UNIVERSITI PUTRA MALAYSIA

DETECTION ON AMBIGUOUS SOFTWARE REQUIREMENTS SPECIFICATION WRITTEN IN MALAY USING MACHINE LEARNING

MOHD FIRDAUS BIN ZAHRIN

FSKTM 2017 1

DETECTION ON AMBIGUOUS SOFTWARE REQUIREMENTS SPECIFICATION WRITTEN IN MALAY USING MACHINE LEARNING

By

MOHD FIRDAUS BIN ZAHRIN

Thesis Submitted to Universiti Putra Malaysia,

in Fulfilment of the Degree of Master Computer Science

June 2017

Abstract of thesis presented to the Senate of Universiti Putra Malaysia in fulfillment of the requirements for Master of Computer Science

DETECTION ON AMBIGUOUS SOFTWARE REQUIREMENTS SPECIFICATION WRITTEN IN MALAY USING MACHINE LEARNING

By

MOHD FIRDAUS BIN ZAHRIN

June 2017

ABSTRACT

Software requirement specification (SRS) document is the most crucial document in software development process. SRS is normally produced during the initial part of software development process and all subsequent steps in software development are influenced by the requirements. This implies that the quality of SRS influences the quality of the software product. However, to produce a good quality SRS document is a challenging task as the requirements are normally specified in Natural Language. Issues in requirement, such as ambiguities or incomplete specification may lead to misinterpretation of requirements which consequently, higher the risk of time and cost overrun of the project. Detecting ambiguity requirements in the initial phase is crucial since the ambiguities in requirements that found late

are more expensive to fix if it were found early. In Malaysia context, most of Malaysian government's SRS are written in Malay language as of the requirement to comply with the Article 152, the Federal Constitution of Malaysia (through PP. Bil. 9, 2009 [1] and SPA Bil. 1, 2006 [2]). Most of the work in detecting ambiguity requirements is conducted specifically in English. Unfortunately, the structure of writing between Malay and English is totally different. Hence, we propose a framework to detect ambiguity on SRS using supervised machine learning technique. Four (4) SRS have been collected as our case study and text mining technique is used to classify the ambiguity and unambiguity requirements. Four (4) algorithms have been evaluated to find the suitable classification algorithm for this purpose. As the result, the Random Forest algorithm is the best algorithm which is measured based on measurement metric i.e. F Measure is 0.89, IR Precision is 0.90, IR Recall is 0.89 and Correct is 89.89%. Based on the result, we developed a prototype tool called detection on ambiguous SRS written in Malay using machine learning. This prototype tool has been evaluated by ten (10) experienced participants consist of Requirement Engineer and System Analyst. As the result, six (6) participants are satisfied and two (2) participants are strongly satisfied with the prototype tool on overall.

Abstrak tesis yang dikemukakan kepada Universiti Putra Malaysia sebagai memenuhi keperluan Ijazah Sarjana Sains Komputer

PENGESANAN KESAMARAN KE ATAS SPESIFIKASI KEPERLUAN PERISIAN YANG DITULIS DI DALAM BAHASA MELAYU MENGGUNAKAN PEMBELAJARAN MESIN

Oleh

MOHD FIRDAUS BIN ZAHRIN

Jun 2017

ABSTRAK

Dokumen spesifikasi keperluan perisian (SRS) adalah dokumen yang paling penting dalam proses pembangunan perisian. SRS biasanya dihasilkan di peringkat awal proses pembangunan perisian dan langkah-langkah seterusnya dalam pembangunan perisian dipengaruhi oleh dokumen ini. Ini menunjukkan bahawa kualiti SRS mempengaruhi kualiti produk perisian. Walau bagaimanapun, untuk menghasilkan dokumen SRS yang berkualiti adalah satu tugas yang mencabar dimana keperluan perisian biasanya dinyatakan dalam Bahasa Natural. Isu-isu dalam keperluan perisian seperti kekaburan atau spesifikasi yang tidak lengkap boleh membawa kepada

