

Automatic Identification of Bug-Introducing Changes

Presented by Peter Rigby

MSR Course

October 12, 2006

Old Technique (SZZ)

- Look in logs and bugzilla to get bug commits that fixed bugs
- Diff fix with older versions -> hunk
- Uses the SCM annotate feature
- Problems
 - Annotation info insufficient
 - Not all modifications are fixes

Measure

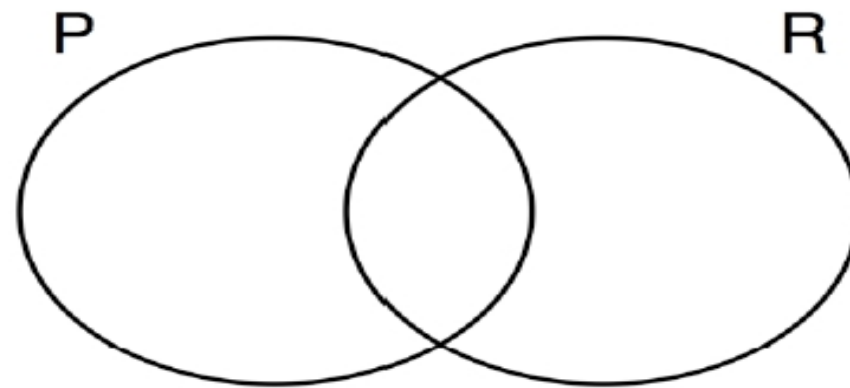


Figure 3. Bug-introducing change sets identified using SZ (P) and with the new algorithm (R)

Assuming R is the more accurate bug-introducing change set, we compute false positives and false negatives for the set P as follows:

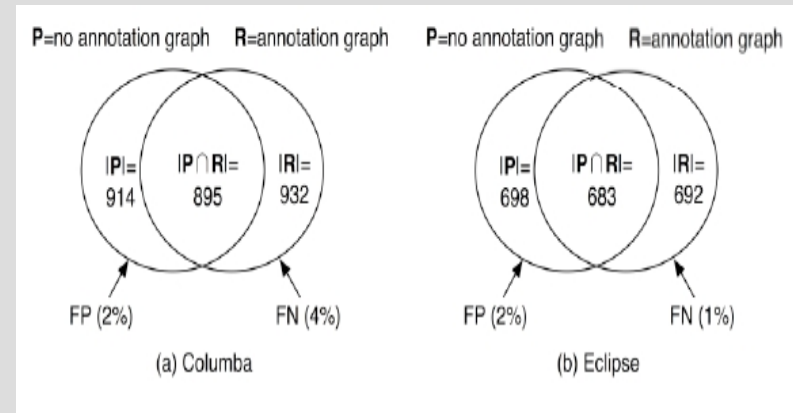
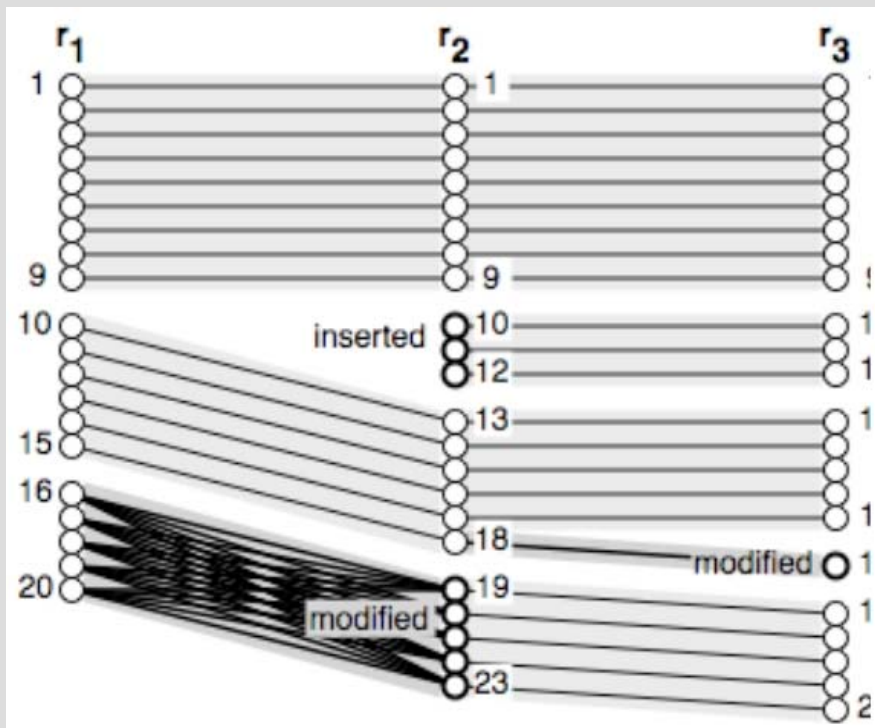
$$\text{False positive (FP)} = \frac{|P - R|}{|P|}$$

