오픈소스 프로젝트의 토픽 모델링을 통한 잠재결함 분석 연구

이정빈*, 이택*, 인호*
*고려대학교 컴퓨터학과
e-mail:{jungbini, comtaek, hoh_in}@korea.ac.kr

Analysis of Potential Bugs using Topic Model of Open Source Project

Jung-Been Lee*, Taek Lee*, Hoh Peter In*
*Dept of Computer Science, Korea University

요 약

하나의 프로젝트에는 다양한 기능과 역할을 가진 소스코드가 존재한다. 그러나 기존 정적 분석 도구들은 이러한 특성을 고려하지 않고, 모든 소스코드에 동일한 탐색 정책과 우선순위를 적용하고 있다. 본 연구에서는 오픈소스 프로젝트로부터 수집한 소스코드들을 토픽모델링을 이용하여 특정 토픽으로 분류하고, 분류된 토픽에 해당되는 코드 안에서 높은 영향력을 갖는 잠재결함(Potential Bug)의 특징을 분석하였다. 이 결과를 바탕으로 개발자에게 개발 중인 소스코드의 특성에 따라 어떤 잠재결함에 더 우선순위를 두어야 하는지에 대한 지침을 제공할 수 있다.

1. 서론

정적 분석 도구는 개발 중인 소스코드 안에 잠재된 결함을 찾는 도구로써, 실제 버그는 아니지만 잠재적으로 실패(Failure)를 유발하거나, 유지보수 비용을 상승시킬 가능성 높은 결함들을 찾아낸다. 그러나 대다수의 정적분석 도구들은 35%~91%의 오탐(false positive)률을 가지고 있어, 그 결과에 대한 신뢰성이 높지 않은 실정이다[1,2]. 이와 같은 이유 중 하나는 다양한 특성을 가진 소스코드가 존재함에도 불구하고, 정적 분석 도구는 동일한 탐색 정책과 우선순위를 적용하고 있기 때문이다.

본 연구에서는 이러한 문제점을 해결하기 위한 기초 연구로써, 소스코드를 토픽별로 분류하여 해당 토픽에 속하는 소스코드들에서 발견되는 유의미한 잠재결함이 존재하는지에 대한 분석 연구를 수행하였다. 기존 연구[3]를 바탕으로 Sourceforge, Eclipse, Git 레파지토리에서 27개의 오픈소스 프로젝트 소스코드들을 수집하고, 토픽 모델링을 통해 특정 토픽을 분류하였다. 분류된 4개의 토픽(Collection, GUI, IO, Network)에 속한 소스코드에서 영향력이 높은 잠재결함들을 추출하고 분석한 결과, 토픽 별로 각기 다른 유의한 잠재결함들이 발견되었다. 이를 통해, 소스파일의 특성에 따라 다른 정책과 우선순위로 정적 분석을 수행해야 할 필요성을 확인하게 되었다.

2. 토픽 모델링을 이용한 잠재결함 추출

기존 연구[3]에서 제안한 토픽 모델링을 이용한 잠재결함 추출 기법에 대한 도식도는 그림1과 같다.

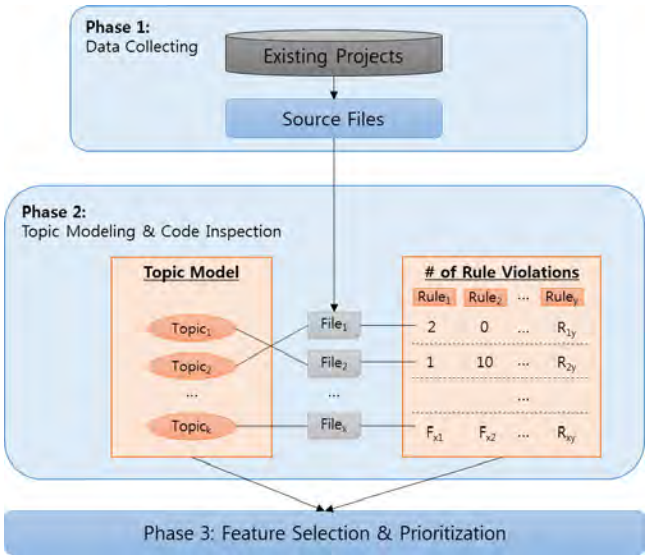

그림 1. 토픽 모델링을 통한 잠재결함 추출 기법[3]

Phase 1에서 분석을 위한 소스 코드를 수집한다. Phase 2에서 수집된 소스코드를 이용하여 토픽 모델링 및 정적분석을 수행한다. 본 논문에서는 토픽 모델링은 토픽 모델링 기법 중에 하나인 LDA[4]를 구현한 Mallet[1]을 이용하였으며, 정적분석은 오픈소스 정적분석 도구 중에 하나인 PMD[2]를 활용하였다. 기존 연구와는 달리 수식 1과 같이 특정 파일의 code change 정보를 Git 및 SVN을 통

1) http://mallet.cs.umass.edu/
2) https://pmd.github.io/

해 수집한 후 고쳐진 잠재 결함의 개수와 해당 파일의 Topic 분포를 곱하여 잠재결함이 어떤 토픽에서 큰 영향력이 있는지 분석하였다.

$$(\text{토픽 } t\text{의 분포}(\%) \times \text{수정된 잠재결함의 수}) \in File\ f$$

수식 1. 파일별 잠재 결함의 영향도

Phase3에서 모든 파일을 대상으로 구한 잠재결함의 영향력을 정규화하여 토픽별로 그 차이를 분석하였다.

