Proposal

January 20, 2019

1 Overview

Software Defect Prediction(SDP) is popular research theme in software domain. The progresses that the application of deep learning in artificial intelligence have inspired many researchers who is expert in SDP. On the other hand, traditional metrics are not able to extract the semantics of program, which is important property of program. In order to handle the semantics of program, code naturaness hypothesis have been proposed recently[1]. Programming language corpora has to similar statistical properties to natural language corpora. Follow this hypothesis, many researchers have applied deep learning model to extract semantics informtion.By extracting ASTs of programs, sequences can be generated;then generate feature by using deep learning model to train embedded words[9].The model can classify the bugs more effective than traditional based metris.Later, Convolution Neural Network was applied to extract the text feature,which is also powerful in SDP[6][5]. Beside cross entropy was employed as a form of metric in SDP[10].

2 Problem & Improvement

However, the previous research did not pre-train the words[9][6].The easiest way to associate a word with a dense vector is to randomly select the vector. The problem with this approach is that the resulting embedded space has no structure[2].It is difficult for deep neural networks to learn about this messy, unstructured embedded space[2].

Our model will utilze Word2vec or BERT technique to pre-train words[8][7][3]. The corpus we selected from open source program ”Big Data”[1]. BERT is the-state-of-art technology to pre-train the word. The rest is our methodology to implement the task.

3 Methodology

Our core technoloy will be base on deep learning. The follow are the step that we implement our model.

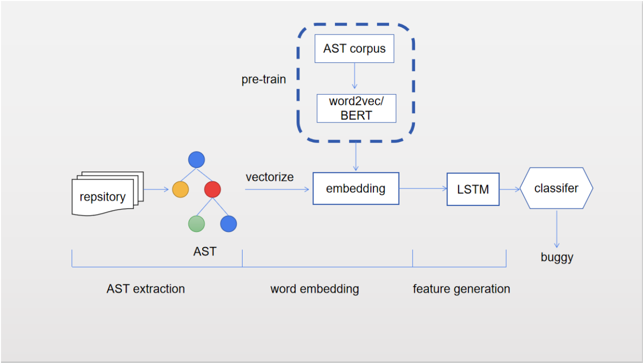

Figure 1:

1. Extract AST from data set. in this step ,some kind of nodes will be filtered out.
2. Tokenize the sequences then vectorize them by using one-hot-encode.
3. After getting vectorized words, we then pre-train the corpus and add its weight space into word embedding layers.
4. Using deep learning model such Long Short-Term Memory model[4] to generate the feature.
5. Classifer will be employed to classify the instances.

Figure 1 is the framwork the our model.

4 Experimental Design

Based on emprical study, Logistics Regression perform well in classification. we will select it as classifier to classify the both deep learning based feature extraction model and traditional feature based model. The comparison group is :

1. **Our model+Logistics Regression**
2. **Traditional metrics+Logistics Regression**

In order to know how well that our model by using the pre-train section. we also compare the model that without pre-training we select Covolution Neural Network(CNN) to extract the feature.The comparison group is previous work[6] and our model. It can be name:

1. **Our Model**
2. **Without Pre-training**

In order to understand how well that Word2vec and BERT that can extract semantics. we also make comparison between count based model such as TF-IDF and Bag of Word(BOW); Then the comparison group is :

1. **Our Model+Logistics Regression**
2. **TF-IDF+Logistics Regression**
3. **BOW+Logistics Regression**

References

[1] Miltiadis Allamanis, Earl T Barr, Premkumar Devanbu, and Charles Sutton. A survey of machine learning for big code and naturalness. *ACM Computing Surveys (CSUR)*, 51(4):81, 2018.

[2] Francois Chollet. *Deep learning with python.* Manning Publications Co., 2017.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[6] Jian Li, Pinjia He, Jieming Zhu, and Michael R Lyu. Software defect prediction via convolutional neural network. In *Software Quality, Reliability and Security (QRS), 2017 IEEE International Conference on*, pages 318–328. IEEE, 2017.

[7] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[8] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[9] Song Wang, Taiyue Liu, and Lin Tan. Automatically learning semantic features for defect prediction. In *Software Engineering (ICSE), 2016 IEEE/ACM 38th International Conference on*, pages 297–308. IEEE, 2016.

[10] Xian Zhang, Kerong Ben, and Jie Zeng. Cross-entropy: A new metric for software defect prediction. In *2018 IEEE International Conference on Software Quality, Reliability and Security (QRS)*, pages 111–122. IEEE, 2018.