Open Source Software Classification Model
Using K-Nearnest Neighbor with Text Elements

Seungchan Back*, Hyunjae Choi*, Yong-Joon Joe*, and †Dong-Myung Shin*
*LSWare Inc., Korea

(E-mail: morgan@lsware.co.kr, esther@lsware.co.kr, eugene@lsware.co.kr, roland@lsware.co.kr)

ABSTRACT A Recent market for open source software is enormous, the classification of open source software can be the basis of various open source software studies. However, due to characterizes of open source software, the classification study of open source software has not been done well. Therefore, this study is part of a flexible approach of open source categorization for emerging new one, we propose open source software classification applying machine learning with considering the characteristics of open source software.

Keywords: classification, open source software, categorization, machine learning, text mining

1. INTRODUCTION

Open source software(OSS) is software that releases the source code, anyone can view and use the code without restriction. At the same time, the classification of software is collecting software having similar functions and create one set, it is utilized as a basis of research such as software recommendations, software evaluation, software filtering, software relationship analysis. In this study, we propose to classify OSS projects for that although the recent OSS market has increased, not managed due to the characteristics of OSS[1]. Thus, it is possible to manage the OSS which has not been classified on existing systems and to achieve a flexible classification of OSS which is not limited to the existing categories when together with a study of the creation of categories of OSS. In chapter 2, we deal with the machine learning based open source classification method proposed in this paper, and in 3 chapter, we conclude with future work and conclusions.

2. Open Source Classification Using Text Elements

OSS has two types of characteristics as follows, since the management entity is not sure as compared with commercial software. One is that OSS profiles are not managed with the absence of profitable institutions. This makes OSS classification difficult, as classic software classification is based on project software profile. For example, classic software classification researches have used import and export statistics and profiles used for contracts, but in open source software, it is difficult to collect such data. The other is that policies are very free for each open source repository. Since it is very free, there are difficulties in selecting common features from OSS for classification. For Github and SourceForge as an example, in case of Github, they recommend Read.me file, SourceForge does not have such data. Also, SourceForge is recommended for Description, but not for Github. it has the ability to describe Profile for SourceForge only. In this way, it is difficult to specify classification feature points in different policies.

Therefore, the method proposed by this paper provides a method of selecting text elements such as ReadMe, Description that belonging to the project as feature points. As shown in Figure 1, First, It is extracted those text elements of an OSS project such as ReadMe, Description, Source Code, xml, and License from which OSS Repository classifying OSS projects. At this stage, all text elements may be selected as excessive feature points, so unnecessary data requires an erasing process. In the process of removing, because we extract all text elements, a considerable number of factors are considered, basically
reserved words that are related with programming languages, special symbols such as parentheses or question marks, language stopwords, words with too much or too little frequency.

The next step is to vectorize the words through the word embedding. There is a prediction-based word embedding which predict using before and after words, and word frequency-based word embedding. Representative methods of them are TF-IDF and Word2Vec. The vectorized factor through word embedding used with machine learning algorithms[2] like K-NN, Decision Tree, and construct a classifier to calculate how a new open source project belongs to a category.

3. Future Work and Conclusions

This study proposes an OSS classification technique based on the textual elements of an open source project using word embedding and machine learning techniques. However, in the proposed method of this research, the conventional classification scheme is specified, and based on this, the open source classification should be used as a basis for learning data. Recognizing these limitations, we are also conducting research on clustering for classification systems that have not been classified. In the future, it is expected that flexible classification systems for open source projects that have not been designated as a result of combining with this study.

ACKNOWLEDGMENT

This research project was supported by the Ministry of Culture, Sports, and Tourism (MCSTT) and by the Korea Copyright Commission in 2018.

REFERENCES