논문 2018-1-2

머신러닝 기반의 오픈소스 SW 카테고리 분류 모델 연구

백승찬*, 최현재*, 윤호영**, 조용준*, 신동명*†

Machine Learning based Open Source Software Category Classification Model

Seung-Chan Back*, Hyunjae Choi*, Ho-Yeong Yun**, Yong-Joon Joe*, Dong-Myung Shin*†

요 약

기업과 개인 여러 방면에서 오픈소스 SW의 사용과 중요성은 날이 갈수록 증가하고 있다. 그러나 사용자를 위한 소프트웨어 서비스인 소프트웨어 평가, 추천, 필터링들의 기반 연구인 소프트웨어 분류에 대해서 오픈소스 SW의 특성에 맞게 유연하게 대처하지 못하고 고정된 분류 체계를 사용하고 있다. 본 연구에서는 오픈소스 SW를 대상으로 분류에 대한 조사와 새로운 오픈소스 SW 범주에 대해서 유연하게 대처할 수 있는 머신러닝 기반의 오픈소스 카테고리 분류 모델에 대해 제안한다.

Abstract

In many respects, the use and importance of open source software in companies and individuals are increasing as the days pass. However, software evaluation for users, software classification of filtering fundamentals research can not deal flexibly according to the characteristics of open source software. They are using a fixed classification system. In this research, we provide a classification model of open source software that can flexibly deal with the classification of open source software and the software category of new open source software.

한글키워드 : 오픈소스 SW, TF-IDF, 소프트웨어 분류, 머신러닝

keywords : Open Source Software, TF-IDF, Software Classification, Machine Learning

1. 서 론

2016년 조사 결과에 따르면 기업의 78%가 오픈소스 소프트웨어(이하 오픈소스 SW)를 사용하고 있고[1], 2020년도의 국내 오픈소스 SW 시장 규모는 2,862억원으로 2015년도에 비해 약 두 배 가까이 성장할 것으로 예상한다[2].

국외 기업으로는 아파치, 소프트웨어 재단, 구글, 페이스북과 국내 기업으로는 네이버 랩스, 우아한 형제들, 삼성전자 등이 자사의 제품을 오픈소스로 배포하고 기술을 이끌며 상업용 SW와 비교해도 오히려 강세를 보이며 오픈소스 SW 생태계를 풍성하게 하고 있다.

이러한 오픈소스 SW 시장의 성장과 더불어

* 엘에스웨어(주)
** 연세대학교 산업공학과
† 교신저자: 신동명(roland@lsware.com)
접수일자: 2018.05.28. 심사완료: 2018.06.12.
게재확정: 2018.06.20.

오픈소스 SW 분류 혹은 카테고리 체계 구성에 대한 연구가 진행 중이다. 기존의 소프트웨어 분류는 유사한 기능 혹은 목적을 가진 소프트웨어들을 하나의 범주로 묶어서 카테고리를 구성하는 것[3]으로 소프트웨어에 대한 질의를 간편하게 제공하는 것뿐만 아니라 소프트웨어 추천, 소프트웨어 평가, 소프트웨어 필터링, 소프트웨어 관계성 분석 등의 연구의 기반으로 활용되고 있다. 그러나 오픈소스 SW 분류 및 카테고리 구성에 관한 연구는 상용 소프트웨어와는 달리 특정 오픈소스 SW에 대한 관리 기관이 불확실하다는 점과 수•출입에 사용되는 산업 분류 체계가 기존의 소프트웨어 현황을 단순화하고 있다는 점에서 기존의 체계를 적용하기 어렵다[4].

본 연구와 유사한 목적으로 독일의 컴퓨터 과학 경진대회 InformatiCup는 2017년도의 주제로 오픈소스 SW 저장소 중 하나인 Github를 대상으로 오픈소스 SW 프로젝트 에 대한 분류를 시행했던 바가 있다. 해당 분류는 신경망 이론부터 다양한 머신러닝 기반의 연구가 진행되었으며 이는 2장의 관련연구에서 다룬다.

