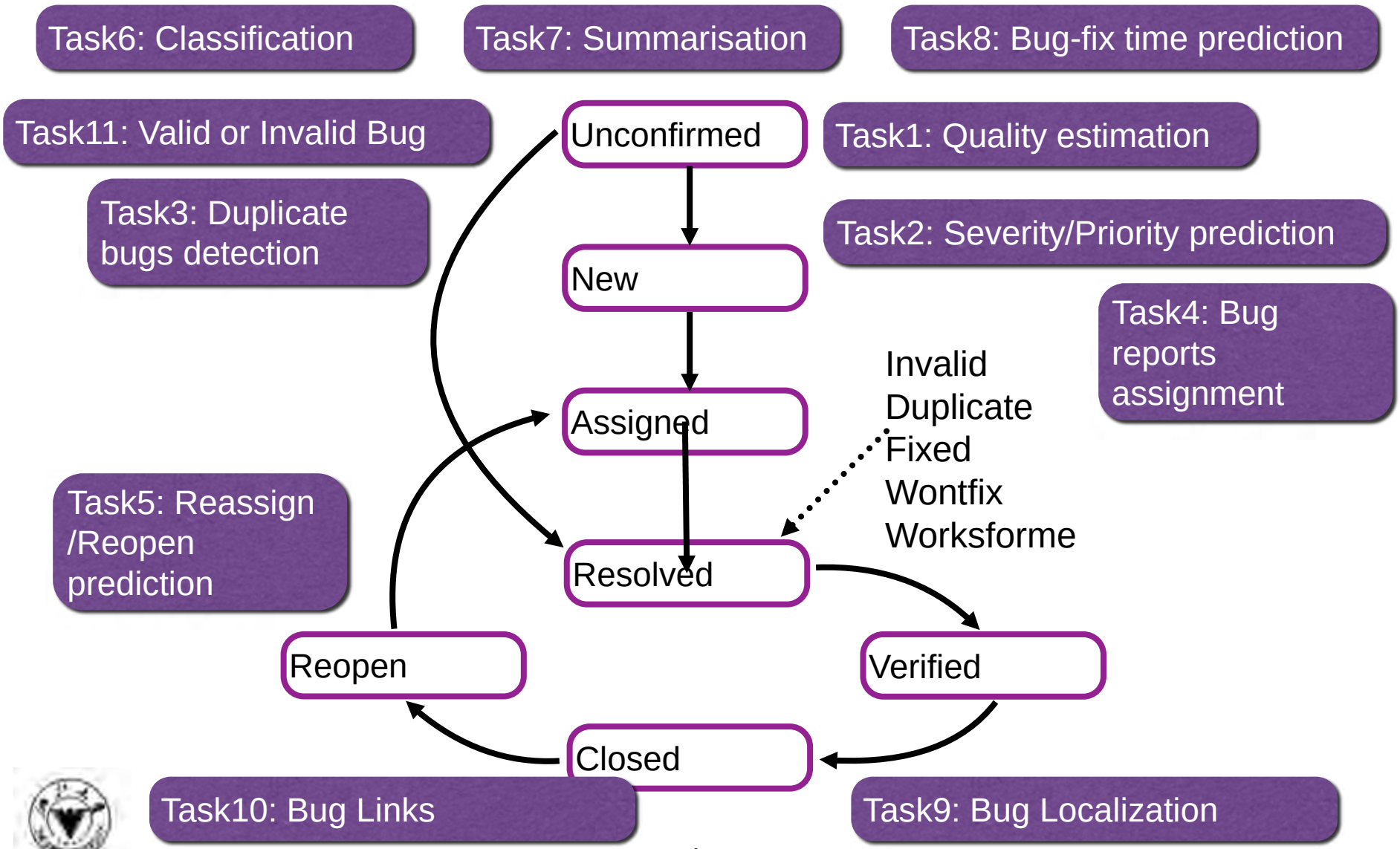
Analysing textual information inside bug reports is easy or hard?

- Easy !
  - Large textual information in software repository.
    - 641K in Mozilla, 18K in Linux, 7K in Apache.
  - Existing techniques from NLP, Information Retrieval, Machine Learning.

Hard !