salah tafsir keperluan yang seterusnya meningkatkan risiko pembangunan sistem yang boleh menyebabkan pelanjutan tempoh masa dan peningkatan kos membangunkan projek. Mengesan keperluan perisian yang samar di peringkat awal adalah penting kerana keperluan perisian yang samar ditemui agak lewat menjadi lebih mahal berbanding jika ia dikesan lebih awal. Dalam konteks di Malaysia, sebahagian besar daripada SRS kerajaan Malaysia ditulis dalam bahasa Melayu sebagai keperluan untuk mematuhi Artikel 152, Perlembagaan Persekutuan Malaysia (melalui PP. Bil. 9, 2009 [1] dan SPA Bil. 1, 2006 [2]). Kebanyakan penyelidikan dalam mengesan keperluan yang samar dilakukan berpandukan Bahasa Inggeris. Namun, struktur penulisan antara Bahasa Melayu dan Bahasa Inggeris adalah sama sekali berbeza. Oleh itu, kami mencadangkan satu kerangka untuk mengesan kesamaran pada SRS menggunakan teknik pembelajaran mesin di bawah seliaan. Empat (4) SRS telah dikumpulkan sebagai kajian kes dan teknik perlombongan teks digunakan untuk mengelaskan kesamaran dan ketidaksamaran keperluan. Empat (4) algoritma telah dinilai untuk mencari algoritma klasifikasi yang sesuai bagi tujuan ini. Sebagai hasilnya, algoritma Random Forest adalah algoritma yang terbaik karana apabila diukur berdasarkan ukuran metrik nilai F Measure ialah 0.89, IR Precision ialah 0.90, IR Recall 0.89 dan nilai peratus ketepatan ialah 89,89%. Berdasarkan keputusan tersebut, kami membangunkan alat prototaip dipanggil pengesanan kesamaran ke atas spesifikasi keperluan perisian yang ditulis di dalam bahasa Melayu menggunakan pembelajaran mesin.

Alat prototaip ini telah dinilai oleh sepuluh (10) peserta yang terdiri daripada Jurutera Keperluan dan Penganalisa Sistem. Hasil daripada penilaian, secara keseluruhannya, dua (2) peserta amat berpuas hati dan enam (6) orang peserta berpuas hati dengan alat prototaip ini.

ACKNOWLEDGEMENT

Alhamdulillah, there is no god but Allah SWT and the Prophet Muhammad is the Messenger of Allah SWT. First of all, thanks and praise goes to Allah SWT for His will and for bestowing me with health, time and maturity of mind to complete this project within the timeline.

A million thanks to Dr. Mohd Hafeez Bin Osman as being my supervisor during preparing proposal and project. He has provided assistance, guidance and constructive comments during the process of completing this project.

I would like to express appreciation to UPM lecturers, facilitators and staffs who have been involved directly or indirectly in this project. Besides that I would like to thank my colleagues and family for endless support in completing this project. Last but not least, I am indebted to the Public Service Department (JPA) and the Government of Malaysia for awarding me scholarship to further Master Degree study in FSKTM, UPM.

APPROVAL

Thesis submitted to the Senate of Universiti Putra Malaysia and has been accepted as fulfillment of the requirement for Master of Computer Science.

Project Supervisor,

Dr. Mohd Hafeez Bin Osman

Department of Software Engineering

Faculty of Computer Science and Information Technology

Universiti Putra Malaysia

Date:

DECLARATION

I hereby confirm that:

- This thesis is my original work;
- Quotations, illustrations and citations have been duly referenced;
- This thesis has not been submitted previously or concurrently for any other degree at any other institutions;
- Intellectual property from the thesis and copyright of thesis are fully-owned by University Putra Malaysia, as according to the University Putra Malaysia (Research) Rules 2012;
- written permission must be obtained from supervisor and the office of Deputy Vice-Chancellor (Research and Innovation) before thesis is published (in the form of written, printed or in electronic form) including books, journals, modules, proceedings, popular writings, seminar papers, manuscripts, posters, reports, lecture notes, learning modules or any other materials as stated in the University Putra Malaysia (Research) Rules 2012;
- There is no plagiarism or data falsification/fabrication in the thesis, and scholarly integrity is upheld as according to the University Putra Malaysia (Graduate Studies) Rules 2003 (Revision 2012-2013) and

the University Putra Malaysia (Research) Rules 2012. The thesis has undergone plagiarism detection software.

Signature: _____ Date: _____

Name and Matric No.: MOHD FIRDAUS BIN ZAHRIN (GS44368)

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

LIST OF ABBREVIATIONS

Abbreviations	Meaning
BOW	Bag of Words
CBR	Checklist Based Reading
CSV	Comma Separated Values
DBR	Defect Based Reading
DT	Decision Tree
HTML	Hypertext Markup Language
IR	Information Retrieval
IT	Information Technology
ME	Maximum Entropy
NB	Naïve Bayes
PBR	Perspective Based Reading
PHP	Hypertext Preprocessor
POS	Part of Speech
RF	Random Forest
RT	Random Tree
ROC	Receiver Operating Characteristic
SGD	Stochastic Gradient Descent
SQL	Structure Query Language
SRS	Software Requirement Specification
SVM	Support Vector Machines
TF-IDF	Term Frequency-Inverse Document Frequency
UPM	Universiti Putra Malaysia

CHAPTER 1

INTRODUCTION

1.1. Research Background

Software requirement specification (SRS) is the foundation and the most crucial document in software development process. SRS is normally produced during the initial part of software development process and all subsequent steps in software development are influenced by the requirements. Hence, high quality software requirements may increase the possibility of high software quality. It is just like the term “garbage in, garbage out” that has been used in programming software which means “If there is a logical error in software, or incorrect data are entered, the result will probably be either a wrong answer or a system crash” [3].

