

# 버그 리포트 사이의 연관 관계를 이용한 PageRank 기반 버그 리포트 요약 기법

김범준\*, 강성원\*, 이선아\*\*

\*KAIST, \*\*경상대학교

e-mail : {bjk129, sungwon.kang}@kaist.ac.kr, saleese@gnu.ac.kr

## A PageRank-based Bug Report Summarization Technique using Bug Report Relationships

Beom-Jun Kim\*, Sung-Won Kang\*, Seonah Lee\*\*

\*KAIST, \*\*Gyeongsang National University

### 요 약

소프트웨어의 유지보수 단계에서 소프트웨어의 버그 리포트는 개발자에게 유용한 정보를 제공한다. 개발자들은 버그 수정이나 변경 내역 열람 등 다양한 작업을 위해 버그 리포트를 열람한다. 하지만, 대화 형식으로 작성되는 버그 리포트의 특징 때문에 버그 리포트는 종종 매우 길거나 장황하여 이를 읽고 이해하기 어려운 경우가 많다. 이러한 문제점을 해결하기 위한 방법으로 버그 리포트의 요약문을 자동으로 생성하는 기법을 제안하였고, 다양한 관련 연구가 진행되었다. 그러나, 기존에 제안된 버그 리포트 요약 기법들은 버그 리포트만의 고유한 특성을 활용하지 않는 경우가 많다. 본 연구에서는 버그 리포트들 사이의 중복(duplicates), 의존(depends-on), 역의존(blocks) 관계들을 이용한 PageRank 알고리즘 기반 버그 리포트 요약 기법을 제안한다. 실험 결과 제안 기법이 기존 버그 리포트 요약 기법보다 요약 품질과 적용 범위 측면에서 뛰어남을 확인하였다.

### 1. 서론

버그 리포트는 개발자에게 소프트웨어 프로젝트의 유지보수에 매우 중요한 정보를 제공한다. 버그 리포트는 소프트웨어 유지보수 작업의 약 40%를 차지하는 버그 수정 작업 시 개발자들의 정보 공유 및 논의의 용도로 사용된다. 또한, 버그 수정 작업이 완료된 이후에도 과거의 변경 이력을 확인하거나 연관된 버그 수정 작업 시 참고하는 등의 다양한 용도로 활용된다[1]. 일반적으로, 버그 리포트는 고유 번호, 제목 등의 정보와 함께 리포터가 직접 작성한 버그에 대한 상세한 설명이 포함되며, 재현 방법이나 해결 방안 등에 관한 개발자들의 논의가 코멘트의 형식으로 이루어진다. 이러한 구조 때문에 버그 리포트가 매우 길고 장황한 경우가 많다. 개발자가 버그 리포트에서 원하는 정보를 얻는데 어려움을 겪는 경우가 많다.

이러한 문제점을 해결하기 위해 버그 리포트의 요약문을 이용하는 방안이 등장하였고, 다양한 연구[2][6][7]가 진행되어 왔다. 하지만, 대부분의 경우 학습 방법을 개선하는데 초점을 맞췄고, 버그 리포트의 특성과 같은 특정 도메인 지식을 활용하는 방법에는 큰 관심을 두지 않았다. 본 연구에서는 버그 리포트의 다양한 메타정보들 중 버그 리포트들 사이의 연관 관계를 이용한 요약 기법을 제안하며, 이를 통해 개발자들에게 보다 높은 품질의 버그 리포트 요약문을 제공할 수 있다.

본 연구에서는 소프트웨어 업계에서 가장 많이 사용되는 3 가지 종류의 버그 리포트 사이의 연관 관계를 이용하였다. 첫 번째는 중복(duplicates) 관계로, 두 가지 버그 리포트가 서로 중복된 버그에 대한 내용을 담고 있음을 의미한다. 두 번째는 의존(depends-on) 관계로, 해당 버그를 해결하려면 연관된 버그 리포트가 먼저 해결되어야 한다는 선행관계를 나타낸다. 세 번째는 의존 관계와 역관계인 역의존(blocks)<sup>1</sup> 관계로, 해당 버그가 해결되어야 연관된 버그들이 해결될 수 있음을 의미한다.

