AN INTEGRATED MODEL OF AUTOMATED ELEMENTARY PROGRAMMING FEEDBACK USING ASSISTED AND RECOMMENDATION APPROACH

SUHAILAN SAFEI

DOCTOR OF PHILOSOPHY

2017

Faculty of Information and Communication Technology

AN INTEGRATED MODEL OF AUTOMATED ELEMENTARY PROGRAMMING FEEDBACK USING ASSISTED AND RECOMMENDATION APPROACH

Suhailan Safei

Doctor of Philosophy

2017

AN INTEGRATED MODEL OF AUTOMATED ELEMENTARY PROGRAMMING FEEDBACK USING ASSISTED AND RECOMMENDATION APPROACH

SUHAILAN SAFEI

A thesis submitted
in fulfilment of the requirements for the degree of Doctor of Philosophy

Faculty of Information and Communication Technology

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2017

DECLARATION

I declare that this thesis entitled "An Integrated Model of Automated Elementary Programming Feedback Using Assisted and Recommendation Approach" is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : .....

Name : Suhailan Safei .....

Date : .....

APPROVAL

I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of Doctor of Philosophy.

Signature : .....

Supervisor Name : Associate Professor Dr. Abdul Samad Shibghatullah

Date : .....

DEDICATION

This thesis is lovingly dedicated to the memory of my late father, Dato' Hj. Safei bin Daud. He may not live long enough to see his son through his Ph.D. journey, but still, he planted the thirst for knowledge and profound impact on me. For that, I thank him dearly. This thesis is also dedicated to my mother, To' Puan Che A'eshah binti Sulaiman, who always pray for my success and give affection to me. Special thanks to all my brothers (Abang Haji, Abang Za, Abang Man, Adik Li), sisters (Kak Da, Kak Mah) and their caring families who continuously equipped me with their generosity, loving kindness and constant support.

I am particularly indebted to my wife, Zuliana binti Mat Ali, for all her love and encouragement. She has always been there with me from day one of my academic career. My gratitude to all my children; Shazwani, Shahiran, Shadi Asyraf and Shakir Ahmad who illuminate my life with lively atmosphere and joy.

My appreciation as well to all my former experts from Sekolah Menengah Agama Sheikh Abdul Malek (SHAMS), lecturers from Universiti Teknologi Malaysia (UTM) and Mr. Abu Bakar bin Hj. Hasan (my former employer) who have shared with me their respective knowledge and expertise which implicitly influenced me to be who I am today.

Without all of this supports, I doubt this research work could have been accomplished in time.

ABSTRACT

Many studies on automated programming assessment tools with automated feedbacks have been addressed to assist students rectifying their solution's difficulty. However, many students will depend on an expert's assistance (e.g. expert) to debug their programs towards meeting the question's requirements. While several studies have produced feedback on the specific need of programming's question using a static template analysis, there is still a lack of an automated programming feedback model that is dynamically enriched through a live assisted feedback from an expert. Thus, this research proposed an integrated programming feedback model of elementary programming question using assisted and recommended approach. The assisted feedback was done by an expert through a similar difficulty analysis of computer programs that were grouped together based on their difficulty features. The features were proposed to enable ranking among computer programs and they were proven to be strongly correlated with the manual ranking of an expert's rubric assessment ($r_s = 0.914$, $p < 2.2e\text{-}16$). Meanwhile, similar difficulty groups of the computer programs were generated using a K-Means clustering algorithm that was enhanced with ranking consideration. This enhancement was evaluated on three ordinal datasets on different application domain covering 67 Java programs, 92 students' marks on computer architecture subject and 456 EUFA's football club coefficient ranking list. The results showed that not only a rank cluster representation was achieved, but the purity value was also increased by 1%. As the computer programs were clustered based on ranking consideration, expert's feedback analysis can be effectively done from worst to least for the programs' difficulty. Hence, two kinds of assisted feedbacks were proposed; general and specific assisted feedback. These feedbacks were automatically indexed using general program features and specific statement pattern rule for automated retrieval on general and specific recommended feedbacks respectively. An experiment was executed real programming lab dataset that consists of 475 elementary programming answer submissions from 67 participants. Expert's assisted feedbacks were provided at the end of a program submission. It shows that the technique has successfully clustered 67 computer programs into 24 similar groups of programming logic's mistake. Based on the groups, general feedbacks were provided on 6 groups covering 33 programs. Then, by using the proposed indexing technique, the same feedback has efficiently recommended to other 148 programs that are having similar mistakes along the lab session. On the other hand, 7 specific feedbacks were provided on 7 computer statements' mistake and were recommended to other 64 programs who were having similar statement mistakes along the lab session. Thus, the proposed technique can effectively help the expert providing continuous and dynamic feedback in rectifying logic's requirement of a problem. Unfortunately, the model is not suitable for complex programming question where their solution logics can be diversified. However, future work on the automatic extraction of acceptable program answer as a solution template may solve such limitation.

