# 효율적인 S/W 유지관리를 위한 Git의 커밋메시지 복합 분류모델 제안

최지훈[O], 김재웅[**], 이윤열[*], 채의근[*], 김준용[***]
[O]공주대학교 컴퓨터공학과,
[*]공주대학교 컴퓨터공학과,
[**]공주대학교 소프트웨어학과,
[***]서울신학대학교 IT융합소프트웨어학과
e-mail: hunnx27@gmail.com[O], jwkim@kongju.ac.kr[**],
{alphaone, ygchae}@kongju.ac.kr[*], musimk@stu.ac.kr[***]

# Proposal of Git's Commit Message Complex Classification Model for Efficient S/W Maintenance

Ji-Hoon Choi[O], Jae-Woong Kim[**], Youn-Yeoul Lee[*], Yi-Geun Chae[*], Joon-Yong Kim[***]
[O]Dept. of Computer Engineering, Kongju National University,
[*]Dept. of Computer Engineering, Kongju National University,
[**]Dept. of Software, Kongju National University,
[***]Dept. of IT Convergence Software, Seoul Theological University

● 요 약 ●

Git의 커밋 메시지는 프로젝트가 진행되면서 발생하는 각종 이슈 및 코드의 변경이력을 저장하고 관리하고 있기 때문에 소프트웨어 유지관리와 프로젝트의 생명주기와 밀접한 연관성을 갖고 있다. 이러한 Git의 커밋 메시지에 대한 정확한 분석 결과는 소프트웨어 개발 및 유지관리 활동 시, 시간과 비용의 효율적인 관리에 많은 영향을 끼치고 있다. 이에 대한 기존 연구로 Git에서 발생하는 커밋 메시지를 소프트웨어 유지관리의 세 가지 형태로 분류하고 매핑하여 정확한 분석을 시도하려는 연구가 진행되었으나, 최대 87%의 정확도를 제시한 연구 결과가 있었다. 이러한 연구들은 정확도가 낮아 실제 프로젝트의 개발 및 유지관리에 적용하기에는 위험성과 어려움이 있는 현실이다. 본 논문에서는 커밋 메시지 분류에 대한 선행 연구 조사를 통해 각 연구들의 프로세스와 특징을 추출하였고, 이를 이용한 분류 정확도를 높일 수 있는 커밋 복합 분류 모델에 대해 제안한다.

**키워드:** 커밋 메시지(Commit Message), 다중분류(Multi-Label Classification), BERT(Bidirectional Encoder Representations from Transformers), 소스 변경(Source Change)

## I. Introduction

소프트웨어 개발과 유지관리 및 소스 버전 관리를 위한 SCM(Source Control Management)은 2000년대 초 리눅스 커널 관리를 위해 Linus Torvalds가 개발한 Git으로 대체되어 현재 주류를 이루고 있다.

이러한 Git은 여러명의 사용자들 사이에 발생하는 이슈 및 코드의 변경이력을 커밋 메시지라는 형태로 저장하고 관리한다.

프로젝트가 진행되면서 쌓이는 커밋 메시지를 정확하게 분석하면 프로젝트의 유지관리와 새로운 프로젝트 계획 시 이를 활용하여 사전에 리스크나 불확실성을 없애 비용과 시간에 대한 효율을 향상시킬 수 있다[1].

이러한 이유로 Git에서 발생하는 커밋 메시지를 분석하는 다양한 연구들이 진행되었고, 특히 커밋 메시지를 Corrective, Perfective 그리고 Adaptive의 3가지 형태로 분류하는 연구가 진행되었다[2].

관련 연구에는 분류의 정확도를 높이기 위해 커밋 메시지에서 단어의 빈도와 소스코드 변경 데이터를 이용해 76%의 정확도를 제시한 연구 결과가 있었다[3].

또한, NLP 트랜스포머 모델기반의 BERT를 이용해 87%의 정확도를 제시한 관련 논문이 발표되었다[4-5].

그러나, 기존연구에서 제시한 87%의 정확도로는 효율적인 개발 및 유지관리에 한계가 있는 것이 현실이다.

