An Approach to Detecting Duplicate Bug Reports using Natural Language and Execution Information

Xiaoyin Wang[1], Lu Zhang[1*], Tao Xie[2*], John Anvik[3] and Jiasu Sun[1]

[1]Key laboratory of High Confidence Software Technologies, Ministry of Education, Institute of Software, EECS, Peking University, Beijing, 100871, P. R. China,
{wangxy06, zhanglu, sjs}@sei.pku.edu.cn

[2]Department of Computer Science, North Carolina State University, Raleigh, NC 27695, USA
xie@csc.ncsu.edu

[3]Department of Computer Science, University of Victoria
PO Box 3055, STN CSC
Victoria, BC, Canada, V8W 3P6
janvik@cs.uvic.ca

ABSTRACT

Categories and Subject Descriptors

Software Engineering

General Terms

Keywords

1. INTRODUCTION

*

2.1 Browser-Closing Bug

Bug-260331 *Bug-239223*
Bug-260331
Bug-239223

Bug-260331: After closing Firefox, the process is still running. Cannot reopen Firefox after that, unless the previous process is killed manually

Bug-239223: (Ghostproc) – [Meta] firefox.exe doesn't always exit after closing all windows; session-specific data retained

-
-
-
-

2.2 Document-Contain-No-Data Bug

Bug-219232
Bug-244372

Bug-244372: "Document contains no data" message on continuation page of NY Times article

Bug-219232: random "The Document contains no data." Alerts

Bug-244372

Bug-219232
Bug-219232

2. MOTIVATING EXAMPLES

2.3 Motivation

3. BACKGROUND

n n

w_i $1 \le i \le n$

i w w w

$w_i = tf_i \times idf_i$

tf_i *term frequency* idf_i

inverse document frequency tf_i

i

idf_i

$idf_i = log\ (Dsum / Dw_i)$

$Dsum$ Dw_i

i

q_1 w w w q_2 w w w

q_1 q_2

$$Sim = \frac{\sum_{i=}^{n} w_i w_i}{\sqrt{\sum_{i=}^{n} w_i \times \sum_{i=}^{n} w_i}}$$

