A DEEP LEARNING FRAMEWORK FOR THE DETECTION OF SOURCE CODE PLAGIARISM USING SIAMESE NETWORK AND EMBEDDING MODELS

BY

MOHAMMED S. M. MANAHI

A thesis submitted in fulfilment of the requirement for the degree of Master of Computing (Computer Science and Information Technology)

Kulliyyah of Information and Communication Technology
International Islamic University Malaysia

NOVEMBER 2021

Abstract

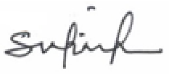
Source code plagiarism represents an ongoing problem that threatens academic integrity and intellectual rights. Various research works on detection approaches have been proposed to overcome prolonged manual inspection as it requires laborious efforts and consumes time. These detection approaches can be categorised into four major domains; software engineering, knowledge discovery, shallow parsing and machine learning. Review of the literature revealed that most of the detection approaches had been evaluated based on the commonly referenced and established six-level classification of source code transformations known as the Faidhi and Robinson spectrum, except for the approaches in the machine learning domain. Thus, this research sought to fill the gap in the absence of a machine learning approach that uses embedding models to detect source code plagiarism and evaluated based on the six-level classification. The objectives of this research are threefold; to extract various embedding sequences as similarity features from source codes using embedding models, to train a Siamese network that learns similarity representations from source code embedding sequences, and to develop a deep learning framework that leverages embedding sequences and Siamese network to identify the most accurate detection based on the standard six-level classification of plagiarism activities defined by Faidhi and Robinson. A deep learning framework that utilised a Siamese network and embedding models is proposed to detect deliberate plagiarism in source codes. The proposed framework split source codes into character-based, word-based and token-based sequences to obtain embedding sequences through Word2Vec and fastText models. These embedding sequences were then used as inputs to the Siamese BLSTM network for learning similarity representations. The experimental results showed that the character-based embedding sequences with Word2Vec, Skip Gram and Negative Sampling (W2V-SGNS) approach and the token-based embedding sequences with FastText, Skip Gram and Hierarchical Softmax (FT-SGHS) approach outperformed the other approaches. The detection results were also found to be able to detect up to level five (i.e., semantic equivalents) of the standard classification. However, future experiments will require a larger dataset and fine-tuning of the Siamese network to reduce overfitting and to improve the generalisation of the trained models on plagiarism attacks.

خلاصة البحث

تشكل السرقة الأدبية في الشيفرات المصدرية تهديداً صارخاً للنزاهة الأكاديمية والحقوق الفكرية. أنشأ باحثون أوائل تصنيفاً من ستةِ مستوياتٍ لأنشطة السرقة الأدبية المتعمّدة في الشيفرات المصدرية والذي أصبح لاحقاً معياراً موحداً لقياس مستوى السرقة الأدبية في الشيفرات المصدرية. اقترحت العديد من الأبحاث السابقة طرقاً للكشف عن السرقة الأدبية في الشيفرات المصدرية للتغلب على التحقق اليدوي الذي يتطلب جهوداً مضنية و وقتاً طويلاً. تنقسم هذه الأبحاث السابقة إلى طرق تعتمد على أربعة مجالاتٍ رئيسة والتي هي هندسة البرمجيات، أساليب المعرفة الاستكشافية، تقنيات معالجة اللغات الطبيعية الضحلة و تعلم الآلة. كشفت الأدبيات عن تقييم غالبية المناهج بناءً على التصنيف الموحد باستثناء مناهج مجال تعلم الآلة. سعى هذا البحث إلى سد الفجوة البحثية المتمثلة في عدم وجود طريقة تعتمد على تعلم الآلة لاكتشاف السرقة الأدبية في الشيفرات المصدرية والتي تقيمّ نتائجها بناءً على التصنيف الموحد. يهدف هذا البحث الى الحصول على تضمينات للشيفرات المصدرية لاستخدامها كميزات تشابه و من ثم تدريب شبكة عصبونية سيامية لتعلم تمثيلات التشابه للشيفرات المصدرية واخيرا لبناء إطار عمل يدمج التضمينات مع الشبكة العصبونية السيامية للتحقق من النتائج بناء على التصنيف الموحد للسرقة الأدبية في الشيفرات المصدرية. اقترح هذا البحث إطار عمل مبني على تقنيات التعلم العميق باستخدم شبكة عصبونية سياميّة و نماذج تضمين اللغة في فضاء المتجهات لاكتشاف أنشطة السرقة الأدبية المتعمّدة في الشيفرات المصدرية. تحصّل إطار العمل المقترح على تسلسلات تضمين لعدةّ أشكال تجزئة للشيفرات المصدرية المتمثلة في التجزئة المبنية على الأحرف، التجزئة المبنية على الكلمات و التجزئة المبنية على الرموز المميّزة باستخدام نموذجيّ التضمين Word2Vec و fastText. بعد ذلك، استخدمت تسلسلات التضمين كمدخلات للشبكة العصبونية السيامية BLSTM لتعلم تمثيلات التشابه بين الشيفرات المصدرية. أشارت النتائج التجريبية إلى تفوق التجربة المبنية على الأحرف المستندة لمعماريّة W2V- SGNS والتجربة المبنية على الرموز المميّزة المستندة لمعماريّة FT-SGHS على باقي تجارب إطار العمل. لاحقاً، تم تقييم هاتين التجربتين بناءً على المعيار الموحد حيث أظهرت نتائج التقييم اكتشاف أنشطة السرقة الأدبية المتعمدة حتى المستوى الخامس من التصنيف. يوصي البحث بإجراء المزيد من التجارب البحثية المستقبلية لصقل و ضبط الشبكة السيامية بهدف تحسين و تعميم اكتشاف الأنشطة المختلفة في السرقة الأدبية المتعمدة للشيفرات المصدرية.

