DUPLICATE BUG REPORT DETECTION USING PRE-TRAINED LANGUAGE MODELS

K.A.Udeshika Sewwandi

(199363T)

Degree of Master of Science in Computer Science

Department of Computer Science and Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

July 2022

DUPLICATE BUG REPORT DETECTION USING PRE-TRAINED LANGUAGE MODELS

Kahandawala Arachchige Udeshika Sewwandi

(199363T)

Thesis/Dissertation was submitted in partial fulfillment of the requirements for the degree MSc in Computer Science specializing in Data Science

Department of Computer Science and Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

July 2022

DECLARATION

I declare that this is my own work and this thesis/dissertation does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other University or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:................................ Date:……………………

The above candidate has carried out research for the PhD/MPhil/Masters thesis/dissertation under my supervision. I confirm that the declaration made above by the student is true and correct.

Name of the Supervisor: Dr. Surangika Ranathunga

Signature of the supervisor: .... Date:……………………

ACKNOWLEDGEMENTS

I would like to show my gratitude to Dr. Surangika Ranathunga for guiding me to initiate and find a better methodology and try out different novel approaches to conduct the research. Her supervision greatly helped me in setting goals and engaging in the research.

I would like to express our greatest gratitude to the Department of Computer Science and Engineering, the University of Moratuwa for providing the support to overcome this effort.

Last but not least, my heartfelt gratitude goes to my parents, husband, and friends who supported me throughout this effort.

ABSTRACT

Software testing and defect reporting are significant factors of software development and maintenance. Defects are identified and reported in a bug tracking system like JIRA, or Bugzilla. Those reported defects are further triaged by an expert who has an understanding of the repository, system, and developers and assigns them to the developers to fix them. During this defect reporting there can be duplicate bugs reported and identifying duplicate bugs is a crucial task. Manual labeling of duplicate defects is time-consuming, may identify defects as duplicate bug reports, and also increases the cost of software maintenance. Therefore automated duplicate bug report detection is very significant. This research proposes a duplicate bug report classification methodology that leverages the Pre-trained language models BERT and XLNet with Multi-Layer Perceptron as the Deep Learning classifier for duplicate bug detection. We tested on publicly available datasets related to Eclipse, NetBeans, and OpenOffice bug reporting datasets. The selected models were shown to outperform the previously proposed systems for the same task. Among them, the approach used with BERT embeddings has shown the best results. Further experiments showed that BERT is capable of domain adaptation –meaning that even when the BERT was fine-tuned with different bug report datasets, it is still capable of detecting duplicate bugs in an unseen dataset. Finally, a multi-stage classification was done using a Convolutional Neural Network model and a BERT model using Eclipse and NetBeans datasets and a combined dataset of Eclipse and NetBeans. The approach used with the combined dataset has outperformed the baseline approach.

Keywords: Duplicate Bug detection, BERT, XLNet, MLP, CNN, Domain Adaptation, Multi-Stage Classification

Table of Contents

DECLARATION i

ACKNOWLEDGEMENTS ii

ABSTRACT iii

TABLE OF CONTENTS iv

LIST OF FIGURES vi

LIST OF TABLES vii

1. CHAPTER 1: INTRODUCTION 1

1.1 Background 1

1.2 Research Statement 2

1.3 Motivation 2

1.4 Objectives 3

1.5 Thesis Structure 3

2. CHAPTER 2: LITERATURE REVIEW 5

2.1 Overview 5

2.2 Text Classification 5

2.2.1 Text Classification using traditional machine learning techniques 6

2.2.2 Text Classification using Deep Learning techniques 6

2.3 Datasets for Duplicate Bug Detection 7

2.4 Information Retrieval for Duplicate Bug Report Detection 7

2.5 Pre-Trained Language Models 8

2.5.1 BERT 8

2.5.2 XLNet 10

2.5.3 Comparison between BERT and XLNet 11

2.5.4 BERT and XLNet based contextual information extraction 11

2.6 Domain Adaptation 12

2.7 Datasets 12

2.8 Evaluation 13

2.8.1 Accuracy 13

2.8.2 Precision 13

2.8.3 Recall (Sensitivity) 13

2.8.4 F1 score 14

3. CHAPTER 3: METHODOLOGY 15

3.1 Text Classification with BERT and XLNet 16
3.1.1 Feature Selection 18
3.1.2 Feature Extraction 18
3.1.3 Preprocessing 19
3.1.4 Classifier Layer 19
3.1.4.1 MLP Model 19
3.1.4.2 CNN Model 20
4. CHAPTER 4: Evaluation 22
4.1 Introduction 22
4.2 Experiment Setup 22
4.2.1 Bug Report Dataset 22
4.3 Baseline 23
4.4 Evaluation 23
4.5 Hyper Parameters 23
4.6 Text Classification with BERT and XLNet 24
4.7 Domain Adaptation Results 25
5. CHAPTER 5: Conclusion 28
6. REFERENCES 29

List of figures

Figure	Description	Page
Figure 1.1	Sample bug report document	1
Figure 2.1	Duplicate bug report pair	5
Figure 2.2	BERT text processing	9
Figure 2.3	Auto Regression Language Model	10
Figure 2.4	Domain Adaptation	12
Figure 3.1	Flow Chart for the entire system	16
Figure 3.2	Proposed Approach for BERT or XLNet with MLP classifier	17
Figure 3.3	Proposed Approach for BERT or XLNet with CNN classifier	17
Figure 3.4	Bug report pairing	18
Figure 3.6	Model used in the research	20
Figure 3.7	Model with CNN	20
Figure 3.8	CNN Model architecture	21
Figure 4.1	Accuracy obtained for different approaches	27
Figure 4.2	Accuracy obtained for NetBeans and Eclipse datasets on different approaches	27

List of tables

Table Description Page

Table 2.1 Data distribution of msr14 dataset 13

Table 4.1 Classification results of our approach and baseline 24

Table 4.2 Results of Classification with CNN approach 25

Table 4.3 Domain adaptation results of our approach 25

Table 4.4 Multi Stage Classification results on combined model, Eclipse and NetBeans approach 26