- Domain specific words/phrases, and meanings (e.g, call a function Vs call a friend)
- Poor quality text
  - Inconsistent
  - Grammar mistakes
  - Incomplete information
- Contain stack trace, source code, link

Lifecycle of a Bug Report



Research Problems

1. Quality estimation
2. Severity/Priority prediction
3. Duplicate bugs detection
4. Bug reports assignment
5. Predict reassign/reopen bug reports
6. Bug reports classification
7. Bug reports summarisation
8. Bug-fix time prediction
9. Information Retrieval Based Bug Localization
10. Bug Links Prediction
11. Valid or Invalid Bug

Research problems & Techniques

- Preprocessing
- Techniques for different research problems

Preprocessing

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<bugzilla version="4.4" urlbase="https://bugs.eclipse.org/bugs/" maintainer="webmaster@eclipse.org">
  ▼<bug>
    <bug_id>84936</bug_id>
    <creation_ts>2005-02-10 15:03:00 -0500</creation_ts>
   ▼<short_desc>
      AbstractDecoratedTextEditor crashes in createPartControl method
    </short_desc>
    <delta_ts>2005-02-18 04:48:31 -0500</delta_ts>
    <reporter_accessible>1</reporter_accessible>
    <cclist_accessible>1</cclist_accessible>
    <classification_id>2</classification_id>
    <classification>Eclipse</classification>
    <product>Platform</product>
    <component>Text</component>
    <version>3.1</version>
    <rep_platform>PC</rep_platform>
    <op_sys>Windows XP</op_sys>
    <bug_status>VERIFIED</bug_status>
    <resolution>FIXED</resolution>
    <bug_file_loc/>
    <status_whiteboard/>
    <keywords/>
    <priority>P2</priority>
    <bug_severity>normal</bug_severity>
    <target_milestone>3.1 M5</target_milestone>
    <everconfirmed>1</everconfirmed>
    <reporter name="Alex Chapiro">achapiro</reporter>
    <assigned_to name="Platform-Text-Inbox">platform-text-inbox</assigned_to>
    <votes>0</votes>
    <comment_sort_order>oldest_to_newest</comment_sort_order>
   ▼<long_desc isprivate="0">
      <commentid>398408</commentid>
      <comment_count>0</comment_count>
      <who name="Alex Chapiro">achapiro</who>
      <bug_when>2005-02-10 15:03:04 -0500</bug_when>
     ▼<thetext>
        I just don't assign preference store to my editor, and get NullPointerException because program does not check it:
        if(getPreferenceStore().getBoolean(AbstractDecoratedTextEditorPreferenceConstants.EDITOR_DISABLE_OVERWRITE_MODE))
        enableOverwriteMode(false); I havent find out that preference store is mandatory.
      </thetext>
    </long_desc>
   ▼<long_desc isprivate="0">
```

- Stemming

Task1: Severity & Priority Prediction

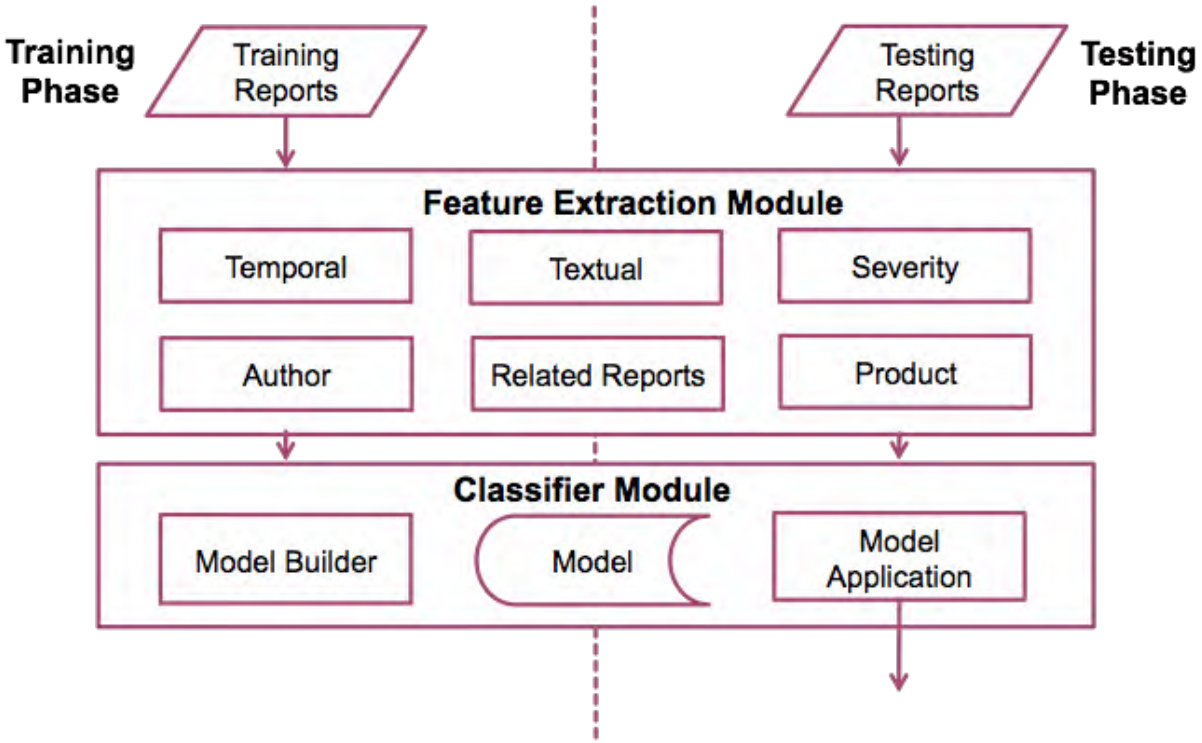
- Specific:
  - Imbalance
  - Ordinal class
  - Multi-class
- Solutions:
  - Extract features+ select feature(option) + classifiers (binary/multi-class)
  - Compute similarities + kNN
- Measurement:
  - Precision/recall/F-measure
  - Speed of achieving stable results
  - Top-feature.

Priority Prediction

Tian et al. DRONE: Predicting Priority of Reported Bugs by Multi-Factor Analysis, ICSM'13.

- Used Information:

- Summary, description, product, component, severity, similar bug reports.



Task2: Duplicate Bugs Detection

Definition:

- Given a bug report/natural language query, detect whether there are existing bug reports report the same bug or not.
- Given a bug report, detect whether it is a duplicate bug report or not.

Motivation:

- Bug reporting in nature is an uncoordinated distributed process.