APPROVAL PAGE

I certify that I have supervised and read this study and that in my opinion, it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a thesis for the degree of Master of Computing (Computer Science and Information Technology)



…………………………………..
Suriani Sulaiman
Supervisor

…………………………………..
Normi Sham Awang Abu Bakar
Co-Supervisor

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a thesis for the degree of Master of Computing (Computer Science and Information Technology)

…………………………………..
Sharyar Wani
Internal Examiner

…………………………………..
Rayner Alfred
External Examiner

This thesis was submitted to the Department of Computer Science and is accepted as a fulfilment of the requirement for the degree of Master of Computing (Computer Science and Information Technology)

…………………………………..
Amir 'Aatieff Bin Amir Hussin
Head, Department of Computer Science

This thesis was submitted to the Kulliyyah of Information and Communication Technology and is accepted as a fulfilment of the requirement for the degree of Master of Computing (Computer Science and Information Technology)

…………………………………..
Abd. Rahman Bin Ahlan
Dean, Kulliyyah of Information and Communication Technology

DECLARATION

I hereby declare that this dissertation is the result of my own investigations, except where otherwise stated. I also declare that it has not been previously or concurrently submitted as a whole for any other degrees at IIUM or other institutions.

Mohammed S. M. Manahi

Signature .... Date 1/11/2021 ....

INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

DECLARATION OF COPYRIGHT AND AFFIRMATION OF FAIR USE OF UNPUBLISHED RESEARCH

THE IMPACT OF MOBILE INTERFACE DESIGN ON INFORMATION QUALITY OF M-GOVERNMENT SITES

I declare that the copyright holders of this dissertation are jointly owned by the student and IIUM.

No part of this unpublished research may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without prior written permission of the copyright holder except as provided below

1. Any material contained in or derived from this unpublished research may be used by others in their writing with due acknowledgement.

2. IIUM or its library will have the right to make and transmit copies (print or electronic) for institutional and academic purposes.

3. The IIUM library will have the right to make, store in a retrieved system and supply copies of this unpublished research if requested by other universities and research libraries.

By signing this form, I acknowledged that I have read and understand the IIUM Intellectual Property Right and Commercialization policy.

Affirmed by Mohammed S. M. Manahi

1/11/2021

……..…………………….. ………………………..

Signature Date

ACKNOWLEDGEMENTS

First and foremost, All praises to Allah Subhanahu Wa Ta'laa, the Most Gracious, and the Most Merciful, Who granted me the knowledge and bestowed His everlasting mercies and bounties upon me during this journey. My humblest gratefulness and appreciation to Him; without Him, I would not have been able to do this research. Alhamdulillah!

