논문 2018-2-1

텍스트 마이닝을 통한 키워드 추출과 머신러닝 기반의 오픈소스 소프트웨어 주제 분류

이예슬*, 백승찬*, 조용준*, 신동명*†

Keyword Extraction through Text Mining and Open Source Software Category Classification based on Machine Learning Algorithms

Ye-Seul Lee*, Seung-Chan Back*, Yong-Joon Joe*, Dong-Myung Shin*†

요 약

오픈소스를 사용하는 사용자 및 기업의 비중이 지속적으로 증가하고 있다. 국외뿐만 아니라 국내에서의 오픈소스 소프트웨어 시장 규모가 급격하게 성장하고 있다. 하지만 오픈소스 소프트웨어의 지속적인 발전에 비해서, 오픈소스 소프트웨어 주제 분류에 대한 연구 거의 이루어지지 않고 있으며 소프트웨어의 분류 체계 또한 구체화되어 있지 않다. 현재는 사용자가 주제를 직접 입력하거나 태깅하는 방식을 사용하고 있으며 이에 따른 오 분류 및 번거로움이 존재한다. 또한 오픈소스 소프트웨어 분류에 대한 연구는 오픈소스 소프트웨어 평가, 추천, 필터링등의 기반 연구로 이용될 수 있다. 따라서 본 연구에서는 머신러닝 모델을 사용하여 오픈소스 소프트웨어를 분류하는 기법에 대하여 제안하고, 머신러닝 모델 별 성능 비교를 제안한다.

Abstract

The proportion of users and companies using open source continues to grow. The size of open source software market is growing rapidly not only in foreign countries but also in Korea. However, compared to the continuous development of open source software, there is little research on open source software subject classification, and the classification system of software is not specified either. At present, the user uses a method of directly inputting or tagging the subject, and there is a misclassification and hassle as a result. Research on open source software classification can also be used as a basis for open source software evaluation, recommendation, and filtering. Therefore, in this study, we propose a method to classify open source software by using machine learning model and propose performance comparison by machine learning model.

한글키워드 : 오픈소스, 주제 분류, 머신러닝, 성능 비교

keywords : open source, category classification, machine learning, performance comparison

* 엘에스웨어(주)
† 교신저자: 신동명(roland@lsware.com)
접수일자: 2018.11.23. 심사완료: 2018.12.07.
게재확정: 2018.12.21.