3. 토픽 별 잠재결함 분석 결과

2장의 Phase 2에서 수행한 토픽 모델링을 통해 설문조사를 거쳐 사람이 구분 가능한 총 4가지(Collection, GUI, IO, Network) 토픽으로 추출하였으며, 각 토픽별로 영향력이 높은 잠재결함을 분석한 결과는 표1과 같다. 영향력이 높은 잠재결함이란, 해당 토픽에 속하는 파일에서 개발자들이 주로 많이 수정했던 잠재결함이라는 뜻이다.

토픽	영향력이 높은 잠재결함
Collection	UselessParentheses, UseVarargs, ArrayIsStoredDirectly
GUI	UnusedLocalVariable, EmptyIfStmt, EmptyStatementNotInLoop, ConfusingTernary, DefaultPackage, ModifiedCyclomaticComplexity, LongVariable, NPathComplexity
IO	UnnecessaryFullyQualifiedName, SimplifyBooleanReturns
Network	UnusedModifier

표 1. 토픽별 영향력이 높은 잠재결함

- **Collection 토픽**: Java collection에 객체를 추가하고 삭제하는 과정에 조건문을 많이 사용하기 때문에 다양한 비교 연산자들이 사용되는데, 이 과정에서 개발자들이 불필요한 괄호들을 많이 추가하게 되었고 이를 주로 수정하여서 'UselessParentheses'와 같은 잠재결함의 영향력이 높게 나타났다.
- **GUI 토픽**: 다른 토픽에 비해 GUI 관련 소스코드는 다른 코드들과 구조적인 차이가 크기 때문에 영향력 있는 잠재결함의 종류가 많았다. 또한, 인터페이스 및 이벤트 구현을 위한 코드들의 깊이(Depth) 관련 Metric이 컸기 때문에 '-Complexity' 관련 잠재결함들이 많이 수정되어 높은 영향력을 갖게 되었다.
- **IO 토픽**: IO 코드들을 테스트하기 위해서 삽입한 Assert와 같은 테스트 관련 코드들을 주로 사용한다. 이때, 클래스 명이 아닌 풀 패키지 명을 함께 사용하여 'UnnecessaryFullyQualifiedName'이라는 잠재결함 많이 발견되었고, 수정되어 영향력이 높게 나타났다.
- **Network 토픽**: 개발자들이 네트워크에서 사용되는 프로토콜 인터페이스나 헤더들을 선언하는 Interface 안에서 'public static final'와 같이 불필요한 modifier들을 선언되었고, 많이 수정되었기 때문에 높은 영향력을 가진 잠재결함으로 분석되었다.

이와 같이 토픽별로 다른 유의미한 잠재결함을 발견하였으며, 정적분석 시, 각기 다른 정책과 우선순위를 가지고 수행해야 할 필요성을 확인하였다.

4. 결론

기존의 정적 분석 도구들은 다양한 기능과 역할을 가지고 있는 소스코드들이 특성을 반영하지 않은 공통적인 정책과 우선순위를 바탕으로 정적 분석을 수행하였다. 본 연구에서는 이러한 문제에 착안하여 오픈소스 프로젝트로부터 소스코드를 수집하여 토픽 모델링을 수행하고, 특정 토픽에서 영향력이 높은 잠재결함들을 분석하였다. 분석 결과 4가지 토픽(Collection, GUI, IO, Network)에서 특정 잠재결함들이 영향력이 높게 나타났으며, 해당 잠재결함들이 높은 영향력을 갖는 이유에 대해서 분석하였다. 본 연구 결과를 이용하여 개발 중인 소스코드의 특성별로 개별적인 정적분석 정책과 우선순위를 적용해야 할 필요성을 확인할 수 있었다.

향후에는 더 다양한 토픽을 추출하고, 잠재결함 분석 알고리즘을 도구에 적용하여 실시간으로 개발 중인 코드에 대한 잠재결함의 발생 확률을 추출해주는 연구를 진행하고자 한다.

사사

이 논문은 2016년도 정부(미래창조과학부)와 정보통신기술진흥센터 재원으로 한국연구재단-차세대정보·컴퓨팅기술개발사업(2012M3C4A7033345) 및 대학 ICT연구센터 육성·지원사업(IITP-2016-R0992-16-1011)의 지원을 받아 수행된 연구임.

참고문헌

[1] C. Csallner, Y. Smaragdakis, T. Xie, "DSD-Crasher: A hybrid analysis tool for bug finding", ACM T Softw Eng Meth (TOSEM) 17.2, 2008.8

[2] S. Heckman, L. Williams, "On establishing a benchmark for evaluating static analysis alert prioritization and classification techniques", Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. ACM, 2008.

[3] 이정빈, 이택, 인호, "토픽모델링을 이용한 잠재결함 특징 추출 기법", 17회 한국 소프트웨어공학 학술대회 논문집 (KCSE 2015), 17권 1호, 2015.1.

[4] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." Journal of machine Learning research, p.993-1022, 2003.1.3