4. THE PROPOSED APPROACH

4.1 Calculating NL-S

4.2 Calculating E-S

■

■

■

4.3 Retrieving Potential Target Bug Reports

4.3.1 Basic Heuristic

$SIM_{combined} = f(SIM_{nlp}, SIM_{exe})$

SIM_{nlp} SIM_{exe}

$SIM_{combined}$ f

$$SIM_{combined} = \frac{SIM_{nlp} + SIM_{exe}}{}$$

4.3.2 Classification-Based Heuristic

4.3.3 Determining Credibility Thresholds

v

v

S D

S b

b

b D b

S v $dup(b, v)$

b v $fp(b, v)$

b

v v $E(v)$

$$E(v) = \sum_{b \in D} dup(b, v) - fp(b, v)$$

4.4 Presenting Potential Target Bug Reports

■

5. EXPERIMENT

$$recall\ rate = \frac{N_{recalled}}{N_{tatal}}$$

5.1 Experimental Setup

-
-
-

5.2 Calibration and Evaluation on Eclipse

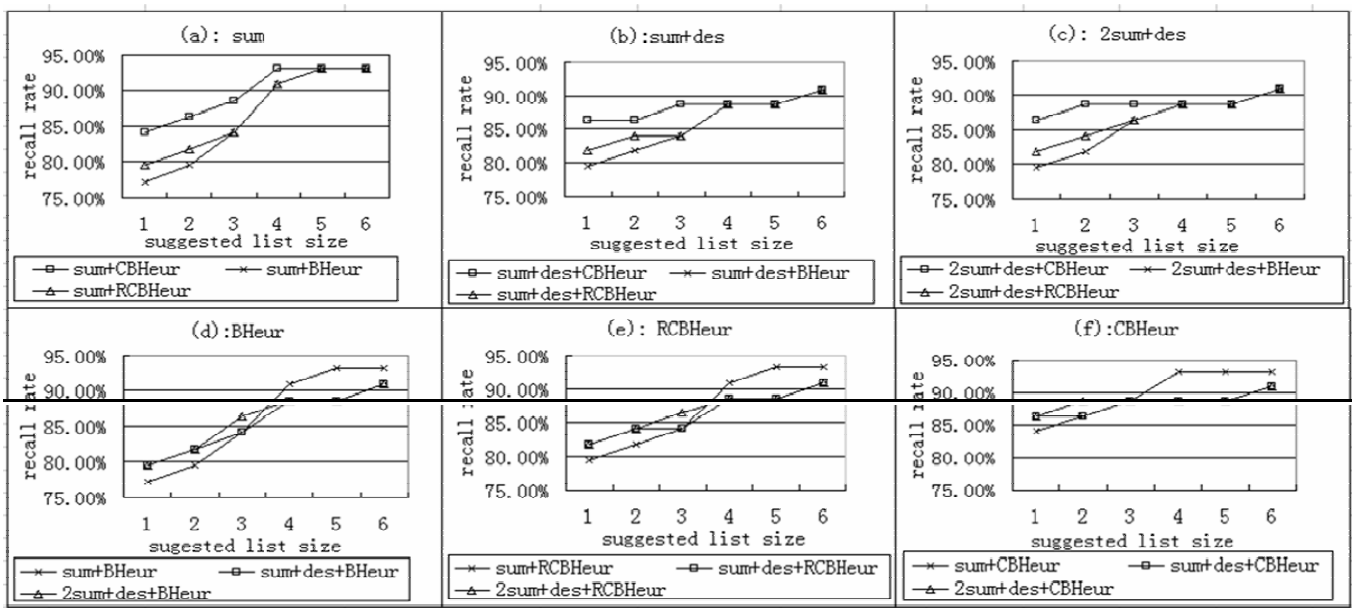

Figure 1: Recall rates using different parameters in Eclipse

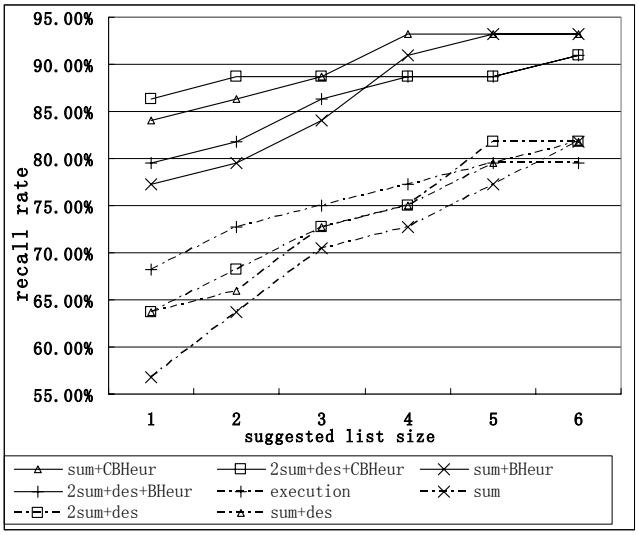

Figure 2: Recall rates using different similarities in Eclipse

5.3 Evaluation on Firefox

inverse duplicate problem

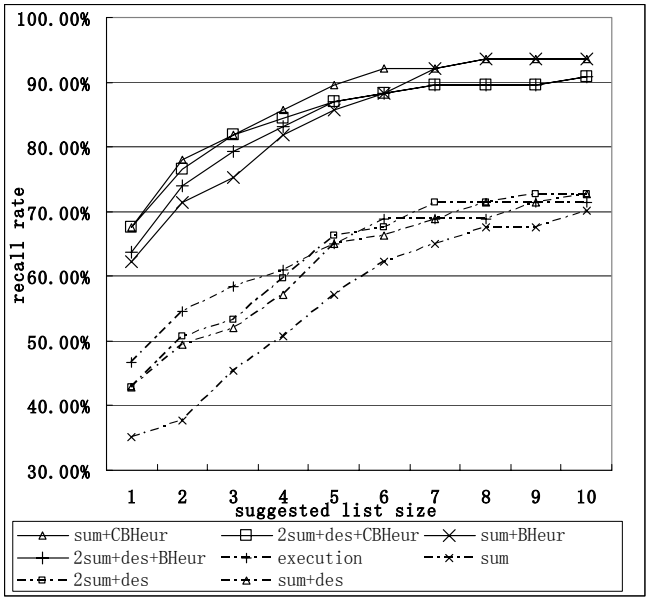

Figure 3: Recall rates using different similarities in Firefox

6. DISCUSSION

6.1 Costs of Using Execution Information

idf

idf

5.4 Threats to Validity

n

m

n m

6.2 Coping with Existing Bug Repositories

7. RELATED WORK

R

i

$$AvgSim_{exe}(i) = \frac{1}{n}\sum_{x \in R} Sim_{exe}(i, x)$$

n R

i

6.3 Evaluation Criteria for Duplicate-Bug-Report Detection

8. FUTURE WORK

9. CONCLUSION

Acknowledgments

10. REFERENCES

In Proc. ICSE.

In Proc. of OOPSLA Workshop on Eclipse Technology eXchange (ETX)

In Proc. ICSE

In Proc. SEKE

IEEE TSE

In Proc. ISSRE

Assisted Detection of Duplicate Bug Reports

In Proc. ISSTA

In Proc. of IEEE Conf. on Visual Language and Human-Centric Computing (VL/HCC)

In Proc. PLDI

In Proc. ICSM,

ACM TOSEM

In Proc. ICSE

Journal of the American Society for Information Science

In Proc. ICSE

In Proc. ICSE,

In Proc. MSR,