I am extremely grateful to my main supervisor Assistant Professor Dr. Suriani Sulaiman and my co-supervisor Associate Professor Dr. Normi Sham Awang Abu Bakar for their extraordinary guidance, invaluable advice, continuous support, and patience throughout my research journey. Their immense knowledge and plentiful experience have encouraged me in all the time of my academic life.

I am sincerely thankful to my family for instilling encouragement, support, inspiration and heartfelt love, especially my beloved Mother. She always had faith that I would indeed finish, even as life events conspired to get in the way! My words could not express how grateful I am to my Mother, Brother and all my family members! My deepest appreciation to you all.

I would also like to thank the Ministry of Higher Education, Malaysia and my research grant members for funding my research which has immensely helped me to pursue my study.

In addition, many thanks to my colleagues, lecturers, staffs and fellows at the research unit of computational intelligence.

Last but not least, I wish to express my appreciation to my friend Samir for his insightful advice and technical help all the time.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ACM	Association of Computing Machinery
ANN	Artificial Neural Network
AST	Abstract Syntax Tree
AWS	Amazon Web Services
BGRU	Bidirectional Gated Recurrent Unit
BLSTM	Bidirectional Long Short-Term Memory
BOW	Bag-Of-Words
BPTT	Backpropagation Through Time
BRNN	Bidirectional Recurrent Neural Network
CBOW	Continuous Bag-Of-Words
CFG	Control-Flow Graph
CNN	Convolutional Neural Network
CSV	Comma-Separated Values
cuDNN	CUDA Deep Neural Network
CV	Computer Vision
FFNN	FeedForward Neural Network
FT-SGHS	fastText using Skip-gram and Hierarchical Softmax
FT-SGNS	fastText using Skip-gram and Negative Sampling
FP	False Positive
FN	False Negative
GRU	Gated Recurrent Unit
GST	Greedy String Tiling
GEC	Graph Edit Distance
GLoVe	Global Vectors for word representations
GPU	Graphical Processing Unit
HPC	High-Performance Computing
IR	Information Retrieval
IIUM	International Islamic University of Malaysia
KICT	Kulliyyah of Information and Communication Technology
K-NN	K-Nearest Neighbour

LSA	Latent Semantic Analysis
LSTM	Long Short-Term Memory
MaLSTM	Siamese LSTM with Manhattan distance
MOSS	Measure of Software Similarity
Nadam	Adam optimiser with Nesterov Momentum
NAG	Nesterov Accelerated Gradient
NCE	Noise Contrastive Estimation
NLP	Natural Language Processing
NMT	Neural Machine Translation
OOP	Object-Oriented Programming
OOV	Out-Of-Vocabulary
OS	Operating System
PCA	Principal Component Analysis
PDG	Program Dependency Graph
PLP	Programming Language Processing
PS	Problem-Set
RBM	Restricted Boltzmann Machine
RBNN	Radial Basis function Neural Network
ReLU	Rectified Linear Unit
RKR-GST	Running Karp Rabin and Greedy String Tiling
RNN	Recurrent Neural Network
SIM	Software Similarity Tester
SLR	Systematic Literature Review
SOM	Self-Organising Map
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TBCNN	Tree-Based Convolutional Neural Network
TF-IDF	Term-Frequency Inverse Document Frequency
TN	True Negative
TP	True Positive
TR	Text Retrieval
WASTK	Weighted Abstract Syntax Tree Kernel
W2V-SGHS	Word2Vec using Skip-gram and Hierarchical Softmax

W2V-SGNS	Word2Vec using Skip-gram and Negative Sampling
YAP	Yet Another Plague

CHAPTER ONE

INTRODUCTION

1.1 INTRODUCTION

This chapter introduces fundamental research statements that constitute a roadmap to the research thesis. These fundamental research statements cover the problem statement, the objectives of the research, the questions of the research, the purpose of the research and the significance of the research. In addition, this chapter addresses more statements such as essential research definitions to provide a holistic view of the later chapters.

