Building a probabilistic model of statement replacements based on an empirical study of real Java bug fixes

Mauricio Soto, Selva Samuel
School of Computer Science, Carnegie Mellon University, Pittsburgh, PA
mauriciosoto@cmu.edu, ssamuel@andrew.cmu.edu

ABSTRACT

Here goes the abstract

Keywords

Automatic error repair; Maintainability; Replacements

1. INTRODUCTION

2. DATASET AND CHARACTERISTICS

How we have mined Github, the regex, etc

2.1 Replacements and QACrashFix

3. BUILDING THE MODEL

4. EVALUATION

5. DISCUSSION AND FUTURE WORK

6. ACKNOWLEDGMENTS

7. REFERENCES

[1] M. Asaduzzaman, C. K. Roy, K. A. Schneider, and M. Di Penta. LHDiff: A Language-Independent Hybrid Approach for Tracking Source Code Lines. *ICSM*, pages 230–239, 2013.

[2] E. T. Barr, Y. Brun, P. Devanbu, M. Harman, and F. Sarro. The plastic surgery hypothesis. In *FSE*, pages 306–317, 2014.

[3] F. DeMarco, J. Xuan, D. Le Berre, and M. Monperrus. Automatic repair of buggy if conditions and missing preconditions with smt. In *CSTVA*, 2014.

[4] R. Dyer, H. A. Nguyen, H. Rajan, and T. N. Nguyen. Boa: A language and infrastructure for analyzing ultra-large-scale software repositories. In *ICSE*, pages 422–431, 2013.

[5] D. Kim, J. Nam, J. Song, and S. Kim. Automatic patch generation learned from human-written patches. In *ICSE*, pages 802–811, 2013.

[6] C. Le Goues, T. Nguyen, S. Forrest, and W. Weimer. Genprog: A generic method for automatic software repair. *IEEE Transactions on Software Engineering*, 38(1):54–72, 2012.

[7] F. Long and M. Rinard. Automatic patch generation by learning correct code. In *POPL*, 2016.

[8] M. Martinez and M. Monperrus. Mining software repair models for reasoning on the search space of automated program fixing. *Empirical Software Engineering*, 20(1):176–205, 2015.

[9] S. Mechtaev, J. Yi, and A. Roychoudhury. DirectFix: Looking for simple program repairs. In *ICSE*, 2015.

[10] Z. Qi, F. Long, S. Achour, and M. Rinard. An analysis of patch plausibility and correctness for generate-and-validate patch generation systems. In *ISSTA*, pages 24–36, 2015.

[11] H. Zhong and Z. Su. An empirical study on real bug fixes. In *ICSE*, pages 913–923, 2015.

	Assert	Break	Continue	Do	For	If	Label	Return	Case	Switch	Synch	Throw	Try	TypeDecl	While
Assert	-	7.48	3.76	0.53	8.30	23.05	0.31	20.04	4.90	4.62	1.30	13.50	7.23	0.03	4.95
Break	1.00	-	4.08	0.60	9.93	26.03	0.13	25.39	2.48	1.57	1.79	8.39	11.73	0.10	6.77
Continue	1.74	9.42	-	1.28	11.39	18.25	0.35	22.60	3.80	2.85	2.17	8.98	9.42	0.11	7.63
Do	0.81	5.26	6.60	-	9.44	14.21	0.18	15.86	3.73	1.67	1.97	5.88	6.39	0.03	27.98
For	0.86	6.28	3.19	0.79	-	22.89	0.09	21.08	5.01	3.34	1.87	10.01	10.71	0.08	13.79
If	1.64	8.43	2.87	0.60	13.49	-	0.24	26.46	7.45	4.80	2.85	9.89	15.11	0.08	6.11
Label	1.30	8.33	7.86	1.11	5.18	22.85	-	15.17	3.05	2.04	14.62	10.45	4.16	0.09	3.79
Return	1.13	9.41	3.11	0.49	13.33	27.24	0.24	-	5.59	3.65	2.55	14.91	12.61	0.12	5.61
Case	0.78	2.84	2.84	0.39	10.27	31.79	0.16	22.40	-	0.46	2.07	7.37	11.69	0.08	6.87
Switch	1.14	2.72	3.80	0.55	11.07	34.14	0.13	21.86	0.75	-	1.53	8.65	9.02	0.05	4.58
Synch	0.80	6.57	2.28	0.43	10.21	24.18	0.05	19.77	6.35	2.07	-	9.16	12.16	0.04	5.93
Throw	2.11	6.57	2.58	0.48	11.87	18.84	0.17	32.28	4.64	3.30	2.74	-	10.08	0.07	4.27
Try	0.71	7.41	3.02	0.66	11.73	27.75	0.11	23.24	5.63	2.65	2.58	8.99	-	0.09	5.42
TypeDecl	0.00	4.51	7.52	1.00	10.28	21.05	0.50	17.79	6.02	1.75	2.01	9.27	11.53	-	6.77
While	0.72	8.02	3.82	1.96	23.16	19.78	0.12	16.48	6.56	3.09	1.64	6.81	7.80	0.04	-

Table 1: Likelihood of replacing a statement type (row) by a statement of another type (column), for Java.