For companies that outsource their software development, software requirements document is the most crucial communication document that specifies the stakeholder’s vision and needs of software to be developed. Hence, a good software requirements document is needed to ensure the software developers understand and able to fulfil the stakeholder needs.

However, to produce a good quality SRS document is a challenging task as the requirements are normally specified in Natural Language. Issues in requirement, such as ambiguities or incomplete specification may lead to

misinterpretation of requirements which consequently, higher the risk of time and cost overrun of the project. Detecting ambiguities requirements in the initial phase is crucial since the ambiguities that found late is more expensive that if it was found early [4]. This implies that the quality of SRS influences the quality of the software product.

1.2. Problem Statement

Issues in software requirements, such as ambiguities or incomplete requirements specification can lead to time and cost overrun in a project [5]. Some of the issues in requirements specification can be manually detected by the requirements engineers and some are not possible. For example, the requirements specification that ambiguous, unclear and incomplete can be detected by the requirement engineer but the requirements that related to the domain knowledge is difficult to be detected. Consequently, an approach that offers requirements engineers' rapid detection to possible defect in specification could contribute valuable feedback [6].

Software requirements defects detection based on requirements template (also known as boilerplate) is feasible based on the work by Arora et. al [7], [8]. As the baseline, Arora et. al used the requirement templates from the ISO/IEC/IEEE 29148 [9] and the template proposed by Pohl and Rupp [10]. However, since requirements in the industry are nearly exclusively written in Natural Language, it is hard to detect the issues in requirements because

natural language has no formal semantics. In the Malaysia context, most of the Malaysia government's SRS are written in Malay. A lot of research has been conducted to solve this problem are focused on English. There are little works focused on Malay. Furthermore, the boilerplates for requirement are not commonly used by Malaysia government agencies that make the requirements review difficult to detect ambiguities in requirements.

1.3. Research Objectives

The objectives of this study are listed as below:

i. To propose a framework for ambiguous requirements detection using classification algorithm.
ii. To select the best classification model. The classification model selection is based on the evaluation of model's accuracy.
iii. To design and implement a prototype tool that support the ambiguity SRS detection proposed framework.
iv. To validate the proposed framework based on the prototype tool.

1.4. Research Scope

This study focus on mining the requirements from four (4) SRS documents written in Malay that provided by Malaysia government Agencies. The classification algorithm is used as framework in developing the prototype tool to facilitate the process.

1.5. Research Questions

There are two (2) research questions have been identified for this study:

RQ1. How to classify the ambiguity requirements?

RQ2. How effective the proposed framework in classifying ambiguity requirements?

1.6. Research Outline

This thesis is divided into six chapters. The following paragraphs provide a brief description of the remaining chapters of this thesis.

Chapter 2 provides a literature review by covering existing study, software requirements specification quality, classification techniques and algorithms. Chapter 3 describes the methodology which include theoretical study, propose a new framework, build a prototype tool and model framework. Meanwhile in Chapter 4, the implementation of the framework and prototype tool development will be covered. The result and discussion will be explained in Chapter 5. Chapter 6 concludes the thesis findings and future work.

REFERENCES

[1] Pekeliling Perkhidmatan Bilangan 9 Tahun 2011, *"Panduan Penggunaan Bahasa Kebangsaan Dalam Perkhidmatan Awam."* Jabatan Perkhidmatan Awam, 2011.

[2] Surat Pekeliling Am Bilangan 1 Tahun 2006, *"Langkah-langkah Memperkasakan Penggunaan Bahasa Kebangsaan Dalam Perkhidmatan Awam."* Jabatan Perkhidmatan Awam, 2006.

[3] "Definition of garbage in, garbage out," 2004. [Online]. Available: http://foldoc.org/GIGO. [Accessed: 12-May-2017].

[4] D. M. Berry, A. Bucchiarone, S. Gnesi, G. Lami, and G. Trentanni, "A new quality model for natural language requirements specifications," *Int'l Work. Requir. Eng. Found. Softw. Qual.*, pp. 1–12, 2006.

[5] D. Méndez Fernández and S. Wagner, "Naming the pain in requirements engineering: A design for a global family of surveys and first results from Germany," *Inf. Softw. Technol.*, vol. 57, pp. 616–643, 2015.

[6] H. Femmer, D. Méndez Fernández, S. Wagner, and S. Eder, "Rapid quality assurance with Requirements Smells," *J. Syst. Softw.*, vol. 123, pp. 190–213, 2017.

[7] C. Arora, M. Sabetzadeh, L. Briand, F. Zimmer, and R. Gnaga, "RUBRIC: A flexible tool for automated checking of conformance to requirement boilerplates," *2013 9th Jt. Meet. Eur. Softw. Eng. Conf. ACM SIGSOFT Symp. Found. Softw. Eng. ESEC/FSE 2013 - Proc.*,

pp. 599–602, 2013.

[8] C. Arora, M. Sabetzadeh, L. Briand, and F. Zimmer, “Automated checking of conformance to requirements templates using natural language processing,” *IEEE Trans. Softw. Eng.*, vol. 41, no. 10, pp. 944–968, 2015.

[9] ISO, IEC, and IEE, “Systems and software engineering — Life cycle processes — Requirements engineering (ISO/IEC/IEEE 29148),” *Iso/ Iec/ Ieee*, pp. 1–83, 2011.

[10] K. P. and C. Rupp, *Requirements Engineering Fundamentals, 1st ed*. Santa Barbara, CA 93103: Rocky Nook, 2011.

[11] A. Davis *et al.*, “Identifying and Measuring Quality in a Software Requirements Specification,” *Ieee*, pp. 141–152, 1993.

[12] B. Anda and D. I. K. Sjøberg., “Towards an inspection technique for use case models,” *Softw. Eng. Knowl. Eng. (SEKE),* 2002.

[13] A. V. Lamsweerde, “Requirements Engineering,” *John Wiley Sons*, 2009.

[14] A. Mavin, P. Wilkinson, A. Harwood, and M. Novak, “EARS (Easy Approach to Requirements Syntax),” *Proc. IEEE Int. Conf. Requir. Eng.*, pp. 317–322, 2009.

[15] A. A. Alshazly, A. M. Elfatatry, and M. S. Abougabal, “Detecting defects in software requirements specification,” *Alexandria Eng. J.*, vol. 53, no. 3, pp. 513–527, 2014.

[16] H. Haron, A. A. A. Ghani, and H. Haron, “A Conceptual Model To

Manage Lexical Ambiguity in Malay," *ARPN J. Eng. Appl. Sci.*, vol. 10, no. 3, pp. 1–8, 2015.

[17] H. Haron and A. A. A. Ghani, "A Method to identify potential ambiguous Malay words through ambiguity attributes mapping: An exploratory study," in *The Fourth Conference of Computer Science and Information Technology (CCST2014)*, 2014, vol. 4, no. 2, pp. 1–8.

[18] H. Haron and A. A. A. Ghani, "A Survey on Ambiguity Awareness towards Malay System Requirement Specification (SRS) among Industrial IT Practitioners," *Procedia Comput. Sci.*, vol. 72, pp. 261–268, 2015.

[19] D. M. . E. K. Berry, "From Contract Drafting to Software Specification: Linguistic Sources of Ambiguity," *Autom. Softw. Eng. 1997 …*, pp. 1–80, 2003.

[20] S. Matsumoto, H. Takamura, and M. Okumura, "Sentiment classification using word sub-sequences and dependency sub-trees," *Proc. 9th Pacific-Asia Conf. Adv. Knowl. Discov. Data Min.*, vol. 05the9, pp. 301–311, 2005.

[21] D. Zhang, H. Xu, Z. Su, and Y. Xu, "Chinese comments sentiment classification based on word2vec and SVMperf," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 1857–1863, 2015.

[22] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentiment reviews using n-gram machine learning approach," *Expert Syst. Appl.*,

vol. 57, pp. 117–126, 2016.

[23] R. Garreta and G. Moncecchi, *Learning scikit-learn: machine learning in python*. Berlin Heidelberg: Packt Publishing Ltd, 2013.

[24] B. Luo, J. Zeng, and J. Duan, “Emotion space model for classifying opinions in stock message board,” *Expert Syst. Appl.*, vol. 44, pp. 138–146, 2016.

[25] P. Ekman and W. V. Friesen, “Constants-Across-Cultures-In-The-Face-And-Emotion.pdf,” *Personal. Soc. Psychol.*, pp. 124–129, 1971.

[26] Quora, “What is a training data set & test data set in machine learning? What are the rules for selecting them?,” 2016. [Online]. Available: https://www.quora.com/What-is-a-training-data-set-test-data-set-in-machine-learning-What-are-the-rules-for-selecting-them. [Accessed: 02-Jun-2017].

[27] Wikipedia, “Precision and Recall,” 2017. [Online]. Available: https://en.wikipedia.org/wiki/Precision_and_recall#F-measure. [Accessed: 04-Jun-2017].

[28] RapidMiner, “RapidMiner tool.” 2006.

[29] University of Waikato, “WEKA tools.” 1993.

[30] T. Fushiki, “Estimation of prediction error by using K-fold cross-validation,” *Stat. Comput.*, vol. 21, no. 2, pp. 137–146, 2011.

[31] D. Cournapeau, “Scikit-Learn,” 2007. [Online]. Available: http://scikit-learn.org/stable/. [Accessed: 18-May-2017].