ENHANCING THE PERFORMANCE OF IR-BASED TRACEABILITY RECOVERY OF REQUIREMENT ARTIFACTS USING NOUN PHRASES

MASHAHI KHALAFALLA DAFAALLA ABDELRAHMAN

FACULTY OF COMPUTER SCIENCE AND IFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2020

ENHANCING THE PERFORMANCE OF IR-BASED TRACEABILITY RECOVERY OF REQUIREMENT ARTIFACTS USING NOUN PHRASES

MASHAHI KHALAFALLA DAFAALLA ABDELRAHMAN

DISSERTATION SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SOFTWARE ENGINEERING

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2020

UNIVERSITY OF MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: **MASHAHI KHALAFALLA DAFAALLA ABDELRAHMAN**

Matric No: **WGC140025**

Name of Degree: **Master of Software Engineering**

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):

Enhancing the Performance of IR-Based Traceability Recovery of Requirement Artifacts Using Noun Phrases

Field of Study: **Software Engineering**

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date:

Subscribed and solemnly declared before,

Witness's Signature Date:

Name:

Designation

ENHANCING THE PERFORMANCE OF IR-BASED TRACEABILITY RECOVERY OF REQUIREMENT ARTIFACTS USING NOUN PHRASES

ABSTRACT

Requirement traceability can be considered as a measure of software quality to help achieve validation, verification, and reusability. Neglecting traceability leads to less maintainable software. Creating traceability links after-the-fact, known as traceability recovery, is a tedious and time-consuming process when it is done manually. Therefore, information retrieval (IR) methods have been used to automatically identify traceability links between the artifacts. However, as a result of limitations of the software engineer and the IR techniques, the performance of the IR methods is negatively affected. There is no IR method that is able to recover traceability links between artifacts with high precision and high recall, such as in Vector Space Model (VSM), the retrieved false positives cause low precision results. Nevertheless, VSM is widely practiced as it considers the simplest linear algebraic method, easy to understand and use for non-IR experts. It allows ranking of documents concurring their probable relevance, and there are many tools and open-source implementations which implement VSM such as RETRO and ReqSimile. The research aims to assist software engineers (analysts) during the process of recovering traceability links between software artifacts by suggesting the appropriate type of phrases, which enhance the performance of IR method. The research objectives are: 1) To investigate IR methods for traceability recovery; 2) To propose a method that achieves high performance (as high recall and precision as possible) in traceability recovery; 3) To empirically validate the proposed method through an experimental analysis to demonstrate its ability to improve the performance (as high recall and precision as possible) in traceability recovery. A comparative experiment is done by extracting noun phrases (NP), verb phrases (VP), and combination of noun and verb phrases (NPVP) from

three benchmarking datasets namely CM1, MODIS, and PINE. VSM is applied, the result is evaluated in terms of recall and precision and the result showed that indexing NP only tends to outperform VP, NPVP, and all terms by achieving high recall and precision as possible.

Keywords: Traceability Recovery, Information Retrieval, Vector Space Model, Software Requirements, Noun Phrases.

PENINGKATAN PRESTASI PEMULIHAN KEBOLEH-KESANAN BERDASARKAN IR KE ATAS ARTIFACT KEPERLUAN DENGAN MENGGUNAKAN FRASA KATA NAMA

ABSTRAK

Keperluan keboleh-kesanan boleh dianggap sebagai ukuran kualiti perisian untuk membantu mencapai validasi, verifikasi, dan kebolehan mengguna semula. Mengabaikan keboleh-kesanan membawa kepada perisian yang tidak dapat dikekalkan. Mencipta pautan keboleh-kesanan selepas-fakta, dikenali sebagai pemulihan keboleh-kesanan, adalah proses yang rumit dan memakan masa apabila ia dilakukan secara manual. Oleh itu, kaedah pengambilan maklumat (IR) telah digunakan untuk mengenalpasti hubungan keboleh-kesanan secara automatik antara artifak. Walau bagaimanapun, akibat daripada batasan jurutera perisian dan teknik IR, prestasi kaedah IR adalah terjejas secara negative. Tidak ada kaedah IR yang dapat memulihkan hubungan keboleh-kesanan pautan antara artifak yang mempunyai ketepatan tinggi dan penarikan balik yang tinggi, seperti dalam Model Ruang Vektor (VSM), positif palsu yang diperolehi menyebabkan keputusan ketepatan yang rendah. Walau bagaimanapun, VSM diamalkan secara meluas kerana ia dianggap sebagai kaedah aljabar linear yang paling ringkas, mudah difahami dan digunakan untuk pakar bukan IR. Ia membolehkan mengatur kedudukan dokumen yang berkaitan, dan terdapat banyak alat dan pelaksanaan sumber terbuka yang melaksanakan VSM seperti RETRO dan ReqSimile. Penyelidikan ini bertujuan untuk membantu jurutera perisian (penganalisis) semasa proses memulihkan pautan keboleh-kesanan antara artifak perisian dengan mencadangkan jenis frasa yang sesuai untuk peningkatkan prestasi kaedah IR. Objektif penyelidikan adalah: 1) Menyiasat kaedah IR untuk pemulihan keboleh-kesanan; 2) Mencadangkan suatu kaedah dalam pemulihan keboleh-kesanan yang dapat mencapai prestasi yang tinggi (sebagai penarikan balik dan ketepatan yang tinggi); 3) Mengesahkan kaedah yang dicadangkan secara empiris melalui analisis

eksperimen yang dapat menunjukkan keupayaannya untuk peningkatkan prestasi (sebagai penarikan balik tinggi dan ketepatan yang jitu) dalam pemulihan keboleh-kesanan. Satu eksperimen perbandingan dilakukan dengan mengekstrak frasa kata nama (NP), frasa kata kerja (VP), dan gabungan frasa kata nama dan kata kerja (NPVP) daripada tiga dataset penanda aras iaitu CM1, MODIS, dan PINE. VSM diterapkan, keputusannya dinilai dari segi penarikan balik dan ketepatan jitu dan keputusannya menunjukkan bahawa pengindeksan NP hanya cenderung mengungguli VP, NPVP, dan semua terma dengan mencapai penarikan balik dan ketepatan yang tinggi .

Kata kunci: Pemulihan Keboleh-Kesanan, Pengambilan Maklumat, Model Ruang Vektor, Keperluan Perisian, Frasa Kata Nama.

ACKNOWLEDGEMENTS

By the Name of Allah, the Most Gracious and the Most Merciful. First of all, I would like to express my thanks and praise to Allah, the Most Merciful and, the Most Compassionate who gave me the willingness and the ability to carry out this study.

I would like to express my sincere gratitude and appreciation to my supervisor Prof. Dr. Lee Sai Peck for the continuous guidance and support during my master's study, also for her patience, motivation, enthusiasm, and immense knowledge that have contributed to the success of this research.

There are some people who made this journey easier with their prayers and encouragement. My deepest gratitude and special thanks to my parents, Khalafalla and Nadiah for their prayers, motivation, continuous support and endless love.

I would like to extend a special thanks and gratitude to my beloved husband Mutwakil who supports me towards this achievement and also my thanks go to my most beloved kids and brothers.

Sincere thanks to all my friends who have been along with me in this journey and for their great help and support.

My Appreciation to all members of Faculty of Computer Science and Information Technology (FSKTM) University of Malaya.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

IR	:	Information Retrieval
IE	:	Information Extraction
RTM	:	Requirements Traceability Matrix
VSM	:	Vector Space Model
LSI	:	Latent Semantic Indexing
PN		Probabilistic Network Model
SVD	:	Singular Value Decomposition
NL		Natural Language
NLP	:	Natural Language Processing
POS	:	Part-of-Speech
NP	:	Noun Phrase
PP	:	Preposition Phrase
VP	:	Verb Phrase
NPVP	:	Combination of Noun Phrases and Verb Phrases
MODIS	:	Moderate Resolution Imaging Spectroradiometer
PINE	:	Program for Internet News & Email
MDP	:	Metrics Data Program
GATE	:	General Architecture for Text Engineering

LIST OF APPENDICES

CHAPTER 1: INTRODUCTION

1.1 Background

Several information retrieval methods have been used to automatically recover traceability links between different types of software artifacts. Despite the advantages of information retrieval methods, any IR method will retrieve false positives, while it will also fail to retrieve some of the correct links (Kchaou et al., 2019). Many empirical studies proved that broadly applied IR-based methods are almost equivalent. Nevertheless, low precision persists the main drawback to using IR for traceability link recovery in practice, such as for VSM, it returns low precision when a high recall is achieved (Chen, Hosking, & Grundy, 2011; Cleland-Huang et al., 2014). This is due to the fact that many important and correct links are missed while unuseful incorrect links are retrieved (false positives). This limitation negatively affects the analyst's confidence in IR method, and the industrial practitioners to adopt IR-based traceability tools consequently.

1.2 Problem Statement and Motivation

Traceability recovery process becomes a tedious task when the performance of the IR-based method is low, and the analyst has to do much effort in evaluating candidate links to discard false positives (Capobianco et al., 2013).

False positive links could be retrieved due to noise in the textual document which IR methods rely on, this results in poor accuracy. False positives and poor accuracy have persuaded researchers to propose enhancement strategies to improve the performance of IR methods.

In the study of Capobianco et al. 2013, the authors proposed to use only nouns extracted from the software artifacts, as nouns provide more indication on the semantics of the document (Ali et al., 2019). Unfortunately, their approach neglects the functions of other words in the sentence (Ali et al., 2019; Wang, Xue, & Chu, 2016). However, considering only one part of speech fails to achieve optimal performance (Mahmoud &

Niu, 2015), although nouns carry more information value, it is not enough to obtain satisfied performance (Wang, Xue, & Chu, 2016).

Furthermore, the approach proposed by Capobianco et al. reduces the precision of retrieval result due to retrieving inappropriate documents. However, The use of phrases is founded to be more effective when explaining the substance of the text than the use of single words (Zou, Settimi, & Cleland-Huang, 2010). Moreover, part-of-speech tagging (POS tagging) method cannot ensure that all terms are correctly tagged since a term possibly will have many POS tags (Wang, Xue, & Chu, 2016).

The above limitations motivate the researcher to propose a method which utilizes phrasing with VSM, with the goal of enhancing the performance of IR-based traceability recovery.

1.3 Research Objectives

The research aims to assist software engineers (analysts) during the process of recovering traceability links between software artifacts by suggesting the appropriate type of phrases to improve the performance of IR method. The research objectives are:

1. To investigate IR methods for traceability recovery.
2. To propose an IR-based method that achieves high performance (as high recall and precision as possible) in traceability recovery.
3. To empirically validate the proposed method through an experimental analysis to demonstrate its ability to improve the performance (as high recall and precision as possible) in traceability recovery.

1.4 Research Questions

The main questions that will be answered by this research are:

1. What are the existing methods that have been used for traceability recovery?
2. Which IR-based method/ enhancement strategy can be used to leverage on VSM to achieve high performance in traceability recovery?

3. Is the proposed method able to improving the performance of IR-based traceability recovery method?

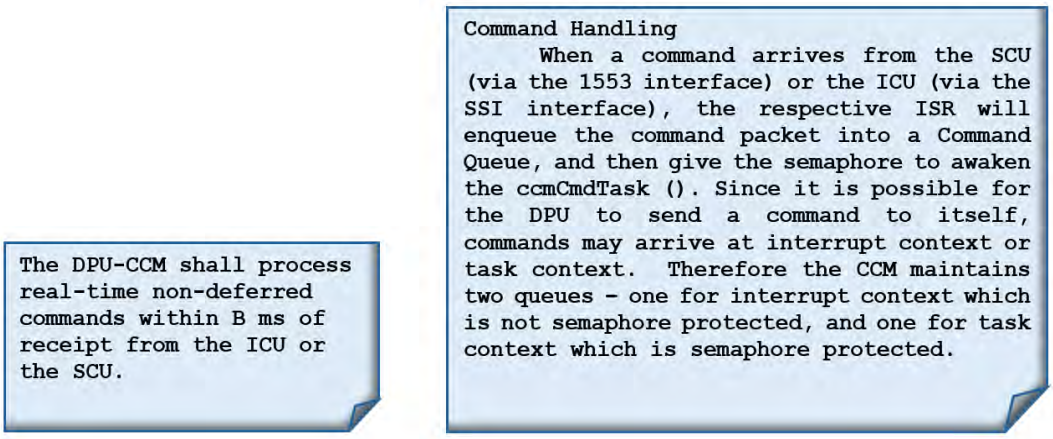
1.5 Research Scope

The scope of this research covers the following points:

a. Traceability links between requirements and use cases, high-level and low-level requirements: In literature, many case studies and experiments have been done to analyze the feasibility of the IR techniques and to improve their performance (Edyed, A. et al., 2010). These studies have already been implemented to various software artifacts including requirements, source code, external documents, etc. (De Lucia et al., 2012).

   This research focuses on recovering traceability links between requirements and use cases, high-level requirements and low-level requirements. Traceability links between requirements artifacts are considered as a measure of system quality and also to ensure validation and verification (Chikh, & Aldayel, 2012). Furthermore, requirements traceability is important in understanding and reducing development risks (Mahmood, Takahashi, & Alobaidi, 2015) such as product effectiveness, the ability to meet the goals, and quality of product. Particularly in the interdisciplinary industry such as software industry in which products are continually being made up-to-date as risks realization raises with experience and new products are built consistent with risk. Consequently, the traceability links among the requirements artifacts are so crucial to lowering the risk and making sure the success of products (Mahmood, Takahashi, & Alobaidi, 2015).