본 논문에서는 사전 연구된 두 모델의 장점인 소스변경과 BERT를 이용한 두 모델을 하이브리드 형태로 혼합하여 선행연구 모델보다 분류 정확도를 높일 수 있는 커밋 복합 분류모델에 대한 연구를 진행하였다.

## II. Preliminaries

### 1. Related works

#### 1.1 소프트웨어 유지관리 분류

다음 Table 1. 은 소프트웨어 유지관리를 세 가지 형태로 분류한 것이다[2].

Table 1. Software Maintenance Type

| Type | Description |
|---|---|
| Corrective | fixing bugs |
| Perfective | improving the system |
| Adaptive | introducing a new features into the system |

첫째, Corrective는 기능 및 비기능에 대해 문제가 되는 버그를 수정하거나 조치하는 것을 말한다.

둘째, Perfective는 시스템 및 디자인을 보완하기 위해 개선하는 범주이다.

셋째, Adaptive는 시스템에 새로운 기능을 도입하는 것을 말한다.

본 논문에서는 커밋 메시지를 위와 같은 세가지 유지관리 형태로 분류하여 커밋과 소프트웨어 유지 관리의 연관 관계를 형성해주는 모델에 대한 연구를 진행하였다.

#### 1.2 소스변경 데이터를 이용한 분류 모델

다음 Fig. 1. 은 소스변경 데이터를 이용한 모델의 프로세스로 76%의 정확도를 나타낸다[3].

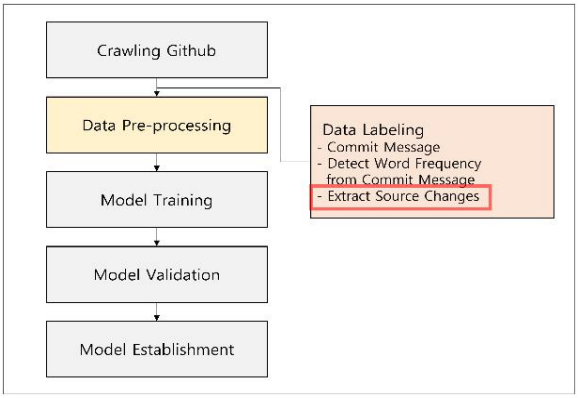


Fig. 1. Classification Process Using Source Change

관련 연구에서는 분류 모델에 데이터 레이블링 소스변경 내용을 추가하여 모델을 구성하였다. 커밋 분류 모델 생성 시 커밋 메시지와 커밋 메시지로부터 나오는 단어의 빈도수를 트레이닝 데이터로 사용하는 모델에 소스변경 데이터를 추가 적용하여 정확도를 높였다.

#### 1.3 BERT모델을 이용한 분류 모델

다음 Fig. 2. 는 BERT모델을 이용한 모델의 프로세스로 87%의 정확도를 나타낸다[5].

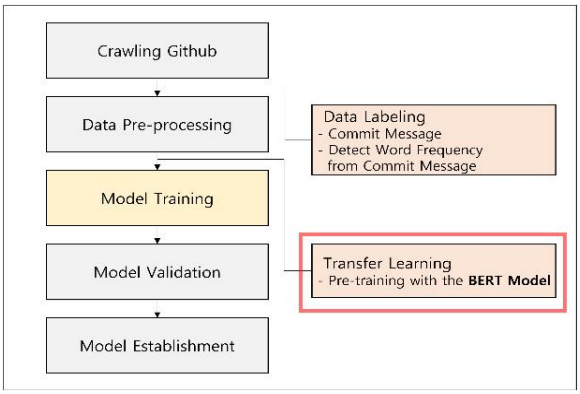


Fig. 2. Classification Process Using BERT Model

관련 연구에서는 분류 모델을 실제 트레이닝 과정에서 BERT 모델로 사전에 트레이닝시킨 후 전이학습으로 모델을 다시 트레이닝시킨다.

보통의 커밋 분류 모델처럼 트레이닝 시키는 데이터는 동일하지만 모델을 최종 트레이닝 전 BERT를 이용한 사전 트레이닝 후 전이학습을 시킴으로써 분류의 정확도를 높였다.

## III. The Proposed Scheme

### 1.1 Classification Process

다음 Fig. 3. 은 본 논문에서 제안하는 복합 커밋 분류 모델의 프로세스이다.