본 논문의 구성은 다음과 같다. 2 장의 관련연구에서는 기존에 제안된 버그 리포트 요약 기법들을 알아보고 3 장에서는 본 논문의 제안 방법인 Relationship-based Bug Report Summarizer(RBRS)를 설명한다. 4 장에서는 실험을 통해 요약 품질을 향상시킬 수 있음을 보이며, 마지막으로 5 장에서는 결론을 맺으며 향후 연구 과제를 기술한다.

### 2. 관련연구

버그 리포트 자동 요약 기법은 Rastkar et al.[2]이 처음 제안하였다. 이들은 버그 리포트가 대화 형식의

<sup>1</sup> blocks 는 '막다'의 의미로, 해당 버그가 연관된 버그의 해결을 막고 있다는 의미를 나타내기 위해 사용되는 용어이다. 하지만, 본 연구에서는 blocks 연관 관계가 의존(depends-on) 관계와 역관계를 가지는 것에 초점을 두어 역의존 관계로 표현하였다.

문서로 취급될 수 있다는 점에서 착안하여 Murray et al.[3]의 회의록 및 이메일 쓰레드 요약 기법을 버그 리포트에 적용하여 지도 학습 기반 요약 기법을 제안하였다. Rastkar et al.은 36 개의 버그 리포트에 대해 직접 요약문을 작성하였고, 이를 학습 및 평가에 활용하였다. 이후 다양한 버그 리포트 요약 연구가 진행된 바 있으며, 지도 학습 기반 요약 기법의 학습 데이터 부족 문제를 해결하기 위해 비지도 학습을 활용한 연구들이 많이 진행되고 있다.

Jiang et al.[4]은 중복 버그 리포트들이 원본 버그 리포트와의 텍스트상의 유사성이 존재함을 이용하여 PageRank 알고리즘을 적용한 버그 리포트 요약 기법을 제안하였다. Jiang et al.[4]의 연구에서 사용된 중복 관계를 포함하여 버그 리포트들 사이에는 다양한 연관 관계가 존재한다. 이슈 추적 플랫폼 JIRA 는 11 종류의 버그 사이의 연관 관계를 정의한 바 있으며, 버그 리포트에 명시적인 링크의 형태로 표현된다. 이러한 연관 관계들은 버그 해결에 도움을 주기 위해서나, 해결 이후 다른 연관된 버그의 해결에 도움을 주기 위한 용도로 작성된다.

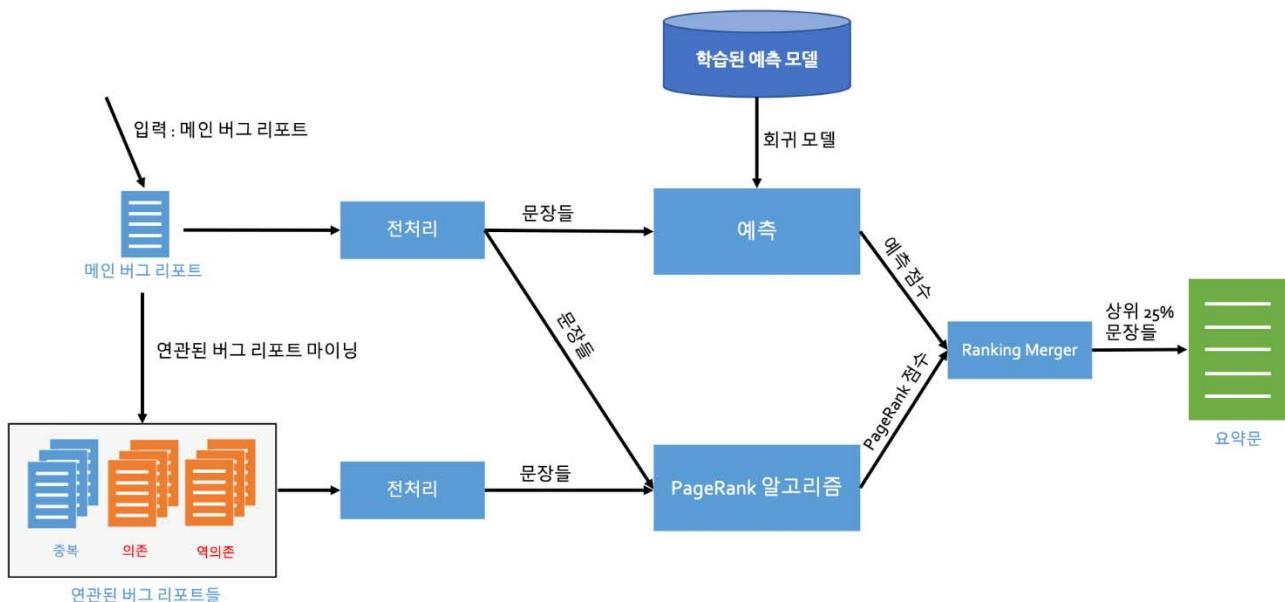
이들은 버그 리포트의 중복 관계들이 포함된 새로운 OSCAR 데이터셋을 만들었고, 이를 바탕으로 Rastkar et al.[1]의 지도 학습 기반 요약 기법과 PageRank 를 적용한 비지도 학습 기반 요약 기법을 혼합하여 버그 리포트 요약 품질을 향상시켰다. 하지만, 임의의 버그 리포트 50,000 개를 대상으로 한 사

