이슈 리포트 자동 레이블링 성능 개선을 위한 데이터 증강 기법의 실증적 연구

김정우[1], 김미수[2]
[1]전남대학교 에너지자원공학과 학부생
[2]전남대학교 인공지능융합학과 교수
jeongwu510@.naver.com, misoo.kim@jnu.ac.kr

An Empirical Study on Data Augmentation Techniques for Improving the Performance of Automatic Issue Report Labeling

Jeong-Wu Kim[1], Misoo-Soo Kim[2]
[1]Dept. of Energy and Resources Engineering, Chonnam University
[2]Dept. of Artificial Intelligence Convergence, Chonnam University

요 약

이슈 리포트 자동 레이블링은 효율적인 소프트웨어 유지보수를 위해 필수적인 작업이다. 그러나 이슈 리포트 데이터 셋은 롱테일 레이블 분포를 가지고 있어, 성능 저하를 초래할 수 있는 불균형 문제가 존재한다. 본 연구에서는 Easy Data Augmentation을 적용하여 롱테일 레이블의 성능을 개선할 수 있는지 실증적으로 검토하였다. 그 결과, 가장 희소한 “good-first-issue” 레이블에서 성능이 크게 개선된 것을 확인하였다.

1. 서론

소프트웨어 이슈 리포트를 자동으로 레이블링하는 작업은 소프트웨어 유지보수와 프로젝트 관리에 매우 중요하다[1]. 그러나 이슈 리포트 데이터 셋은 상위 레이블에 비해 대다수 레이블이 적은 "롱테일" 분포를 보인다. 예를 들어, bug와 wontfix가 전체 샘플의 66%를 차지하며, 나머지 레이블은 학습 데이터가 매우 적어 성능 저하의 원인이 된다.

이를 해결하기 위한 방법으로 데이터 증강 기법 중 EDA(Easy Data Augmentation)를 검토한다[2]. EDA는 텍스트 분류 성능을 향상시키는 간단하고 효과적인 기법으로, 본 연구에서는 이슈 리포트의 롱테일 레이블 성능 개선에 효과적인지 실증적으로 분석한다.

2. 실험 데이터셋

본 연구에서는 Heo et al. (2023)이 구축한 데이터 셋을 사용하였다[3]. 이 데이터 셋은 GitHub 내의 다양한 소프트웨어 프로젝트에서 발생한 이슈리포트를 포함한다. 우리는 몇 가지 절차를 통해 최종 실험 데이터를 구성하였다.

GitHub 프로젝트의 레이블은 크게 프로젝트 관리자가 설정한 커스텀 레이블과 GitHub에서 기본적으로 제공하는 기본 레이블로 나뉜다. 커스텀 레이블은 변동이 크기 때문에, 본 연구에서는 기본 레이블을 가진 180,339개의 샘플을 1차로 선별하였다. 다중 레이블 샘플의 경우, 데이터 증강 시 레이블 간 일관성 문제가 발생할 수 있다. 이를 방지하기 위해 5,503개의 다중 레이블 샘플을 제외하고, 최종적으로 174,836개의 이슈 리포트를 실험에 사용하였다. 표 1은 최종 데이터 셋을 요약한 것이다.

<표 1> 최종 데이터 셋의 분포

	Label	Count	Percentage
1	bug	66,199	37.9
2	wontfix	50,647	29.0
3	invalid	23,367	13.4
4	enhancement	14,125	8.1
5	duplicate	13,936	8.0
6	question	3,168	1.8
7	documentation	2,105	1.2
8	help-wanted	1,029	0.6
9	good-first-issue	260	0.1
	Total	174,836	100

3. 연구 방법

3.1. 증강 기법 적용

우수한 분류 성능을 유지하면서 성능이 저조한 롱테일 레이블의 성능을 개선할 수 있는지 검증하기

위하여 f1-score가 0.5 이하이면서 샘플 수가 10,000개 이하인 샘플 (question, help-wanted, good-first-issue)에 증강 기법을 적용하였다.