본 연구의 목적은 수집된 오픈소스 SW를 대상으로 해당 프로젝트가 가지고 있는 소스코드에서 의미없는 단어는 제거하고 TF-IDF와 머신러닝 기법을 활용해 오픈소스 SW를 분류하는 것이고 구성은 다음과 같다. 2장에서는 기존 분류 연구에서 사용된 기법들과 오픈소스 SW 분류대회인 InformatiCup에서 텍스트 처리 및 머신러닝 기법을 사용한 연구들에 대해서 서술하고 제3장에서는 본 연구에서 제안하는 오픈소스 SW 분류 기법에 대해 제시한다. 4장에서는 본 논문의 실험을 5장에서는 마지막으로 본 연구에 대한 결론 및 한계점으로 끝을 맺는다.

2. 관련 연구

2.1 분류 기법

데이터 전처리와 자연어 처리를 위하여 널리 적용되는 분류 기법으로 워드 임베딩(Word-Embedding) 기법과 머신 러닝(Machine Learning) 기법이 있다. 워드 임베딩 기법은 여러 문서로 이뤄진 문서군의 단어를 분석하여 핵심어를 추출하거나 문서들 사이의 유사성을 구하는 것을 목적으로 한다[5]. 머신 러닝 기법은 인공지능의 한 분야로, 데이터로부터 데이터의 패턴을 검증하고 스스로 학습하여 새로운 데이터에 대한 예측을 제시하는 것이다.

워드 임베딩은 크게 단어 예측 기반 기법과 단어 빈도수 기반이 있으며, 대표적인 단어 빈도수 기반 기법으로는 TF-IDF가 있다. TF는 아래 수식과 같이 단어의 빈도를 뜻하며, 특정 단어가 문서 내에서 얼마나 자주 등장하는지를 나타내는 값이다.

$$w_{i,j} = TF_{i,j} \times \log(\frac{N}{DF_i})$$

$TF_{i,j}$ = number of occurrences of i in j

DF_i = number of documents containing i

N = total number of documents

TF-IDF는 단순하게 단어의 빈도와 중요도는 비례한다고 해석할 수도 있지만, 단일 문서가 아닌 다수의 문서가 있는 경우에는 이는 다르게 해석가능하다. 단어 자체가 문서군 내에서 자주 사용되는 경우는 해당 단어는 흔하게 등장하는 것을 의미, 즉 중요도가 낮다고 해석할 수 있다. 이를 문서의 빈도를 뜻하는 DF라고 하며, 이 값의 역수를 IDF라고 한다. TF-IDF는 TF와 IDF의 곱한 값으로 수식으로 표현하면 위와 같다. DF 값을 역수로 변환하게 되면 곡선의 형태를 띄기 때문에 로그값을 취하여 TF의 값과 동일하게 선형의 값을 갖게 변환하는 과정을 거친다.

머신러닝은 지도학습(Supervised Learning),

비지도학습(Unsupervised Learning), 강화학습(Reinforcement Learning)의 방법이 있으며, 분류 모델은 지도학습에 속하게 된다[6].

분류 모델 알고리즘은 기존에 존재하는 데이터와 카테고리와의 관계를 학습하여 새로 관측된 데이터의 카테고리를 판별하는 구조로 대표적으로 의사결정 트리, 회귀 트리, KNN 기법이 존재한다.

의사결정 트리는 초기 지점인 Root 노드에서 분기를 거듭해나가는 형태를 보여준다. 새로운 데이터가 특정 터미널 노드에 속한다는 정보를 확인한 뒤 해당 터미널 노드에서 가장 빈도가 높은 범주에 새로운 데이터를 분류하게 된다. 하지만 의사결정트리의 출력 값은 반드시 이진(binary)형태여야 한다는 제약조건이 있기 때문에 오픈소스 SW 카테고리 분류 모델에 적용하기 위해서는 변형이 필요하다.

