Decluttering Challenge Report

Current Topics in Software Engineering: Automating Software Engineering

Lukas Pagitz, Bernhard Nitsch, Boda Wen

Universität Klagenfurt
Department of Informatics Systems

1 Introduction

The Decluttering Challenge (DeClutter) is an international challenge with the goal to develop an automated tool to "identify unnecessary software documentation at the class or file level". [1]

Comments are either considered as informative or non-informative. A non-informative comment is defined as: "non-information is a comment that is completely uninformative and hence useless/should be removed (in the perspective of documentation decluttering)". The exact descriptions can be found on the DeClutter GitHub page.

For the challenge, a training data set was provided to allow the tool to learn which comments are informative and which not. The file "declutter-gold_DevelopmentSet.csv" included 1050 rows and was later on replaced by by file "train_set_0520.csv" with 1311 rows. These data sets contain links to code lines and their respective comments and the information if the comment is considered as informative or not (by the authors of the challenge).

The team working on this tool implementation decided to use Python as programming language as it provides an easy way to iterate through .csv files and is fast at analysing code using existing libraries.

2 Development environment

Version 3.8 of python was used for development. Important! The 64-bit version is required. The tool does not work with the 32-bit version.

Please follow the following steps to setup the environment for the tool.

1. Install 64-bit version of Python [3]
2. Install pandas for reading csv files
   - pip install pandas
3. Install spaCy (NLP tool) and its EN language model
   - pip install -U spaCy
   - python -m spacy download en_core_web_sm
4. Install javalang to allow Python to understand Java language syntax
   - pip install javalang

3 Project Structure

3.1 download_code

3.2 preprocess_data

3.3 extract_comment_and_code

3.4 train

4 Approach and experiments

4.1 Download of Java files

The very first step we took was to download the files used for the challenge so that they could be analyzed later on.

4.2 Get Code

4.3 Text To Number

4.4 Tokenize

4.5 Remove words from vocabulary set

4.6 Training data

5 Techniques and resources used

6 Result analysis

References

1. DeClutter on GitHub, https://github.com/dysdoc/declutter. Last accessed 30 May 2020
2. DeClutter on Kaggle, https://www.kaggle.com/c/declutter20v2/overview/. Last accessed 30 May 2020
3. Python, https://www.python.org/downloads/. Last accessed 30 May 2020