EDA 기법은 네 가지 주요 방법을 통해 데이터를 증강한다. 이에는 동의어 치환 (synonym replacement, sr), 랜덤 삽입 (random insertion, ri), 랜덤 교환 (random swap, rs), 그리고 랜덤 삭제 (random deletion, rd)가 포함된다. 증강을 위해서는 문장의 변형 빈도를 설정하기 위한 알파 값 (α)과, 원본 문장 당 몇 개의 샘플을 증강할지 설정해야 한다. 본 연구에서는 Wei et al. (2019)가 제안한 EDA 기법의 권장 설정을 따라 α를 0.1로 설정하고, 각 원본 문장당 4개의 증강 문장을 생성하였다[2].

3.2. 레이블링 모델 학습

모델 학습을 위해 데이터 셋을 80:10:10의 비율로 학습, 검증, 테스트 셋으로 분할하였으며, 데이터 불균형을 고려하여 stratify 기법을 적용하였다.

레이블링 모델은 Fang et al. (2023)이 제안한 RTA 모델을 기반으로 한다[4]. RTA는 GitHub, Jira, Bugzilla 등 다양한 버그 추적 시스템으로부터 수집된 약 250,000개의 버그리포트를 바탕으로 사전 학습된 모델로, 여러 소프트웨어 관련 다운스트림 작업에서 도메인 특화된 성능을 발휘할 수 있다.

4. 실험 결과

표 2는 데이터 증강 기법이 각 레이블에 미친 영향을 요약한다.

<표 2> 데이터 증강에 따른 성능 변화
(bold: 각 레이블 별 가장 좋은 성능)

	base	sr	ri	rs	rd
bug	**0.92**	**0.92**	**0.92**	**0.92**	**0.92**
wontfix	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
invalid	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**
enhancement	**0.92**	0.87	0.90	0.88	0.86
duplicate	**0.70**	**0.70**	0.69	**0.70**	**0.70**
question	**0.47**	0.27	0.37	0.35	0.31
documentation	**0.87**	0.85	0.84	0.83	0.85
help-wanted	**0.56**	0.47	0.46	0.46	0.50
good-first-issue	0.42	0.59	**0.67**	0.64	0.49

good-first-issue(GFI) 레이블은 기본 성능이 0.42에서 sr(0.59), ri(0.67), rs(0.64), rd(0.49)로 크게 향상되었으며, 특히 ri는 약 58%의 성능 개선을 보여 가장 우수한 결과를 나타냈다. 반면, question과 help-wanted는 데이터 증강 후 오히려 성능이 하락하였다. 상위 성능을 보이는 레이블들(예: bug, wontfix, invalid)도 증강 전후 성능 변화가 거의 없거나 약간의 하락이 있었다.

5. 결론

데이터 증강 기법은 전체 데이터 중 0.1%에 불과한 GFI와 같은 특정 레이블에서 성능을 크게 향상시켰다. GFI는 신규 기여자가 쉽게 이해하고 해결할 수 있도록 복잡한 내용이 배제되고 명확한 지침만 포함하는 경우가 많아, 증강 후에도 성능이 안정적으로 유지될 가능성이 높다. 반면, question과 help-wanted 레이블에서는 성능 저하가 관찰되었다. 이는 모든 레이블에서 증강이 일관되게 성능을 향상시키지 않음을 시사하며, 각 레이블의 특성에 맞는 증강 기법의 선택이 필요함을 강조한다.

사사

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 소프트웨어중심대학사업(2021-0-01409)과 과학기술정보통신부 및 정보통신기획평가원의 인공지능융합혁신인재양성사업(IITP-2023-RS-2023-00256629), 대학ICT연구센터사업(IITP-2024-RS-2024-00437718)의 연구 결과로 수행되었음

참고문헌

[1] Wang, Jun, et al. "Personalizing label prediction for GitHub issues." Information and Software Technology, vol. 145, 2022, p. 106845.

[2] Wei, Jason, and Kai Zou. "Eda: Easy data augmentation techniques for boosting performance on text classification tasks." Proceedings of the EMNLP-IJCNLP 2019, Hong Kong, 2019, pp. 6382 - 6388.

[3] Heo, Jueun, and Seonah Lee. "An empirical study on the performance of individual issue label prediction." Proceedings of the 2023 IEEE/ACM 20th International Conference on Mining Software Repositories (MSR), Melbourne, Australia, 2023, pp. 228-233.

[4] Fang, Sen, et al. "RepresentThemAll: A universal learning representation of bug reports." Proceedings of the 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), Melbourne, Australia, 2023.