회귀 트리는 이진값이 아닌 실제 값 출력하는 모델로, 학습하는 방법이 크게 다르지 않아 의사결정트리보다는 적합한 모델이다. 또한 하나의 의사결정트리를 사용하는 것이 아니라 한 번에 여러 개의 의사결정트리를 만들어서 각각의 의사결정트리들이 내리는 결정을 종합적으로 판단하는 앙상블 학습 기법도 있다.

마지막으로 K-Nearest Neighboring(KNN)은 가까운 데이터는 같은 속성일 가능성이 크다고 가정하는 모델로, 새로운 데이터가 들어오면, 학습된 데이터 중 그 데이터와 가장 가까운 k개의 데이터를 뽑는 알고리즘이다. 뽑은 k개의 데이터들의 속성을 관측하고 그 중 가장 많은 속성을 새로운 데이터의 속성으로 할당하게 된다. 비교적 구현이 쉽고 변형하기에도 간단하다는 장점이 있다.

2.2 InformatiCup 2017 competition

InformatiCup 2017은 지능화된 사용자 서포트 서비스의 발전을 도모하기 위해 현존하는 최대의 레파지토리인 GitHub의 레파지토리가 포함하고 있는 요소에 따라 DEV, HW, EDU, DOCS, WEB, DATA, OTHER의 7개의 주제로 분류하고, 정밀도(Precision)와 회수율(Recall)을 사용하여 수상작을 선정하는 독일에서 열리는 컴퓨터과학 경진대회이다.

S.Marcus[7]외 1인이 연구하고 개발한 ClassifyHub 연구는 InformatiCup 2017의 문제를 해결하기 위해 레파지토리가 포함하고 있는 요소에 따라 약분류기(Weak Classifier)를 구성하고, 약분류기들의 분류 결과에 평균치를 이용하는 앙상블 학습(Ensemble Learning)[8]을 이용해 강분류기(Strong Classifier)를 구축하였다.

File Extension, ReadMe, MetaData, Language, Language Details, Name, Commit Message, Repository Structure을 약분류기의 데이터 요소로 삼았으며 이 중에서 텍스트 데이터와 관련된 요소는 ReadMe, CommitMessage, RepositoryStructure이다.

텍스트 데이터를 사용한 분류기들은 Word Embedding 기법 중 단어의 빈도수만을 계산하는 Bag Of Words 표현을 사용해서 텍스트를 수치화 하였다. 출현 빈도수가 모든 문서 내에서 0%와 100%에 가깝거나 불용어에 해당하는 단어들을 제거하고 Bag Of Words 표현으로 된 단어간 Jaccard Distance이 적용된 K-Nearnest Neighborhood을 활용해서 레파지토리의 카테고리 확률을 사용하였다.

S.Marcus[7]의 연구에서는 텍스트 데이터 이외에도 Fork 수, Wiki 페이지 포함 여부, 저장소 사이즈, Like 수 등 다양한 요소를 활용해서 약분류기를 추가로 구성하였고 결과적으로 정밀도 59.90%와 회수율 58.41%로 InformatiCup 2017의 수상작 중 하나로 선정되었다.

Git Better는 Gierke.W[9]외 1인이 연구한 분

류기의 이름이며 ClassifyHub와 유사하게 GitHub API를 이용해 메타데이터, 레파지토리 설명, ReadMe 파일과 소스코드를 하나의 요소로 활용하여 각각 약분류기를 구성하였다. Git Better가 약분류기의 결함으로 사용하는 강분류기는 다수결 규칙 알고리즘(Majority Rule Algorithm)이 적용된 앙상블 학습 알고리즘[10]을 적용하여 구축하였다.