- Different users or developers might report the same bug.
- Duplicate bug reports cost extra time to triage and fix.

IR-based, bug report as input

Sun et al., Towards More Accurate retrieval of duplicate bug reports, ASE'11.

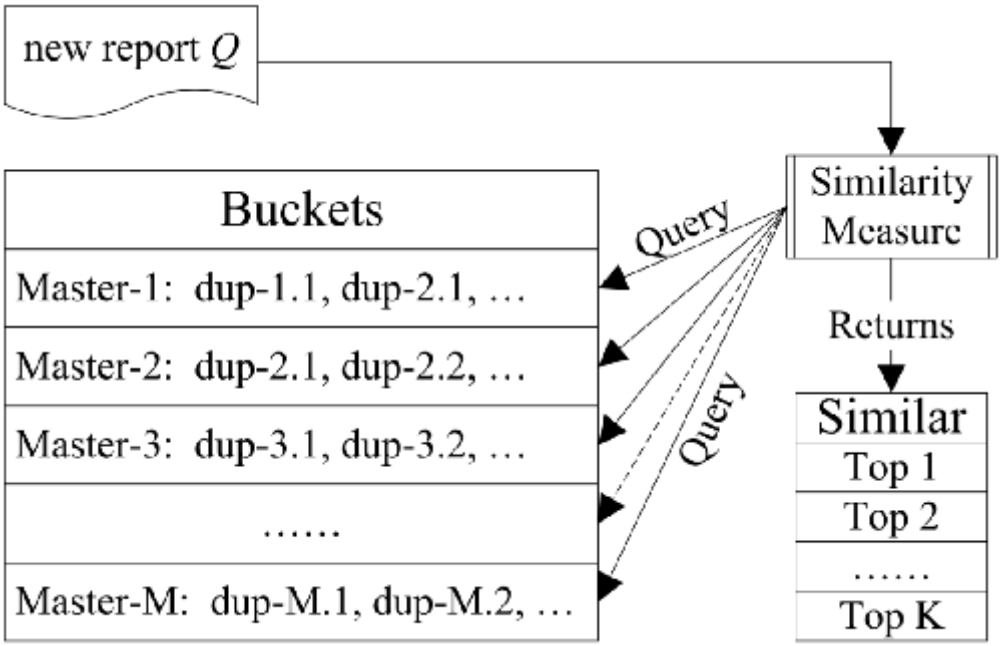
Task:

Retrieve top-N buckets with highest similarity.

Idea:

Bug repository has a hashmap-like data structure, each bucket present a bug.

- Key: master reports
- Value: corresponding duplicate reports



IR-based, bug report as input

Sun et al., Towards More Accurate retrieval of duplicate bug reports, ASE'11.

Used Information:

Unigram/bigram in summary, description, product, component, type, priority, version.

Approach:

- Modified BM25F for long query
- Proposed REP by considering structure of bug reports
- Tuning parameters using Gradient Descent.

$$BM25F_{ext}(d,q) = \sum_{t \in d \cap q} IDF(t) \times \frac{TF_D(d,t)}{k_1 + TF_D(d,t)} \times W_Q$$

$$\text{where } W_Q = \frac{(k_3+1) \times TF_Q(q,t)}{k_3 + TF_Q(q,t)} \quad (4)$$

$$TF_Q(q,t) = \sum_{f=1}^{K} w_f \times occurrences(q[f],t) \quad (5)$$

$$REP(d,q) = \sum_{i=1}^{7} w_i \times feature_i$$

Task3: Bug Reports Assignment

Definition:

Given a new bug report, find the most appropriate fixer.

Motivation:

Developers find it's hard to allocate hundreds of new bug reports per day.

Key point:

- How to define the suitable developer?
  - Expert in related domain
  - Developer who has fixed similar bug report
  - Expert + time cost

Task3: Bug Reports Assignment

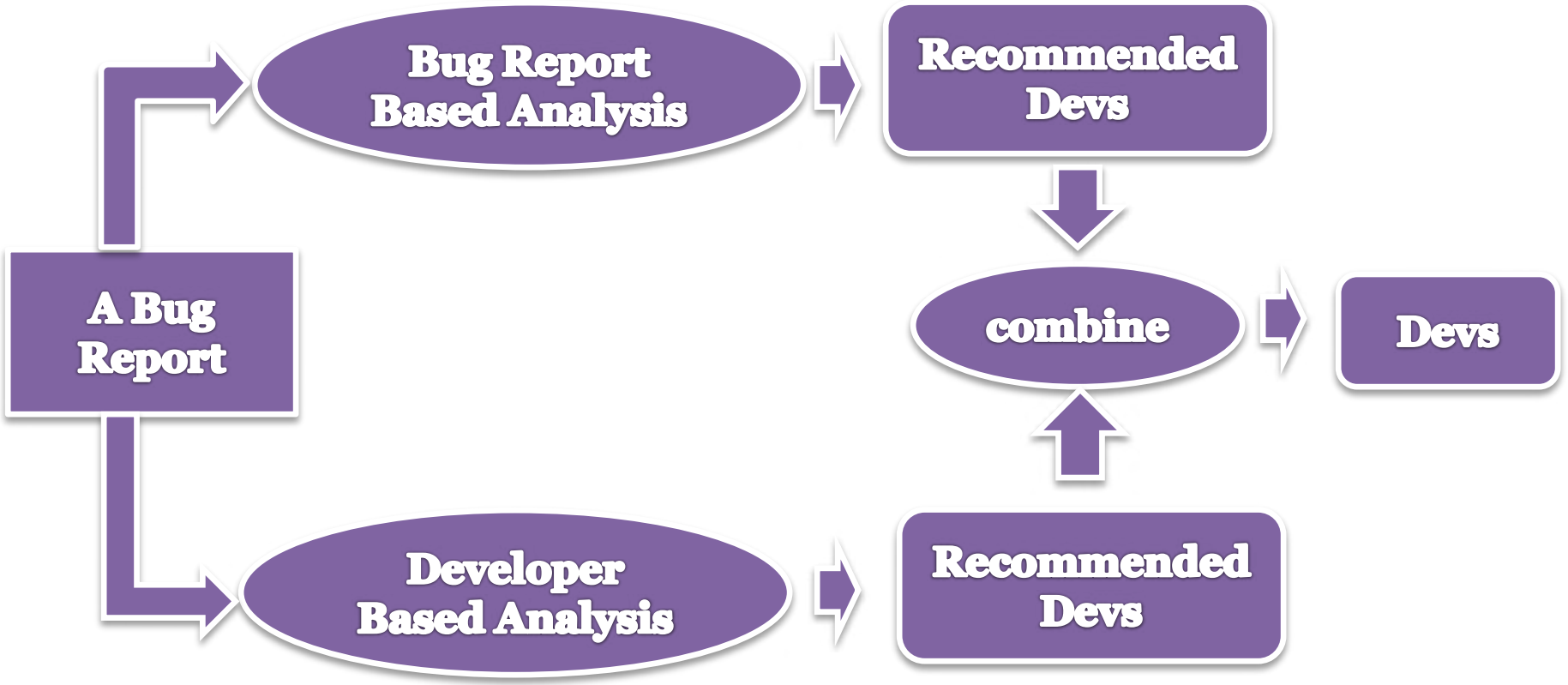
General Approach:

- Machine Learning (feature+classifiers for each developer)
- Information Retrieval
  - Compute similarities between new bug report and developer profiles
  - Fuzzy set theory

Accurate Developer Recommendation for Bug Resolution

WCRE 2013, JSEP

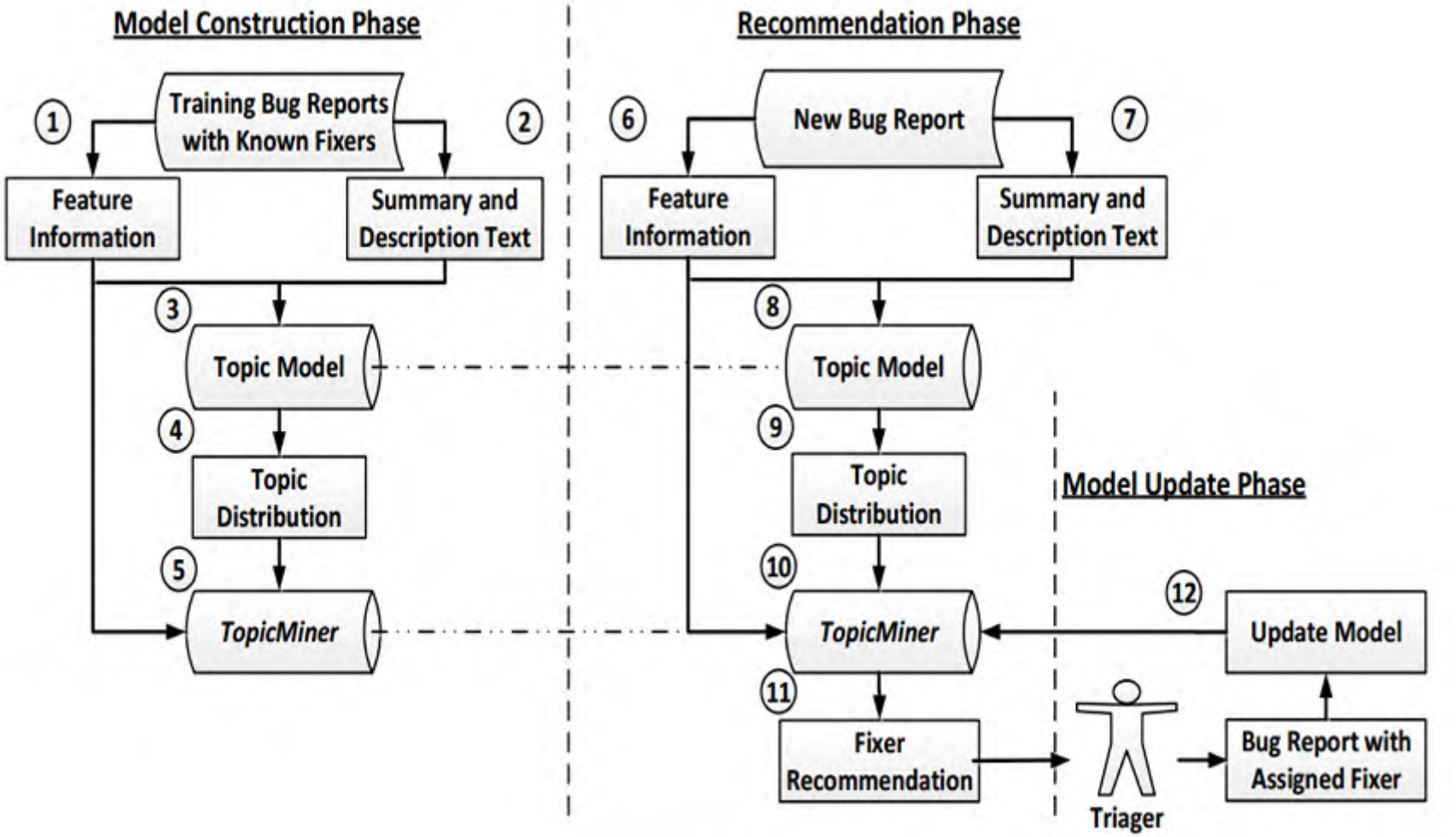
DevRec: A Composite Method



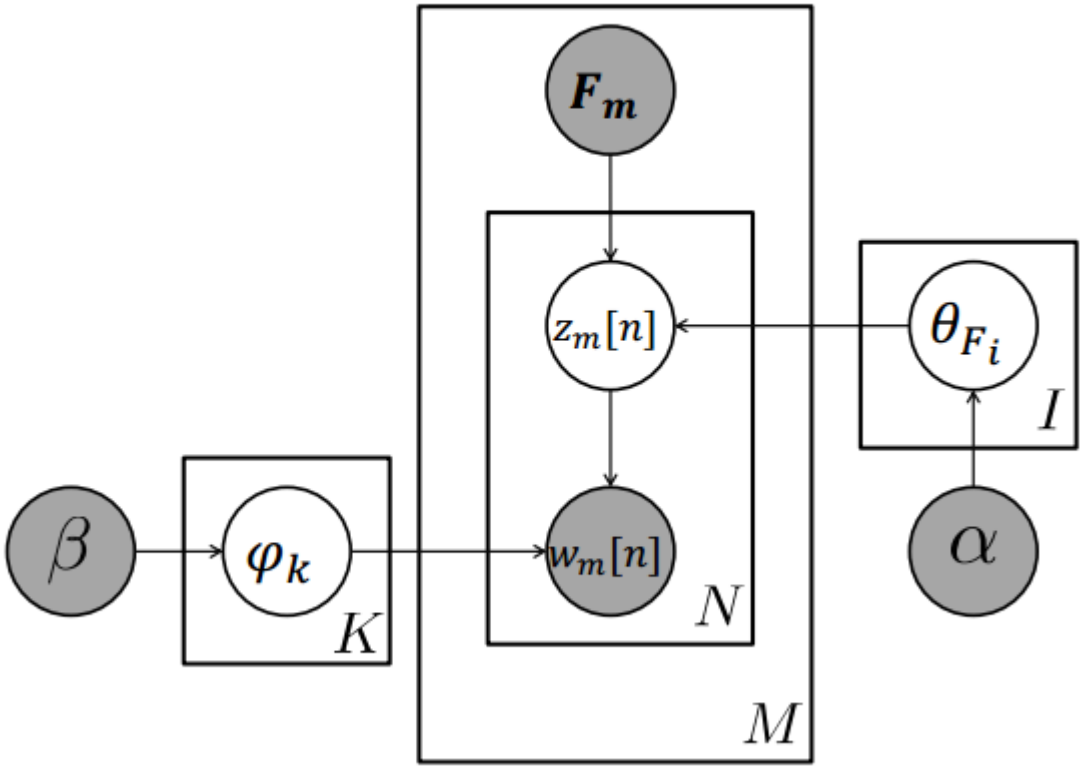
Improving Automated Bug Triaging with Specialized Topic Model

Accepted, TSE

Bug Triaging based on Topic Modeling



Multi-feature Topic Model



Task4: Predict Reassign/Reopen Bug Reports

Definition:

Predict whether a bug will be reassigned, or be reopen earliered.

Motivation:

- Reassign/Reopen open happens (around 20%).
- Takes time for human to detect.

Task4: Predict Reassign/Reopen Bug Reports

Specific:

- Explore good features
- Post-submitting features helps but usually late.

Solution:

- Select reasonable features+classifiers.[Lamkanfi et al.'13,Shihab'12,Xia'13]
- Features: Work Habit, Bug Fix, Bug report, People.

Measurement:

10-fold cross validation. Precision, recall, F-measure.