ABSTRAK

Banyak kajian telah dibuat untuk membangunkan alatan sokongan bagi automasi maklumbalas penyelesaian bagi kesilapan aturcara yang dibuat oleh pelajar. Namun, pelajar masih memerlukan bantuan guru untuk mengenalpasti kesilapan berkaitan dengan keperluan spesifik sesuatu soalan. Walaupun terdapat banyak kajian dalam mengautomasi maklumbalas berkaitan keperluan sesuatu soalan menggunakan kaedah analisa skima jawapan, model yang mampu mengautomasi dan mengindek maklumbalas langsung daripada pakar masih belum diteroka sepenuhnya. Sehubungan itu, penyelidikan ini mengkaji model maklumbalas bagi soalan pengaturcaraan melalui pendekatan integrasi maklumbalas bantuan pakar dan maklumbalas saranan. Maklumbalas bantuan tersebut dibuat melalui analisa kesukaran membangunkan aturcara yang dikumpul mengikut ciri persamaan berdasarkan skema arahan komputer. Apabila ciri persamaan yang dicadangkan ini digunakan dalam menghasilkan penarafan pada 67 aturcara komputer, didapati ia mempunyai korelasi yang tinggi terhadap laporan penarafan yang dibuat menggunakan penilaian rubrik oleh pakar (r_s = 0.914, $p < 2.2e-16$). Sementara itu, pengumpulan aturcara berdasarkan ciri tersebut telah dibuat menggunakan teknik K-Means yang ditambahbaik dengan elemen penarafan. Tiga set data ordinal daripada domain aplikasi berlainan telah diuji menggunakan teknik tersebut meliputi 67 aturcara Java, 92 markah pelajar bagi subjek senibina komputer dan 456 senarai kedudukan kelab bolasepak EUFA. Eksperimen menunjukan bukan sahaja perwakilan kluster bertaraf dapat dicapai, malah analisis nilai purity juga menunjukkan peningkatan sebanyak 1%. Dengan adanya elemen penarafan dalam teknik pengklusteran, ia membolehkan analisa untuk sesuatu maklumbalas difokuskan daripada kumpulan aturcara yang paling banyak kepada yang paling kurang bermasalah. Maklumbalas bantuan pakar ini meliputi maklumbalas bantuan umum dan bantuan khusus. Ia diindek menggunakan jarak cirian aturcara dan penapisan pernyataan yang digunakan untuk menghasilkan automasi maklumbalas dikenali sebagai maklumbalas saranan umum dan khusus. Eksperimen telah dilaksanakan menggunakan data yang mengandungi 475 jawapan daripada 67 peserta dalam makmal pengaturcaraan. Maklumbalas pakar telah disediakan di akhir penghantaran jawapan aturcara. Teknik ini telah berjaya mengumpulkan 67 aturcara kepada 24 kumpulan yang mempunyai kesilapan logik. Maklumbalas umum telah diberikan hanya kepada 6 kumpulan meliputi 33 aturcara. Ia telah disarankan kepada 148 aturcara lain yang mempunyai kesilapan yang sama sepanjang pelaksanaan sesi makmal tersebut. Selain itu, 7 maklumbalas khusus diberikan pada 7 kesilapan pernyataan komputer dan berjaya disarankan kepada 64 aturcara lain. Oleh itu, teknik ini berkesan membantu pakar menyediakan maklumbalas berterusan dan dinamik dalam mengenalpasti keperluan logik bagi satu permasalahan. Walaubagaimanapun, model ini tidak sesuai untuk digunakan pada soalan yang melibatkan logik yang kompleks. Namun, kekangan ini dapat diatasi melalui kajian mendatang dalam pengekstrakan automatik aturcara jawapan yang betul sebagai templet penyelesaian.