례조사 결과, 중복 관계를 가지는 버그 리포트는 전체의 5.2%에 불과하였고, 이는 제안 기법의 적용 범위가 매우 좁다는 한계점을 시사한다.

Robert et al.[5]는 사례 조사를 통해 중복 관계를 포함한 3 종류의 연관 관계를 분류하였으며, 이는 버그 리포트 사이의 의존 관계와 역의존 관계를 포함한다. 실제로, Bugzilla 등의 이슈 추적 플랫폼에서는 해당 3 종류의 연관 관계들 외에는 거의 사용되지 않는 경우가 많다. 또한, 50,000 개의 버그 리포트를 대상으로 진행된 사례조사 결과, 전체의 5.2%만이 가지던 중복 관계와는 달리 위의 두 의존 관계는 약 전체의 약 35.1%만큼의 버그 리포트가 가짐을 확인하였다. 본 연구에서는 Jiang et al.의 연구를 확장하여 버그 리포트 사이의 3 종류의 연관 관계를 분석 및 활용하여 버그 리포트 요약문을 자동 생성하는 기법을 제안한다.

### 3. RBRS

본 연구의 제안 방법 RBRS 의 개요는 그림 1 과 같다. RBRS 는 크게 3 단계로 구성되며, 3.1 절에서는 버그 리포트의 전처리 과정을, 3.2 절에서는 버그 리포트 그래프를 이용한 PageRank 기법을, 3.3 절에서는 본 기법과 지도 학습 기반 Rastkar et al.의 요약 기법을 통합하는 과정을 설명한다.



(그림 1) RBRS 개요

### 3.1. 버그 리포트 전처리

버그 리포트 전처리 과정은 3 단계로 구분된다. 첫 번째는 버그 리포트 필터링으로, META 버그 리포트와 같이 실제 버그에 대한 설명이 담겨있지 않은 버그 리포트를 제외시킨다. 두 번째 단계는 버그 리포트의 코멘트 필터링이다. 버그 리포트가 일반적인 문서와 큰 차이를 보이는 점 중 하나는 버그 리포트에는 자연어뿐만 아니라 코드 조각, 스택 추적 등의 자연어가 아닌 문장들이 혼재하며, 또한 문장으로 완결되지 않는 키워드의 나열과 같은 자유로운 표현이 가능하다는 점이다. 이러한 정보를 노이즈로 간주하여 제거하고자 하는 연구들이 진행된 바 있으며[6][7], 본 연구에서도 해당 코멘트들을 제거하였다. 마지막 단계는 문장 분리 및 불용어 제거, 어간 추출 과정이다. 본 기법에서는 자연어 처리 패키지 NLTK 의 100 여개의 불용어 목록을 이용하였으며, 추가로 버그 리포트에서 흔히 사용되는 "bug", "build", "tag", "developer" 등의 불용어들을 추가로 정의하였다.