Automatic, High Accuracy Prediction of Reopened Bugs

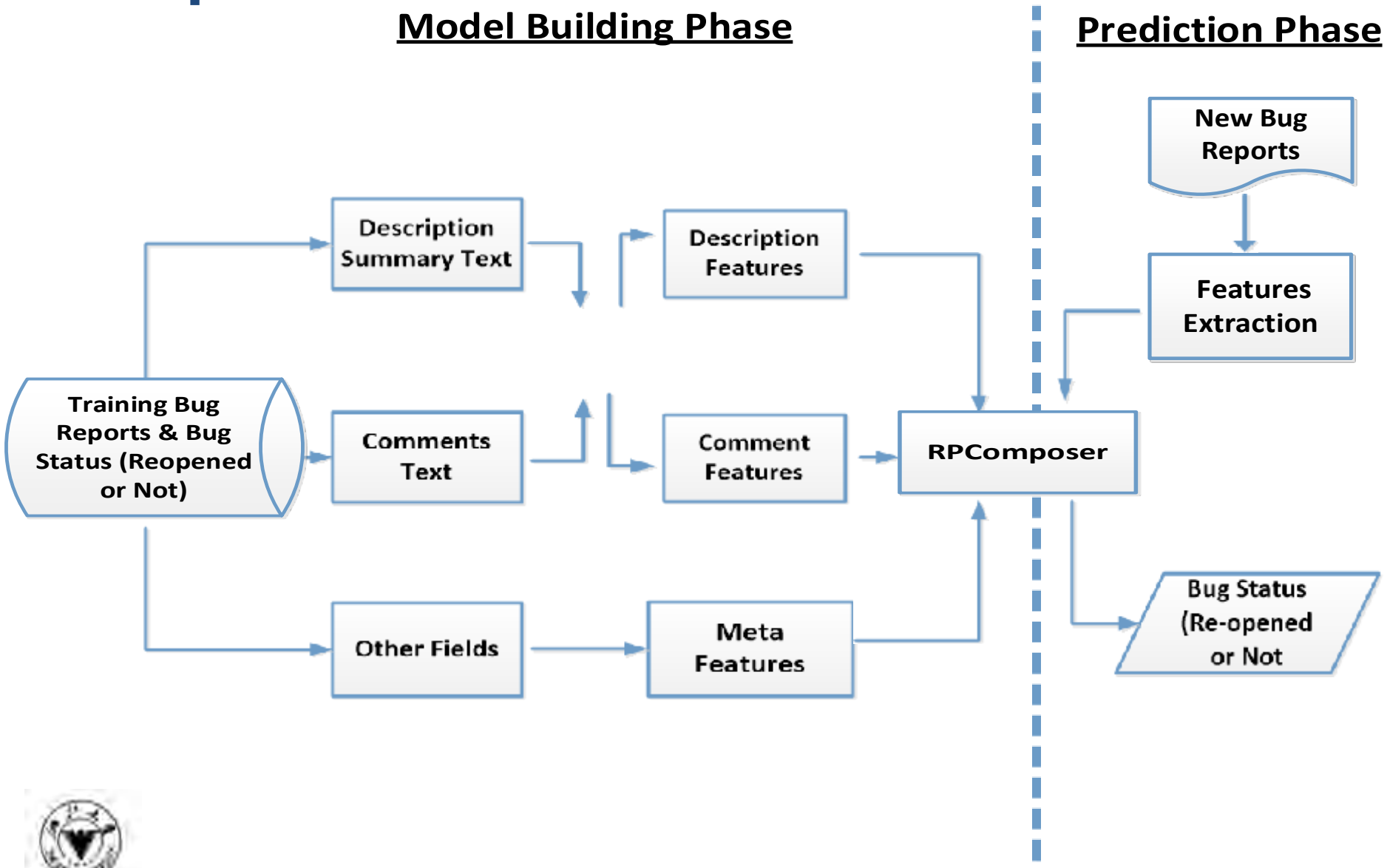
Accepted by Automated Software Engineering Journal

Some Difficulties

Only 16%, 6%, and 26 % of the bug reports in Eclipse, OpenOffice, and Apache HTTPD are reopened

Class Imbalance Phenomenon!

ReopenPredictor: Overall Framework



Automated Bug Report Field Reassignment and Refinement Prediction

Accepted by IEEE Trans. On Reliability, CSMR-WCRE 2014

General Root Cause

- New Bug Report Correction
  - When a bug report is submitted, some fields could be wrongly assigned.
- Progressing in the Process
  - During bug triaging, some bug fields get reassigned
  - Different from that of new bug report correction, the fields are not wrongly assigned
- Admin Batch Operations
  - Administrators also reassign some fields in the bug reports to better organize the project

Fields in A Bug Report

- Product
- Component
- Priority
- Severity
- Assignee
- Status (reopen or not)
- Platform
- Version
- .....