1.2 BACKGROUND

The era of technological advancements inevitably witnesses massive and prevalent information exchange to facilitate sustainable human life (Okul, Aksu, & Aydin, 2019; Maltby, 2011). Nevertheless, the easily obtainable information everywhere had created an environment in which works or ideas of others can easily be copied and violated. These violations commonly refer to plagiarism which threatens intellectual rights and academic integrity (Agrawal & Sharma, 2017; Hourrane & Benlahmar, 2017; Joy, Cosma, Yau, & Sinclair, 2011). The word plagiarism is a derivative from the Latin word *plagiarius*, which means *literary theft* (Agrawal & Sharma, 2017). The plagiarist, who commit plagiarism, illegitimately claims their ownership of the plagiarised content and denies acknowledging the original owner (Sulistiani & Karnalim, 2018; Zhao, Xia, Fu, & Cui, 2015; Durić & Gašević, 2013). Therefore, plagiarism represents a challenging phenomenon to preserve authenticity.

Students of computing disciplines study programming courses to construct analytical and logical skills. A significant part of their assessment is based on individual assignments in which various problem-sets have to be solved to ensure that each student has a proper comprehension of the programming logic (Mišić, Protić, & Tomašević, 2017). Some students fail to fulfil the required assignments, so they turn to cheat to submit solutions (Huang, Song, & Fang, 2020). This form of cheating in the context of programming assignments is known as source code plagiarism. Source code plagiarism refers to partial or complete reuse of someone else's source codes without acknowledging the source code's original owner (Karnalim, Budi, Toba, & Joy, 2019; Karnalim, 2017; Cosma & Joy, 2008). However, Mišić, Protić and Tomašević (2017) as well as Mirza and Joy (2017) reported that source code plagiarism could either be accidental or deliberate. Accidental plagiarism activities are caused by coincidental reuse or the grey area of understanding plagiarism. On the contrary, intentional plagiarism represents actions with intended premeditation (Mišić, Protić, & Tomašević, 2017; Wilkinson, 2009).

The deliberate plagiarism in source codes severely violates the principles of the Association of Computing Machinery (ACM). The ACM concerns ethical, reliable and safe computing practices. Their principles urge the students and the professionals to acknowledge others' contributions, respect others' privacy, maintain copyrights and avoid source code violations (The Association of Computing Machinery, 2018). Therefore, source code plagiarism represents explicit offences to the ACM principles.

In response to the source code plagiarism threats, various research works proposed detection approaches for source code plagiarism (Agrawal & Sharma, 2017; Chong, 2013). These detection approaches scrutinise students' submissions automatically as compared to the manual inspection that is a laborious and time-consuming process.

Although the detection approaches are feasible through various techniques such as matching algorithms and data mining, recent machine learning achievements have motivated researchers to designate the use of machine learning algorithms in software engineering (J. Zhang et al., 2019). State-of-the-art machine learning practices for source code analysis are known as Programming Language Processing (PLP) (Mou, Li, Zhang, Wang, & Jin, 2016). PLP utilises a subdiscipline of machine learning known as deep learning to perform several source code analysis tasks such as similarity classification (Tufano et al., 2019).

Deep learning leverages deep neural networks that consist of interconnected artificial neurons in stacked hidden layers that learn to transform data points into feature representations (Trask, 2019; Goodfellow, Bengio, & Courville, 2017). Deep learning has been achieving breakthroughs in many research fields such as Computer Vision (CV) and Natural Language Processing (NLP) (Miotto et al., 2017; Hordri, Samar, Yuhaniz, & Shamsuddin, 2017). NLP is a computer science field of research that utilises linguistics, Information Retrieval (IR), feature engineering and machine learning to empower machines to interpret textual data (Deng & Liu, 2018; Goldberg, 2017).

Word embedding is one of the NLP applications that achieved revolutionary improvement to word analogy and similarity relationships in vector space (Trask, 2019). Word2Vec, Global Vectors for word representations (GloVe) and fastText are widely known word embedding models (Lane, Howard & Hapke, 2019). Word embedding models are linguistic computational models that represent words based on their correlations. Each vector is a representational form of a word in the n-dimensional space where close distance vectors indicate words of shared meanings (Allen & Hospedales, 2019; Le, 2016). Word embedding models can predict the similarity relationships of words by learning the correlations of contiguous words for a given context. However, these models alone are insufficient to find similarities in long-term dependencies (i.e. non-consecutive words).