Figure 1.1 shows an example of high-level and low-level requirements obtained from CM1 NASA dataset.



(a) High-level requirement SRS5.12.2.2 **(b) Low-level requirement DPUSDS5.12.1.3.2**

Figure 1.1: Example of high-level and low-level requirements obtained from CM1

b. VSM will be used in this research. It is the simplest linear algebraic method, easy to understand and use for non-IR experts. It allows ranking of documents concurring their probable relevance, and there are many tools and open-source implementations which implement VSM such as RETRO and ReqSimile (Zou, Settimi, & Cleland-Huang, 2010; Brodén, 2011; Al-Saati, & Abdul-Jaleel, 2015; Nyamisa, Mwangi, & Cheruiyot, 2017; Panichella et al., 2016).

VSM has been utilized in previous research to create trace links between requirements, requirement and source code, manual page and source code, UML diagram and source code, test cases and source code, and defect reports and source code (De Lucia et al., 2012).

TraceLab[1] is a framework designed to support creating experiments traceability using visual modeling environment, with both existing and user-defined executable components, in the field of traceability recovery, as well as other

[1] Funded by the National Science Foundation and is developed at DePaul University with collaborating partners at Kent State University, University of Kentucky, and the College of William and Mary.
https://github.com/CoEST/TraceLab-CDK

software engineering tasks (Dit, Moritz, & Poshyvanyk, 2012; Keenan. et al. 2012; Cleland-Huang, Czauderna, & Hayes, 2013). Due to its stability, scalability, portability, high performance, quality, and easy to install and use for new users (Cleland-Huang, Gotel, & Zisman, 2012), TraceLab is used in this research to apply VSM on selected artifacts.

c. Phrasing: many enhancement strategies have been introduced in literature to improve the performance of IR methods such as thesaurus, glossary and relevance feedback. In this research, phrasing will be used to augment VSM. Phrasing generates high value in differentiating between true and false positives links (Mahmoud & Niu, 2015), it can help to get a more accurate description of document's content than single words. Using single words usually retrieves unrelated documents while phrases can give a stronger sign that two artifacts should be linked. Then false positives can be reduced, and therefore, the precision of retrieval results is improved (Zou, Settimi, & Cleland-Huang, 2010).

d. Part of Speech Tagger and Chunker: POS tagging will be used to assign part of speech to each word in the document. As a term possibly will have many POS tags, a chunker will be performed to guarantee that all terms are correctly tagged. Furthermore, to extract noun and verb phrases. GATE[2] as a framework for text processing that contains various plugin natural language processing (NLP) tools (Cunningham, et al. 2013; Cunningham, et al. 2014) is used. Part of speech tagger (POS tagger), chunker, and other preprocessing tools from GATE are used in this research to extract the phrases, in addition to C++ program which is used in this research to put noun phrases and verb phrases into separated folders.

[2] https://gate.ac.uk/

1.6 Research Significance

In the literature, many techniques have been proposed to improve the accuracy of IR-based traceability methods, and to reduce errors and time consumed by traceability recovery process, such as extracting terms from the artifacts, preprocessing, domain-specific terms, project glossary, thesaurus, and smoothing filters. These techniques have been used to improve the precision of IR-based traceability recovery methods. Using such techniques can help to augment IR methods, thus provide a need to investigate further. This works as a motivation for the researcher to contribute in enhancing and improving the result achieved by standard VSM.

This research introduces a comparative experiment, to suggest an effective phrase type that can be used to enhance the performance (as high recall and precision as possible) in traceability recovery. The result of this experiment can help to create traceability links between requirement artifacts with high precision, as requirements are very important to ensure developing successful software products, trace the influence of changes to the requirements, and lower the risk for products that need to be updated and modified constantly such as safety-critical systems. Furthermore, it will assist the software engineers (analysts) in the process of traceability links creation by improving the quality of retrieved links. This will reduce time and efforts consumed by the analyst to filter out unwanted links, since time consumption is also an important point that always takes a major concern (Al-Saati & Abdul-Jaleel, 2015; Sundaram, 2010).

As a consequence, augmenting the IR method and enhancing its result will make the analyst more confident in the IR methods. This will also encourage the industrial practitioners to adopt IR-based traceability tools.

1.7 Thesis Outline

This thesis is organized into six chapters which are as follows: Chapter 1 provides the background of the study, problem statement and motivation, research objectives, research

questions, research scope and research significance. Chapter 2 presents the literature review in the area of requirements traceability, information retrieval methods, traceability recovery, IR-based traceability recovery process, enhancement strategies for IR-based traceability, performance metrics for IR-Based traceability recovery and tradeoff between recall and precision. Chapter 3 discribes the methodology followed in this research to achieve research objectives, the proposed method enhance IR-based traceability recovery, planning for the comparative experiment and datasets for the experiment. Chapter 4 presents details of implementing the proposed method to enhance the performance of IR-based traceability recovery. Chapter 5 describes the results of the experiment and result discussion and validation. Chapter 6 discusses the fulfilment of research objectives, strengthes and contribution, limitations and future work.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

In this chapter, the related literature is reviewed in order to determine and identify the background of the study and research gap. In addition, to propose a clear understanding of research area such as requirement traceability, its benefits and challenges, information retrieval methods, their categories, advantages and limitations, as well as enhancing the performance of IR-based traceability recovery. Various sources have been reviewed to come up with the proposed method to enhance the performance of IR-based traceability recovery of requirements artifacts.

2.2 Requirement Traceability

Requirement Traceability is defined by Gotel and Finkelstein as follows: "*the ability to describe and follow the life of a requirement, in both forward and backward directions (i. e., from its origin to its subsequent deployment and use, and through all periods of on-going refinement and iteration in any of these phases"* (Gotel & Finkelstein, 1994).

2.2.1 The Significance and Benefits of Traceability

Requirements traceability is considered as a reason behind the software engineering efficiency (Borg, 2016). It recognized as an attribute of system quality in software development that helps to achieve various system quality aspects such as adequacy, understandability, and maintainability (Chikh, & Aldayel, 2012). While ignoring traceability causes less maintainable software due to inconsistencies and omissions (Winkler, & Pilgrim, 2010). Some of the benefits and advantages of establishing and using traceability are explained in the following points:

a. Traceability supports various software engineering activities associated with developing software such as verification and validation, change management, impact analyses which helps developers to recognize how a proposed change

impacts the current system, and regression testing (Cleland-Huang, Gotel, & Zisman, 2012).

b. Traceability is able to help the software engineer in many tasks such as system comprehension, knowledge transfer, and process alignment (Borg, 2016).

c. Traceability supports numerous critical activities. Utilizing pre-requirements traceability to confirm that a product meets the stakeholders' requirements or it fulfills a government rules, is an example of these activities (Cleland-Huang, Gotel, & Zisman, 2012).

d. Traceability is used to create and understand the relations between requirements and low-level products such as design, source code, and test cases.

e. Traceability is very useful when the artifacts are being reused. It helps to recognize parts for the new requirements and the development of software systems (Cleland-Huang, Gotel, & Zisman, 2012; Brodén, 2011).

Traceability can be established through the following ways:

1. Manually: where the creation and maintenance of traceability links done by a human (Cleland-Huang, Gotel, & Zisman, 2012). However, the manual tracing requires a lot of labor, tedious task, time-consuming, and error-prone. As a result, successful traceability is hardly developed and followed in practice, as practitioners often fail to perform an effective manual tracing process. (Ali et al,. 2019).

2. Semi-automatic tracing using both automated tools and human activities.

3. Automatically using automated techniques, methods and tools (Cleland-Huang, Gotel, & Zisman, 2012).

2.2.2 Traceability Issues

Achieving successful traceability in practice faced by numerous issues that make it difficult. Many organizations and companies fail to achieve proper traceability documentation in their software projects. The study of Saiedian, Kannenberg and Morozov (2013) addressed that organizations find difficulties in both understanding the principles of traceability and practicing traceability in the software development life cycle. These issues are classified as:

- Social issues which are related to communication between project stakeholders. (Cleland-Huang, Gotel & Zisman, 2012).
- Technical issues which are relevant to the creation, maintenance, and utilization a lot of traceability links (Cleland-Huang, Gotel & Zisman, 2012)

In many instance, the non-developers have to create the requirements traceability matrix (RTM), which is the main component for traceability analysis, after-the-fact. This is because documentation of traceability throughout the progress of the development has been neglected, incomplete or not detailed enough. This task is very tiresome to conduct manually, error prone and time consuming even though requirement specifications often consist of a significant number of requirements (Brodén, 2011).

2.2.3 Traceability Creation

Traceability creation is the process of linking two artifacts or more than two attifacts, by establishing trace links between them, for the purposes of tracing. This process can be done manually, semi-automatically or automatically. So there are different approaches and techniques that promote the develeopment of trace links and also different level of efficiency and effectiveness. Despite there are many ways to create trace links, validation is critical, as it is concerned with determining the viability

of traceability creation process and ensuring the reliability of the whole trace (Cleland-Huang, Gotel & Zisman, 2012).

Cleland-Huang, Gotel & Zisman (2012), determined the Generic traceability process model which illustrates the crucial activities that help to establish, use, and maintain traceability links with explanation of the inputs, outputs, and necessary resources. Figure 2.1 shows the generic traceability process model.

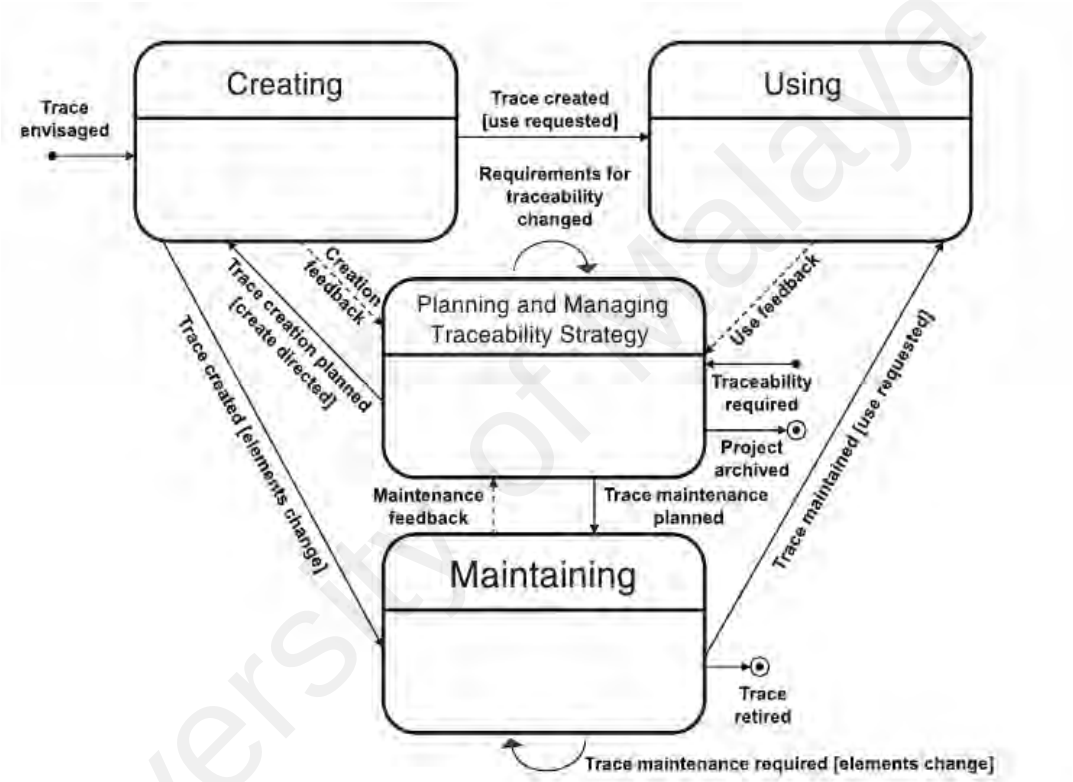


Figure 2.1 A generic traceability process model

(Cleland-Huang, Gotel, & Zisman, 2012)

Trace Elements is classified as trace artifacts and trace links. Trace artifact is a unit of data that can be trace. It can be defined as either a source artifact or as a target artifact (e.g., high-level requirements, low-level requirements, source code, etc.) (Cleland-Huang, Gotel, & Zisman, 2012).

Trace link is a specified association between the source artifact and the target artifact that may or may not include information about link type or other semantic

attributes. The direction of trace link can be a primary trace link direction, reverse trace link direction, or bidirectional trace link (Cleland-Huang, Gotel & Zisman, 2012). Figure 2.2 shows the elements of trace.

Figure 2.2 Elements of trace

(Cleland-Huang, Gotel & Zisman, 2012)

In the literature, there are many approaches and methods developed to support traceability that are based on information retrieval, reference model and rule-based approach (Chikh, & Aldayel, 2012).

2.3 Information Retrieval Methods

Information Retrieval (IR) is defined as "finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)" (Manning et al. 2008). Information retrieval (IR) initiated in the 1950s to reduce information overload by providing only the information that related to user's query. This approach generates a list of corresponding documents for a specific query from a set of documents, by selecting key terms from each document in the corpus, and then measuring the similarities between the query terms and the key terms from each document.