Gierke.W[9] 연구는 레파지토리 요소 특징에 따라 분류기를 나누었는데 하나의 요소는 레파지토리의 수치 표현에 해당하는 메타데이터(Numberic Metadata Of Repository)이고 하나는 ReadMe, Description, 소스코드의 텍스트 데이터이다. 수치 표현 메타데이터는 GitHub REST API와 GitHub GraphQL에서 수집하는 Number Of User, Number Of Issues, Number Of Requests, Number Of Forks 등 Numberic 값이 붙어서 수치적으로 표현할 수 있다는 것이 특징이다.

소스 코드, 커밋 메시지, Wiki, 확장자 이름 등을 학습 데이터로 사용한 텍스트 데이터 분류기는 워드 임베딩으로 TF-IDF 기법을 활용해서 텍스트 전처리를 수행하였고 데이터 처리에 혼란을 가져올 수 있는 불용어와 빈도 수가 적거나 너무 많은 데이터는 미리 삭제하였다. 그리고 Linear SVM(Surpport Vector Marchine)[11]을 활용해서 텍스트 데이터의 약분류기를 구축하였다.

S.Marcus[7]와 Gierke.W[9]의 연구의 결과는 InformatiCup 2017에서 우수 프로젝트로 입상된 연구이고, 텍스트 데이터와 머신러닝을 활용한 분류기의 적합성을 보여주었다. 그러나 위 두 개의 연구가 분류한 7개의 카테고리는 기존의 카테고리 체계와는 많은 차이를 보인다.

3. TF-IDF기법과 머신러닝을 이용한 오픈소스 SW 분류 기법

본 연구는 이전의 오픈소스 SW 프로젝트의 내적 정보를 바탕으로 유연하게 카테고리 체계를 지정할 수 있는 모델[12]의 비균형 오픈소스 SW 분류를 TF-IDF 기법과 머신러닝을 적용해 구현하고, 적용한 머신러닝 알고리즘의 결과 비교를 통해 적합성을 분석을 진행하였다.

3.1 오픈소스 SW 분류 모델

선행연구인 오픈소스 SW 카테고리 분류 모델[12]은 그림 1과 같이 오픈소스 SW 저장소에 있

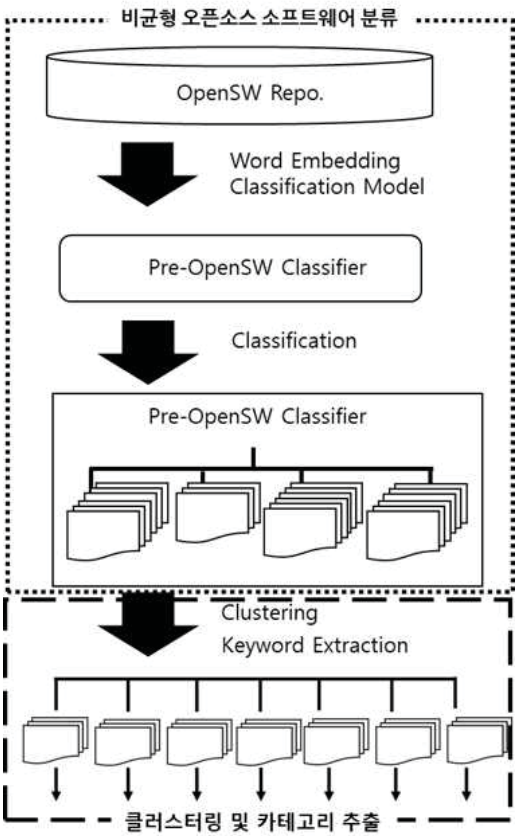

그림 1. 오픈소스 SW 카테고리 분류 모델
Fig. 1. OSS category classification Model

는 오픈소스 SW 프로젝트를 대상으로 (1) 비균형 오픈소스 SW 분류 과정을 통해 일차적인 분류를 시행하고, 하나의 카테고리에 속하는 오픈소스 SW프로젝트의 개수가 평균과 비교해 많은 차이를 보이는 분류를 대상으로 K-Means, DBSCAN, 병합적 클러스터링 알고리즘, 분할적 클러스터링 알고리즘 등을 통해 (2) 클러스터링 및 카테고리 추출 과정을 진행하는 모델이다.