A Bug Report

Bug 227547 - Unit test org.netbeans.modules.cnd.classview.QuoteTestCase.testQuote failed

Status: RESOLVED FIXED
Product: cnd
Component: -- Other --
Version: 7.4
Hardware: All All

Issue Type: DEFECT
Priority: P2 (vote)
Target Milestone: 7.4
Assigned To: Alexander Simon

Reported: 2013-03-16 07:31 UTC by Alexander Simon
Modified: 2013-04-02 09:31 UTC (History)
CC List: 3 users (show)

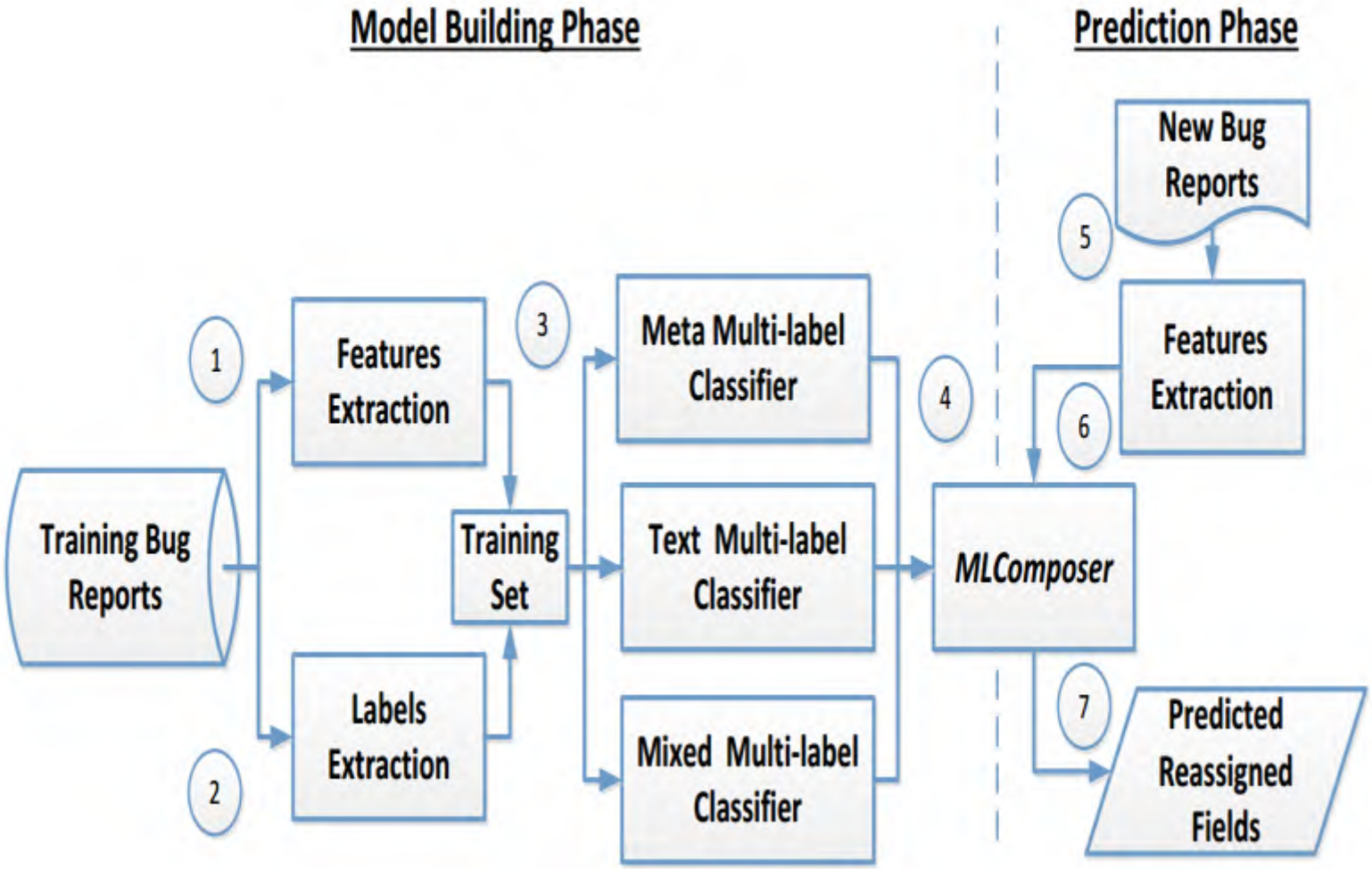
Alexander Simon 2013-03-16 07:31:46 UTC Description

```
Test filed since change set:
http://hg.netbeans.org/main-golden/rev/e7e9a8f3cea7
Commit for bug #227050
```

Fields Get Reassigned

vv159170	2013-03-18 09:17:10 UTC	Priority	P2	P1
vv159170	2013-03-18 09:19:38 UTC	Priority	P1	P2
jtulach	2013-03-18 13:57:23 UTC	Assignee	jtulach	vv159170
mmirilovic	2013-03-20 08:00:55 UTC	CC		mmirilovic
		Whiteboard	73patch-candidate	73patch2-candidate
vv159170	2013-03-20 14:23:03 UTC	Component	Options&Settings	Code
		Assignee	vv159170	issues
		Product	platform	cnd
		QA Contact	issues	issues
alexvsimon	2013-03-20 16:40:53 UTC	Component	Code	Options&Settings
		Assignee	issues	jtulach
		Product	cnd	platform
		QA Contact	issues	issues
jtulach	2013-03-21 08:31:59 UTC	CC		jtulach
		Component	Options&Settings	-- Other --
		Assignee	jtulach	alexvsimon
		Product	platform	cnd
		QA Contact	issues	issues

Overall Framework



Task5: Bug Reports Classification

Definition:

Identify the type of a bug report.

Key point:

- What is the type?
  - The component affected by the bug. (multiple components, more than 10)

Task5: Bug Reports Classification

Specific:

- Preprocess, no need for stop word removal.
- Imbalance, some small component might not have enough data for training.

Solutions:

- Select features+classifiers [Antoniol et al.'08]
- Distance based.

Measurement:

10-fold cross validation. Precision, recall, F-measure.

Predict Affected Component

- Somasundaram et al. Automatic Categorisation of Bug Reports Using Latent Dirichlet Allocation, ISEC’12.

- Used Information:
  - Terms in description.
- Approaches:
  - SVM(TF-IDF)
  - For bug reports in training dataset, learn topics using LDA
    - SVM with LDA output as features
    - LDA + Kullback Leibler divergence (Compute centroid of topics for each component. i.e., average document topic probability of all reports. Compute DKL(P||Q))

Task6: Bug-Fix Time Prediction

Definition:

Predict time needed to fix a bug report/bug reports. (New->Resolved)

Motivation:

Help project managers schedule bug fixing process, allocate resource, make better plan for release.

Task6: Bug-Fix Time Prediction

Specific:

Very hard if only use textual information. Need status, history.

Solution:

- k-Nearest Neighbour. [Wei et al'07]
- Discrete fix time to categories+select feature+classifier [Panjer'07,Marks'11]
- Location aspect, reporter aspect, and description aspect

Measurement:

- Average absolute residual.
- Accuracy.

Task 7: IR-based Bug Localization

- Definition:
  - Automatically Detect the Source Code Files Related to a Bug
- Motivation:
  - Not require program execution information.
  - Initial bug report as a query
  - Rank the source code files by their relevance to the query

Where Should the Bugs Be Fixed?

- Zhou et al. Where Should the Bugs Be Fixed? ICSE’12.

Bug ID: 80720
Summary: **Pinned console** does not remain on top
Description:
*Open two **console views**, ... **Pin** one **console.** Launch another program that produces output. Both **consoles display** the last launch. The **pinned console** should remain **pinned.***

Source code file: **ConsoleView.java**

```
public class ConsoleView extends PageBookView
    implements IConsoleView, IConsoleListener {...
      public void display(IConsole console) {
          if (fPinned && fActiveConsole != null) { return;}
      } ...
      public void pin(IConsole console) {
            if (console == null) {  setPinned(false);
            } else {
                if (isPinned()) { setPinned(false); }
               display(console);
               setPinned(true);
            }
      }
}
```

$$rVSMScore(q,d) = g(\#term)\times\cos(q,d)$$

$$= \frac{1}{1+e^{-N(\#terms)}} \times \frac{1}{\sqrt{\sum_{t\in q}((\log f_{tq}+1)\times\log(\frac{\#docs}{n_t}))^2}}$$

$$\times \frac{1}{\sqrt{\sum_{t\in d}((\log f_{td}+1)\times\log(\frac{\#docs}{n_t}))^2}} \qquad (7)$$

$$\times \sum_{t\in q\cap d}(\log f_{tq}+1)\times(\log f_{td}+1)\times\log\left(\frac{\#docs}{n_t}\right)^2$$

Conclusion

Conclusion

- Target Bug tracking System:

- Bugzilla, Jira, Industry

- Target Project:

- Eclipse, Mozilla, OpenOffice, JBoss, GNOME, Mylyn, Debian, Launchpad, Firefox, Jazz, Chrome, Blackberry.

- Used Information:

- Terms in title, description, comments. Other filed value like severity, priority, component, product, reporter, assignee.

Reference

1. Anvik et al. Reducing the effort of bug report triage: Recommenders for development-oriented decisions. ACM Transactions on Software Engineering and Methodology (TOSEM)
2. Tamrawi et al. "Fuzzy set and cache-based approach for bug triaging."Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering. ACM, 2011.
3. Sun et al. "Towards more accurate retrieval of duplicate bug reports." Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering. 2011.
4. Tian et al. "Improved duplicate bug report identification." Software Maintenance and Reengineering (CSMR), 2012 16th European Conference on. IEEE, 2012.
5. Liu et al. "Has this bug been reported?." Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering. ACM, 2012.
6. Amoui, Mehdi, et al. "Search-based duplicate defect detection: an industrial experience." Proceedings of the Tenth International Workshop on Mining Software Repositories. IEEE Press, 2013.
7. Lamkanfi et al. "Comparing mining algorithms for predicting the severity of a reported bug." Software Maintenance and Reengineering (CSMR), 2011 15th European Conference on. IEEE, 2011.
8. Tian et al. "DRONE: Predicting Priority of Reported Bugs by Multi-Factor Analysis." ICSM, 2013.
9. Somasundaram et al., ” Automatic categorization of bug reports using latent dirichlet allocation." Proceedings of the 5th India Software Engineering Conference. ACM, 2012.
10. Antoniol et al. "Is it a bug or an enhancement?: a text-based approach to classify change requests." Proceedings of the 2008 conference of the center for advanced studies on collaborative research: meeting of minds. ACM, 2008.
11. Rastkar et al., ” Summarizing software artifacts: a case study of bug reports." Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1. ACM, 2010.
12. Lotufo et al. ” Modelling the ‘hurried’ bug report reading process to summarize bug reports." Software Maintenance (ICSM), 2012 28th IEEE International Conference on. IEEE, 2012.
13. Mani et al. "Ausum: approach for unsupervised bug report summarization." Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering. ACM, 2012.
14. Weiss et al. "How long will it take to fix this bug?." Proceedings of the Fourth International Workshop on Mining Software Repositories. IEEE Computer Society, 2007
15. Giger et al. "Predicting the fix time of bugs." Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering. ACM, 2010.
16. Lamkanfi et al. ” Predicting Reassignments of Bug Reports-An Exploratory Investigation." Software Maintenance and Reengineering (CSMR), 2013 17th European Conference on. IEEE, 2013.
17. Xia et al., "A Comparative Study of Supervised Learning Algorithms for Re-opened Bug Prediction." Software Maintenance and Reengineering (CSMR), 2013 17th European Conference on. IEEE, 2013.
18. Bettenburg et al. "What makes a good bug report?." Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering. ACM, 2008.

谢　谢！