2.3.1 IR Categories

IR techniques are divided into three categories of models (Brodén, 2011):

a. Set-theoretical models: In this type of models, a sets of words are used to interpret the documents and similarity is measured based on set-theoretical operations and methods.

b. Algebraic models: for this context, models use vectors or matrices to represent documents and queries. The most widely used model in practice is Vector Space Model (VSM). It represent documents and queries in multi-dimensional space and assign weight to each term using Term Frequency- Inverse Document Frequency (TF-IDF). The similarity can be calculated using distance functions.

   Latent Semantic Indexing (LSI) is a further development of the basics in VSM. It is able to handle the effects of polysemy and synonyms. Both LSI model and VSM are based on the term-document matrix method. LSI uses Singular Value Decomposition (SVD) to reduce the dimension space and make the matrix more manageable. While the execution of SVD, the parameter *k* has to be defined to identify the right size of the new matrix. However, defining the right size for the matrix is exhaustive task and required high computational cost to reduce the noise and include important information. (Brodén, 2011; Borg, 2016).

c. Probabilistic models: This type of models deal with the process of retrieval as a probabilistic implication and measure the similarity as the probability that a document is appropriate for a specified query. By having assumption that the probability of the outcome rely on document representation and the given query (Brodén, 2011). The widely practiced models from this category are the Probabilistic Inference Networks and Binary Independence retrieval Model (BIM) (Brodén, 2011).

Figure 2.3 shows the categories of IR models for traceability recovery.

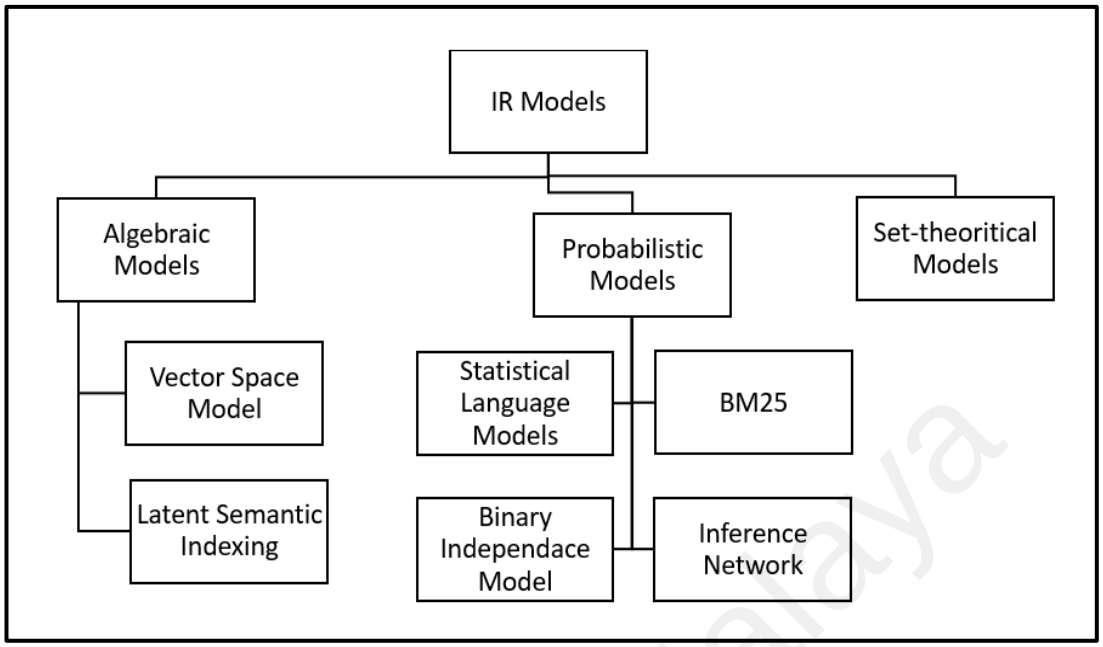


Figure 2.3 Taxonomy of IR models in traceability recovery

The study of Borg, Runeson & Ardö (2014) indicated that Algebraic Models is the most frequently applied models during the last decade, particularly VSM. As many comparing research demonstrated that VSM obtained the best results (Ali et al., 2019).

VSM is widely practiced. The reason behind this interest is that it considers the simplest linear algebraic method, automatically extract information from the corpus (Zou, Settimi, & Cleland-Huang, 2006), easy to understand and use for non-IR experts. It allows ranking of documents concurring their probable relevance, and there are many tools and open-source implementations which implement VSM such as RETRO and ReqSimile (Zou, Settimi, & Cleland-Huang, 2010; Al-Saati, & Abdul-Jaleel, 2015; Nyamisa, Mwangi, & Cheruiyot, 2017; Panichella et al., 2016; Brodén, 2011).

Although LSI is developed after VSM and has several advantages, it does not constantly surpass VSM in traceability recovery applications, and has many drawbacks, for instance, performing the mathematical technique singular value decomposition (SVD) that requires high computational costs. In addition, defining the

optimal dimension *k* is an open question and can be computationally exhaustive (Zou, Settimi, & Cleland-Huang, 2010; Mahmoud & Niu, 2015; Brodén, 2011). Furthermore, Zou, Settimi, & Cleland-Huang (2010) indicated that LSI usually does better on applications with a large number of artifacts and textually rich datasets to avoid losing important information during performing dimension reduction.

For all the above reasons, VSM over LSI is selected to be used in this research.

2.3.2 Vector Space Model (VSM)

VSM is developed in the 1960. Queries and documents in VSM are represented as vectors that described in the space of all terms $T = \{t_1, t_2, \ldots, t_n\}$ these terms have been gathered from the set documents.

Document *d* is demonstrated as a vector $\vec{d} = (w_{1,d}, w_{2,d}, \ldots, w_{n,d})$ where $w_{i,d}$ is the term weight that assigned to the term *i* where *n* is the total number of terms in the term space. All document's vector are then represent together into term-document matrix. The query *q is* represented as a vector $\vec{q} = (w_{1,q}, w_{2,q}, \ldots, w_{n,q})$.

The weight reveals the importance of the term in representing the query or the document content. The weighting scheme *tf-idf*, where *tf* is for term frequency and *idf* is for inverse document frequency, this scheme is used to calculate the term weight. The *tf* of a term t_i in a document *d* is usually calculated as $tf(t_i, d) = \frac{freq(t_i,d)}{|d|}$ where $freq(t_i,d)$ represents the frequency of the term in the specific document and the length of the document is utilize for normlization . The *idf* of a term t_i, idf_{ti}, can be measured as $\log_2(\frac{n}{ni})$ (i.e. $idf_{ti} = \log_2(\frac{n}{ni})$) , where *n* represents the total number of the documents and n_i is stand for the number of documents where t_i occurs. $w_{i,d}$ is stand for term weight related with the term *i* in the document *d*; then it can be calculated as $w_{id} = tf(t_i, d) \times idf_{ti}$. When the term repeted many times in the specific document and is enclosed in a few documents then it is considered relevant for representing a document's content and the weight assigned to this term will be high. The similarity score *sim(d,q)* is

usually used to compute the relevance between the query q and the document , it can be defined as the cosine of the angle formed by their corresponding vectors, and is computed as: $sim(d,q) = \frac{\sum w_{i,d} \times w_{i,q}}{\sqrt{\sum w_{i,d}^2 \times \sum w_{i,q}^2}}$ (Zou, Settimi, & Cleland-Huang, 2010, Cheruiyot, 2017; Panichella et al., 2016). Figure 2.4 represents an example of three vectors for three documents.

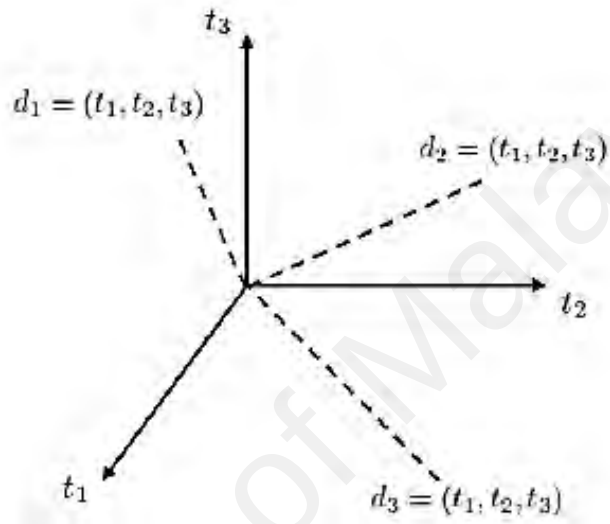


Figure 2.4 An example of three vectors for three documents

2.4 Traceability Recovery

Traceability recovery is defined as follows: "trace recovery is an approach to create trace links after the artifacts that they associate have been generated or manipulated" (Borg, Runeson, & Ardö, 2014).

Creating traceability links after-the-fact manually is a tedious and time-consuming procedure, on account of that most of the artifacts are usually written in natural language. The main challenges that faces traceability recovery are: (Cleland-Huang, Gotel, & Zisman, 2012).

- The artifacts are represented in different formats and at various abstraction levels
- The semantics of the links is understood in a different way by several people.

- The format of the data for software engineering artifacts is not defined, so the approaches such as database and data analysis centered are considered impractical.

However, the textual data is the one type of data that present in all software artifacts. Most of the traceability tools are now based on extracting and analyzing the textual data due to many artifacts are containing textual parts that describe the semantics of the artifacts.

So when the textual part of two artifacts refers to similar concepts, this means that the two artifacts are relevance and could create a traceability link between them. (Cleland-Huang, Gotel, & Zisman, 2012)

To extract and analyze this textual data from the software artifacts, the IR techniques had been adopted by the researcher, due to the fact that IR-based methods use the similarity score between the texts from the software artifacts to recover traceability links between those artifacts (Diaz et al., 2013).

2.5 IR-Based Traceability Recovery Process

Generally, to recover traceability links between source and target artifacts manually, this process starts by reading through all of the documents in the artifacts, compare them against each other to find if there a relevance between them and they influence each other. By the end of this step in the traceability, the candidate link lists would be generated.

In the next step, the analyst evaluates and examines the generated candidate links carefully, to assign the value “link” for the correct link or “no link” for the false-positive links.

It is noticeable that IR can prompt the traceability process by generating the candidate link lists automatically, without needing an analyst to go through all of the documents over and over again manually (Brodén, 2011). Therefore, various IR methods have been adapted in different tools to recover traceability links between software artifacts.

Figure 2.5 demonstrates the IR-based traceability recovery process from the study of Bavota et al. (2014). At first, the artifacts indexing process starts by extracting words from the artifact's content. To index the artifacts and construct a term-by-document matrix, a text normalization phase is performed to remove out white spaces and non-textual tokens followed by splitting source code identifiers. Also in the artifact indexing process, a stop word function and/or a stop word list are utilized to reject common words that are not valuable. After extracting the required terms, a morphological analysis can also be performed to return each word to its origin. By the end of the indexing process, the term-by-document matrix will be ready. Then any IR method can be used to compare the set of source artifacts against the set of target artifacts to compute and rank the similarity of all potential pairs of artifacts (Bavota et al., 2014).

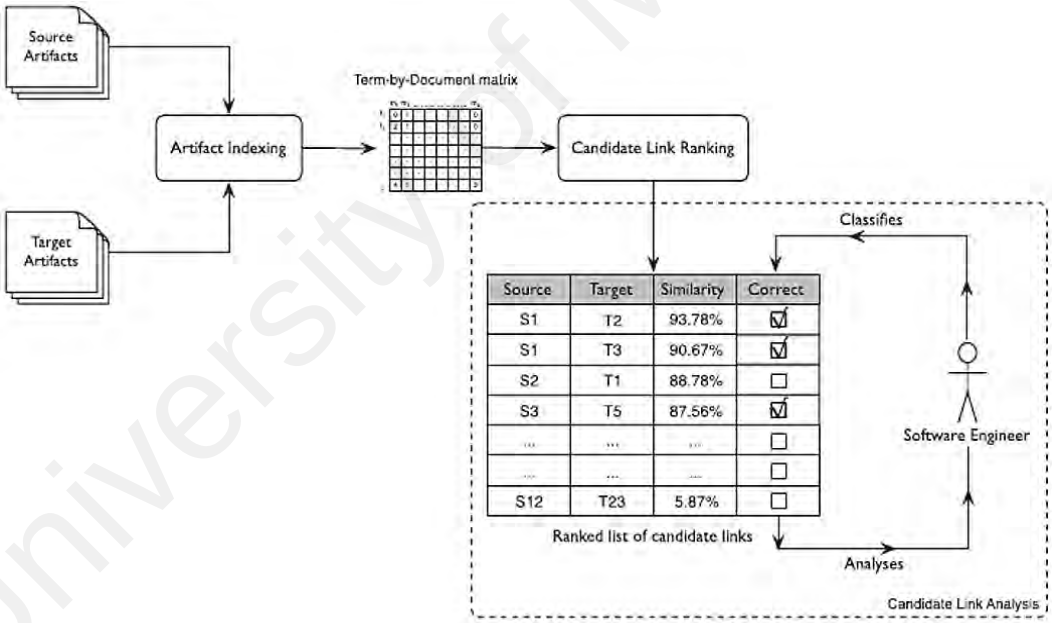


Figure 2.5 IR-based traceability recovery process

Since the IR techniques rely on textual information to recover trace links, this textual information may include noise; so IR techniques could recover unwanted links that lead to poor accuracy (Ali et al., 2019).