본 연구는 선행연구[12]에서 제안한 오픈소스 SW 카테고리 분류 모델을 기반으로 구현적인 이슈를 다루며 5장에서는 오픈소스 SW 저장소 중 하나인 소스포지(Source Forge)의 카테고리 분류체계를 대상으로 (1) 비균형 오픈소스 SW 분류과정을 TF-IDF 기법과 머신러닝 기법을 적용한 뒤 비교 및 분석하는 결과를 보여준다.

3.2 머신러닝을 이용한 오픈소스 SW 분류 기법

TF-IDF 기법과 머신러닝을 이용한 오픈소스 SW 분류 기법은 소프트웨어 분류 체계에 맞게 분류 된 오픈소스 SW 카테고리를 기준으로 새로운 오픈소스 SW 프로젝트가 입력 값으로 사용되면 적합한 분류를 배정해주는 작업이다.

이를 위한 오픈소스 SW 분류기의 구축은 총 3단계로 이루어져 있다. 먼저 오픈소스 SW 분류기의 학습을 위해 분류된 오픈소스 SW가 있는 저장소를 선정하고 자료를 수집한다.

사용자들이 가장 많이 이용하고 있는 오픈소스 SW 저장소는 깃허브와 소스포지이다. 깃허브의 경우 오픈소스 뿐만 아니라 공개될 수 있는 문서, 사진, 파일 등 모든 정보에 대해 저장하는 저장소로 현재 7,500만 개의 프로젝트를 보유하고 있다. 깃허브는 프로젝트에 대한 메타데이터, 설명, ReadMe 파일 등을 GitHub API를 통해 사용할 수 있게 지원하고 있다.

깃허브에는 소수의 프로젝트를 추천하기 위한 집합 혹은 프로젝트의 해시 태그를 이용해서 주체자가 추가하는 방식을 사용하고 있는데 지도학습의 분류체계로 사용하기에는 어려움이있다.

소스포지는 1999년에 출시된 오픈소스 SW 저장소로 깃허브가 모든 종류의 파일에 대해 공개 레파지토리로의 활용성을 갖는다면 소스포지는 소프트웨어 개발을 공개적으로 관리하고 배포해주는 목적을 지니고 있어 깃허브보다 규모는 작지만 개발과 소스코드 중심의 오픈소스 SW와 관련되어 있다. 뿐만 아니라 소스포지는 20개의 상위 분류와 370개의 하위 분류에 대한 카테고리 개념이 있어 데이터를 수집함에 있어 지도학습에 적합한 분류체계로 사용할 수 있다.

분류기 구축의 다음 과정은 오픈소스 SW 저장소에서 하나의 카테고리에 대해 파일 확장자를 기반으로 소스코드를 수집하고, 수집한 소스코드들을 하나의 문서로 취합해서 문서 내 단어의 빈도 수인 TF(Term Frequency) 값을 계산하고, 같은 방식으로 다른 카테고리마다 소스코드를 수집하여 하나의 문서로 취합한 후에 하나의 카테고리마다 다른 카테고리 문서들과 비교하여 DF(Document Frequency) 값을 계산한다.

분류기 구축의 마지막 단계는 카테고리 별로 구성된 TF-IDF 리스트릴 기반으로 머신러닝 알고리즘인 K-NN, 의사결정트리, 회귀 트리 등의 머신러닝 알고리즘 기법을 이용해서 분류기의 학습을 거친다. 이를 통해 분류기는 입력 값에 해당하는 오픈소스 SW 프로젝트가 소스코드를 통해 질의를 수행하면 소스코드와 분류기의 유사도 비교를 통해 오픈소스 SW 프로젝트가 주어진 카테고리 중 어느 카테고리에 속하는 지를 확률 값으로 계산을 통해 보여준다.

4. 실험 과정 및 결과

4.1 분류 체계 및 오픈소스 SW 저장소 선정

본 연구는 상위 오픈소스 SW 저장소 중 분류 20개와 하위 분류 370개로 분류를 진행하고, 오픈소스 SW에 대한 자료만을 관리하는 소스포지를 저장소로 선정하고, 이미 지정되어있는 분류 중 본 기법의 적합성을 실험하기 위해서 분류 별 교집합이 상대적으로 적은 8개의 분류(Accounting, BBS, Board Games, Boot, File Transfer Protocol(FTP), Information Analysis, Side Scrolling Arcade Games, Testing)를 선정해 실험을 진행하였다.

표 1. 실험 소스포지 프로젝트 수 및 카테고리
Table 1. Number of SourceForge Projects and Categories

Category	# of OSS
Accounting	175
BBS	205
Board Games	138
Boot	176
File Transfer Protocol(FTP)	137
Information Analysis	213
Side Scrolling Arcade Games	170
Testing	222

표 1은 해당 연구를 위해 수집한 카테고리와 오픈소스 SW 프로젝트의 개수를 보여준다.

실험 과정은 그림 2와 같이 3장에서 언급한대로 (1) 각 카테고리의 모든 오픈소스 SW 프로젝트의 c, cpp, java, h 파일 등의 을 비롯한 추출하고, 하나의 카테고리에 속하는 모든 소스코드를 하나의 문서로 통합하였다. (2)(3) 소스코드에 해당하는 예약어 및 불용어를 제거하고 각 카테고리는 하나의 문서가 되고 문서 내에서 TF값을 구하고 다른 카테고리와 비교해서 DF 값을 계산하여 TF-IDF 값을 계산하였다.

4.2 카테고리 별 TF-IDF 리스트

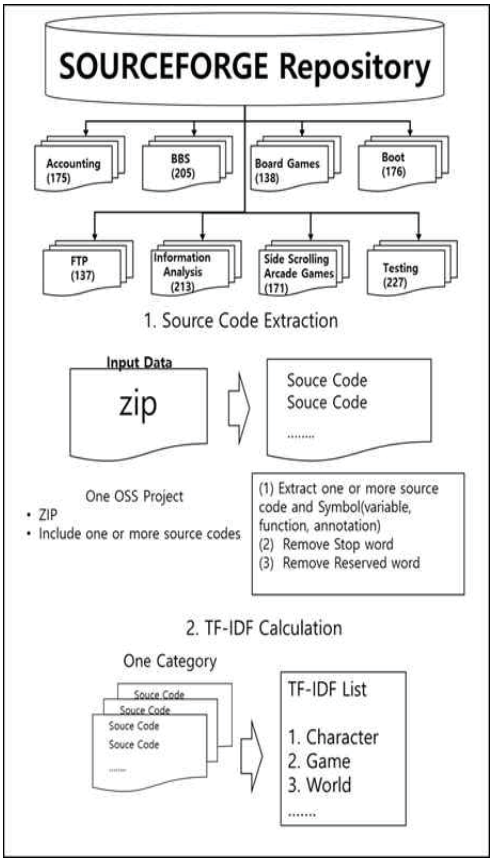

그림 2. 오픈소스 SW 카테고리 분류 실험과정
Fig. 2. Course of OSS Category classfication

소스포지의 오픈소스 SW 프로젝트를 Accounting, BBS, Board Games, Boot, File Transfer Protocol(FTP), Information Analysis, Side Scrolling Arcade Games, Testing으로 나누어 각 카테고리 별 TF-IDF 리스트를 추출하였다.

실험에 활용할 오픈소스 SW 프로젝트는 카테고리별로 250개씩 수집하였으나, 소스코드의 부재, 프로젝트 파일의 손상, Read Me 파일의 부재 등의 문제가 있는 프로젝트는 제외하였다. 표 3은 분석한 프로젝트의 수와 해당 프로젝트에 존재하는 텍스트 파일 수에 대한 정보이다.

표 2. 소스포지 카테고리별 단어 빈도 분석(상위 10개)
Table 2. Word frequency analysis by SourceForge cartegories

순위	Accounting	BBS	Board Games	Boot	FTP	Info. Analysis	SSAG	Testing
1	document	execute	game	refind	sitecommands	term	sprite	fff(duplicate)
2	license	bbcode	player	file	package	idf	posx	hello
3	software	required	version	status	skip	stemmed	posy	genus
4	calendar	matches	diether	license	ftpclientconnection	getidf	massive	parameter
5	company	templates	license	paypal	deprecated	cristina	brick	tree
6	foundation	permission	card	copyright	authentication	carlo	item	cairo
7	copyright	pagination	file	rodsbooks	connection	carlos	middle	relating
8	email	property	program	conf	ftpslaveconnection	fabio	passive	time
9	bookkeeping	configuration	gtk	partition	ftpclientcommand	camacho	underground	summary
10	voucher	assertfalse	chess	manager	slaveserver	petrillo	tolua_s	person

표 3. 카테고리 별 프로젝트 및 파일 수
Table 3. Number of projects and files

Category	프로젝트 수	파일 수
Accounting	171	27,436
BBS	201	35,447
Board Games	134	7,853
Boot	172	9,273
FTP	133	6,570
Info.Analysis	209	23,032
SSAG	166	10,791
Testing	218	48,533
합계	1,404	168,935

표 4. 카테고리 분류 모델별 인식률 비교
Table 4. Comparison of recognition rates by model

Category	Cosine Similarity	K-NN
Accounting	70%	90%
BBS	50%	80%
Board Games	100%	100%
Boot	40%	60%
FTP	80%	90%
Info.Analysis	70%	80%
SSAG	90%	100%
Testing	60%	70%

4.3 카테고리 분류 모델별 결과 비교

각 카테고리 별로 추출된 TF-IDF 리스트를 기반으로 전형적인 유사도 알고리즘인 Cosine유사도 기법과 머신 러닝 기법인 K-NN 분류 기법을 TF-IDF Weight로 활용하여 분류 모델을 학습한 내용을 비교 실험하였다. 실험 데이터는 정확한 분류 정보를 알고 있는 오픈소스 SW 프로젝트를 (프로젝트별로 10개씩 총 80개를 실험하였으며, 카테고리 분류 모델을 통해 분류한 결과는 표 4와 같다.

표 4에서 확인할 수 있듯이 TF-IDF 빈도 분석 후 Cosine 유사도를 통해, 유사성을 분석하여 카테고리를 분류하는 방법과는 달리 머신러닝 기법을 활용하여 분류하는 것이 높은 성능을 보였다.

Board Games 카테고리와 Side Scrolling Arcade Games 카테고리와 같이 특징이 명확한 카테고리에서는 Cosine 유사도와 K-NN 모두 정확하게 분류하는 결과를 보였다. 반대로 Boot 카테고리와 Testing 카테고리는 상대적으로 낮은 인식률을 나타냈는데, Game 카테고리와 달리 두드러진 특징성이 부족한 것으로 예상된다.

비슷한 이유로 정확히 분류가 되지 않은 프로젝트들은 각 카테고리의 특징성이 부족하고, 단일 카테고리에만 종속되지 않고, 여러 카테고리와 겹치는 내용들이 존재하는 카테고리일 것이다. 추후 카테고리별로 정확히 특징을 추출하여, 인식률을 높이는 분류 모델을 연구할 예정이다.

5. 결론 및 한계점

본 연구는 머신러닝 기반의 오픈소스 SW 카테고리 분류에 대한 연구로 이전 선행 연구인 오픈소스 SW 카테고리 분류 모델 연구의 실질적인 이슈 및 다양한 머신러닝 알고리즘을 통해 그 효과성을 비교하고 관련된 이슈를 탐색하고자하는 연구이다.

본 연구는 소스코드 이외에도 ReadMe, Description, Tag 등의 텍스트 데이터를 통해 분류기를 구축할 때 활용할 수 있다. 또한 다양한 분류기를 구성할 수 있으면 앙상블 학습같은 분류기의 결합을 예상할 수도 있다. 그러나 카테고리간 오픈소스 SW 프로젝트의 교집합이 다양하게 생기는 문제에 대해서는 아직 해결하지 못했다. 이와 같은 오픈소스 SW 카테고리 분류 모델 개발을 통해 선행 연구에서 제안한 오픈소스 SW의 발전에 맞게 유연한 카테고리 분류를 생성하는 연구로 발전할 수 있을 것이라고 기대된다.

Acknowledgement

본 연구는 문화체육관광부 및 한국저작권위원회의 2018년도 저작권기술개발사업의 연구결과로 수행되었음

참 고 문 헌

[1] Black Duck, “The tenth annual future of open source survey”, Black Duck Software https://www.blackducksoftware.com/2016-future-of-open-source, Oct. 2016.

[2] 정보통신산업진흥원(NIPA), "2016년 공개 SW 시장조사 보고서", 2017.

[3] 김예솔, “프로그램 간 효율적인 유사성 분석을 위한 특징정보 기반의 소프트웨어 분류 기법”, 단국대학교, 2015.

[4] 김종배, 조재홍, 김태열, "오픈소스 SW 비즈니스 모델", 한국통신학회지(정보와통신), 제35권, 제5호, pp.52-59, May, 2018.

[5] Melamud, Oren, Omer Levy, Ido Dagan, "A simple word embedding model for lexical substitution", Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, pp.1-7, 2015.

[6] Witten, I. H., Frank, E., Hall, M. A., Pal, C. J, "Data Mining: Practical machine learning tools and techniques", Morgan Kaufmann, 2016.

[7] S, Marcus, M Vosgerau, "ClassifyHub: An Algorithm to Classify GitHub Repositories", Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz). Springer, Cham, pp.373-379, 2017.

[8] S, Giovanni, J. F. Elder, "Ensemble methods in data mining: improving accuracy through combining predictions", Synthesis Lectures on Data Mining and Knowledge Discovery 2.1, pp.1-126, 2010.

[9] W. Gierke, B. Sebastian. “https://github.com/WGierke/git_better”, Mar, 2017.

[10] Kolter, J. Zico, Marcus A. Maloof, "Dynamic weighted majority: An ensemble method for drifting concepts" Journal of Machine Learning Research 8, pp.2755-2790, Dec, 2010.

[11] Chang, Yin-Wen, Chih-Jen Lin, "Feature ranking using linear SVM", Causation and Prediction Challenge, Dec, pp.53-64 2008.

[12] 백승찬, 윤호영, 조용준, 신동명, "TF-IDF 기법 기반의 오픈소스 SW 카테고리 분류 모델 연구", 한국소프트웨어감정평가학회 제28회 춘계학술대회발표대회 논문집, pp , May, pp. 11-13, 2018.

저 자 소 개

백승찬(Seung-Chan Back)

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

최현재(Hyunjae Choi)

2018년 성균관대학교
전자전기컴퓨터공학과 석사
2018년-현재 엘에스웨어(주) 선임
<주관심분야> 정보보호, 네트워크 보안

윤호영(Ho-Yeong Yun)

2012년 한성대학교 산업경영공학과 학사
2016년-2018년 엘에스웨어(주) 선임
2012년- 현 재 연세대학교
산업공학과 박사과정
<주관심분야> 최적화 이론, 알고리즘

조용준(Yong-Joon Joe)

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

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