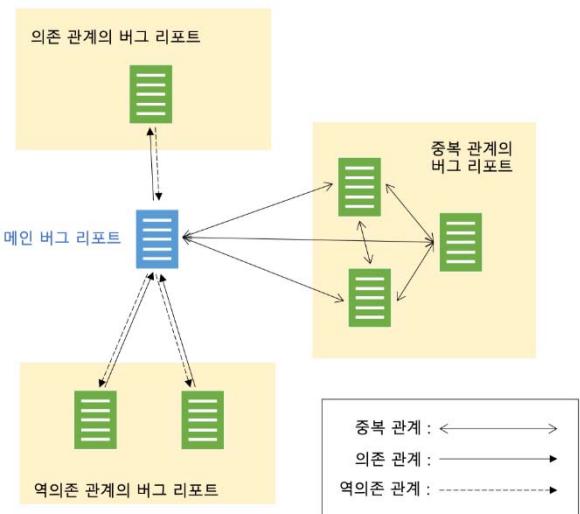
### 3.2. 버그 리포트 그래프 기반 PageRank 알고리즘

PageRank 알고리즘은 Sergey et al.[8]이 제안한 웹 문서에 상대적 중요도에 따라 가중치를 부여하는 방법이다. 이는 검색 엔진 연구를 위해 개발되었으며, 이를 바탕으로 구글 검색 엔진이 개발 및 서비스되었다. PageRank 알고리즘은 웹 페이지 사이의 연결 관계에 주목하여, 웹 페이지에 존재하는 링크 구조를 이용한다. 웹 페이지의 중요도를 계산하는 용도 이외에도 Rada et al.[9]은 PageRank 알고리즘을 바탕으로 문서 요약 및 키워드를 추출하는 알고리즘인 TextRank 를 개발하였다. 이는 ‘한 문서 내에서 다른 문장을 과 유사한 문장은 중요한 문장이다’라는 가정을 바탕으로 문장이나 단어의 중요도를 측정하고 우선순위화 한다.

버그 리포트의 문장들의 중요도를 측정하기 위해 PageRank 알고리즘을 적용하려면 먼저 연관 버그 리포트들을 수집해야한다. 본 연구에서는 요약의 대상이 되는 버그 리포트에 중복, 의존, 역의존 관계가 명시적으로 링크 되어있는 버그 리포트들을 수집하였다. 이후, 수집된 버그 리포트들을 이용하여 버그 리포트 그래프를 형성하는데, 그래프의 각 노드는 버그 리포트들의 문장을 나타내며, 에지는 해당 문장들이 속한 버그 리포트들 사이의 관계를 나타낸다. 관계에 따른 버그 리포트 그래프 형성 과정은 아래와 같으며, 그 한 예를 그림 2 가 보여준다. 그래프 형성이 완료되면 모든 노드에  $1/(노드의 개수)$  값만큼의 초기 PageRank 점수를 부여하고 PageRank 알고리즘을 시행한다.

두 문장이 속한 버그 리포트의 관계가

- **중복 관계일 경우:** 양방향 에지로 연결하고, 가중치는 두 문장의 자카드 유사도이다.
- **의존 관계일 경우:** 출발 노드로 향하는 단방향 에지로 연결하고, 역방향으로 역의존 관계 에지를 추가하며, 가중치는 두 문장의 자카드 유사도이다.



(그림 2) 연관 관계 기반의 버그 리포트 그래프

- **역의존 관계일 경우:** 출발 노드를 빠져나가는 단방향 에지로 연결하고, 역방향으로 의존 관계 에지를 추가하며, 가중치는 두 문장의 자카드 유사도이다.
- **동일한 버그 리포트일 경우:** 양방향 에지로 연결 하며 가중치는 1이다.
- **없는 경우:** 에지를 연결하지 않는다.