1. 서 론

오픈소스 소프트웨어는 저작권이 존재하지만, 소스코드가 공개된 프로그램으로 누구나 자유롭게 사용하며 수정하고 재배포할 수 있는 소프트웨어이다. 정보통신산업진흥원의 2018 공개 SW 기업 편람에 따르면 오픈소스 SW 시장 규모는 1,890억 규모로 추정되며 이는 1,602억 규모였던 2016년 대비 17.9% 증가 성장한 규모이고 2021년에는 시장규모가 더 성장하여 3,430억 원에 도달할 것으로 전망한다[1]. 오픈소스 소프트웨어의 대표적인 예로는 리눅스, 안드로이드 등이 있다. 시장규모가 커짐에 따라 오픈소스 SW를 사용하고 개발하는 기업이 증가하는 추세이다. 대표적으로 네이버 랩스, 우아한 형제들, 삼성전자 등도 오픈소스 개발에 참여하고 있으며 그 외에도 최근 많이 사용되는 데브옵스(Devops), 퍼블릭 클라우드, 마이크로서비스(Microservice), 딥 러닝 대표 라이브러리인 구글의 텐서플로우 등이 모두 오픈소스 기반이다. 또한, 개방성 정책과는 상당히 먼 기업인 애플마저 2015년 12월 자사가 사용하는 프로그래밍 언어 스위프트(Swift)를 오픈소스로 공개할 것이라고 발표했다[2]. 이처럼 다수 기업에서 오픈소스를 사용하고 개발하며 이를 기본 사양으로 여기고 있다. 블랙덕소프트웨어에 의한 결과에 따르면 응답자의 78%가 오픈소스 소프트웨어를 기반으로 사업을 운영하고 있다고 답했다[3]. 오픈소스 소프트웨어에 참여하는 기업들이 증가하는 추세이며, 개발자 또는 일반 사용자들도 오픈소스를 사용하는 비중이 늘어나고 있다. 하지만 오픈소스 소프트웨어 주제에 대한 체계적인 분류 체계가 존재하지 않는다. 정보통신산업진흥원은 “공개SW 프로파일을 통해 공개 SW 통계를 수집하고 분류하려고 하지만 2016년 통계 이후 이루어지지 않고 있다” 말했다. 현재는 오픈소스 소프트웨어의 분류를 사용자가 직접 입력을 하거나 태깅하는 방식을 사용하고 있다. 이는 사용자가 주제를 직접 입력함으로써 번거로움을 초래하며 주제가 오 분류 될 수 있고 통합적인 분류 체계를 가지지 못하는 문제점이 있다. 이렇게 오픈소스 소프트웨어 환경이 커지는 상황에서 오픈소스 소프트웨어 분류 도구는 소프트웨어 환경의 지속적인 변화와 성장을 위해 필수적으로 갖춰야 하는 기술이다. 이는 다양한 오픈소스 소프트웨어를 쉽게 검색하고 활용할 수 있을 것으로 예상하며 더 나아가 소프트웨어 추천, 평가, 필터링, 관련성 분석 연구의 기반이 될 것 이다.

2. 관련 연구

2.1 TF-IDF 기법

검색엔진이나 텍스트 마이닝에서 흔히 볼 수 있는 기법인 TF-IDF 기법은 문서 내의 단어들을 중요한 단어와 중요하지 않은 단어들에 다르게 가중치를 주며 계량화하여 문서의 유사도나 중요도를 계산하는 기법이다. TF(Term Frequency)는 특정 단어가 문서 내에서 나타나는 빈도수를 나타낸 값으로, 단어가 문서 내에 자주 나타날 때 증가하며 DF(Document Frequency)는 특정한 단어가 전체 문서 내에 포함되는 문서의 수를 나타내는 값이며 IDF (Inverse Document Frequency)는 그 역의 값을 의미한다. TF-IDF는 단일 문서에서 많이 나오지 않고 여러 문서에서 자주 등장하면 단어의 중요도는 낮아진다. TF-IDF 값이 큰 단어일수록 문서의 아이덴티티 (Identity)를 더 높게 반영한다고 할 수 있다.

2.2. 머신러닝 모델 별 특징

표 1. 머신러닝 모델 별 특징
Table 1. Features of machine learning model

머신러닝 모델	모델 별 특징
로지스틱 회귀 (Logistic regression)	• 범주형 데이터를 종속 변수로 취급하며 입력 데이터의 결과가 특정 분류로 나뉘는 분류 기법 • 훈련 속도가 빠르지만 일반화 성능이 떨어짐
선형 지원 벡터 분류 (Linear SVC)	• 데이터 공간을 가우시안 커널을 이용하여 고차원 특징 공간으로 이동시키는 기법 • 손실 함수를 선택할 때 더 많은 유연성을 가지며 많은 수의 샘플로 확장해야함
나이브 베이즈 분류 (Naive bayes classification)	• 특성들 사이에 독립을 가정하는 기법 • 이론이 어렵지 않고 구현이 간단하고 복잡한 상황에서 잘 작동하기 때문에 다양한 분야에서 사용되는 기법 • 특성 별로 개별 취급해 파라미터를 학습하여 각 특성에서 클래스 별로 결과를 나타내기 때문에 효율적
랜덤 포레스트 (Random forest)	• 앙상블 학습 방법의 일종으로 분류, 회귀 분석 등에 사용되는 기법 • 훈련 과정으로부터 구성된 다수의 결정 트리로부터 분류 혹은 예측 값을 출력하는 기법 • 로우 데이터를 별도의 가공 없이 테스트 데이터로 바로 사용할 수 있음 • 데이터 세트가 적을 경우 효율적이지 않고 다른 단일 모델에 비해 예측하는데 많은 시간이 소요
SGD Classifier (Stochastic Gradient Descent Classifier)	• 퍼셉트론에 쓰이는 일반적인 최적화 방법 • 한번에 하나의 오 분류된 데이터만을 이용하여 가중치를 조정하는 기법
Gradient Boosting Classifier	• Boosting에 Gradient Descent를 접목시킨 머신러닝 기법 • 함수의 기울기를 측정하여 기울기가 낮은 쪽으로 이동시키면서 극값에 이를 때까지 반복하는 방법 • Boosting이란 단순한 모델들을 결합하여 단계적으로 학습함으로써 이전 모델의 약점을 점점 보완해 가는 모델