ACKNOWLEDGMENTS

"He grants wisdom to whom He wills, and he whom wisdom is granted receives indeed a benefit of overflowing; but none will grasp the message except those of understanding"

Quran al-Baqarah (2):269

"Whoever removes a worldly grief from a believer, Allah will remove from him one of the griefs of the Day of Resurrection. And whoever alleviates the need of a needy person, Allah will alleviate his needs in this world and the Hereafter. Whoever shields (or hides the misdeeds of) a Muslim, Allah will shield him in this world and the Hereafter. And Allah will aid His slave so long as he aids his brother. And whoever follows a path to seek knowledge therein, Allah will make easy for him a path to Paradise."

Prophet Muhammad (P.B.U.H.)

Undertaking this Ph.D. research has been a truly life-changing experience for me and it wouldn't have been possible without the support and guidance that I received from many people. Foremost, I am indebted to my supervisor, Associate Professor Dr. Abdul Samad bin Shibghatullah from the Faculty of Information Communication and Technology, Universiti Teknikal Malaysia Melaka (UTeM) for his continuous support throughout my Ph.D. journey with inspiration and aspiration through the years of my research as the road ahead. I would like to express my sincere gratitude as well to Associate Professor Dr. Burhanudin bin Mohd Aboobaider for his guidance and assistance all over the way to the completion of this study.

A big thank you to all my fellow academics from Universiti Sultan Zainal Abidin (UniSZA) for sharing their expertise and experience, especially to Assoc. Prof. Dr. Mokhairi Makhtar, Nazirah Abd Hamid and Prof. Dr. Mohd Nordin Abdul Rahman. I would like to acknowledge UniSZA and the Ministry of Higher Education (MOHE) Malaysia for granting the financial support and giving me the opportunity to embark on this journey under the 'Skim Latihan Akademik IPTA (SLAI) programme'.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

AF - Assisted Feedback
AHP - Analytic Hierarchy Process
AOPC - Automated Online Programming Contest
AST - Abstract Syntax Tree
CBR - Case-Based Reasoning
CFG - Context Free Grammar
COA - Computer Organization and Architecture
CW - Coursework
CT - Cognitive Tutor
DAO - Data Access Object
EML - Error Model Language
FCC - Feedback, context and content
GRF - General Recommended Feedback
IDEAS - Interactive domain-specific exercise assistants
IPO - Input-Process-Output
ICR - Instruction count ratio
IGR - Instruction-gram ratio
PAC - Problem Analysis Chart
RF - Recommended Feedback
RS - Recommender System
SRF - Specific Recommended Feedback
TOPSIS - Technique for Order of Preference by Similarity to Ideal Solution
WSM - Weighted Sum Method

LIST OF PUBLICATIONS

Journals

Suhailan Safei, Abdul Samad Shibghatullah and Burhanuddin Mohd Aboobaider, (2014). *A Perspective of Automated Programming Error Feedback Approaches in Problem Solving Exercise*. Journal of Theoretical and Applied Information Technology (JATIT), 70(1), pp. 121-129 *(SCOPUS)*

Suhailan Safei, Abdul Samad Shibghatullah, Burhanuddin Mohd Aboobaider and Mohd Kamir Yusof (2015). *A Targeted Ranking-Based Clustering Using AHP K-Means*. International Journal of Advances in Soft Computing and Its Application (IJASCA), 7 (3). pp. 100-113. *(SCOPUS)*

Proceeding

Suhailan Safei, Abdul Samad Shibghatullah and Burhanuddin Mohd Aboobaider (2014). *Enhanced k-Mean Clustering Algorithm Using Expert Features Weighting for Assisted Programming Feedback*. The 6th International Conference on Postgraduate Education. Melaka, Malaysia, 17-18 December 2014. Universiti Teknikal Malaysia Melaka.