### 3.3. Ranking Merger

PageRank 알고리즘 시행 결과 요약 대상 버그 리포트의 모든 문장들에 PageRank 점수가 부여된다. 본 연구에서는 해당 점수를 그대로 이용하지 않고, Rastkar et al.의 지도 학습 기반 회귀 모델을 통해 얻은 문장들의 점수를 가중치를 두어 합산한 결과를 각 문장의 최종 점수로 계산하였다. 이는 연관 관계가 없는 경우에는 해당 방법을 적용하기 어렵고, 두 방법을 통합하여 더욱 높은 성능을 보일 수 있기 때문이다. 가중치는 실험을 통해 최적의 값을 찾는 방법을 사용하였으며 이는 버그 국지화(bug localization) 등의 다양한 연구 분야에서 널리 사용되는 방식이다. 최종 점수가 부여되면 이를 바탕으로 모든 문장들 중 점수가 높은 상위 25%만큼의 문장들을 추출하여 요약문을 구성하였으며, 이는 기존 버그 리포트 요약 연구에서 가장 적합한 길이로 간주된다.

## 4. 평가

기존 연구에서 버그 리포트 사이의 연관 관계들 중 중복 관계만이 버그 리포트 사이의 텍스트 유사성을 이용하여 버그 리포트 요약 품질을 향상시키는데 이용된 바 있다. 본 연구는 의존, 역의존 관계의 버그 리포트들 사이에도 텍스트 유사성이 존재하며 이를 요약 품질 향상에 이용할 수 있음을 보이고자 한다. 따라서, 본 기법과 중복 관계를 활용한 Jiang et al.의 기법을 요약의 정확도와 범위 측면에서 비교한다.

기존의 버그 리포트 요약 연구에서 일반적으로 사

용되는 실험 데이터는 Rastkar et al.이 제안한 BRC 데이터이다. 하지만, BRC 데이터의 36 개의 버그 리포트들 중 연관 관계를 가지는 버그 리포트의 수는 13 개에 불과하며, Jiang et al.은 실험을 위해 MBRC 와 OSCAR 데이터를 구축하여 실험을 진행한 바 있다. 본 연구에서는 명확한 비교를 위해 3 종류의 연관 관계를 모두 가지는 36 개의 버그 리포트를 임의로 선정하여 RBRS 데이터를 구축하여 실험하였다.

실험은 두 가지로 구성된다. 4.1 절에서는 중복 관계가 존재하며, 동시에 의존, 또는 역의존 관계가 존재하는 경우로, Jiang et al.의 제안 기법이 적용 가능한 버그 리포트들을 실험 대상으로 한다. 4.2 절에서는 중복 관계가 존재하지 않지만, 의존, 또는 역의존 관계가 존재하는 경우로, Jiang et al.의 제안 기법을 적용하기 힘든 버그 리포트들을 대상으로 한다.

#### 4.1. 중복 관계가 존재하는 경우

버그 리포트에 중복 관계가 존재하여 Jiang et al.의 제안 기법이 적용 가능한 경우의 실험 결과는 표 1과 같다.

<표 1> 중복 관계가 존재하는 경우 실험 결과

	정밀도	재현율	F1 점수
기존 연구	49.71%	49.77%	49.74%
제안 방법	51.69%	51.82%	51.76%

의존, 또는 역의존 관계를 추가적으로 이용함에 따라 정밀도와 재현율, F1 점수가 약 2%만큼의 상승을 보였다. 이는 오차 범위 내로, 실제로 유의미한 상승이라고 보기 어려우며, 의존 관계를 이용하는 것이 요약 품질에 큰 영향을 미치지 않는다고 분석된다.

#### 4.2. 중복 관계가 존재하지 않는 경우

버그 리포트에 중복 관계가 존재하지 않아 기존 연구가 적용되기 힘든 경우의 실험 결과는 표 2와 같다.