Unfortunately, any IR method fails to retrieve some of the correct links, at the same time it may retrieve false positives. this is due to either the IR methods or the software

engineer. Consequently, enhancing the performance of traceability recovery methods has turned into one of the primary challenges in traceability management. However, it is able to be achieved by reducing the number of false positives. Too many false positives retrieved makes the traceability recovery procedure a tiresome mission, because the software engineer has to spend much time to discard false positives rather than to trace correct links (Capobianco et al., 2013). Phrasing will be used in this research to overcome this problem, as this technique generates high value in differentiating between true and false positives links (Mahmoud & Niu, 2015).

2.6 Enhancement Strategies for IR-based Traceability

Prior research introduces many enhancement strategies that have been used to improve the performance of IR based traceability recovery methods. Each strategy proved its effectiveness in improving the retrieval results somehow. However, every strategy has its own drawbacks and restrictions, firtheremore, it may be only valid to specific datasets. Thesaurus, glossary and clustering are described in this section.

2.6.1 Thesaurus

Thesaurus is a method that is frequently utilized in the fields of IR in order to minimize the number of term mismatches. It has a simple form which is a set of triples (t, t', α) where t and t' are stand for thesaurus terms and α is the similarity coefficient between them. An example of a simple thesaurus triple could be (“car”, “vehicle”, 0.8). The tf-idf weighting method had been extended by this method in many research (Pinheiro, 2004).

Despite its usefulness, general thesaurus has many drawbacks such as introducing noises and does not improve classifications (Mahmoud & Niu, 2015; Rago, Marcos, & Diaz-Pace, 2017), cannot handle abbreviations and acronyms. In addition, it is exhausting and time consuming to build a thesaurus because it requires extensive knowledge on the specific project (Zou, Settimi, & Cleland-Huang, 2010).

Domain-specific thesaurus has a high performance in comparison with general thesaurus but it can quickly become out-of-date because it needs to keep tracking the vocabulary whenever changes happen in the project (Mahmoud & Niu, 2015).

2.6.2 Clustering

Clustering relies upon finding that right links appear to exist within a hierarchy of physical clusters. Such clusters can be considered as a logical collections of software. Clustering strategy hypothesizes that when a link is identified between the query and the document which is referred to a logical cluster, other documents within that cluster are probably related to that query. The conception is called *document side clustering*. In similar way, *query side clustering* can be applied also to individual documents and clusters of queries when a link between a document and one of the queries in the same cluster is identified (Zou, Settimi, & Cleland-Huang, 2010).

2.6.3 Glossary

Every project glossary captures terms and phrases which can be regarded as more expressive for identifying relations between documents comparing to general terms. This is due to that it can describe the specific meaning of the project. It is considered as a new enhancement technique, since the information in the project glossary can be utilized to increase term weight and phrases weight that included in the project glossary. This strategy increases the relevance ranking of documents that inclosing glossary items and subsequently improves the precision. However, this strategy will not be effective if the project comes without a glossary or the existing glossary is not being consistently followed in the software development (Zou, Settimi, & Cleland-Huang, 2010).

2.6.4 A comparison of different enhancement strategies

Figure 2.2 shows the number of times an enhancement strategy has been reported in publications as general and the black bars presents the reporting in publications

which apply algebraic models (Borg, Runeson, & Ardö, 2014). The most frequently applied enhancement strategy is relevance feedback, followed by thesaurus, clustering results based on document structure to improve demonstration of the recovered links. Relevance feedback is implemented by allowing the analyst to analyze trace retrieval outcome from the tracing tool and re-execute an enhanced search query (Borg, Runeson, & Ardö 2014). Although this strategy achieves an improvements, it has drawbacks such as the additional work required from the analyst for enhancing results (Zou, Settimi, & Cleland-Huang 2010). Since this research aims to assist the analyst, phrasing is used in this research.. Phrasing is described as a sequence of two words or more, it had been used by Zou et al., (2006) and Chenand Grundy, (2011).Other enhancement strategies such as up-weighting terms according to their existence in the project glossary, are repetitively used. In addition, machine learning approaches, query expansion, analyses of call graphs, regular expressions, and smoothing filters had been also applied in the literature (Borg, Runeson & Ardö, 2014).

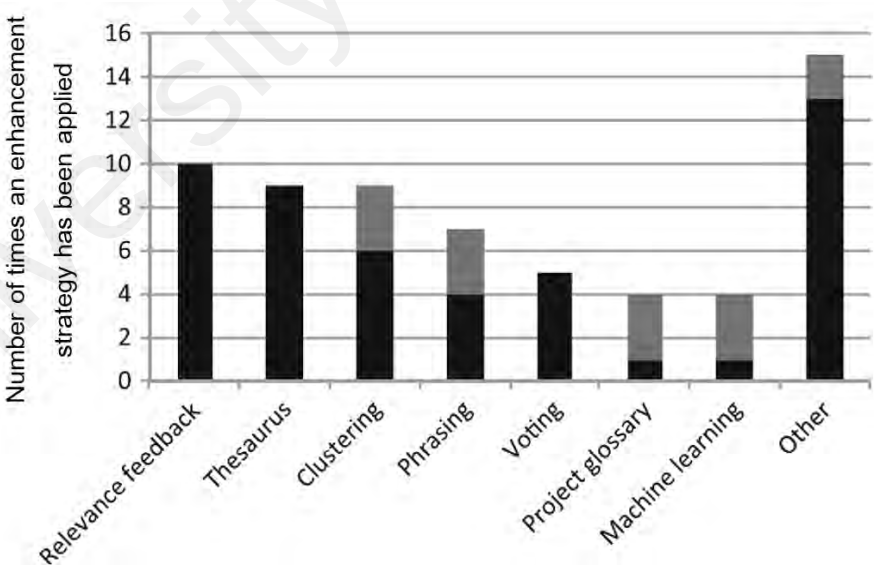


Figure 2.6 Enhancement strategies for IR-based trace recovery

(Borg, Runeson, & Ardö, 2014)

2.7 Natural Language Processing

Natural language processing techniques (NLP) have been used in several software engineering tasks. Liddy (2001) defined the natural language (NL) and NLP as follows: “NL text is text written in a language used by humans to communicate to one another”, and “NLP is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications” (Borg, Runeson, & Ardö, 2014; Arunthavanathan et al., 2016). There are several levels that have been used to extract meaning from textual documents or spoken languages such as mentioned in the study of Chowdhury (2001) and Liddy (2001):

a. Phonology level

This level deals with the analysis of speech sounds in the words and across it, where the spoken input is then analyzed and encoded into a digitized signal to be interpreted by several rules or by comparison to the particular language model being used.

b. Morphology level

The morphology is the study of words. The component of word is treated within this level, which is comprised of small units of meaning called morphemes. As an example, a word can be analyzed into prefix, root, and suffix. The meaning can be recognized and carried by each morpheme in order to get the meaning and represent it.

c. Lexical level

This level deals with the word's lexical meaning. The meaning of individual words is perceived by both humans and NLP systems, POS tagging process is used in this level to assign a single part-of-speech tag to each word. This level may require

a lexicon, which may be simple, or complex and this will be determined by the NLP system. As well as, in case the words have only one possible meaning, the semantic representation used for replacing that meaning.

d. Syntactic level

It reveals the grammatical structure of the sentence by analyzing the words in a sentence. Therefore, grammar and parser are both essential requirements. The result is a representation of the sentence which exposes the relationships of structural dependency between the words. Some NLP applications require a full parser while others require partial parse of sentences. In most languages, Syntax conveys meaning because of both order and dependency contribute to meaning.

e. Semantic level

This level defines the potential meanings of a sentence. It concerns the relation between word-level meanings in the sentence.

This level of processing can include the semantic disambiguation of words with multiple senses by selecting only one sense of polysemy words to be involved in the semantic representation of the sentence. Various methods can be performed to achieve the disambiguation such as sense frequency, local context, and using pragmatic knowledge.

f. Discourse level

Syntax and semantics levels deal with units of sentence-length, but discourse level of NLP deals with units of text longer than a sentence. It focuses on the properties of the whole text to convey meaning by connecting the component of the sentences. Discourse level has many types of processing that can occur, such as anaphora resolution and discourse/text structure recognition.

g. Pragmatic level

It studies how the context of the text can contribute to understanding additional meaning. This level needs considerable world knowledge. So several NLP applications may apply knowledge bases and inferencing modules.

There are various applications that utilize NLP, such as Information Retrieval (IR), Information Extraction (IE), Question-Answering, Summarization, Machine Translation, and Dialogue Systems.

2.7.1 Part-of-Speech Tagging

POS tagger analyzes the textual documents and assigns tags to all terms in the corpora according to the context and grammar of a sentence (Ali et al., 2019).

In the literature, POS tagging has been used to augment and improve many software engineering tasks. It is used by Abebe and Tonella (2010) to extract domain concepts and relations from program identifiers to create an ontology which is then used to improve concept location. Etzkorn et al. (1999) also utilized POS tagging to generate a software module summarization (Capobianco et al., 2013).

In order to improve the accuracy of IR methods, efforts had been done to eliminate false positives links. One of these techniques is using part-of-speech tagging to extract and index specific terms such as nouns or verbs and reject the other, or assign a different term weight. As this technique can help to remove unwanted terms. As a result the noise will be reduced and the accuracy will improve (Panichella, De Lucia, & Zaidman, 2015; Brodén, 2011; & Ali et al., 2019). In the study of Capobianco et al. (2013), they indexed only nouns from the software artifacts to improve the accuracy of IR traceability recovery, their approach presented improvement in some cases while it also indicated worse result than the base line in other cases (Ali et al., 2019). Nouns and verbs have a significant role in describing the semantics of a software artifact (Wang, Xue, & Chu, 2016; Borg, Runeson, & Ardö, 2014), they have been used in the study of Zhao et al. (2003) and Zhou, & Yu (2007). Moreover, in the study of

Mahmoud, & Niu (2010), they demonstrated that verbs describe the functionality of the software. According to this, they increased the weight of verbs in the TFIDF weighting scheme. They investigated the role of semantic in improving IR-based traceability performance. The results demonstrated that indexing only nouns attains a significantly higher recall than indexing verbs only, but there is still a negative effect on recall. At the same time, this approach is considered useful if precision is preferred over recall. Ali et al. (2019) demonstrated that using only adjectives optains a lower accuracy than using other terms.

In the case of extracting nouns ony, Capobianco et al. (2013) extracted only nouns to eliminate the noise from the software artifacts. Their results attained lower accuracy in some cases than an IR-based technique (Ali et al., 2019). However, in the study of Ali et al. (2019), considering noun-based indexing improved the accuracy in 21% of the cases, while in 79% of the cases noun-based indexing provides lower accuracy than the baseline.

Ali et al. (2019) addressed that using only verbs achieves an extreme decrease in precision and recall. This is due to that losing some of the semantic information results in poor accuracy. They addressed that using a combination of nouns and verbs gives better results than the baseline approach in 58% of the cases.

2.7.2 Phrasing

Phrasing is an approach to perform indexing according to the Bag-of-Word model (Croft et al., 1991). A phrase is defined as a sequence of two or more words which are supposed to represent the document content more accurately than single words.

Some of IR models presented the documents using single terms, such as VSM and Probabilistic Network (PN) models. Nevertheless, sometimes using single terms cannot describe the document content perfectly because a single word may have a wide range of concepts, and this may lead to low precision due to irrelevant documents

retrieved. For this reason, phrases approach is condered as an effective approach for explaning the substance of the text (Zou, Settimi, & Cleland-Huang, 2010).

In the literature, Zou et al. (2010) included the use of extracting phrases from requirements using POS tagger to dynamically trace requirements using PN model. Their approach depends the glossary of the project to discover more phrases and weight the contribution of key phrases and terms (Ali et al., 2019). To improve the lexical component in the hybrid clustering, Thijs, Glänzel, & Meyer (2015) used syntactic parsing to extract nouns and noun phrases only from abstracts and titles, as it represent subjects, objects, predicative expressions or prepositions in sentences.

2.7.2.1 Phrase Detection and Extraction Methods

Strategies for identifying phrases are defined as a syntactical method and statistical method. Statistical methods utilize frequency and co-occurrence of terms to identify phrases. The maximum length of a phrase, the proximity of the occurrence of the phrase components, and document frequency threshold are considered as key parameters that need to be defined in the statistical method. Thus, The retrieval accuracy could be affected by defining these parameters greatly, as well as each individual documet corpus needs to define the optimized parameters. The quality of the detected phrases is another concern for statistical methods, as it is often not satisfactory due to the construction of inappropriate phrases. In addition, some good phrases are missed (Zou, Settimi, & Cleland-Huang, 2010). Syntactical methods is considered more accurate in identifying phrases because it is construct phrases using grammatical structure and relationships between words in a sentence (Zou, Settimi, & Cleland-Huang, 2010).

2.7.2.2 Chunking

It is called shallow parsing or partial parsing. It is one of the NLP technique that divides the text into chunks or groups of words that compose a grammatical unit, such

as noun phrase (NP), verb phrase (VP), or preposition phrase (PP) (Kang, Mulligen, & Kors, 2011). To perform the chunking process, there are two approaches, a rule-based approach, and statistical approach. In the rule-based approach, the chunker is composed of a set of regular expression statements. Using rule-based approach makes systems easier to be developed because do not need a training corpus. However, it is difficulte to be adapted in new domains.

On the other hand, statistical approach using statistical machine learning methods makes reusing the statistical systems in a new domian easier , therefore, it requires a large training corpus (Kang, Mulligen, & Kors, 2011; Wang, Xue, & Chu, 2016).

Pipeline of text processing components is used in natural language processing (NLP) applications to extract information from the textual documents.the performance of each sub-component in the pipline effects the performance of the NLP applications. Chunking is based on the sentence annotation, token annotation, and part-of-speech (POS) (Kang, Mulligen, & Kors, 2011). The relationship between the chunks can reduce the errors which might be produced during POS tagging (Wang, Xue, & Chu, 2016).

Kang et al. (2011) assessed the performance and usability of six chunkers (GATE chunker, Genia Tagger, Lingpipe, MetaMap, OpenNLP, and Yamcha) in terms of noun phrase and verb phrases chunking. They found that OpenNLP achieves the best performance in NP and VP chunking. Regarding the usability, they found that Lingpipe and OpenNLP get the best score. Due to this fact, OpenNLP tools are used in this research. Both GATE which is a framework for text processing that contains many NLP tools and OpenNLP is used in this research.

2.8 Performance Metrics for IR-Based Traceability Recovery

The quality of traceability data is defined using several factors such as granularity level, recall and precision, number of false retrieved, and level of coverage attained for

the system (Cleland-Huang, Gotel, & Zisman, 2012). IR-based traceability recovery method's performance is usually evaluated using the collection of retrieved links over the collection of relevant links. Generally, the retrieved links and the relevant links do not exactly equivalent. Due to the fact that any IR method may fail in retrieving some of the relevant links (correct links), while in contrast, it may also retrieve links that are not relevant (false positives) (De Lucia et al., 2012).

The performance of IR methods' links retrieval can be measured using two well-known metrics, namely recall and precision (Hayes, Dekhtyar, & Sundaram, 2005; De Lucia et al., 2012; Shin, Hayes, & Cleland-Huang, 2015; Koehrsen, 2018; Hayes et al., 2018).

"Recall is the ratio between the number of links that are successfully retrieved and the number of links that are relevant" (De Lucia et al., 2012):

$$Recall = \frac{|\{relevant\ links\} \cap \{retrieved\ links\}|}{|\{relevant\ links\}|}$$

Recall is not enough to evaluate the performance, there is a need to measure the number of non-relevant links as well. Therefore, Precision has been used, it takes into account all retrieved links.

"Precision is the fraction of the links retrieved that are relevant to the source artifact" (De Lucia et al., 2012):

$$Precision = \frac{|\{relevant\ links\} \cap \{retrieved\ links\}|}{|\{retrieved\ links\}|}$$

It is a worth noting that "precision" in the IR field has different meaning and usage from the accuracy and precision in other fields of science and technology. As accuracy in the field of science is defined as "the degree of conformity of a measured or calculated quantity to its actual (true) value", precision is "the degree to which further measurements

or calculations show the same or similar results" (Baeza-Yates and Ribeiro-Neto, 1999; De Lucia et al., 2012).

Both recall and precision have values in the interval of [0, 1]. When the recall value is 1, this means that all links have been retrieved (may include non-relevant links). When the precision is 1, it indicates that all retrieved links are relevant (there may be missing links).

Average precision is another metric used in this research to measure the performance. It is known as the mean of the precision scores after retrieving each relevant document, and for missing correct links, it gives zero precision. It combines recall and precision for the retrieval results (De Lucia, et al., 2012; Zhang E., Zhang Y. 2009).

$$Average\ Precision = \frac{\sum_r P@r}{R}$$

where $P@r$ is the precision score at each relevant retrieved document and R is the number of relevant documents.

2.9 Tradeoff between Recall and Precision

Precision is involved with accomplishing a low number of false positives (incorrect links). While *Recall* is involved with accomplishing a low number of false negatives (missing links). The ideal candidate link list desires to obtain 100% recall and 100% precision. But, it is very difficult to gain this ideal result (Sundaram, 2007), due to the fact that seeking to gain 100% recall via returning all possible links, leads to decrease precision by retrieving more false postitives (incorrect links). In contrast, trying to enhance precision might also leading to retrieve less true positives(correct links) which result in decrease recall (Zou, Settimi, & Cleland-Huang, 2010). Therefore, there is usually trade-off between precision and recall. For that reason, for any project, managers ought to take into account whether recall or precision is more vital. For example, for safety critical systems, recall is likely to be more crucial. On this kind of systems, engineers will no longer want to take the risk of missing a link, and they are ready to

analyse and remove false positives. On the other hand, non-safety critical systems which have a time limit may prefer to choose precision (Koehrsen, 2018).

For search engines like Google, Bing and Yahoo, precision over recall is preferred. For instance, users who use these search engines would look at the top 20 retrieved documents to check whether they are enough for their desired information, not caring about whether a few valid documents are missing or not (Zou, Settimi, & Cleland-Huang, 2010).

In this research, the researcher wants to achieve as a high precision and recall as possible.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction

This research aims to assist software analysit by proposing a method to enhance the performance of IR-based trceability recovery, based on indexing noun phrases from software artifacts instead of indexing all terms.

This chapter discribes the methodology followed in this research to achieve research objectives.

3.2 Research Methodology

The methodology for this research is illustrated in Figure 3.1. It consists of four phases: Literature Review, Identify Research Gap, Method to Enhance IR-based Traceability Recovery, and Results Evaluation.

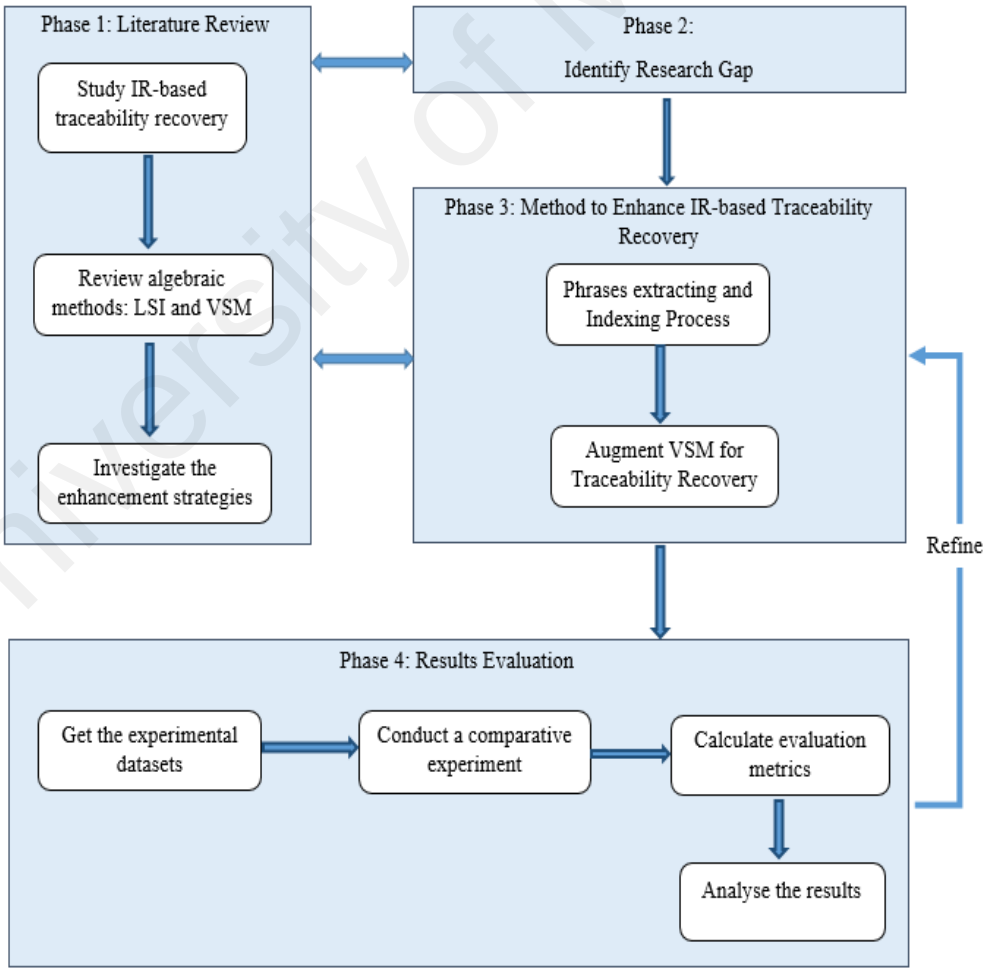


Figure 3.1: Research Methodology

3.2.1 Phase 1: Literature Review

The first phase is literature review in which IR-based traceability recovery is reviewed to identify its usefulness, and further algebraic information retrieval methods are reviewed to demonstrate their strength and weakness in traceability recovery process. In addition, the enhancement strategies that augment IR methods are investigated.

3.2.2 Phase 2: Identify Research Gap

After reviewing literature about traceability recovery, investigating IR method, and enhancement strategies, Research gap and problem statement is identified in the second phase.

3.2.3 Phase 3: Design and Development of a Method to Enhance The IR-based Traceability Recovery

The third phase is the proposed method to enhance the IR-based traceability recovery. In this phase, indexing noun phrases is used as an enhancement strategy to augment the selected IR method for traceability recovery. The proposed method is described in detail in Section 3.3.

3.2.4 Phase 4: Result Evaluation

In the fourth phase, to evaluate the proposed method, a comparative experiment is conducted to compare the performance of proposed noun phrase indexing with other indexing strategies. Experimental datasets are collected, and frameworks and tools are used to conduct a comparative experiment for evaluation purpose. The evaluation metrics are calculated to compare between the indexing noun phrases and other indexing strategies. These metrics are namely recall, precision and Average Precision. Recall is defined as the fraction of the number of links that are successfully retrieved to the number of links that are relevant. Precision is the ratio of the links retrieved that are relevant and the source artifact. Average Precision is defined as the mean of the

precision scores after each relevant document is retrieved (Hayes, Dekhtyar, & Sundaram, 2005; De Lucia et al., 2012; Shin, Hayes, & Cleland-Huang, 2015; Koehrsen, 2018; Hayes et al., 2018). Then the result is analyzed to refine the proposed method.

3.3 The Proposed Method to Enhance IR-based Traceability Recovery

In order to enhance the performance of IR-based traceability recovery, this research proposes a method that modifies artifacts indexing process by indexing noun phrases rather than indexing all terms or nouns as a single term. Nouns involve more information value and have a semantic role. In addition, phrases can help to get a more accurate description of document's content than single words.

The overall process of the proposed method is shown in Figure 3.2. It is classified into two phases.

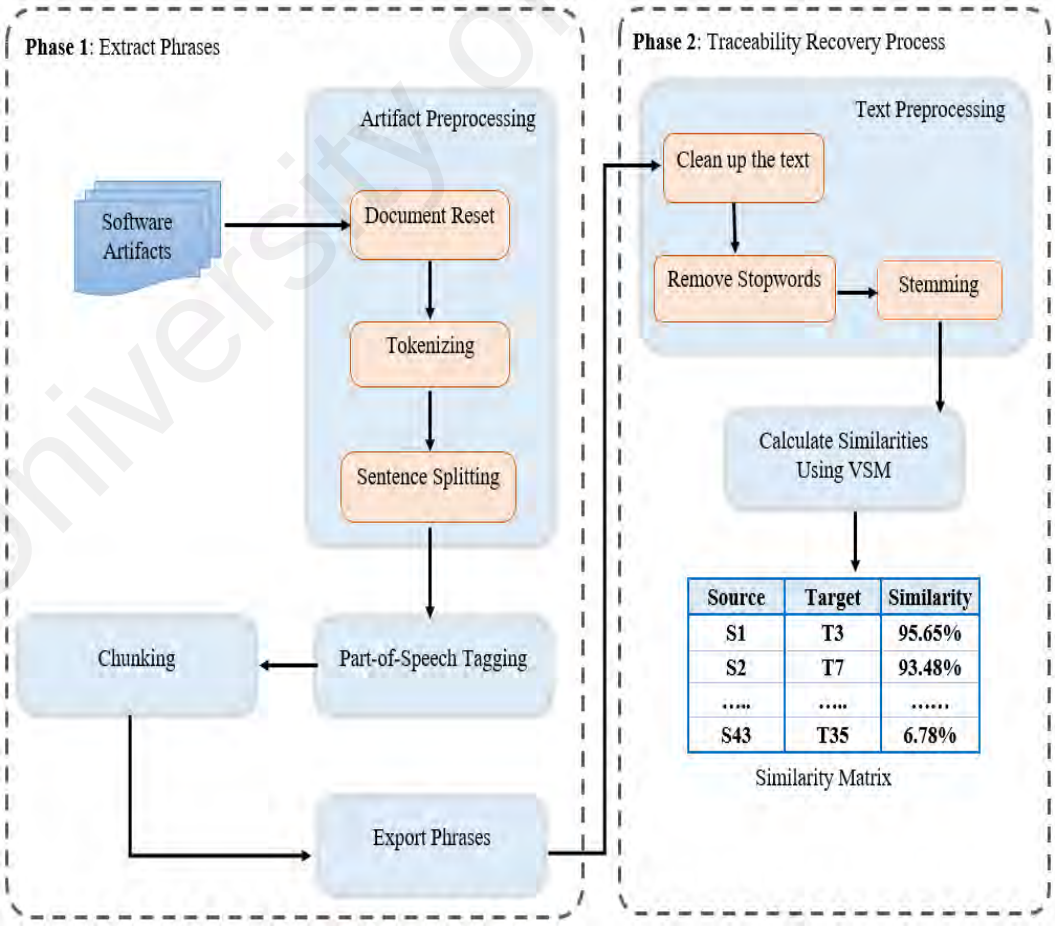


Figure 3.2: Proposed Method's Process flow

3.3.1 Phase 1: Extract phrases

Phase 1 is to extract phrases from software artifacts. It consists of 4 steps, Artifacts Preprocessing, Part-of-Speech Tagging, Chunking, and Exporting Phrases. All steps are presented in appendix A.

i. Artifacts Preprocessing

The textual artifacts are preprocessed. The first preliminary process is Document Rest. This process separates the annotation sets from the text document and makes it possible to rest the document to its original state.

The second process is Tokenizing. This process is used to add token annotations to each word, symbol, etc. and space token annotations to the white space in the text.

Splitting Sentences is the third process. It used to add sentence annotations and split annotations.

ii. Pat-of-Speech Tagging

It takes the textual artifacts as an input to assign part of speech tags such as (noun, verb, adjective …etc.) to each term in the document. POS tagging process used to assign POS tags to each token annotation. This process requires token and sentence annotations.

iii. Chunking

As a term possibly will have many POS tags, a chunker will be performed to guarantee that all terms are correctly tagged and also to define different chunks such as NP and VP.

Sentence annotation, token annotation, and category feature (tagged terms) must be presented at runtime, so POS tagger is necessity.

iv. Export Phrases

After chunking the textual artifacts and identifying the desired phrases, these phrases are exported and indexed to be used for enhancing traceability recovery.

3.3.2 Phase 2: Traceability Recovery Process

This phase is to create traceability links between source and target artifacts. It consist of two steps:

i. Text Preprocessing

The text artifacts have to preprocess first then forward them to the IR technique, such as eliminate non-characters, remove stop word, stemming, etc.

ii. Calculate Similarities Using VSM

Filtered and preprocessed artifacts will be forwarded to VSM. Different weights will be assigned to the terms based on their occurrences in documents using TF_IDF (Term frequency_ Inverse document frequency). The term weight is computed as the multiplication of TF and IDF. The term weight considered high when the term occurs many times within a document, and is contained in a small number of documents. Otherwise, it considered low, if the term occurs few times in a document, or occurs in many documents or all documents. The similarity score *sim (d,q)* is usually used to compute the relevance between the query *q* and the document *d*, it can be defined as the cosine of the angle formed by their corresponding vectors, and is computed as: $sim(d,q) = \frac{\sum w_{i,d} \times w_{i,q}}{\sqrt{\sum w_{i,d}^2 \times \sum w_{i,q}^2}}$ (Zou, Settimi, & Cleland-Huang, 2010).

3.4 Planning for the Comparative Experiment

High-level and low-level requirements are used in this research are collected from CM1 and MODIS datasets. For PINE, two high-level artifacts are used, requirements and use cases. All datasets are text documents written in English. The answer set is created and provided by the developer for all datasets.

To compare the performance of each type of phrases in enhancing IR-based traceability recovery, the traceability links between artifacts are recovered according to the following indexing strategies:

a. Indexing all terms

b. Indexing noun phrases only

c. Indexing verb phrases only

d. Indexing both noun and verb phrases

All artifact files are preprocessed to eliminate all non-characters, remove stop words, and then stem to the roots. For the last three strategies, this preprocessing is done after phrases have been extracted using GATE.

VSM is used to calculate the similarities between source and target artifacts, by assigning different weight to each term according to its appearance in the document, and use cosine similarity score to compute the similarity.

Recall, precision, and average precision evaluation metrics is utilized to evaluate the result.

3.5 Dataset for the experiment

To conduct the experiment, three datasets namely MODIS, CM1, and PINE datasets are used. Modis refers to NASA's Moderate Resolution Imaging Spectroradiometer. This dataset consists of 19 high-level requirements and 49 low-level requirements selected from two high-level and low-level requirements documents publicly available by NASA.

It has been modified and the answer set was created by Dr.Hayes and Dr.Dekhtyar (Sayyad, Menzies 2005; Sundaram, Hayes, & Dekhtyar 2005).

CM1 is a NASA spacecraft instrument for data collection and processing. NASA Metrics Data Program (MDP) provided this dataset to the public. It contains 235 high-level requirements and 220 low-level requirements. Low-level requirements determine the detailed design. CM1-subset which contains 22 high-level requirements and 53 low-level requirements, is used in this research. It also has been modified and the answer set was created by Dr.Hayes and Dr.Dekhtyar (Sayyad, Menzies 2005; Sundaram, Hayes, & Dekhtyar 2005). The MODIS and CM1 datasets have been used in many studies (Hayes, Dekhtyar, & Sundaram, 2005; Sundaram, 2007; Capobianco et al., 2013; Borg, Runeson, & Ardö, 2014; Hayes et al., 2018).

PINE is and email management system developed at the University of Washington. Pine dataset was created by Sultanov and Hayes (Sultanov, H., & Hayes, J. H. 2010).

Table 3.1 demonstrates details about used datasets.

Table 3.1: Dataset overview

Dataset	Origin	Language	Artifacts	
			Type	Number
MODIS	NASA	English	High-level requirements	19
			Low-level requirements	49
CM1	NASA	English	High-level requirements	22
			Low-level requirements	53
PINE	University of Washington	English	High-level Requirements	49
			Use Cases	51

CHAPTER 4: IMPLEMENTATION OF THE PROPOSED METHOD

4.1 Introduction

This chapter presents details of implementing the proposed method to enhance the performance of IR-based traceability recovery, to achieve as high recall and precision as possible.

4.2 Implementing the method to enhance IR-based traceability recovery

This research proposes a method that modifies artifacts indexing process by indexing noun phrases rather than indexing all terms or nouns as a single term. The overall process of the proposed method is presented in the previous chapter Figure 3.2, it classified into two phases. The implementation details is as the following:

4.2.1 Phase 1: Extract phrases

To extract phrases from software artifacts, 4 steps is followed, Artifacts Preprocessing, Part-of-Speech Tagging, Chunking, and Exporting Phrases. Figure 4.1 shows processing resources that have been set to perform this phase. An example of uploading documents into GATE is shown in Figure 4.2.

i. Artifacts Preprocessing

Document Rest process is performed to separates the annotation sets from the text document and makes it possible to rest the document to its original state.

OpenNLP Tokenizer is used to add token annotations to each word, symbol, etc. and space token annotations to the white space in the text.

OpenNLP Sentence Splitter is used to add sentence annotations and split annotations.

ii. Pat-of-Speech Tagging

OpenNLP Tagger is used in this process to assign POS tags to each token annotation. This process requires token and sentence annotations.

iii. Chunking

OpenNLP Chunker is used in this experiment to define noun phrases and verb phrases.

Sentence annotation, token annotation, and category feature (tagged terms) must be presented at runtime, so POS tagger is necessity. Noun phrases only, verb phrases only, and combination of noun and verb phrases annotation sets are demonstrated in Figure 4.4, 4.5, and 4.6 respectively.

iv. Export Phrases

To export NP and VP from the documents, Gate plugin Flexible Exporter is used. It allows the user to choose the name of the annotation set. Then the document will be saved in its original format appended with the chosen annotation sets.

Runtime parameters have to be set such as annotationSetName, annotationTypes, dump types, and outputDirectoryUrl to define the directory where the document is exported.

Figure 4.7 demonstrates an example of exported file consisting of noun phrases and verb phrases along with appended annotation sets all in one file. To separate this into two files, one file contains noun phrase only and the other file contains

verb phrase only, C++ program is developed. The program is presented in appendix B.

Figure 4.8 shows running C++ program on use cases from PINE dataset to separate noun and verb phrases into two different files. An example of separated noun and verb phrases files from PINE dataset are shown in Figure 4.9.

4.2.2 Phase 2: Traceability Recovery Process

This phase is to create traceability links between source and target artifacts. It consist of three steps:

i. Text Preprocessing

Text preprocessing step include three processes 1) Clean up text, to filter out all non-character such as punctuation marks, numbers etc. 2) Remove stopwords, to remove common stop words such as ‘a’, ‘the’, ‘is’, ‘are’, ‘will’, etc. it utilizes a previously prepared stop words list. 3) Stemming, it is very important in Information Retrieval, it is used to reduce different forms of a word to its root. For example the word ‘receiv’ is the stem of ‘receive’, ‘receives’, ‘received’, ‘receiving’, etc.

ii. Calculate Similarities Using VSM

Figure 4.10 illustrates how phase 2 is done using TraceLab. Firstly import software artifacts, stop words list, and the answer set. Second, do some text preprocessing and filtering by eliminating all non-characters and removing common stop words. Then snowball stemmer algorithm is used to reduce each word to its root. The next process is using VSM to calculate the similarities. Appendix C describes more about artifacts preprocessing.

Figure 4.1 shows processing resources that have been set to perform phrase extraction phase.

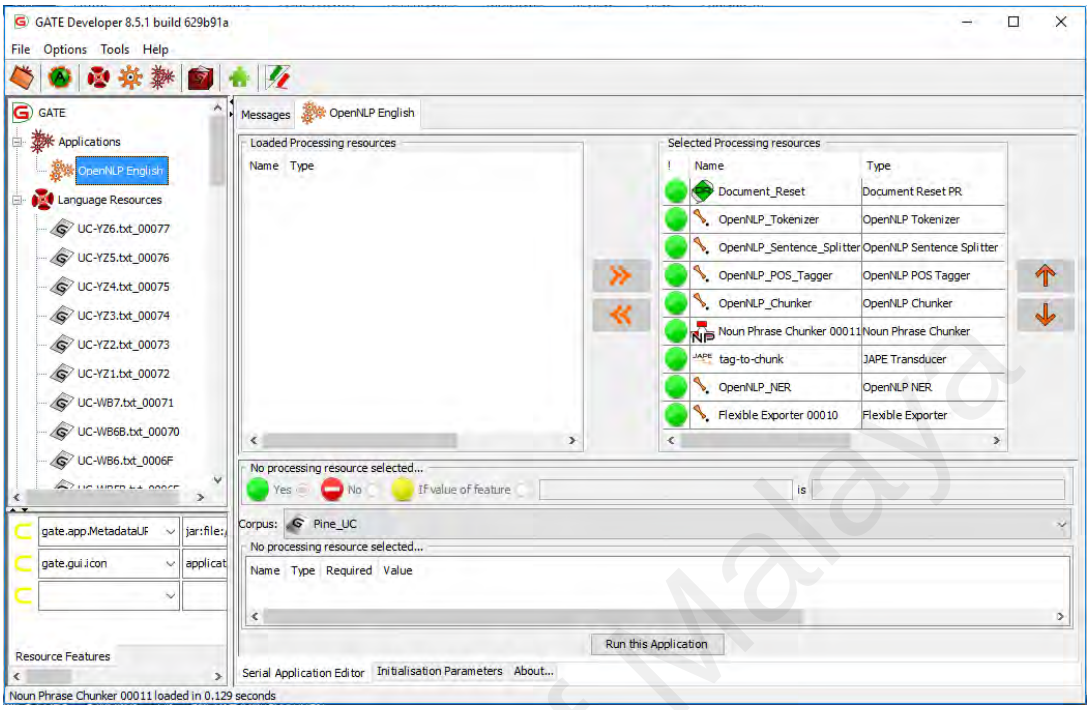


Figure 4.1: Set Processing Resources

Figure 4.2 shows an example of uploading documents into GATE

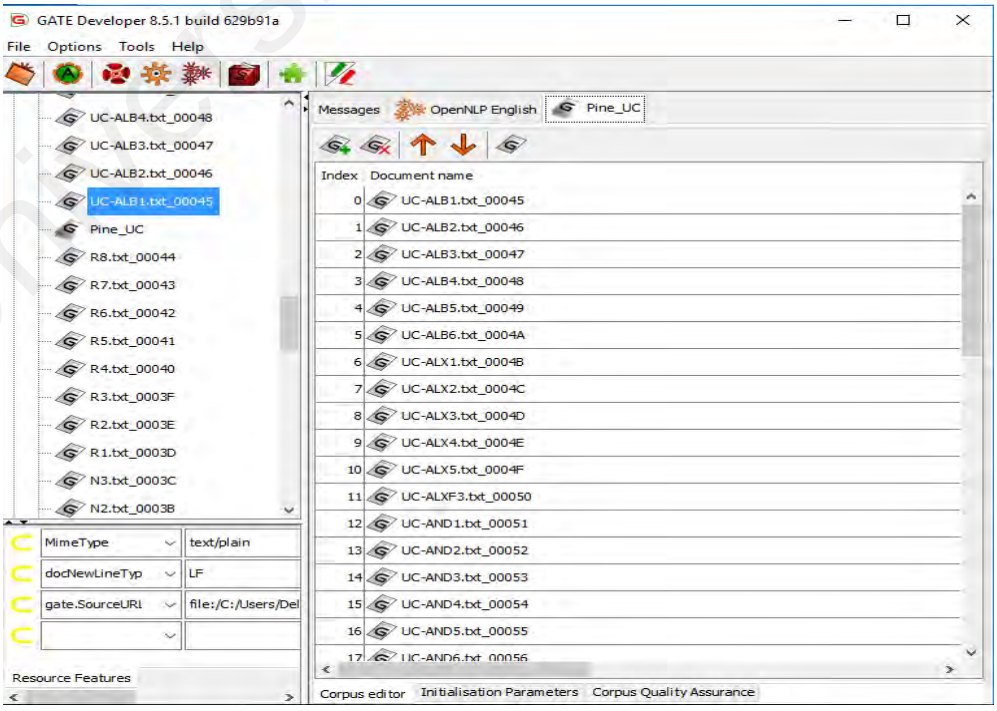


Figure 4.2: Uploading documents into GATE

After uploading the documents into GATE and set all the resources, Figure 4.3 shows running OpenNLP on 51 documents.

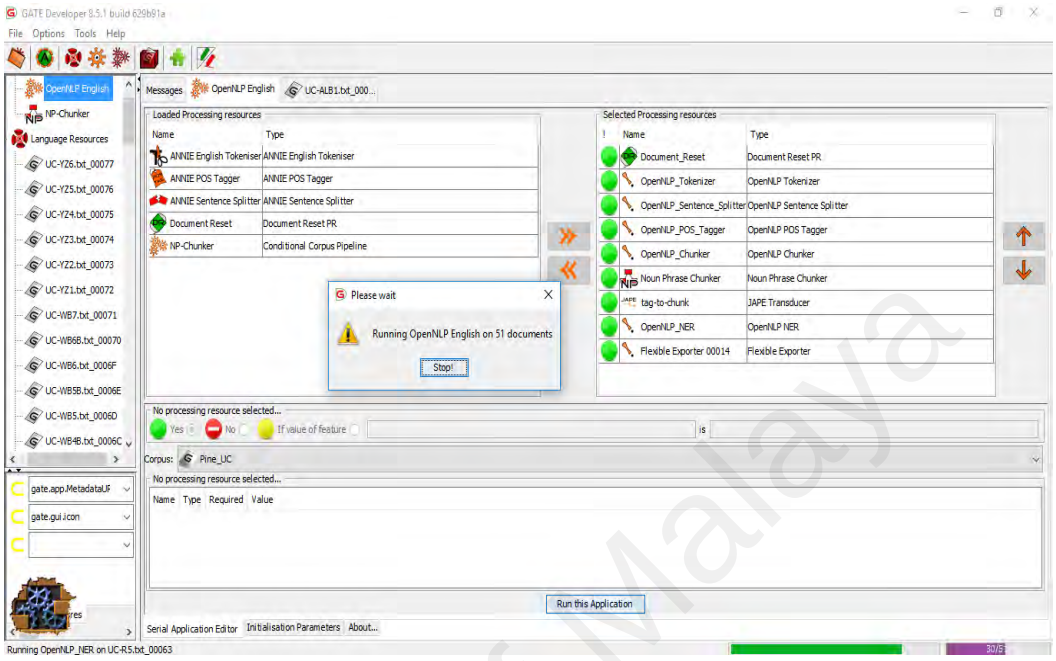


Figure 4.3: Run OpenNLP on 51 Documents

Figure 4.4 demonstrates verb phrases annotation sets.

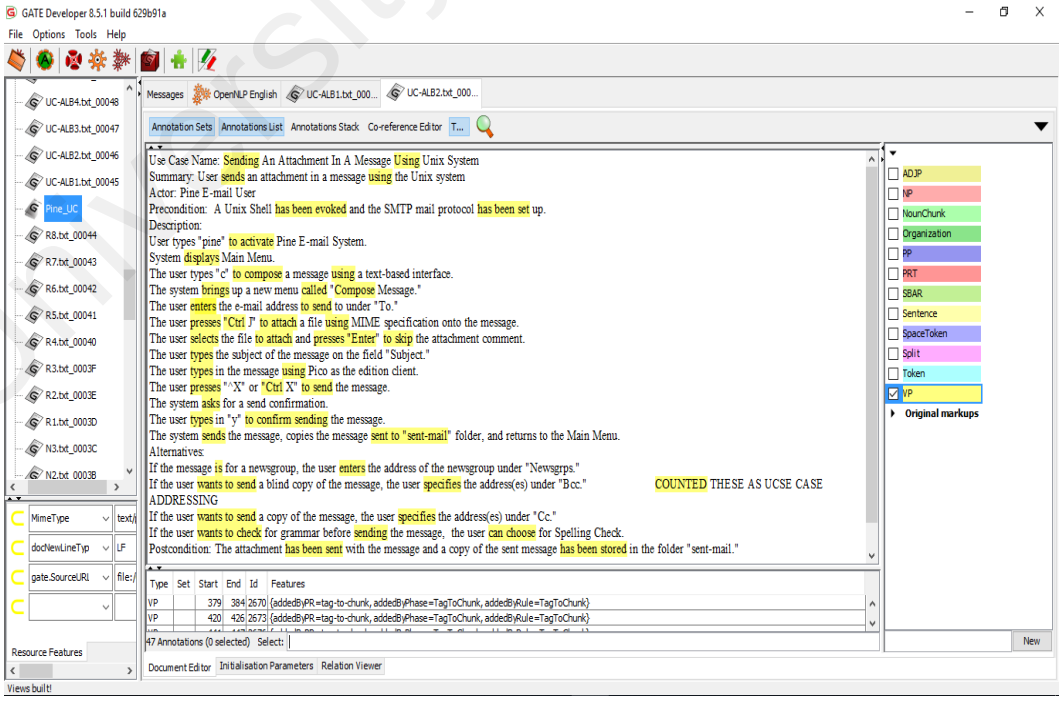


Figure 4.4: Verb Phrases Annotation Set

Figure 4.5 demonstrates noun phrases annotation sets.

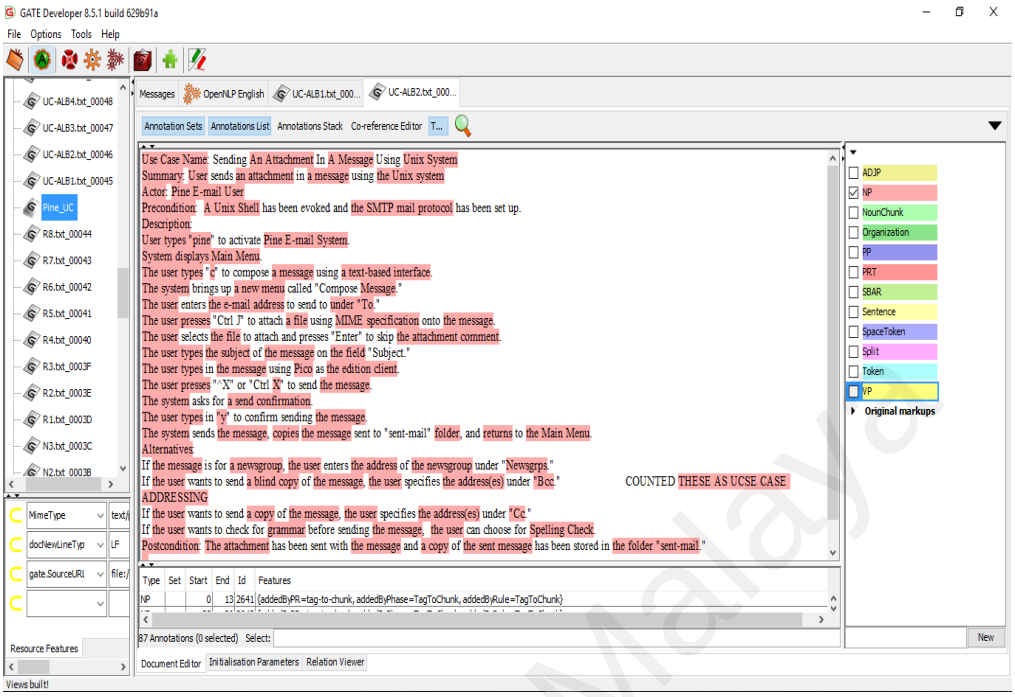


Figure 4.5: Noun Phrases Annotation Set

Figure 4.6 demonstrates a combination of noun phrases and verb phrases annotation sets.

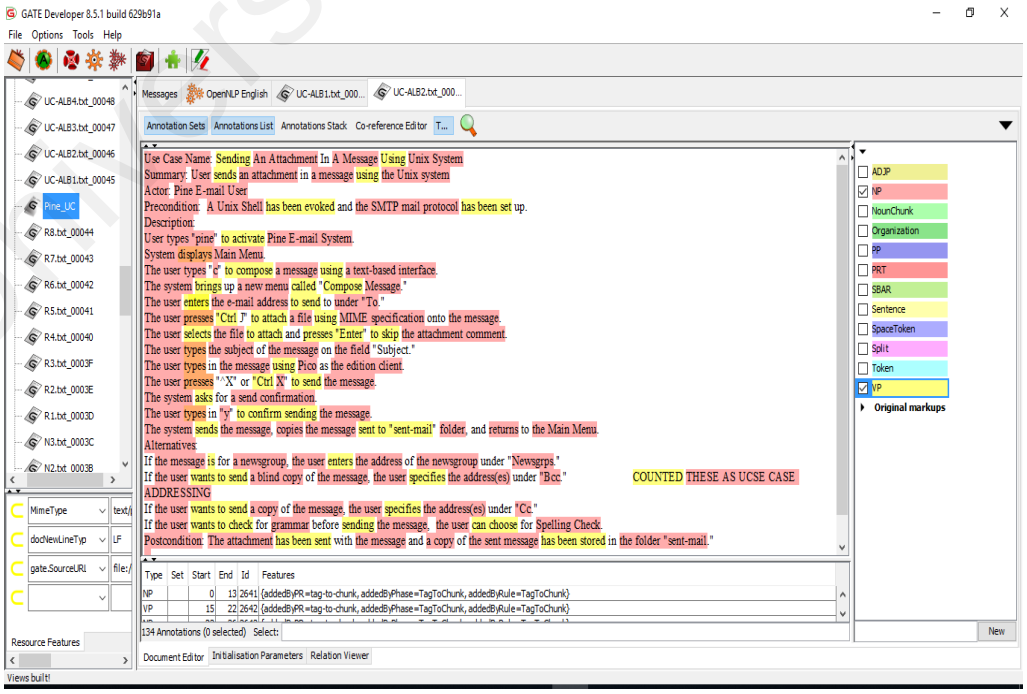


Figure 4.6: Verb and Noun Phrases Annotation Set

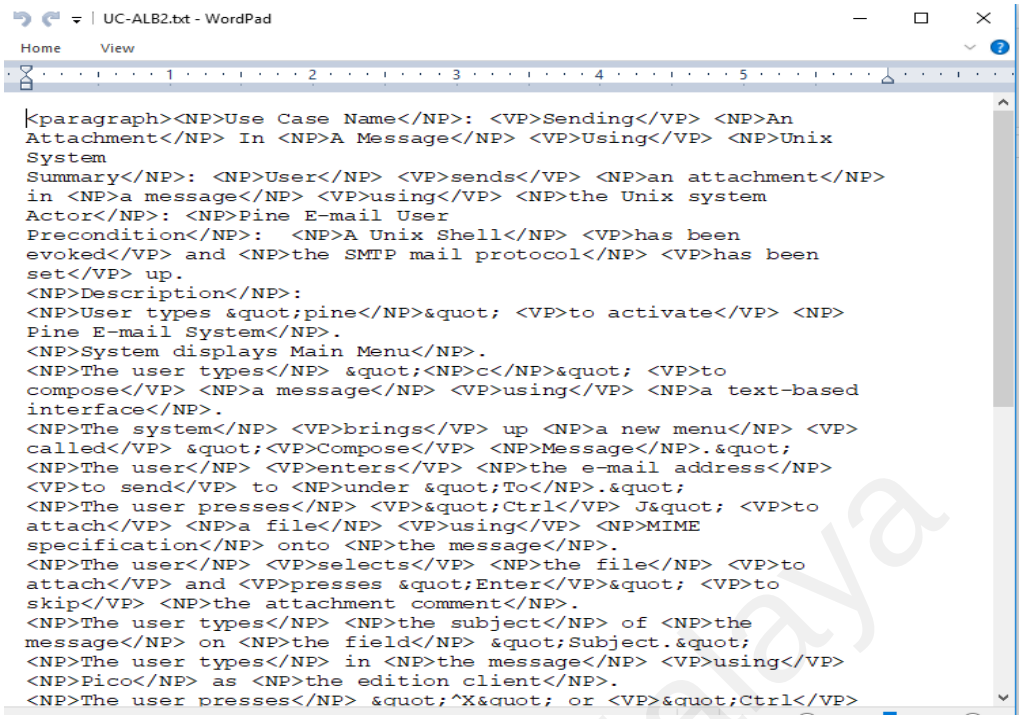


Figure 4.7: Example of an Exported Annotation set

Figure 4.8 shows running C++ program on use cases from PINE dataset to separate noun and verb phrases into two different folders named as NP Only and VP Only.