CHAPTER 1

INTRODUCTION

1.0 Motivation

A new plan for 2015-2020 higher education development was launched to produce graduates who are able to think critically and innovatively with problem-solving initiatives and an entrepreneurial mindset. As for ICT graduates, computer programming is one of the core subjects that could grant these skills and attributes (Taheri et al., 2015). In many examples, those who are very competent in programming tend to write their own computer programs and eventually able to earn some cash on their own efforts. On the other hand, as mentioned by Cyberjaya's chairman, Tan Sri Mustapha Kamal, it is expected that 120,000 of the workforce in ICT with critical thinking skills from 2020 onwards (Azura, 2013).

Unfortunately, learning computer programming seems to be difficult among students. There are high failure rate among students in introductory programming in higher learning institution (Suliman et al., 2011; Ribeiro et al., 2014). Although most students are able to understand reading an existing program, but that does not mean that they are capable of solving any encountered programming errors during code writing (Swigger and Wallace, 1988). Many students will encounter difficulty in developing programming logics in solving a problem. As programming logic is a thinking skill, only continuous practice and experience of doing problem-solving exercises can develop the skill among the students.

In leveraging students to practice programming, a lot of automated programming assessment tools with automated feedbacks have been continuously developed (Ihantola et al., 2010; Le et al., 2013). From there, students could constructively build up the programming syntax and logic skills after experiencing some errors and mistakes. Such feedback messages will be instantly given by a compiler or any automated programming tools. The feedback is essential in helping students to stimulate both of their conscious and unconscious knowledge towards a correct program solution in meeting a problem's specification (Huitt, 2004).

However, current practice of the automated feedback is still insufficient to the novice students, especially when involving problem-solving techniques on a variety of programming exercises. They still require feedback from an expert to help them to interpret the automated feedback or rectify their problem-solving based mistakes. The expert can provide more dynamic feedback contents (what, why and how) and is able to interact with students in various contexts (when and where). Unfortunately, it seems difficult for the expert to provide individual feedback for each mistake during the programming lab exercises, especially when involving with a large group of students. In most cases, the mistakes or errors during a programming lab are repeated among individuals that require the expert to repeat the same feedback. Considering this situation, the feedback from the expert during a lab session could be effectively delivered by gathering the same programming difficulty into certain groups. This scenario should warrant for a further research to model an automated programming feedback model that can incorporate live expert's feedback and assistance during a programming lab session. This model should also promote an indexing mechanism to associate such expert's assisted feedback to be reactivated as a recommended feedback based on a similar difficulty case.

1.1 Background

Researchers have put efforts to automate computer-programming feedback that would not only cover on the syntax or semantic errors but also the program structure, styles and performance. Feedbacks are instantly provided via text messages when errors or mistakes are found in the students' program. However, when it comes to a feedback that is related to a problem's specification, it gives more challenges for a feedback to be modelled as the specification is varies. Thus, many resources are need to be spent especially for the expert to model a feedback on each set of the specification. It also difficult to predict and model all the possible feedback's cases especially when a new problem's specification arises. This is due to the automated programming feedback model that is constructed based on a pre-defined feedback library. All the feedback context and content needs to be determined prior or after a programming lab execution, but not during the lab execution.

Although the automated feedback model has been effectively used to assist user in detecting certain programming mistakes, the feedback is still not effective as the expert's feedback model especially in the rectifying problem's specification mistake (Queirós et al., 2012; Rubio-sánchez et al., 2014). This is some of the limitation of the model that lack of method to assess the program intention. For example, although rectifying mistake will relate to the output formatting is easy, most novice students may have difficulties in rectifying the mistake (Rubio-sánchez et al., 2014; Pieterse, 2013). The automated feedback usually does not know which part of the program that represents a specific goal of the problem's specifications. However, the goal of a program can be assessed by performing a static analysis to the program. The analysis requires a solution template to be provided and matched with the program. Then, a goal-based feedback can be specifically reported based on the discrepancy location of the template and the program.

