Long Method and God Class smell detection using features from Code2Seq model

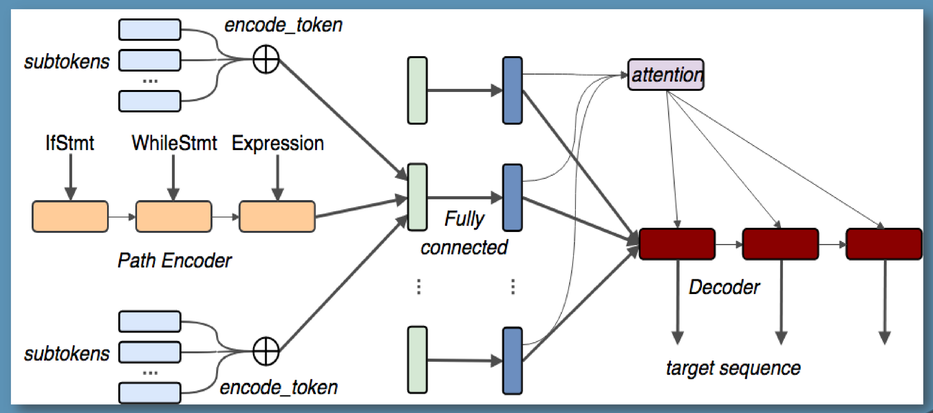
Mitar Perović SW71/17

Problem & Method

Detect smell in code using 320-dimensional feature vectors extracted from the code2seq model. Forward the vectors as input to ML classification algorithms: SVM, Random Forests and Logistic Regression. The output of a classifier is binary classification - 1 for code smell, and 0 for non code smell.

Code2Seq model

The model inputs a Java method and then parses it as an Abstract Syntax Tree tokens as an input for the encoder. The decoder then produces a feature vector of an input.



Dataset & Preprocessing

Dataset used is MLCQ, containing god class and long method code smelly examples, as well as clean ones. Data is divided 80% for training and 20% for testing. The training dataset is balanced, using an over-sampling technique, SMOTE, before fitting the classifiers.

Long Method results

SVM	Precision	Recall	F1 score
code smell	0.52	0.42	0.46
non-smell	0.92	0.95	0.94
R. Forests			
code smell	0.56	0.36	0.44
non-smell	0.92	0.96	0.94
Logistic R.			
code smell	0.44	0.68	0.54
non-smell	0.95	0.88	0.92

God Class results

SVM	Precision	Recall	F1 score
code smell	0.32	0.28	0.30
non-smell	0.91	0.92	0.91
R. Forests			
code smell	0.48	0.32	0.38
non-smell	0.91	0.95	0.93
Logistic R.			
code smell	0.2	0.7	0.31
non-smell	0.94	0.62	0.74

Improvements

God class vector has been calculated as a mean of all method vectors. There could be used other approaches. We could provide additional input to ML classifiers, aside of the feature vector. For example number of lines of code for a long method. Another approach would be feeding the vector to neural network with some rule based features, that we manually extract.