2.3. InformatiCup 2017 competition

Informaticup은 독일의 컴퓨터 과학협회에서 주최하는 경진대회의 일종으로 매년 컴퓨터 과학과 밀접한 주제를 선정하여 해당 과제의 학습, 프레젠테이션, 문제 해결 능력을 경쟁하는 대회이다. Informaticup 2017의 과제는 git 기반으로 운영되는 Microsoft의 오픈소스 공개 저장소인 Github를 인공 지능 및 데이터 마이닝을 적용하여 자동으로 레파지토리를 분류하는 것이다. 본 대회의 목적은 내용이나 컨트리뷰트 활동에 근거해 레파지토리를 자동으로 분류하고 태그 함으로써 더 나은 검색 결과를 제공함에 있다. 레파지토리를 DEV, HW, EDU, DOCS, WEB, DATA의 6개의 주제로 분류하고, Precision과 Recall을 측정하여 성능을 평가한다.

본 대회에서 우승을 차지한 Andreas Grafberger[4]외 3인의 연구에서는 훈련 데이터를 수집하기 위하여 GitHub API를 이용해 메타데이터, 프로젝트 설명, ReadMe 파일, 소스 코드에 대한 데이터를 수집하고 수치 표현에 해당하는 데이터와 텍스트 표현에 해당하는 데이터를 구분한 뒤 레파지토리에 필요한 정보를 가진 웹사이트를 설정하고 데이터베이스 서버에 저장된 결과 데이터 샘플을 분류하는 작업을 하여 2000개 이상의 분류된 데이터를 훈련 데이터 샘플로 사용했다. 분류기 구축을 위하여 사용된 신경망 이론으로는 경사 하강법을 적용한 LSTM(Long Short-Term Memory)[5], 비 선형 RBF-Kernal 기법을 사용한 SVM(Support Vector Machine)[6], Naive Baye와 앙상블 기법으로 Random Forest, Gradient Tree Boosting[7] 기법을 활용했다. 카테고리별로 40개의 레파지토리를 사용하여 분류기를 검증하였다.

본 연구는 약 58%의 분류 정확도를 보였다. 이는 실제로 카테고리를 자동 분류하기 위해 적용하기에는 낮은 정확도로 판단되며 또한 6개의 카테고리로 오픈소스 소프트웨어를 분류하기에는 한계가 있을 것으로 보인다.

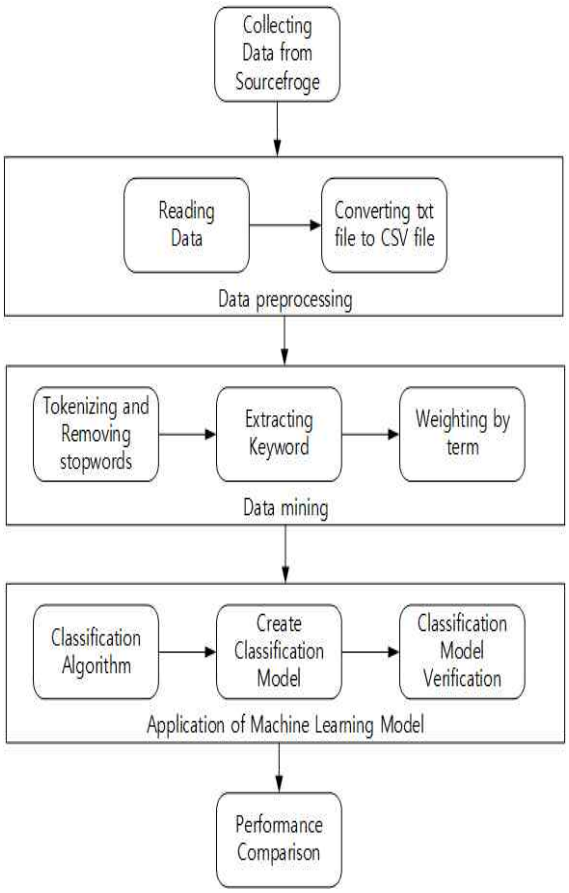

그림 1. 오픈소스 소프트웨어 주제 분류를 위한 머신러닝 모델 성능비교 구조도

Fig 1. The Structure of performance comparison for open source software category classification depended on machine learning models

3. 분석 및 설계

본 연구는 오픈소스 소프트웨어에 대한 주제 분류에 대한 모델을 제안하고, 다양한 머신러닝 모델을 적용하여 그 성능을 비교 분석하는 것에 대해 제안한다. 오픈소스 소프트웨어 주제 분류를 위한 머신러닝 모델 성능 비교를 위한 모델은 크게 다섯 가지인 데이터 수집 단계, 데이터 전처리 단계, 데이터 마이닝 단계, 머신러닝 모델 적용 단계, 성능 평가 단계로 나뉜다.

3.1. 데이터 전처리

이 단계에서는 오픈소스 소프트웨어 프로젝트의 텍스트 파일인 Readme, Metadata, CommitMessage, Repository Structure를 읽어 들여 머신러닝 입력 데이터로 사용하기 위해 CSV 파일로 변환하며 각 오픈소스 소프트웨어 프로젝트와 주제를 매핑 시킨다.

3.2. TF-IDF 기법을 사용한 데이터 마이닝

오픈소스 소프트웨어의 주제를 보다 정확하게 분류하기 위하여 TF-IDF 기법을 사용하여 오픈소스 소프트웨어 주제 분류에 사용될 데이터 세트의 단어 형태소를 분석하여 어간을 추출하고 컴퓨터 언어별 특징을 고려하여 키워드 혹은 예약어를 불용어로 선정하고 선정된 불용어 (예: abstract, arguments, break, boolean, class 등) 및 특수문자를 제거한다. 또한, 영어의 일반적인 대명사를 제거함으로써 노이즈를 줄이는 과정을 가진다. 그 후 데이터 세트를 분석하고 분류 모델에 적용하기 위해 각 오픈소스 소프트웨어의 데이터 세트에 포함된 텍스트를 수치로 이루어진 특징 벡터로 변환한다.

3.3. 머신러닝 모델 적용

오픈소스 소프트웨어 주제를 분류하기 위해 머신러닝 기법의 하나인 지도학습 기법을 사용한다. 지도학습은 데이터에 대해 명시적 정답을 나타내는 레이블을 준 상태에서 학습시키는 방법으로 본 연구에서는 데이터로 오픈소스 소프트웨어의 Readme, Metadata, CommitMessage, Repository Structure와 같은 텍스트 파일 사용하며 데이터에 대한 레이블로는 각 오픈소스 소프트웨어의 주제를 사용한다. 본 연구에서 주제는 오픈소스 소프트웨어 저장소인 소스포지의 카테고리를 기반으로 동일한 주제들의 오픈소스 소프트웨어들 간의 응집도는 상대적으로 높고, 다른 주제와 결합도가 상대적으로 낮아 각 오픈소스 소프트웨어 간의 경계가 확실한 Accounting, BBS, Board Games, File Transfer Protocol (FTP), Information Analysis, Side Scrolling Arcade Game, Testing을 활용하며 이를 데이터 세트의 레이블로 사용한다. 오픈소스 소프트웨어 주제를 분류하기 위해 머신러닝 모델 중 로지스틱 회귀(Logistic regression), 선형 지원 벡터 분류(LinearSVC), 나이브 베이즈 분류(Navie Bayes Classification), 랜덤 포레스트(Random forest), SGDClassifier(Stochastic Gradient Descent Classifier), GradientBostingClassifier)를 적용시키며 모델별 성능을 비교 분석한다.

3.4. 모델 평가

k 개의 fold를 만들어서 교차 검증을 진행하는 k-교차(k-fold cross validation) 기술을 적용해 모델을 평가하는 기법을 적용한다. 훈련될 훈련 세트, 최적의 매개 변수를 찾는데 사용되는 유효성 검증 세트, 모델의 성능을 평가하는 테스트 세트로 데이터를 세 부분으로 나누는 것이 일반적이지만 검증 세트와 테스트 세트를 같이 사용함으로써 훈련 데이터양을 늘려 정확도를 늘릴 수 있다. 본 연구에서는 k=5개의 검증 세트를 만들어서 모델 평가를 진행하며 scikit-learn의 유틸리티 함수인 cross_val_score를 사용하여 평가한다.

4. 실험

4.1. TF-IDF기법을 사용한 데이터 마이닝

오픈소스 소프트웨어 프로젝트 파일 중 텍스트 파일을 분류하고 언어별 예약어, 영어 대명사, 불용어 처리 등을 하는 전처리 과정을 거친 뒤 각 주제별 특징을 파악하기 위해 표 2와 같이 상위 10개 키워드를 추출해 냈다. TF-IDF 기법을 사용해 추출된 주제 별 키워드를 살펴본 결과 아이디 혹은 의미 없는 변수를 나타내는 몇 개의 값을 제외하고는 대부분 주제의 특징을 대표할

수 있는 키워드들이 추출된 것을 확인할 수 있다.

표 2. 상위 10개 키워드 추출 결과
Table 2. Top 10 Keyword Extraction Results

순위	Accounting	BBS	Board Games	Boot	FTP	Info. Analysis	SSAG	Testing
1	derby	sql_query	game	fat	ftp	Swierczek	sprites	testresult
2	income	replies	board	booting	slave	svm	enemies	tests
3	ledger	phpex	cardset	refind	dataproxy	rs422	sdl_rect	test
4	diesel	bbcode	queen	bscript	slaves	national	sprite	eclipse
5	saldo	admin	knight	bootable	mkd	plugin	game	xml
6	expense	avatar	bishop	partitions	retr	mwt	sdl_surface	appsettings
7	petrol	phpbb	rook	bios	rmd	instruments	enemy	junit
8	debit	forum_id	chess	partition	ident	pk	ogg	testname
9	boos	forum	king	mbr	cubnc	labview	sdl	testcase
10	accounts	posts	pawn	boot	bnc	tulp2g	ck1	framework

추출된 키워드들을 TF-IDF 기법을 통해 특징 벡터값으로 변환시키고 그 결과를 시각화시킨 결과는 그림 2와 같다. 그림 2의 결과를 통해 보면 주제별로 그래프 상 가까운 위치에 군집되며 각 주제에 속하는 단어들은 비슷한 특징 벡터값을 가지는 것을 확인할 수 있다.

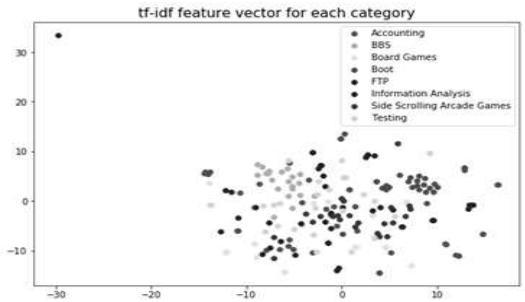

그림 2. 주제 별 특징 벡터 값 시각화
Fig 2. Visualization of subject-specific vector values

4.2. 주제 별 예측 결과

실제 주제(X축)와 검증 데이터 세트를 통해 예측된 주제(Y축)를 비교한 결과는 그림 3과 같다. 그림을 통해 살펴보면, Boot 주제에서 예측이 가장 정확하며 Board Games와 Information Analysis 주제에서 실제 주제가 아닌 다른 주제로 예측하는 경우가 가장 많이 발생했다. Boot 주제의 상위 키워드는 주제의 특성을 잘 드러내는 단어들로 구성된 반면에 Information analysis 주제는 의미 없는 단어들이 다수 등장하며 Testing 주제로 예측하는 경우가 많으며, Board Games 주제와 SSAG 주제는 유사한 키워드들로 구성되어 정확한 예측에 어려움을 지니는 것을 확인할 수 있다.

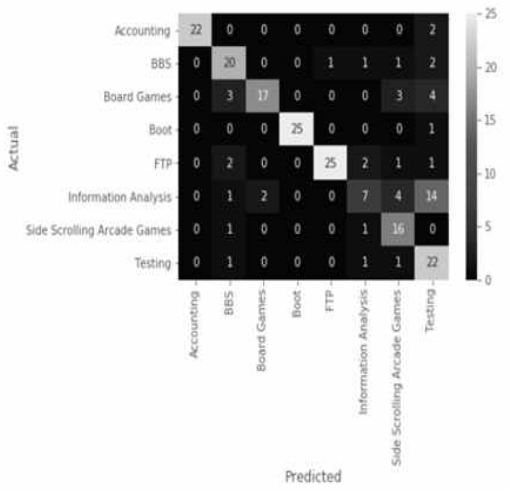

그림 3. 주제 별 예측 결과
Fig 3. Forecasts by Category

4.3. 머신러닝 모델 별 성능 비교

오픈소스 소프트웨어 주제 분류를 위해 다양한 머신러닝 모델을 적용했다. 머신러닝 모델 중 데이터에 대해 명시적 정답을 나타내는 레이블을 준 상태에서 학습시키는 방법인 지도학습 중 분류기법에 사용되는 모델들을 본 실험에 적용했다. 실험에 사용된 모델은 로지스틱 회귀, 선형

지원 벡터 분류, 나이브 베이즈 분류, 랜덤 포레스트, SGDClassifier, GradientBoostingClassifier를 적용해 보았으며, 머신러닝 모델별 적용 결과는 그림 4와 표 3과 같다. 모델 별 정확도 순서로는 SGDClassifier, GradientBoostingClassifier, 랜덤 포레스트, 나이브 베이즈 분류, 선형 지원 벡터 분류, 로지스틱 회귀 순으로 나타났다. 가장 정확도가 높은 모델인 SGDClassifier 모델은 평균 77%의 분류 정확도를 보였으며 검증 모델별 최고 정확도로 82%의 분류 정확도를 보였다. 가장 분류 정확도가 낮은 모델은 로지스틱 회귀 모델로 최고 73% 분류 정확도와 평균 64.5%의 정확도를 보였다. 가장 성능이 좋은 SGDClassifier와 가장 성능이 떨어지는 로지스틱 회귀 모델의 평균 정확도는 약 17.5%의 분류 성능 차이를 나타냈다.

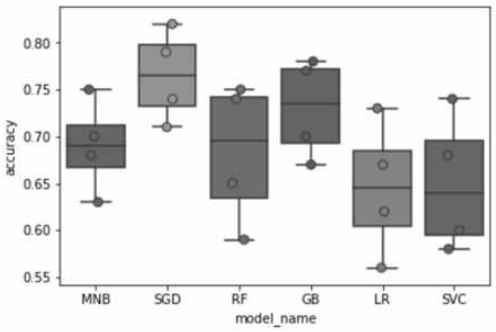

그림 4. 모델 별 적용 결과
Fig 4. Application results by model

표 3. 모델 별 적용 결과
Table 3. Application results by model

머신러닝 모델	1	2	3	4
MNB	63%	75%	68%	70%
SGD	82%	74%	71%	79%
RF	59%	65%	74%	75%
GB	78%	77%	70%	67%
LR	67%	73%	62%	56%
SVC	60%	74%	68%	58%

4.4. 기존연구와의 성능 비교

기존연구와의 성능을 비교하기 위해서 분류 모델 평가 척도 중 f-score를 사용한다. f-score는 데이터 분포를 고려한 지표로 precision(정밀도)과 recall(재현율)을 사용하여 계산한다. Precision은 출력 결과가 정답을 얼마나 맞혔는지를 나타내는 지표이다. 한편, recall은 출력결과가 실제 정답 중에서 얼마나 맞혔는지를 나타내는 지표다. F-score란 큰 의미상으로 보면 precision과 recall에 대한 평균인데, 그냥 평균을 내면 값의 왜곡 현상이 생기기 때문에, 가중치를 주어 평균값을 구한다. 표 4는 오픈소스 소프트웨어 주제 분류의 기존연구인 Informaticup 2017 실험 결과를 나타낸 표이며 표 5는 본 연구의 실험 결과이다.

표 4. Informaticup2017 실험 결과
Table 4. Informaticup2017 experimental results

카테고리	precision	recall	f1-score
DEV	0.47	0.9	0.62
HW	1	0.17	0.29
EDU	0.83	0.71	0.77
DOCS	0.5	0.33	0.40
WEB	0.67	0.67	0.67
DATA	0	0	0
OTHER	0	0	0
Avg/Total	0.58	0.58	0.58

표 5. 오픈소스 소프트웨어 주제 분류 실험 결과
Table 5. Open source software subject classification experiment results

카테고리	precision	recall	f1-score
Accounting	0.83	0.70	0.76
BBS	0.59	0.68	0.63
Board Games	0.92	0.77	0.84
Boot	1.00	0.64	0.78
FTP	0.55	0.68	0.61
Info.Analysis	0.68	0.72	0.70
SSAG	1.00	0.96	0.98
Testing	0.72	0.96	0.82
Avg/Total	0.79	0.76	0.77

기존연구와 본 연구의 주제를 분류하는 방법과 분류된 주제의 개수가 다르므로 정확하게 성능을 비교하기는 어렵지만, 단순히 주제별 평균 f1-score를 비교했을 때, 본 연구는 기존연구와 비교하면 약 19% 향상된 분류 성능을 보인다고 볼 수 있다.

5. 결론

본 논문에서는 다양한 머신러닝 모델을 사용하여 오픈소스 소프트웨어 주제를 분류하는 기법에 대하여 제안하고 머신러닝 모델별 성능 비교에 관하여 연구한다. 이를 통해 기존에 사람이 주제를 직접 입력하거나 태깅함으로써 발생하는 오류나 번거로움 없이 오픈소스 소프트웨어 분류를 자동화하여 기존의 방식보다 빠르고 정확한 분류 작업이 가능할 것으로 보인다. 또한, 다양한 오픈소스 소프트웨어를 쉽게 검색하고 주제별로 오픈소스 소프트웨어를 검색할 수 있을 것이다.

다양한 머신러닝 모델에 오픈소스 소프트웨어 주제 분류를 적용한 결과 SGDClassifier 모델이 최고 82%, 평균 77%를 보이며 가장 높은 정확도를 보였다. 모델 별 정확도 순서는 SGDClassifier, GradientBoostingClassifier, 랜덤 포레스트, 나이브 베이즈 분류, 선형 지원 벡터 분류, 로지스틱 회귀 순으로 나타났다. 기존연구인 Informaticup 2017과의 성능비교에서도 주제 분류 수를 늘렸음에도 약 19%정도 향상된 성능을 확인할 수 있었다.

하지만 여전히 9개로 제한된 주제로는 모든 오픈소스 소프트웨어를 분류해내기 어려워서 더 많은 분류 체계를 만들어서 확장 및 적용을 시키는 방안이 필요하며 분류 정확도를 향상하기 위한 추가적인 연구도 필요하다. 또한, 오픈소스 소프트웨어 분류 연구를 기반으로 사용자에게 적당한 오픈소스를 추천해 주는 모델은 다음 연구를 통해 진행될 예정이다.

Acknowledgement

본 연구는 문화체육관광부 및 한국저작권위원회의 2018년도 저작권기술개발사업의 연구결과로 수행되었음

참 고 문 헌

[1] 정보통신산업진흥원, “2018 공개SW 기업 편람”, 2017.

[2] Apple, “Apple Releases Swift as Open Source”, https://www.apple.com/newsroom/2015/12/03Apple-Releases-Swift-as-Open-Source/, 2015.12.

[3] Black Duck Software. “The tenth annual future of open source survey”, http://nbvp.northbridge.com/2016-future-open-source-survey-results, 2016.

[4] A. Grafberger, M. Leimstadtner, S. Grafberger, M. Keßler, “Documentation: GitHub Classifier for the InformatiCup 2017”, https://github.com/Ichaelus/Github-Classifier, 2017.

[5] S. Hochreiter, J. Schmidhuber, “Long short-term memory”, Neural computation, vol.9, no.8, pp.1735-1780, 1997.

[6] M.A. Hearst, S.T Dumais, E. Osuna, J. Platt, B. Scholkopf, “Support vector machines”, IEEE Intelligent Systems and their applications vol.13, no.4, pp.18-28, 1998.

[7] T. G. Dietterich, A. Ashenfelter, Y. Bulatov, “Training conditional random fields via gradient tree boosting”, Proceedings of the twenty-first international conference on Machine learning. ACM, p. 28, 2004.

저 자 소 개

이예슬(Ye-Seul Lee)

2018년 숭실대 융합소프트웨어학과 석사
2018년-현재 엘에스웨어(주) 선임
<주관심분야> 머신 러닝, 딥 러닝, 빅 데이터, 보안

조용준(Yong-Joon Joe)

2011년 큐슈대학교 전기정보공학과 학사
2016년 큐슈대학교 정보학과 박사과정 수료
2016년-현재 엘에스웨어(주) 선임
<주관심분야> 게임이론, 분산 최적화 이론, 인공지능, 블록체인

백승찬(Seung-Chan Back)

2015년 한국산업기술대학교
컴퓨터공학과 학사
2017년 서울시립대학교 컴퓨터과학과 석사
2017년-현재 엘에스웨어(주) 선임
<주관심분야> 소프트웨어 공학, 소프트웨어 테스팅, 블록체인

신동명(Dong-Myung Shin)

2003년 대전대학교 컴퓨터공학과 박사
2001년-2006년 한국정보보호진흥원
응용기술팀 선임연구원
2006년-2014년 한국저작권위원회
저작권기술팀 팀장
2014년-2016년 한국스마트그리드사업단
보안인증팀 팀장
2016년- 현재 엘에스웨어(주) 연구소장/이사
<주관심분야> 오픈소스 라이선스, 시스템/네트워크보안, SG인증/보안, SW취약점분석・감정, 블록체인