Unfortunately, customising a program debugger such as using the solution template resource is costly to be built (Tam, 2011). As there are varieties of correct solution templates that can solve a problem, it is inefficient for the expert to design all the feedback cases on the different template's structure. Moreover, in a programming lab exercise, guiding a correct structure of the program logics among the novice students should be concerned more than the capability of automated feedback in detecting a mistake (Lee and Ko, 2011).

In contrast, the expert's feedback model does not only capable of alerting a mistake, but also it can suggest what is missing in answering the problem. Considering this, the live expert's feedback is still the best feedback model compared to the automated programming feedback (Rubio-sánchez et al., 2014; Queirós et al., 2012; Mungunsukh and Cheng, 2002). The expert can provide a more specific feedback to be tailored to a specific content (what, why and how) based on the students' background. They can also interact in a more dynamic context (when and where) in assisting students to rectify a programming problem. Furthermore, continuous guidance and instant feedback by the expert are among the key factors (Brito and Sá-soares, 2014) in learning how to program a solution. Without the proper feedback or guidance, students will fail to learn and construct their own knowledge on programming (Yadin, 2011). Unfortunately, guiding students can be a challenging task for the expert particularly when it involves many groups of students. It is also too costly to maintain a continuous interactive feedback with students (Wang et al., 2012). The expert needs to provide a specific feedback on each student based on a quick assessment of the student's computer program. Some of the computer program's assessment may take a longer time to be analysed, especially when the program's statements are not well organized. As a result, not many students can be entertained in a

programming lab session.

Considering there are different programming lab sessions that will be using a same question, a same expert's feedback maybe repeated on a similar mistake made by the other students. This problem also occurs in the expert's feedback model of online forum discussion where there are multiple questions and answers that are duplicated on a same issue. Furthermore, as the expert is assisted feedback in the electronic forum are not indexed as a recommender feedback, students are required to self-explore the discussion contents in seeking a solution to their program's mistake. This kind of approach is not suitable for a novice student that may has no clue on the mistake.

This research is focusing on developing a new model of automated programming feedback by integrating expert's assisted feedback into a recommended feedback. This model is meant to support continuous growth of feedback resources made by the expert in rectifying students' difficulties during elementary programming exercises. It enhances the effectiveness of the automated feedback content with the continuous growth of new expert's assisted feedback. In order to ease the expert's analysis in identifying the programming mistakes on a large group of students, ranking and clustering analysis of computer program is needed. The analysis will help the expert to extract and select certain features of the computer programs for automatically recommending the existing feedback based on a similar association index.

There are several important areas where this research makes an original contribution to the body of knowledge and solution to the domain of study. The study aims to model an automated programming feedback that integrates the expert's assisted feedback as a recommended feedback. In realizing the model, contributions are made on the automated computer program features and ranking-based clustering algorithm for

sorting and grouping the computer programs towards correctness level based on their problem-solving solution acceptability. The research is also contributing to a recommender system algorithm by proposing a new index mechanism to associate a feedback with a computer program's pattern. Hopefully, this research can eventually assist students to self-rectify their problem-solving based difficulty during a programming lab session without the need of an expert.

1.2 Problem Statement

Learning computer programming is one of a compulsory skill for a computer science student. According to constructivism learning theory, rather than having a knowledge transmission, one can understand better through his own experience or practice (Brito and Sá-soares, 2014). Therefore, in order to master the programming skill, lots of programming exercises need to be practised (Kwiatkowska, 2016). Nowadays, many automated programming tools are available to help the student in developing a computer program (Ihantola et al., 2010; Vaessen et al., 2014; Gulwani et al., 2014). A student can submit a computer program on a problem-solving exercise while the tool will instantly provide automated feedback to highlight any encountered errors or mistakes during the compilation or execution of the program. However, when it comes to a mistake that related to a question specification, current automated feedback is still lacking and insufficient as compared to the expert's feedback (Pieterse, 2013; Rubio-sánchez et al., 2014). Among the feedback models are template-based parser, test case automation and intelligent agent. However, the template-based parser and test case automation requires all the feedback's elements to be pre-designed by an expert before a programming lab exercise is conducted among the students. Meanwhile, the feedback using the intelligent agent tends to be too