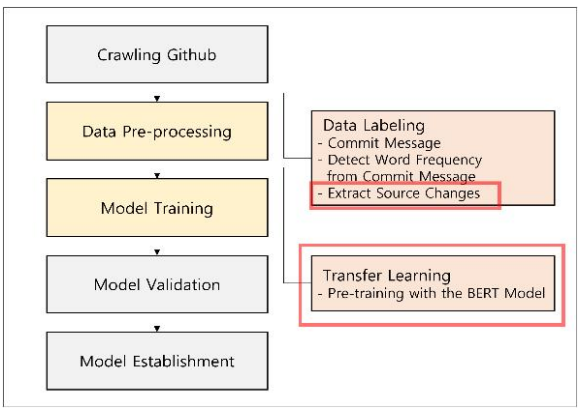


Fig. 3. Proposed Classification Process

본 논문에서는 관련 연구를 통해 제시한 두 가지 논문에서 정확도를 높였던 특징을 이용해서 복합 분류 모델을 설계하였다. 프로세스의 진행과정은 다음과 같다.

첫째, 데이터 수집을 위해 Github에서 별점이 높은 프로젝트의 Git 정보들을 크롤링한다.

둘째, 수집된 데이터의 커밋 메시지에서 단어 빈도와 소스변경 내용을 추출하고, 전체 데이터에 대해 레이블링하는 전처리 작업을 진행한다.

셋째, 레이블링 작업이 완료되면 홀드아웃 교차검증을 위해 8:2로 트레이닝 데이터 셋과 테스트 데이터 셋을 분류한다.

넷째, 트레이닝 데이터 셋은 BERT모델을 이용해 사전 트레이닝을 시키고, 생성된 모델을 전이 학습을 시켜 새로운 분류 알고리즘을 생성한다.

다섯째, 트레이닝하지 않은 20%의 테스트 데이터를 생성한 알고리즘으로 하이퍼 파라미터를 튜닝해 가면서 90% 이상의 정확도를 나타내는 모델을 생성한다.

## IV. Conclusions

본 논문은 기존연구에서 제시한 커밋 분류 모델의 정확도가 실제 프로젝트의 개발 및 유지관리에 적용하기에는 효율성에서 떨어지는 문제점으로 인해 실 프로젝트에 적용하기에는 한계가 있어 최소 90%이상의 정확도를 구현할 수 있는 모델에 대한 연구를 진행하였다.

정확도를 높이기 위해 관련연구들을 조사하고 이 중 소스변경 데이터를 이용한 분류 모델과 BERT를 이용한 분류 모델에서 정확도를 높인 특징만 추출하여 복합 분류 모델을 설계하였다.

향후 본 연구를 지속적으로 진행하여 제안모델을 구현함으로써 최소 90% 이상의 정확도를 확보하고, 이를 통해 프로젝트를 통합 관리할 수 있는 솔루션 설계 및 구현에까지 연구를 확장해 나갈 것이다.

## REFERENCES

[1] Mockus and Votta, "Identifying reasons for software changes using historic databases," Proceedings 2000 International Conference on Software Maintenance, pp. 120-130, 2000. doi: 10.1109/ICSM.2000.883028.

[2] S. Gharbi, M. W. Mkaouer, I. Jenhani, and M. B. Messaoud, "On the Classification of Software Change Messages Using Multi-Label Active Learning," in Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, 2019, pp. 1760–1767. 2019. doi: 10.1145/3297280.3297452

[3] S. Levin, and A. Yehudai, "Boosting Automatic Commit Classification Into Maintenance Activities By Utilizing Source Code Changes," PROMISE: Proceedings of the 13th International Conference on Predictive Models and Data Analytics in Software Engineering, pp. 97–106, November. 2017. doi: 10.1145/3127005.3127016

[4] A. Adhikari, A. Ram, R. Tang, and J. Lin, "DocBERT: BERT for Document Classification," arXiv, 2019. doi: 10.48550/ARXIV.1904.08398

[5] M. U. Sarwar, S. Zafar, M. W. Mkaouer, G. S. Walia and M. Z. Malik, "Multi-label Classification of Commit Messages using Transfer Learning," 2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), pp. 37-42, 2020, doi: 10.1109/ISSREW51248.2020.00034.