<표 2> 중복 관계가 존재하지 않는 경우 실험 결과

	정밀도	재현율	F1 점수
기존 연구	44.50%	44.79%	44.64%
제안 방법	50.91%	50.93%	50.92%

의존, 또는 역의존 관계를 이용할 때 약 6.4%의 품질 향상을 보였다. 통계적으로 유의미한 향상임을 보이기 위해 맨-휘트니 U 검정을 시행한 결과 p-value 값이 0.038로 0.05보다 작아 통계적으로 유의미하다고 분석된다. 따라서, 버그 리포트의 의존 관계는 중복 관계가 없는 경우 버그 리포트 요약 품질 향상에 도움을 줄 수 있으며, 이는 전체 버그 리포트의 약 32%에 해당한다.

### 5. 결론

본 연구는 버그 리포트의 3 가지 연관 관계(중복, 의존, 역의존)를 이용하는 PageRank 기반 버그 리포트

자동 요약 방법을 제안하였다. 그리고 제안 방법을 평가하기 위해 버그 리포트의 연관 관계 정보를 포함하는 36 개의 버그 리포트에 대한 코퍼스를 구축하였으며, 기존 연구인 PRST [4] 와 본 방법을 비교하였다. 실험 결과 본 연구에서 제안하는 방법이 요약문의 정밀도, 재현율, F1 점수 측면에서 기존연구보다 최대 6.4% 높아서 본 연구가 기존 연구보다 높은 품질의 버그 리포트 요약문을 생성함을 확인하였다.

본 연구의 공헌은 다음과 같다. 첫째, 우리는 버그 리포트의 3 가지 종류의 연관 관계 정보를 사용하는 PageRank 기반 버그 리포트 요약 방법을 제안하였다. 둘째, 버그 리포트의 연관 관계를 이용한 버그 리포트 요약 연구에 사용될 수 있는 코퍼스를 구축하였다. 마지막으로, 제안한 방법을 실제 도구로 구현하고 기존 연구와 비교하는 시뮬레이션을 수행하여 우리가 제안한 연관 관계를 이용하는 변경 추천 방법이 기존 방법보다 높은 품질의 요약문을 생성함을 보였다.

본 연구에서 제안한 연관 관계 기반 버그 리포트 요약 코퍼스는 크기가 작고 비전문가에 의해 작성되었다는 한계점을 가지고 있다. 이러한 데이터의 신뢰성 문제는 기존 버그 리포트 요약 관련 연구들에서도 발생하는 문제로, 향후 연구에서는 이러한 문제를 해결하기 위해 보다 큰 규모의 버그 리포트 코퍼스를 구축할 계획이다. 또한, 보다 높은 요약 품질을 위한 전처리 기법에 대한 연구를 수행할 계획이다.

### 참고문헌

- [1] Boehm, Barry, and Victor R. Basili. "Software defect reduction top 10 list." Foundations of empirical software engineering: the legacy of Victor R. Basili, 426.37 (2005): 426-431.
- [2] Rastkar, Sarah, Gail C. Murphy, and Gabriel Murray. "Summarizing software artifacts: a case study of bug reports." 2010 ACM/IEEE 32nd International Conference on Software Engineering. Vol. 1. IEEE, 2010.
- [3] Murray, Gabriel, and Giuseppe Carenini. "Summarizing spoken and written conversations." Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2008.
- [4] Jiang, He, et al. "PRST: a pagerank-based summarization technique for summarizing bug reports with duplicates." International Journal of Software Engineering and Knowledge Engineering, 27.06 (2017): 869-896.
- [5] Sandusky, Robert J., Les Gasser, and Gabriel Ripoche. "Bug Report Networks: Varieties, Strategies, and Impacts in a F/OSS Development Community." MSR. 2004.
- [6] Huai, Beibei, et al. "Mining Intentions to Improve Bug Report Summarization." SEKE. 2018.
- [7] Mani, Senthil, 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.
- [8] Page, Lawrence, et al. The PageRank citation ranking: Bringing order to the web. Stanford InfoLab, 1999.
- [9] Mihalcea, Rada, and Paul Tarau. "Textrank: Bringing order into text." Proceedings of the 2004 conference on empirical methods in natural language processing. 2004.