User Manual for the Bug Report Classification Tool

Elisa Ciocarlan
School of Computer Science

UNIVERSITY OF BIRMINGHAM

23rd March, 2025

1 Introduction

The proposed Bug Report Classification Tool is a Python-based program designed to automate the classification of bug reports. The classifier uses a Support Vector Machine (SVM) in combination with Term Frequency-Inverse Document Frequency (TF-IDF).

It operates in two phases:

1. **Training phase :** Train the classification model using a dataset.
2. **Classification phase :** Classify a given text file (**.txt**) as either "performance-related bug" or "not performance-related bug".

2 Requirements

To run the project's code, Python 3.6+ is required along with the libraries listed in the **requirements.txt** file.

3 Tool Overview

The tool consists of two Python files located in the **src** directory:

1. **train_model.py :** This script is used to train the model on a dataset.
2. **bug_report_classifier.py :** This script takes a text file as an argument and classifies it as "performance-related bug" or "not performance-related bug".

4 Installation and Setup

4.1 Download the tool

Clone or download the repository from GitHub (https://github.com/elisacio/ISE_project.git).

4.2 Install dependencies

The dependencies are provided in the file **requirements.pdf**. They can be installed by using the following command:

```
$ pip install -r requirements.txt
```

5 Training the classification model

5.1 Prepare the dataset (optional)

In this tool, we already provide datasets of bug reports from five different projects (pytorch, tensorflow, keras, incubator-mxnet and caffe). However, the model can be trained on other datasets. This only requires adding or removing datasets from the **datasets** directory.

A dataset should be in CSV format and should contain at least the three following columns:
- 'Title': contains the title of the bug report.
- 'Body': contains the detailed description of the issue.
- 'class': contains the label. If the bug is performance-related, the label is 1; otherwise, it is 0.

5.2 Run the training script

The training script can be run using the following command:

```
$ python src/train_model.py
```

The datasets on which the model is trained are automatically retrieved from the **datasets** directory and the classifier model (**classifier.plk**) and vectorizer (**vectorizer.plk**) files are automatically saved in the **models** directory.

6 Classifying Bug Reports

6.1 Input text file

Create a text file containing the title and the body of the bug report you want to classify.
Examples of bug report text files are provided in the **examples** directory:
- Eclipse_bug_report.txt: is an example of a non-performance-related bug report.
- Firefox_bug_report.txt: is an example of a performance-related bug report.

6.2 Run the classification script

The classification script can be run using the following command:

```
$ python src/bug_report_classifier.py path/to/your_file.txt
```

The input text file to classify must be given as an argument in the command line.

6.3 Output

The script will print the classification result in the terminal. If the given bug report is performance-related, the classification result will be *positive*, otherwise it will be *negative*.

6.4 Example

In this section, we provide an example of how to classify a bug report. To classify the Firefox_bug_report.txt file, enter the following command:

```
$ python src/bug_report_classifier.py examples\Firefox_bug_report.txt
```

The classification result should then appear in the terminal. For the command above, the following output will be displayed:

```
-------------------------------------------------------------------

       -----    CLASSIFICATION RESULT : POSITIVE    -----

The provided bug report is classified as performance bug-related.


-------------------------------------------------------------------
```