$$\text{False negative (FN)} = \frac{|R - P|}{|R|}$$

New approach

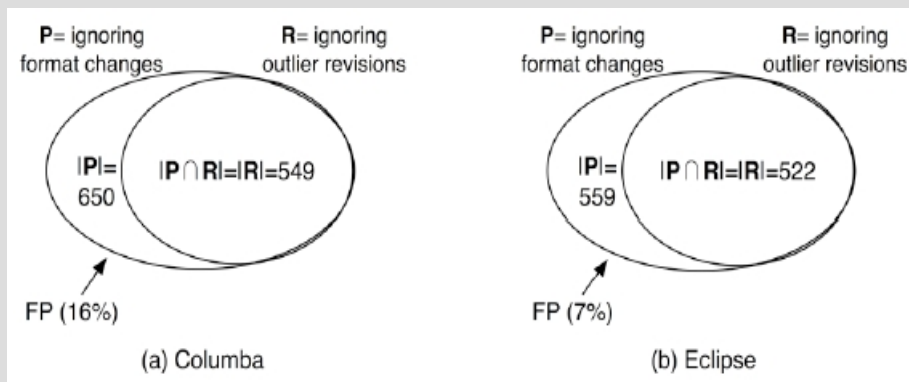
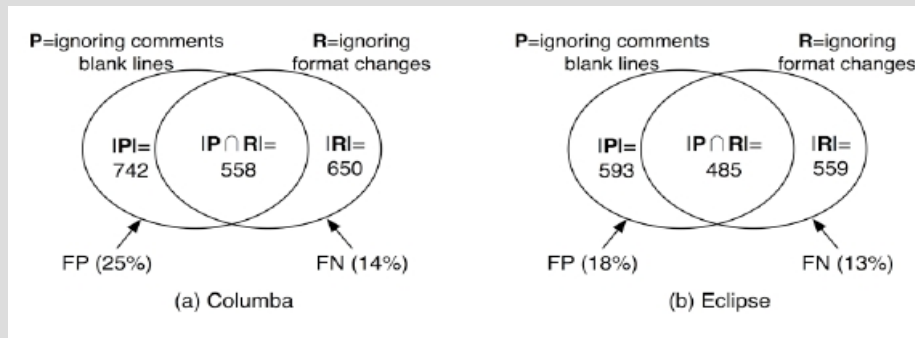
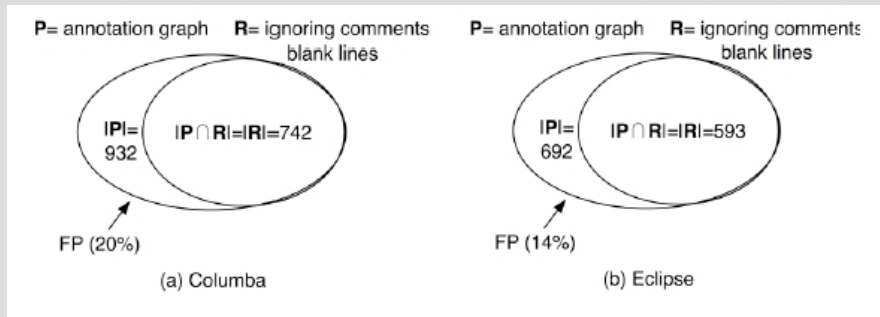
- Create annotation graph
- Ignore comments and blank line changes
- Ignore format changes
- Ignore changes involving many files
- Manually examine bug fix changes

- Create annotation graph



Other techniques

- Ignore non-behavior
- Ignore outliers



Final Result

- 38-51% false positives
 - No manual -> 36-48%
- 14-15% false negatives
- Show dangerous locations in code
- Full bug intro to fix times

Conclusion

- Strengths
 - Aren't hiding anything
 - Show incremental improvements
 - Apply to real problems
- Weaknesses
 - Many bugs not actually reported on bugzilla or recognized in commit log
 - Still not a perfect way to find origin.