Long Short-Term Memory (LSTM) is a deep neural network that utilises a memory state to capture the semantics of long-term dependencies (Karpathy, Johnson & Fei-Fei, 2015). LSTM network consists of three gates: input gate, forget gate, and output gate that determine the important information to retain (Hochreiter & Schmidhuber, 1997). The problem of similarity classification of documents combines the word embedding models and the LSTM network in which the embeddings constitute the input for the LSTM network to learn similarity features of documents. The pre-trained embeddings which represent similarity representations for a context of short-term dependencies are fed to the LSTM network to initialise the input for learning similarity representations for the long-term dependencies of documents.

Standard neural network architecture learns feature representations from one input point at the same time. However, learning similarity representations for plagiarism detection require learning from two input point simultaneously. Therefore, Mueller and Thyagarajan (2016) as well as Neculoiu, Versteegh, and Rotaru (2016) proposed Siamese LSTM networks as an architecture to learn sentence similarities. The Siamese network is a twin neural network that shares identical hyperparameters to learn similarity representations from two input points.

1.3 STATEMENT OF THE PROBLEM

Source code plagiarism is a severe ongoing problem in programming courses. Students of computing disciplines have been shown to have the tendency to commit such malpractice (Joy et al. 2012; Mišić, Protić & Tomašević, 2017; Pawelczak, 2018). Faidhi and Robinson (1987) conducted fundamental research on source code plagiarism in which they established a six-level spectrum for source code plagiarism attacks. The six-level spectrum has become a standard classification for source code plagiarism activities. It classifies plagiarism activities into two different categories. The first category involves the lower three levels, which indicate lexical modifications to the original source code such as routine transformations. These lexical modifications barely require programming skill (Agrawal, Jain, & Uttam, 2020; Muddu, Asadullah & Bhat, 2013; Bejarano, García, & Zurek, 2013). The second category involves the higher three levels, which indicate structural changes to the original source code. These structural modifications require prior knowledge and experience in programming (Maryono, Yuana, & Hatta, 2019; Durić & Gašević, 2013). The detection of source code plagiarism in the second category represents a challenging task due to the modifications of the source codes' structural characteristics (Bandara & Wijayarathna, 2011; Maryono et al., 2019).

Various research works have proposed different plagiarism detection approaches in source codes. These approaches leverage four major domains: software engineering, knowledge discovery, shallow parsing NLP and machine learning. The literature revealed that all domains were mostly evaluated based on Faidhi and Robinson's spectrum (1987) except for the machine learning domain. Hence, none of the existing machine learning approaches has been evaluated based on the spectrum. The machine learning approaches however achieved more accurate detection results in comparison with other domains as well as against well-known detection engines such as JPlag and MOSS engines (Heres 2017; Yasaswi Katta 2018; Yasaswi Katta, Purini, & Jawahar 2017). Therefore, the current research gap is the absence of a machine learning approach to detect source code plagiarism based on Faidhi and Robinson's spectrum (1987).

This research aims to bridge the gap by establishing a framework based on deep learning techniques to detect source code plagiarism based on Faidhi and Robinson's spectrum (1987). The research uses a Siamese neural network as the base model for the framework and explores applying different configurations of embedding models to various forms of source code sequences to obtain source code embedding sequences to be fed as inputs to the base model. Each input represents an experiment for the base model to conduct to determine which embedding sequence produces the most accurate detection results based on Faidhi and Robinson's spectrum (1987).

1.4 RESEARCH OBJECTIVES

This research proposes to train various forms of source code sequences using two embedding models and feed the pre-trained embedding sequences as inputs to a Siamese network to detect plagiarism. Thus, the objectives are threefold:

RO1 To extract various embedding sequences as similarity features from source codes using embedding models

RO2 To train a Siamese network that learns similarity representations from source code embedding sequences.

RO3 To develop a deep learning framework that leverages embedding sequences and Siamese network to identify the most accurate detection based on the standard six-level classification of plagiarism activities defined by Faidhi and Robinson.

1.5 RESEARCH QUESTIONS

The questions reflect the research objectives; thus, this research aims to answer the following questions: