A FRAMEWORK FOR AUTOMATED QUALITY ASSESSMENT OF SOFTWARE REQUIREMENT SPECIFICATION BASED ON PART-OF-SPEECH TAGGING, MULTI-AGENT *K*-MEANS CLUSTERING AND CASE-BASED REASONING

MOHAMMED AHMED JUBAIR

A thesis submitted in fulfillment of the requirement for the award of the Degree of Doctor of Philosophy in Information Technology

Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia

JANUARY 2022

DEDICATION

This thesis is dedicated to:

The sake of Allah, my Creator.

My great teacher and messenger, Mohammed (May Allah bless
and grant him), who taught us the purpose of life;

My great parents, who lead me through the valley of darkness with the light of hope
and support;

My beloved brothers and sisters;

To all my family, the symbol of love and giving;
My friends who encourage and support me;

All the people in my life who touch my heart;

I dedicate this research.

My reverend mother

"Muna Abdulwaheed Abdulkareem"

Humanitarian, Compassionate, Religious, and a True Mother

"M" is for the million things she gives me,
"O" means only that she is growing old,
"T" is for the tears she shades to save me
"H" is for her heart of purest gold,
"E" is for her eyes, with love-light shining,
"R" means right, and right she will always be,
Put them all together, they spell "MOTHER",
A word that means the world to me.

ACKNOWLEDGEMENT

Firstly, all praise and thanks to Allah for His Divine Guidance and help in completing this research project. My Salaam and Gratitude to the Beloved Prophet Mohammed (Peace and Blessings of Allah Be upon Him) who was sent by Allah to be a great teacher of humankind. I would like to express my sincere gratitude to my supervisor, Assoc. Prof. Ts. Dr. Aida Mustapha for her encouragement, ideas, and support throughout the research project.

I appreciate and cherish the moments I have spent with her since the start of this research. We have had productive discussions and constructive arguments over the theories and concepts proposed in this thesis. I would like to thank my co-supervisors, Dr. Salama A. Mostafa and Dr. Mohamad Aizi Salamat for their constant support, guidance, and motivation. It would never have been possible for me to take this work to completion without their incredible support and encouragement.

I must also express my very profound gratitude to my father Ahmed Jubair, my mother Muna Abdulwaheed, my brother Ali Ahmed, sisters, all my friends especially Mustafa Hamid and Ahmed Salih, and members of the Faculty of Computer Science and Information Technology (FSKTM) for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Finally, I would like to thank the Ministry of Higher Education for partially sponsoring my study through the Fundamental Research Grant Scheme (Project No.: FRGS/1/2019/ICT01/UTHM/02/1) with the title “A Hybrid Approach for Fully Automated Software Requirement Specification Technique based on Multi-Agent Systems and Case-based Reasoning”.

ABSTRACT

Software Requirement Specification (SRS) is an imperative process in a Software Engineering (SE) cycle, where its role is to document functional and non-functional requirements and to establish the tasks that a particular system is set to accomplish. Because a badly written SRS has an expensive impact on the entire project, the success or failure of any software product depends on the quality of the SRS document. Recent advancements in the field have explored automated extraction of quality attributes in SRS documents such as the Reconstructed ARM and the Rendex models. However, automating the quality assessment process poses major challenges, which requires advanced Natural Language Processing (NLP) algorithms to extract the quality features, interpreting the context of the features, formulating the assessment metrics, and documenting the shortcomings as well as possible improvements. Recent automated models also attempted to assess the quality of the SRS based on a small number of quality attributes and indicators due to the limitation in extracting quality attributes that require specific indicators from the SRS. To address this gap, this thesis proposes an Automated Quality Assessment of SRS (AQA-SRS) framework by integrating NLP for feature extraction, Multi-Agent System (MAS) with *K*-means for features clustering, and Case-based Reasoning (CBR) for process management. This framework assessed the SRS documents by automatically extracted 11 quality attributes and their corresponding 11 quality indicators through a deep analysis of the SRS textual content. This process is performed through the Multi-Agent *K*-means (MA-*K*-means) model for handling the automatic evaluation of the AQA-SRS framework. The performance of the AQA-SRS framework is evaluated by comparing the results against the state-of-the-art techniques as well as human experts based on two standard SRS datasets. The results showed the AQA-SRS framework reliably handled the assessment of 11 quality attributes and their corresponding 11 quality indicators with Krippendorff's Alpha 0.78 for the agreement with software engineering experts.

ABSTRAK

Spesifikasi Keperluan Perisian (SRS) merupakan satu proses penting di dalam kitaran Kejuruteraan Perisian (SE), yang mana peranannya adalah untuk mendokumentasi keperluan fungsian dan bukan fungsian serta membangunkan tugasan bagi mencapai tujuan sesebuah sistem. Oleh kerana SRS yang tidak ditulis dengan baik boleh memberikan impak yang merugikan kepada keseluruhan projek, kejayaan atau kegagalan mana-mana perisian adalah bergantung kepada kualiti dokumentasi SRS. Kemajuan terkini bidang SE telah mengkaji pengekstrakan atribut kualiti daripada dokumen SRS secara automatik seperti model *Reconstructed ARM* dan *Rendex*. Walau bagaimanapun, proses mengautomasikan penilaian kualiti adalah sangat mencabar, serta memerlukan algoritma Pemprosesan Bahasa Tabii (NLP) termaju bagi mengekstrak ciri-ciri kualiti yang diperlukan, menterjemah konteks maksud ciri-ciri tersebut, memformulasi metrik penilaian, dan mendokumentasi sebarang kekurangan serta penambahbaikan. Model automasi terkini juga cuba untuk mengakses kualiti SRS berdasarkan ciri dan petunjuk kualiti yang sedikit disebabkan keterbatasan dalam proses pengekstrakan ciri-ciri yang memerlukan indikator spesifik sesebuah SRS. Bagi menangani kekurangan ini, tesis ini mencadangkan sebuah Kerangka Penilaian Kualiti SRS secara Automatik (AQA-SRS) dengan mengintegrasikan teknik NLP bagi pengekstrakan ciri-ciri, Sistem Multi-Agen (MAS) dengan *K*-Means bagi pengelompokan ciri-ciri, serta Penaakulan berasaskan Kes (CBR) bagi menguruskan proses ini. Kerangka ini menilai dokumen SRS dengan mengekstrak 11 atribut serta 11 indikator kualiti secara automatik melalui analisis bermakna ke atas kandungan teks. Proses ini dijalankan melalui model *Multi-Agent K-means* (MA-*K*-means) bagi mengurus penilaian automatik kerangka AQA-SRS tersebut. Prestasi kerangka AQA-SRS dinilai dengan membandingkan hasil dengan teknik terkini serta pakar bidang melalui dua dataset piawai SRS. Keputusan menunjukkan kerangka AQA-SRS mengurus penilaian 11 atribut dan 11 indikator kualiti secara dipercayai dengan keseragaman Krippendorff's Alpha 0.78 berbanding pakar Kejuruteraan Perisian.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDIX

LIST OF SYMBOLS AND ABBREVIATIONS

RE	-	Requirement Engineering
SRS	-	Software Requirement Specification
SQA	-	Software Quality Assurance
SDLC	-	Software Development Life Cycle
SATC	-	Software Assurance Technology Center
Q^a	-	Quality Attributes
Q^i	-	Quality Indicators
AI	-	Artificial Intelligence
NLP	-	Natural Language Process
MAS	-	Multi-Agent System
CBR	-	Case-Based reasoning
ARM	-	Automated Requirement Measurement
AQA-SRS	-	Automated Quality Assessment of SRS
FAQA-metrics	-	Fully Automated Quality Assessment metrics
MA-*K*-means	-	Multi Agent-*K*-means
IT	-	Inspection Technique
RT	-	Review Techniques
DBR	-	Defect-based reading
PBR	-	Perspective-based reading
UBR	-	Usage-based reading
CLBR	-	Checklist-based reading techniques

LIST OF PUBLICATIONS

(i) **Jubair, M. A.,** Mostafa, S. A., Mustapha, A., & Hafit, H. (2018, August). A survey of multi-agent systems and case-based reasoning integration. In 2018 International Symposium on Agent, Multi-Agent Systems and Robotics (ISAMSR) (pp. 1-6). IEEE.

(ii) **Jubair, M. A.,** Mostafa, S. A., Mustapha, A., Amana H. & Hassan, M. H. (2019). Fully automated quality assessment metrics for Software requirement specifications. AUS journal, 26(1), 188-197.

(iii) **Jubair, M. A.,** Mostafa, S. A., Mustapha, A., Salamat, M. A., and Hassan M. H., (2020). Digging Deeper into Quality Assessment for Software Requirement Specifications. Journal of Critical Reviews, JCR. 2020; 7(12): 3869-3875.

(iv) Mostafa, S. A., Gunasekaran, S. S., Khaleefah, S. H., Mustapha, A., **Jubair, M. A.,** & Hassan, M. H. (2019). A fuzzy case-based reasoning model for software requirements specifications quality assessment. International Journal on Advanced Science, Engineering and Information Technology, 9(6), 2134-2141.

CHAPTER 1

INTRODUCTION

1.1 Overview

Requirement Engineering (RE) is a process that involves a set of activities from collection, analysis, specification, to validation of user requirements in the form of natural language (Davis *et al*., 2011). The implementation of RE is carried out throughout the early stage of the software development life cycle. In the process of RE, the most crucial factor is the Software Requirement Specification (SRS) documents, which is the main outcome of the process (Aurum & Wohlin, 2003). SRS is a set of requirements that describes the features and properties of the desire software product. It has numerous advantages to the software developers, it represent the functional and non-functional specification of the desired product, states the scope of the project, reduces the development effort, and removes any misunderstanding in the early stage (Wilson *et al*., 1999).

In the SRSs document, all the expected capabilities and functionalities that must be present in a software system are explicitly stated. In addition to these, the document also spells out the limits of the system. While a requirement can be described as an objective that a system must meet, a specification is a description of how the objective must be met (Jani, 2010). One main challenge in preparing SRS documents is the complex writing structures to describe the requirements (Mostafa & Jani, 2011). A poor requirement of any product leads to failing the products because the quality of SRSs is the determinant of the quality of any kind of product. In addition, the stakeholders' needs, as well as the limitations, are also reflected in the SRS (Wilson *et al*., 1997).

Nonetheless, to document requirements, natural language is still the primary means. Requirements in natural language can be created and understood by all stakeholders without additional effort and specific requirements engineering background. However, natural language poses the risk of being imprecise. Poorly written requirements have an expensive impact on the whole project. Incomplete or ambiguous requirements generate additional effort due to unnecessary feedback loops. In the end, bad requirements lead to misinterpretations and finally to the wrong product (Femmer *et al*., 2017).

Software Quality Assurance (SQA) is a set of processes that utilize to protect the quality of any software delivered by monitoring the processes of software engineering in different stages, which ultimately leads to, or at least gives confidence, high-quality software products (Parnas & Lawford, 2003). SQA expands on the entire software development cycle (e.g., SDLC), which rely on the design of software, coding, testing, and release management. The main focus of the SQA plan is to ensure a system or service aligns with the requirements defined in the SRS. Therefore, SRS quality evaluation is critical to identify the level of quality and faults in the very starting steps of the software development process (Thitisathienkul & Prompoon, 2015). The success of a project is strongly determined by the presence of a set of statements that clearly define the requirements of a system. In other words, the success of software projects is significantly influenced by requirements (Ali *et al*., 2018).

The quality of SRS must be guaranteed to achieve a successful software project. Besides, there are no hard and fast rules associated with the production of the SRS document. However, there are given authorities that have suggested sets of information and details that must be possessed by a high-quality SRS. One of such authorities includes the Institute of Electrical and Electronics Engineers (IEEE) that has provided the recommended practice for SRS (IEEE-SA Standards Board, 2000). Since the quality of SRS determines the quality of a given product, it is crucial to have a robust requirement that leads to the achievement of quality and highly efficient software (Nordin *et al*., 2017). For this purpose, Wilson *et al*. (1997) proposed a set of Quality Attributes (Q^a) and Quality Indicators (Q^i) that used to assess the quality of the SRS document. To achieve their goal, they defined a group of word/phrase that indicate the defect in the SRS document and distributed these features by indicators.

These Q^a and Q^i are used to ensure the SRS has precise requirements; the researchers proposed several techniques to detect the defect in SRS document based on the Q^a and Q^i such as the work of Jani and Islam (2012); Alshazly *et al*. (2014); Haque *et al*. (2019), and Femmer *et al*. (2018). These approaches can be categorized into three main categories, which are, automated, semi-automated, and non-automated or manual inspection. Manual inspection is considering a popular technique used to assess the quality of the SRS document. However, it is time-consuming for different reviewers to inspect and integrate the results of the review manually. Since the completion of a review cycle is achieved within several days or even weeks, the author of the requirements must wait a long time before receiving feedback. Additionally, the reviews of different individuals' consequent inconsistency (Saavedra *et al*., 2013). The implication of these problems affects the assessment quality of the SRS and increases the cost.

To address these issues, different Artificial Intelligence (AI) and statistical methods have been used (i.e. case-based reasoning, natural language process, correlation coefficient, multi-agent system). For instance, in 2012, Jani & Islam, proposed semi-automated methods based on Case-based Reasoning (CBR) that combine the manual and automated methods to reduce the inspection time, cost, and workload. The technique has made a significant contribution in reducing the time, cost, and workload required for the process of assessment. However, there is still a need for improvement because humans (experts) still interact with their methods to assess the quality of the document (Rossanez & Carvalho, 2016). There are several AI and statistical methods that are used to measure the quality of the SRS document automatically (i.e. using fuzzy logic, neural network, genetic algorithm, and so on). These methods employ to propose an automated approach, such as the work of (Génova *et al*., 2013; Bakar *et al*., 2016; Thitisathienkul & Prompoon, 2014). However, all the works only considered a limited number of Q^a such as ambiguity, completeness and correctness. This leads to the need for a comprehensive model that deals with different methods or techniques like; pre-processing, features extraction, analysis and assessment to handle a wider range of Q^a (Rine & Fraga, 2015; Hisaszumi *et al*., 2019; Li *et al*., 2018; Stephen & Mit, 2017).

In general, some motivations show the need for automatic review, and these motivations are given as follows. First, an automatic review is fast in terms of different aspects of quality; instant feedback is one of the benefits of automatic reviews. For instance, in the current configuration processes are received by the requirements scout, and then feedback for a paragraph is given about 500ms. This way, it is possible to obtain immediate feedback. Second, an automatic review is low in cost. The high cost of performing manual reviews remains the major problem associated with the process of manual reviews that come along with a comprehensive analysis. Thus, it becomes paramount to have a technique that can be used in obtaining feedback at a low cost; this is surely a promising advantage. Finally, an automatic review is consistent. With the manual method, two different reviewers can provide different results on different occasions for a requirement artifact. Even though this can be advantageous in terms of quality factors such as if the artifact satisfies a given guideline, this subjectivity paves the way for inconsistencies. An automatic review, on the other hand, is capable of providing consistent results for a requirement artifact on different occasions.

To overcome these problems, an automated technique has been proposed in this thesis that includes several statistical and AI methods to minimize the role of the human expert. However, there is no standard Q^a and Q^i that can assess the quality of the SRS in an automated manner. For this reason, this thesis focuses on three main points. Firstly, define a new group of Q^a and Q^i that can be assessed in an automated way. Secondly, construct a framework that is able to assess the quality of the SRS in an automated manner. Finally, this thesis proposes a model based on collaboration between Multi-Agent System and K-means clustering algorithm, that used to clustering the extracted feature based on the corresponding Q^a and Q^i.

This thesis attempts to mimic the steps of the reviewers when assessing the quality of the SRS document. First, the reviewers read the document, and they take a general idea about it. Then, they focus on finding the defect in the SRS document. Next, generate a report that shows the overall quality of the SRS. The proposed work also involves Natural Language Processing (NLP) techniques to understand the SRS document and extract the text from the documents. The AI and statistical methods are used to assess the SRS quality and generate an assessment report. This technique saves time, reduces the cost, workload and produces consistency assessment.

1.2 Problem Statement

Developers and stakeholders can understand requirements written in natural language. However, the usage of natural language in writing the Software Requirement Specification (SRS) along with human writing skills poses the risk of producing low quality and poorly written SRS. The low-quality SRS creates severe problems for the development process of software systems, which eventually causes additional costs and makes unnecessary processing loops (Thitisathienkul & Prompoon, 2015). Therefore, the quality of SRS documents is one of the critical factors for determining project success or failure (Wang *et al*., 2013; Femmer *et al*., 2017).

The popular approach of manual inspection of the SRS documents by multiple reviewers and then integrating the review results is a challenging task. One review cycle often takes days or weeks to be completed. Meanwhile, the author of the requirements has to wait a long time before receiving the assessment report. The result of these problems is that reviews are often only performed sporadically or only superficially (Antinyan & Staron, 2017). To tackle these challenges, many researchers proposed different automated models or tools to assess the quality of the SRS document. All existing models attempt to assess the quality of the SRS based on a limited number of quality attributes and indicators, this is due to several factors, including the assessment of each quality attribute requires extracting specific indicators or features and there is an overlapping of features between the quality attributes.

For instance, the work of Siahaan & Umami (2012) introduced an approach to detect the forward reference in the SRS document by using NLP. Carlson & Laplante (2014) reconstructed the Automated Requirement Measurement (ARM) model (Wilson *et al*., 1997) to increase the quality of the SRS document by assessing three Q^as which are ambiguity, complete, and understandability. Antinyan & Staron (2017) introduced the Rendex model to assess the understandability of the document. Nonetheless, these models are still lacking in terms of contextual features, which affect the process of constructing relationships between different features, the depth of the analysis, and the assessment process. Existing works neglect contents such as tables, text analysis, and more importantly relationships between the quality features.

In addition, quality indicators or features need to be grouped into clusters in order to make it possible to measure the corresponding Q^a and Q^i. The work of Mezghani *et al*. (2018b) used several NLP operators such as Part-of-Speech (POS) and noun chunking to extract features from requirements. It then uses a *K*-means clustering algorithm to categorize the features into groups of quality indicators. These indicators are used to assess redundancy and inconsistency among the requirements documents. In their work, the optimal value of *k* was found based on inertia and statistical gap.

However, the *k*-means algorithm finds locally optimal solutions concerning the clustering error. It is a fast-iterative algorithm that has been used in many clustering applications. It is a point-based clustering method that starts with the cluster centers initially placed at arbitrary positions and proceeds by moving at each step of the cluster centers to minimize the clustering error. The main disadvantage of the method lies in its sensitivity to the initial positions of the cluster enters. Therefore, to obtain near-optimal solutions using the k-means algorithm several runs must be scheduled differing in the initial positions of the luster enters (Likas *et al*., 2011). Also, handling a bigger number of quality indicators and attributes need an advanced algorithm that can draw the boundaries for the shape of clusters and avoid falling in local optimal.

1.3 Research Objectives

The main objective of this research is to propose a framework that utilizes to assess the quality of the SRS document in an automated manner. The following points will achieve the desired objective:

- To propose an Automated Quality Assessment of SRS (AQA-SRS) framework by integrating part-of speech-tagging for textual feature extraction, Multi-Agent and *K*-means for clustering, and Case-Based Reasoning (CBR) for assessment.
- To formulate a Multi-Agent *K*-means (MA-*K*-means) model for handling the measurements of the SRS quality in the AQA-SRS framework.
- To test and evaluate the performance of the AQA-SRS framework for clustering evaluation by implementing the PURE and Reconstruction ARM datasets, as well as the wine and synthetic datasets.

1.4 Research Scope

This work focuses on assessing the quality of the SRS document in an automated way to get a high-quality review and assessment to the SRS in lesser time, lower cost, lesser workload, and higher consistency of the review. The scope of this research concentrates on defining new metrics that comprise a group of quality attributes Q^a (i.e. complete, correct, unambiguity) and Q^i (like; imperative, continuances, directives) that can be assessed in an automated manner.

AQA-SRS comprises four main methods, which feature extraction using Natural Language Processing (NLP) approaches feature clustering using the Multi-Agent Systems (MAS) with *K*-means algorithm, and Case-based Reasoning (CBR) for process management and evaluation. The Java programing language is used to implement the proposed model. The performance of the AQA-SRS framework will be tested by using unlabeled standard datasets that consist of 79 SRS document and a Reconstruction ARM dataset that contain four pre-assessed SRS documents.

The SRS assessment datasets are prepared to provide an environment for testing particular problems, which are addressed in this work. The evaluation metric to the AQA-SRS framework performance focuses on assessing the reliability by measuring the agreement percentage of the proposed AQA-SRS framework against human experts. The classification of the requirement of functional and non-functional is out of the scope of this research. Additionally, quality attributes (Q^i) that require expert review or semantic analysis are also beyond the scope of this research.

1.5 Thesis Organization

This research presents the work of proposing an automated AQA-SRS framework that can assess the quality of the SRS document in an automated manner. This chapter presented the problem statement, objectives, and scope. The remainder of the chapters are organized as follows:

Chapter Two: This chapter presents a comprehensive literature review of this research. It first presents an introduction of the SRS document, a summary of the quality attributes and indicators, challenges, and vulnerabilities. Also, the chapter

REFERENCES

Abad, Zahra Shakeri Hossein, Oliver Karras, Parisa Ghazi, Martin Glinz, Guenther Ruhe, and Kurt Schneider. 2017. "What Works Better? A Study of Classifying Requirements." *Proceedings of the 2017 IEEE 25th International Requirements Engineering Conference, RE 2017* (1):496–501.

Aceituna, Daniel, Gursimran Walia, Hyunsook Do, and Seok Won Lee. 2014. "Model-Based Requirements Verification Method: Conclusions from Two Controlled Experiments." *Information and Software Technology* 56(3):321–34.

Ahmad, S., U. Anuar, and N. A. Emran. 2018. "A Tool-Based Boilerplate Technique to Improve SRS Quality: An Evaluation." *Journal of Telecommunication, Electronic and Computer Engineering* 10(2–7):111–14.

Akbar, Muhammad Azeem, Nasrullah, Muhammad Shafiq, Jawad Ahmad, Muhammad Mateen, and Muhammad Tanveer Riaz. 2019. "AZ-Model of Software Requirements Change Management in Global Software Development." *2018 International Conference on Computing, Electronic and Electrical Engineering, ICE Cube 2018*.

Ali, Syed Waqas, Qazi Arbab Ahmed, and Imran Shafi. 2018. "Process to Enhance the Quality of Software Requirement Specification Document." *2018 International Conference on Engineering and Emerging Technologies, ICEET 2018* 2018,1–6.

Alshazly, Amira A., Ahmed M. Elfatatry, and Mohamed S. Abougabal. 2014. "Detecting Defects in Software Requirements Specification." *Alexandria Engineering Journal* 53(3):513–27.

Antinyan, Vard and Miroslaw Staron. 2017. "Rendex: A Method for Automated Reviews of Textual Requirements." *Journal of Systems and Software* 131:63–77.

Antinyan, Vard, Miroslaw Staron, Anna Sandberg, and Jorgen Hansson. 2017. "A Complexity Measure for Textual Requirements." *Proceedings of the 26th International Workshop on Software Measurement, IWSM 2016 and the 11th International Conference on Software Process and Product Measurement,* *Mensura 2016* 148–58.

Anu, Vaibhav, Wenhua Hu, Jeffrey C. Carver, Gursimran S. Walia, and Gary Bradshaw. 2018. “Development of a Human Error Taxonomy for Software Requirements: A Systematic Literature Review.” *Information and Software Technology* 103(June):112–24.

Arellano, Andres, Edward Zontek-Carney, and Mark A. Austin. 2015. “Frameworks for Natural Language Processing of Textual Requirements.” (December).

Arora, Chetan, Mehrdad Sabetzadeh, Arda Goknil, Lionel C. Briand, and Frank Zimmer. 2015. “NARCIA: An Automated Tool for Change Impact Analysis in Natural Language Requirements.” *2015 10th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC/FSE 2015 - Proceedings* 962–65.

Aslin Jenila, P. S. and P. Ranjana. 2011. “Design Pattern Prediction Techniques: A Comparative Analysis.” *Proceedings of the International Conference on Nanoscience, Engineering and Technology, ICONSET 2011* 574–77.

Aurum, Aybüke and Claes Wohlin. 2003. “The Fundamental Nature of Requirements Engineering Activities as a Decision-Making Process.” *Ist* 45(14):945–54.

Aybuke, Aurum, Petersson Hakan, Wohlin Claes, Aybuke Aurum, Håkan Petersson, and Claes Wohlin. 2002. “State-of-the-Art : Software Inspections after 25 Years.” *Software Testing Verification and Reliability* 12(3):133–54.

Bakar, Noor Hasrina, Zarinah M. Kasirun, and Norsaremah Salleh. 2015. “Feature Extraction Approaches from Natural Language Requirements for Reuse in Software Product Lines: A Systematic Literature Review.” *Journal of Systems and Software* 106:132–49.

Bakar, Noor Hasrina, Zarinah M. Kasirun, and Norsaremah Salleh. 2016. “Terms Extractions: An Approach for Requirements Reuse.” *IEEE 2nd International Conference on InformationScience and Security, ICISS 2015* 31–34.

Bassi, Youssef. 2012. “A Simulation Model for the Spiral Software Development Life Cycle.” *International Journal of Innovative Research in Computer and Communication Engineering* 02(05):3823–30.

Belsis, Petros, Anastasios Koutoumanos, and Cleo Sgouropoulou. 2014. “PBURC: A Patterns-Based, Unsupervised Requirements Clustering Framework for Distributed Agile Software Development.” *Requirements Engineering* 19(2):213–25.

Belsis, Petros, Anastasios Koutoumanos, Cleo Sgouropoulou, Manel Mezghani, Juyeon Kang, and Florence Sèdes. 2018. "PBURC: A Patterns-Based, Unsupervised Requirements Clustering Framework for Distributed Agile Software Development." *13th International Workshop on Natural Language Processing and Cognitive Science* 10859 LNCS(September):27–33.

Boehm, Barry W. and Rony Ross. 1989. "Theory-W Software Project Management: Principles and Examples." *IEEE Transactions on Software Engineering* 15(7):902–16.

Cambria, Erik and Bebo White. 2014. "Jumping NLP Curves: A Review of Natural Language Processing Research." *IEEE Computational Intelligence Magazine* 9(2):48–57.

Carlson, Nathan and Phil Laplante. 2014. "The NASA Automated Requirements Measurement Tool: A Reconstruction." *Innovations in Systems and Software Engineering* 10(2):77–91.

Castañeda, Verónica, Luciana Ballejos, and Ma Laura Caliusco. 2012. "Improving the Quality of Software Requirements Specifications with Semantic Web Technologies." *WER 2012 - 15th Workshop on Requirements Engineering* (January).

Chowdhury, Ranak Roy, Muhammad Abdullah Adnan, and Rajesh K. Gupta. 2019. "Real Time Principal Component Analysis." *Proceedings - International Conference on Data Engineering* 2019-April:1678–81.

Cunningham, Pádraig. 1994. "CBR : Strengths and Weaknesses." 2:517–23.

Davis, Alan, Scott Overmyer, Kathleen Jordan, Joseph Caruso, Fatma Dandashi, Anhtuan Dinh, Gary Kincaid, Glen Ledeboer, Patricia Reynolds, Pradip Sitaram, Anh Ta, and Mary Theofanos. 2011. "Identifying and Measuring Quality in a Software Requirements Specification." *Software Requirements Engineering* 194–205.

Demilie, Wubetu Barud and Automated Class. 2020. "Implemented Stemming Algorithms for Six Ethiopian Languages Implemented Stemming Algorithms for Six Ethiopian Languages." (August):5–10.

Denger, C., D. M. Berry, and E. Kamsties. 2003. "Higher Quality Requirements Specifications through Natural Language Patterns." *Proceedings - IEEE International Conference on Software- Science, Technology and Engineering, SwSTE 2003* 80–90.

Diamantopoulos, Themistoklis and Andreas Symeonidis. 2018. "Enhancing Requirements Reusability through Semantic Modeling and Data Mining Techniques." *Enterprise Information Systems* 12(8–9):960–81.

Doe, John. 2011. *IEEE Recommended Practice for Software Requirements Specifications*.

Fabbrini, F., M. Fusani, S. Gnesi, and G. Lami. 2001. "An Automatic Quality Evaluation for Natural Language Requirements." *Seventh International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'01)* 1:4–5.

Falessi, Davide, Giovanni Cantone, and Gerardo Canfora. 2013. "Empirical Principles and an Industrial Case Study in Retrieving Equivalent Requirements via Natural Language Processing Techniques." *IEEE Transactions on Software Engineering* 39(1):18–44.

Femmer, Henning. 2017. "Automatic Requirements Reviews - Potentials, Limitations and Practical Tool Support." *In International Conference on Product-Focused Software Process Improvement (Pp. 617-620). Springer, Cham.* 482–96.

Femmer, Henning. 2018. "Requirements Quality Defect Detection with the Qualicen Requirements Scout." *CEUR Workshop Proceedings* 2075.

Femmer, Henning, Daniel Méndez Fernández, Stefan Wagner, and Sebastian Eder. 2017. "Rapid Quality Assurance with Requirements Smells." *Journal of Systems and Software* 123:190–213.

Femmer, Henning, Michael Unterkalmsteiner, and Tony Gorschek. 2017. "Which Requirements Artifact Quality Defects Are Automatically Detectable? A Case Study." *Proceedings - 2017 IEEE 25th International Requirements Engineering Conference Workshops, REW 2017* 400–406.

Ferguson, R., Goldenson, D., Fusani, M., Fabbrini, F., & Gnesi, S. 2005. "Automated Natural Language Analysis of Requirements and Specifications." *INCOSE (International Council on System Engineering) International Symposium*.

Ferrari, Alessio, Beatrice Donati, and Stefania Gnesi. 2017. "Detecting Domain-Specific Ambiguities: An NLP Approach Based on Wikipedia Crawling and Word Embeddings." *Proceedings - 2017 IEEE 25th International Requirements Engineering Conference Workshops, REW 2017* 393–99.

Ferrari, Alessio, Gloria Gori, Benedetta Rosadini, Iacopo Trotta, Stefano Bacherini, Alessandro Fantechi, and Stefania Gnesi. 2018. "Detecting Requirements Defects with NLP Patterns: An Industrial Experience in the Railway Domain." *Empirical Software Engineering* 23(6):3684–3733.

Ferrari, Alessio, Giorgio Oronzo Spagnolo, and Stefania Gnesi. 2017. "PURE: A Dataset of Public Requirements Documents." *Proceedings - 2017 IEEE 25th International Requirements Engineering Conference, RE 2017* 502–5.

Fränti, Pasi and Sami Sieranoja. 2018. "K-Means Properties on Six Clustering Benchmark Datasets." *Applied Intelligence* 48(12):4743–59.

Fu, Jianxi and Yuanlue Fu. 2012. "Case-Based Reasoning and Multi-Agents for Cost Collaborative Management in Supply Chain." *Procedia Engineering* 29:1088–98.

Garousi, Vahid, Ahmet Coşkunçay, Onur Demirörs, and Ali Yazici. 2016. "Cross-Factor Analysis of Software Engineering Practices versus Practitioner Demographics: An Exploratory Study in Turkey." *Journal of Systems and Software* 111:49–73.

Génova, Gonzalo, José M. Fuentes, Juan Llorens, Omar Hurtado, and Valentín Moreno. 2013. "A Framework to Measure and Improve the Quality of Textual Requirements." *Requirements Engineering* 18(1):25–41.

Haque, Md Ariful, Md Abdur Rahman, and Md Saeed Siddik. 2019. "Non-Functional Requirements Classification with Feature Extraction and Machine Learning: An Empirical Study." *1st International Conference on Advances in Science, Engineering and Robotics Technology 2019, ICASERT 2019* 2019(Icasert).

Hisazumi, Kenji, Yuedong Xiao, and Akira Fukuda. 2019. "Feature Extraction from Japanese Natural Language Requirements Documents for Software Product Line Engineering." *Proceedings - Companion of the 19th IEEE International Conference on Software Quality, Reliability and Security, QRS-C 2019* 322–29.

Husain, Mohd Shahid and M. Akheela Khanum. 2016. "Word Sense Disambiguation in Software Requirement Specifications Using Wordnet and Association Mining Rule." *ACM International Conference Proceeding Series* 04-05-Marc:4–7.

Husain, Mohd Shahid and Mohd Rizwan Beg. 2015. "Advances in Ambiguity Less NL SRS : A Review." (March 2015):221–25.

Hussain, Ishrar, Olga Ormandjieva, and Leila Kosseim. 2007. "Automatic Quality Assessment of SRS Text by Means of a Decision-Tree-Based Text Classifier." *Proceedings - International Conference on Quality Software* 209–18.

IEEE-SA Standards Board. 2000. "IEEE Recommended Practice for Architectural

Description of Software-Intensive Systems." *IEEE Std* 1471–2000:1–23.

IEEE. 1998. "Recommended Practice for Software Requirements Specification, IEEE Std 830-1993, IEEE Computer Society." *Software Engineering Standard Comminttee of the IEEE Std Computer Society* Revision o:32.

Irshad, Mohsin, Kai Petersen, and Simon Poulding. 2018. "A Systematic Literature Review of Software Requirements Reuse Approaches." *Information and Software Technology* 93(September 2017):223–45.

ISO/IEC/IEEE. 2011. "IEEE/ISO/IEC 29148-2018 - ISO/IEC/IEEE International Standard - Systems and Software Engineering -- Life Cycle Processes -- Requirements Engineering." 1–94.

Jani, Hajar and Tariqul Islam. 2012. "A Framework of Software Requirements Quality Analysis System Using Case-Based Reasoning and Neural Network." *Proceedings - 2012 6th International Conference on New Trends in Information Science, Service Science and Data Mining (NISS, ICMIA and NASNIT), ISSDM 2012* 152–57.

Jani, Hajar Mat. 2010. "Applying Case-Based Reasoning to Software Requirements Specifications Quality Analysis System." *2nd International Conference on Software Engineering and Data Mining, SEDM 2010* (January 2010):140–44.

Jubair, Mohammed Ahmed, Salama A. Mostafa, Aida Mustapha, and Hanayanti Hafit. 2018. "A Survey of Multi-Agent Systems and Case-Based Reasoning Integration." *International Symposium on Agents, Multi-Agent Systems and Robotics 2018, ISAMSR 2018* (August).

Kamalrudin, Massila and Safiah Sidek. 2015. "A Review on Software Requirements Validation and Consistency Management." *International Journal of Software Engineering and Its Applications* 9(10):39–58.

Kamsties, Erik, Daniel M. Berry, and Barbara Paech. 2001. "Detecting Ambiguities in Requirements Documents Using Inspections." *Engineering* (July):1–12.

Kassab, Mohamad, Colin Neill, and Phillip Laplante. 2014. "State of Practice in Requirements Engineering: Contemporary Data." *Innovations in Systems and Software Engineering* 10(4):235–41.

Khan, R. A. and K. Mustafa. 2008. "Secured Requirement Specification Framework (SRSF)." *American Journal of Applied Sciences* 5(12):1622–29.

Kirner, Tereza G. and Janaina C. Abib. 1997. "Inspection of Software Requirements Specification Documents: A Pilot Study." *ACM SIGDOC Annual International*

Conference on Computer Documentation, Proceedings (January 1997):161–71.

Kitchenham, B. and S. L. Pfleeger. 1996. "Software Quality: The Elusive Target." *IEEE Software* 13(1):12–21.

Knauss, Eric and Christian El Boustani. 2008. "Assessing the Quality of Software Requirements Specifications." *Proceedings of the 16th IEEE International Requirements Engineering Conference, RE'08* 341–42.

Koo, Seo Ryong, Poong Hyun Seong, Junbeom Yoo, Sung Deok Cha, and Yeong Jae Yoo. 2005. "An Effective Technique for the Software Requirements Analysis of NPP Safety-Critical Systems, Based on Software Inspection, Requirements Traceability, and Formal Specification." *Reliability Engineering and System Safety* 89(3):248–60.

Laat, Martin DE. 2019. "Design and Validation of a Software Requirements Specification Evaluation Checklist."

Laitenberger, Oliver and Jean Marc Debaud. 2000. "An Encompassing Life Cycle Centric Survey of Software Inspection ISERN-98-32." *Journal of Systems and Software* 50(1):5–31.

Lami, G. 2005. "QuARS: A Tool for Analyzing Requirements." *Software Engineering Measurement and Analysis Initiative* (CMU/SEI-2005-TR-014 ESC-TR-2005-014):33.

Leau, Yu, Wooi Khong Loo, Wai Yip Tham, and Soo Fun Tan. 2012. "Software Development Life Cycle AGILE vs Traditional Approaches." 37(Icint):162–67.

Li, Yang, Sandro Schulze, and Gunter Saake. 2018. "Extracting Features from Requirements: Achieving Accuracy and Automation with Neural Networks." *25th IEEE International Conference on Software Analysis, Evolution and Reengineering, SANER 2018 - Proceedings* 2018-March:477–81.

Likas, Aristidis, Nikos Vlassis, and Jakob Verbeek. 2011. "The Global K-Means Clustering Algorithm." *ISA Technical Report Series*.

Linghu, Bin and Feng Chen. 2014. "An Intelligent Multi-Agent Approach for Flood Disaster Forecasting Utilizing Case Based Reasoning." *Proceedings - 2014 5th International Conference on Intelligent Systems Design and Engineering Applications, ISDEA 2014* 182–85.

Lueninghoener, Cory. 1994. "Case-Based Reasoning : Foundational Issues , Methodological Variations , and System Approaches , as Presented by Praveen Guddeti." *AI Communications* 7(1):39–59.

MacDonell, S. G., Kyongho Min, and A. M. Connor. 2005. “Autonomous Requirements Specification Processing Using Natural Language Processing.” *International Society for Computers and Their Applications - 14th International Conference on Intelligent and Adaptive Systems and Software Engineering, IASSE 2005* 266–70.

Magee, Jeff and Jeff Kramer. 1996. “Dynamic Structure in Software Architectures.” *Proceedings of the ACM SIGSOFT Symposium on the Foundations of Software Engineering* (May):3–14.

Mall, R. 2018. “Fundamentals of Software Engineering.” *PHI Learning Pvt. Ltd.*

Manek, Patricia Gertrudis and Daniel Siahaan. 2019. “Noise Detection in Software Requirements Specification Document Using Spectral Clustering.” *JUTI: Jurnal Ilmiah Teknologi Informasi* 17(1):30.

Manning, Christopher D., John Bauer, Jenny Finkel, and Steven J. Bethard. 2014. “The Stanford CoreNLP Natural Language Processing Toolkit.” *Aclweb.Org* 55–60.

Mántaras, Ramon López D. E., David Mcsherry, David Leake, Barry Smyth, Susan Craw, Mary L. O. U. Maher, and Michael T. Cox. 2005. “Retrieval , Reuse , Revision , and Retention in Case- Based Reasoning.” *The Knowledge Engineering Review* 20(3):215–40.

Medeiros, Juliana, Miguel Goulao, Alexandre Vasconcelos, and Carla Silva. 2017. “Towards a Model about Quality of Software Requirements Specification in Agile Projects.” *Proceedings - 2016 10th International Conference on the Quality of Information and Communications Technology, QUATIC 2016* 236–41.

Mellado, Daniel, Eduardo Fernández-Medina, and Mario Piattini. 2010. “Security Requirements Engineering Framework for Software Product Lines.” *Information and Software Technology* 52(10):1094–1117.

Mesbahi, Nadjib, Okba Kazar, Saber Benharzallah, Merouane Zoubeidi, and Samir Bourekkache. 2015. “Multi-Agents Approach for Data Mining Based K-Means for Improving the Decision Process in the ERP Systems.” *International Journal of Decision Support System Technology* 7(2):1–14.

Mezghani, Manel, Juyeon Kang, and Florence Sèdes. 2018a. “A Clustering Approach for Detecting Defects in Technical Documents.” *13th International Workshop on Natural Language Processing and Cognitive Science* 2018(September):27–33.

Mezghani, Manel, Juyeon Kang, and Florence Sèdes. 2018b. “Using K-Means for Redundancy and Inconsistency Detection: Application to Industrial

Requirements." *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 10859 LNCS(June):501–8.

Mostafa, S. A., S. S. Gunasekaran, S. H. Khaleefah, A. Mustapha, M. A. Jubair, and M. H. Hassan. 2019. "A Fuzzy Case-Based Reasoning Model for Software Requirements Specifications Quality Assessment." *International Journal on Advanced Science, Engineering and Information Technology* 9(6).

Mostafa, Salama A. and Hajar Mat Jani. 2011. "Online Checklist-Based Approach to Software Requirements Specifications Quality Analysis." (February):21–22.

Mukherjee, Rajendrani and K. Sridhar Patnaik. 2018. "A Survey on Different Approaches for Software Test Case Prioritization." *Journal of King Saud University - Computer and Information Sciences*.

Naeem, Afrah, Zeeshan Aslam, and Munam Ali Shah. 2019. "Analyzing Quality of Software Requirements: A Comparison Study on NLP Tools." *2019 25th International Conference on Automation and Computing (ICAC)* (September):1–6.

Nordin, Azlin, Nurul Husna Ahmad Zaidi, and Noor Asheera Mazlan. 2017. "Measuring Software Requirements Specification Quality." *Journal of Telecommunication, Electronic and Computer Engineering* 9(3-5 Special Issue):123–28.

Ormandjieva, Olga, Ishrar Hussain, and Leila Kosseim. 2007. "Toward a Text Classification System for the Quality Assessment of Software Requirements Written in Natural Language." *SOQUA'07: Fourth International Workshop on Software Quality Assurance - In Conjunction with the 6th ESEC/FSE Joint Meeting* 39–45.

Parnas, David L. and Mark Lawford. 2003. "Inspection's Role in Software Quality Assurance." *IEEE Software* 20(4):16–20.

Parra, Eugenio, Christos Dimou, Juan Llorens, Valentín Moreno, and Anabel Fraga. 2015. "A Methodology for the Classification of Quality of Requirements Using Machine Learning Techniques." *Information and Software Technology* 67:180–95.

Rago, Alejandro, Claudia Marcos, and J. Andres Diaz-Pace. 2016. "Identifying Duplicate Functionality in Textual Use Cases by Aligning Semantic Actions." *Software and Systems Modeling* 15(2):579–603.

Rajabi, Mehdi, Morteza Saberi, Ofer Zwikael, and Elizabeth Chang. 2020. “Computers & Industrial Engineering Ambiguous Requirements : A Semi-Automated Approach to Identify and Clarify Ambiguity in Large-Scale Projects.” *Computers & Industrial Engineering* 149(January):106828.

Rashwan, Abderahman. 2015. “Automated Quality Assurance of Non-Functional Requirements for Testability.”

Reuss, Pascal, Klaus Dieter Althoff, Alexander Hundt, Wolfram Henkel, and Matthias Pfeiffer. 2015. “Multi-Agent Case-Based Diagnosis in the Aircraft Domain.” *CEUR Workshop Proceedings* 1520:43–52.

Rine, David C. and Anabel Fraga. 2015. “Chunking Complexity Measurement for Requirements Quality Knowledge Representation.” *Communications in Computer and Information Science* 454:245–59.

Ropponen, Janne and Kalle Lyytinen. 2000. “Components of Software Development Risk: How to Address Them? A Project Manager Survey.” *IEEE Transactions on Software Engineering* 26(2):98–112.

Rossanez, Anderson and Ariadne M. B. R. Carvalho. 2016. “Semi-Automatic Checklist Quality Assessment of Natural Language Requirements for Space Applications.” *Proceedings - 7th Latin-American Symposium on Dependable Computing, LADC 2016* 123–26.

Saavedra, Roxana, Luciana Ballejos, and Mariel Ale. 2013. “Software Requirements Quality Evaluation: State of the Art and Research Challenges.” 1850–2792.

Sabriye, Ali Olow Jim’ale, Wan Mohd Nazmee Wan Zainon. 2017. “A Framework For Detecting Ambiguity In Software Requirement Specification.” *2017 8th International Conference on Information Technology (ICIT)* 1431–33.

Sequeda, Juan F. 2007. “A Taxonomy of Verification and Validation of Software Requirement and Specifications.” 1–13.

Shiv Company. n.d. “IT International Solutions.” *Https://Shivitsolutions.Com/*.

Siahaan, Daniel and Izzatul Umami. 2012. “Natural Language Processing for Detecting Forward Reference in a Document.” *IPTEK The Journal for Technology and Science* 23(4):138–42.

Stellman, Andrew and Jennifer Greene. 2014. *Applied Software Project Management: Estimation*.

Stephen, E. and E. Mit. 2017. “Framework for Measuring the Quality of Software Specification.” *Journal of Telecommunication, Electronic and Computer*

Engineering 9(2–10):79–84.

Stokes, D. A. 1991. "Requirements Analysis." *In Software Engineer's Reference Book. Butterworth-Heinemann.* 1–16.

Thitisathienkul, Patra and Nakornthip Prompoon. 2014. "Quality Assessment Method for Software Development Process Document Based on Software Document Characteristics Metric." *2014 9th International Conference on Digital Information Management, ICDIM 2014* 5(4):182–88.

Thitisathienkul, Patra and Nakornthip Prompoon. 2015. "Quality Assessment Method for Software Requirements Specifications Based on Document Characteristics and Its Structure." *Proceedings - 2nd International Conference on Trustworthy Systems and Their Applications, TSA 2015* 51–60.

Tiwari, Saurabh and Atul Gupta. 2015. "A Systematic Literature Review of Use Case Specifications Research." *Information and Software Technology* 67:128–58.

Vaish, Nancy and Ashish Sharma. 2018. "Semi-Automated System Based Defect Detection in Software Requirements Specification Document." *2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering, UPCON 2018*.

Verma, Kunal, Alex Kass, and Reymonrod Vasquez. 2014. "Using Syntactic and Semantic Analyses to Improve the Quality of Requirements Documentation." *Semantic Web* 5(5):405–19.

Waldo, Whitson G. 2013. *Managing Embedded Software Development*.

Wang, Yongwei, Michael Lees, Wentong Cai, Suiping Zhou, and Malcolm Yoke Hean Low. 2009. "Cluster Based Partitioning for Agent-Based Crowd Simulations." *Proceedings - Winter Simulation Conference* 1047–58.

Wang, Yue, Irene L. Manotas Gutièrrez, Kristina Winbladh, and Hui Fang. 2013. "Automatic Detection of Ambiguous Terminology for Software Requirements." *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 7934 LNCS:25–37.

Wiegers, K., & Beatty, J. 2013. "Software Requirementse." *Pearson Education.*

Wilson, William M. 1999. "Writing Effective Natural Language Requirements Specifications." (February).

Wilson, William M., Linda H. Rosenberg, and Lawrence E. Hyatt. 1997. "Automated Analysis of Requirement Specifications." 161–71.

Yuen, Kevin Kam Fung. 2014. "A Hybrid Fuzzy Quality Function Deployment

Framework Using Cognitive Network Process and Aggregative Grading Clustering: An Application to Cloud Software Product Development." *Neurocomputing* 142:95–106.

Zakree, Mohd, Ahmad Nazri, Siti Mariyam Shamsudin, and Azuraliza Abu Bakar. 2008. "An Exploratory Study of the Malay Text Processing Tools in Ontology Learning." *ISDA '08 Proceedings of the 2008 Eighth International Conference on Intelligent Systems Design and Applications* 375–80.

VITA

Mohammed Ahmed Jubair was born in Iraq. He received a B.Sc. degree in Computer Engineering from Al-Marrif University, Iraq, in 2012. After the completion of the B.Sc. degree, he worked as a teaching assistant in Al-Marrif University from 2012-2014. In 2017, He received the M.Sc. degree from the Computer Science Networks Department, Universiti Kebangsaan Malaysia (UKM), Malaysia. He received a Ph.D. degree from the Faculty of Computer Science and information technology, Universiti Tun Hussein Onn Malaysia (UTHM), Malaysia in 2021. His research interests are in the area of Software Engineering, Artificial Intelligent, Communication Systems, Networking, and Machine Learning.