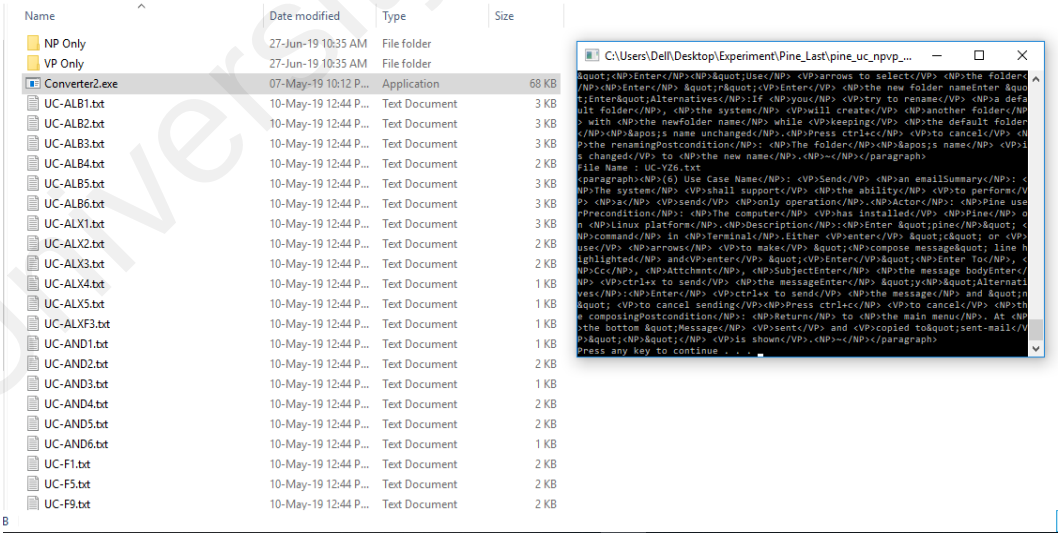


Figure 4.8: Separate Noun and Verb Phrases

An example of separated noun and verb phrases files from PINE dataset are shown in Figure 4.9

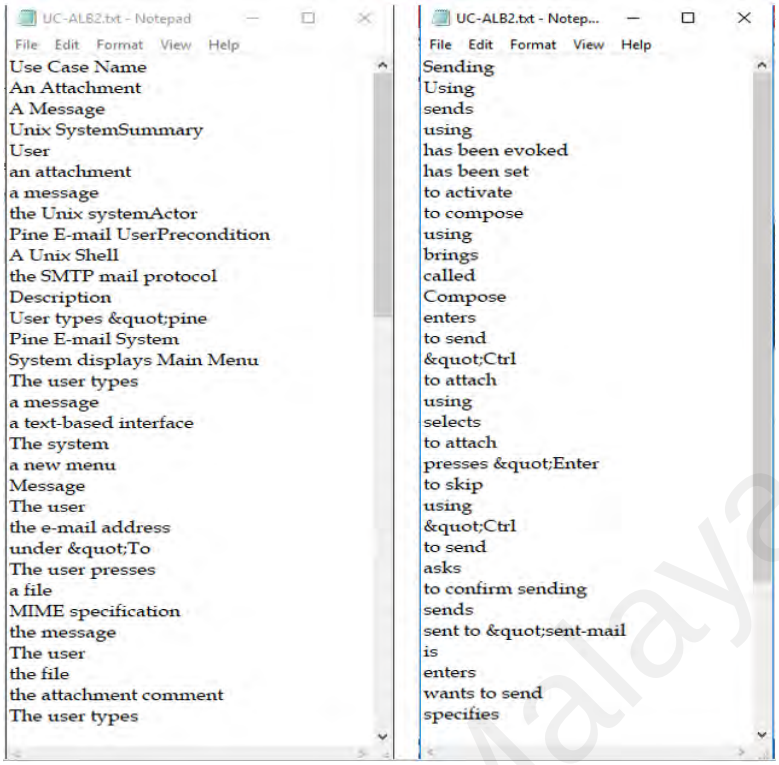


Figure 4.9: Example of Separated NP and VP Files from PINE

Figure 4.10 illustrates how text preprocessing steps and calculating the similarity using VSM are done using TraceLab.

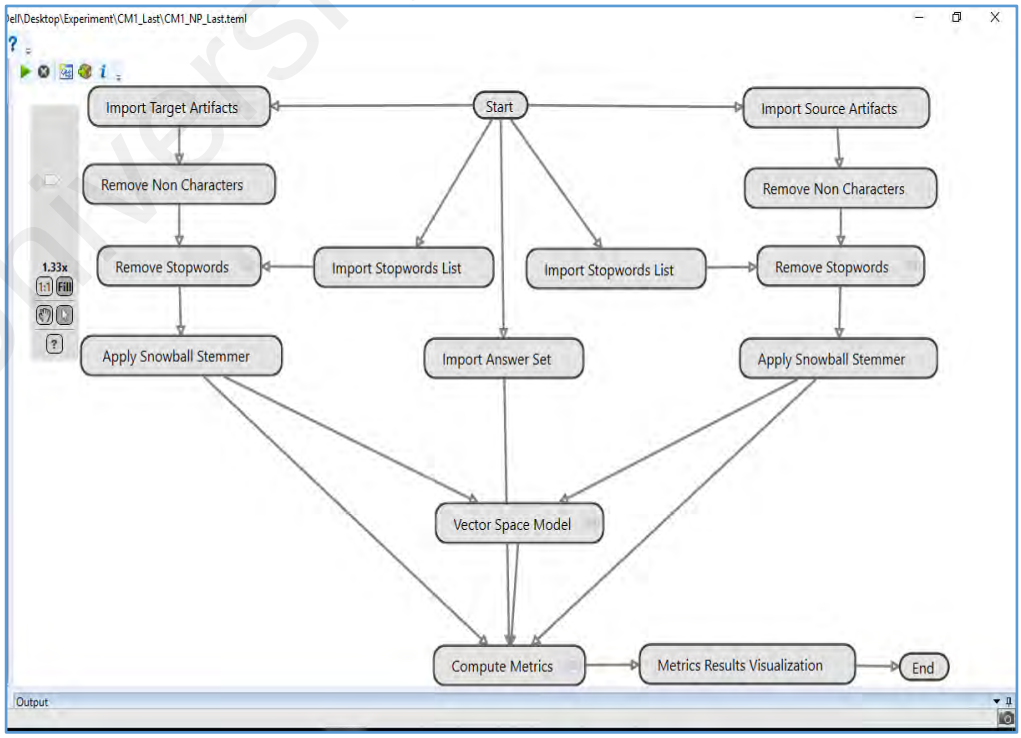


Figure 4.10: Traceability recovery Process

CHAPTER 5: RESULTS AND DISCUSSION

5.1 Introduction

In this chapter, 5.2 describes the results of the experiment conducted to compare indexing different types of phrases to enhance IR-based traceability recovery performance. The comparison and discussion about the results obtained from many evaluation metrics is described in 5.3, to validate the method of indexing noun phrases to enhance the performance of IR-based traceability recovery.

5.2 Results Analysis

The following subsections describe the results obtained from three datasets which are CM1, MODIS and PINE to compare between indexing various types of phrases to enhance IR-based traceability recovery performance.

5.2.1 Result of CM1 Dataset

Figures 5.1, 5.2, 5.3, 5.4 describe the results obtained by applying VSM on the first dataset CM1, when high-level requirements traced to low-level requirements. It shows the effect of indexing all terms, NP only, VP only, and combination of NP and VP on the IR performance. The performance metrics Average precision, Recall, Precision, and precision at recall 100% are used. The results are presented in appendix C.

The boxplots on Figure 5.1 indicate the average precision for each strategy. Indexing NPVP achieved higher median than indexing NP. Indexing NP and All terms did better than indexing VP because 50% of the average precision above 0.5, whereas 75% of the average precision is below 0.42 when index VP only. Indexing NP considerably performed better than indexing all terms and VP for average precision because 25% of NP scored a 0.833 or above.

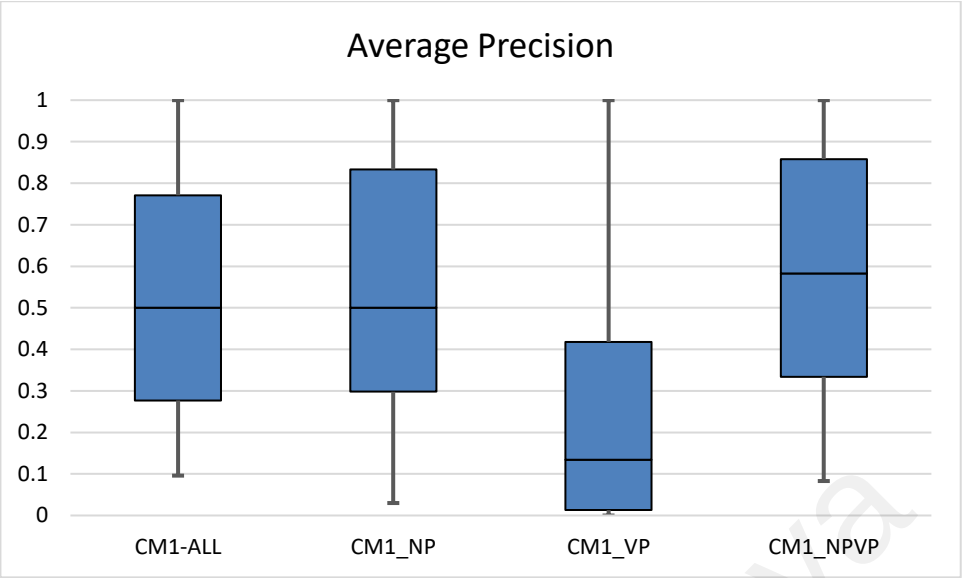


Figure 5.1 Average precision_CM1

The boxplot on Figure 5.2 describes recall. All terms, NP, and NPVP strategies achieved the higher recall than VP. Because 100% of the three strategies scored above 0.75 while VP strategy obtained the lowest recall because 50% was below 0.75.

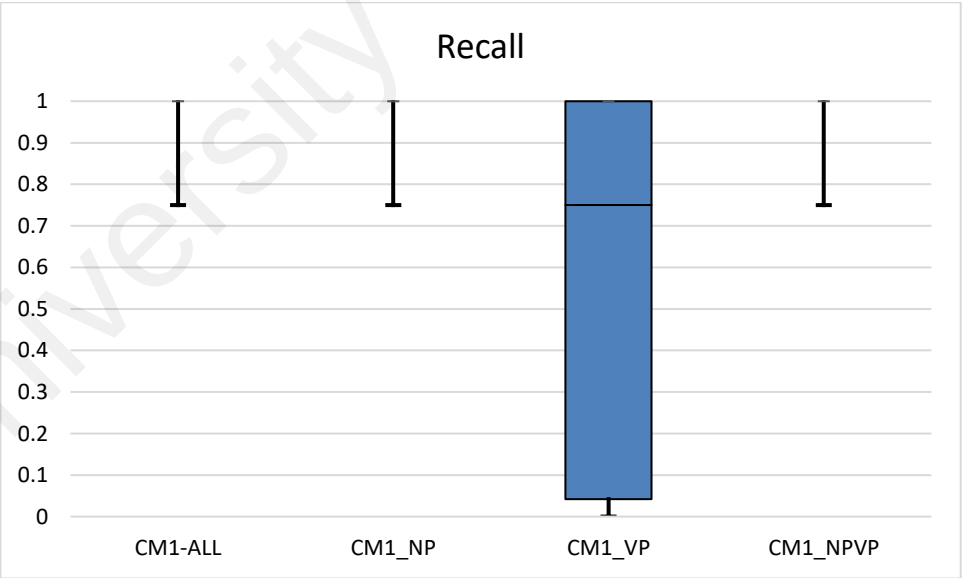


Figure 5.2 Recall_CM1

The boxplots on Figure 5.3 describes precision. It shows that NP strategy did better than All terms and NPVP strategies because 50% of precision using NP scored 0.048,

VP had 50% of precision above 0.049 but it is still had the lower consistent due to more variation.

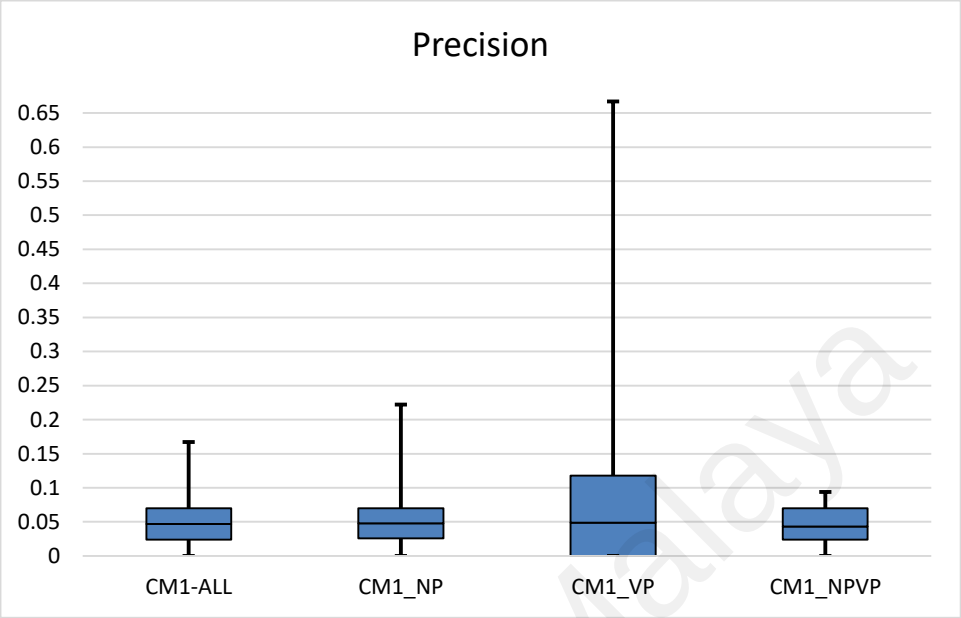


Figure 5.3 Precision_CM1

When recall is 100% in Figure 5.4, the highest precision achieved when indexing NPVP because 75% of precision value is above 0.167, followed by NP because 75% scored 0.125 or above. VP obtained the lowest precision value at recall 100%.

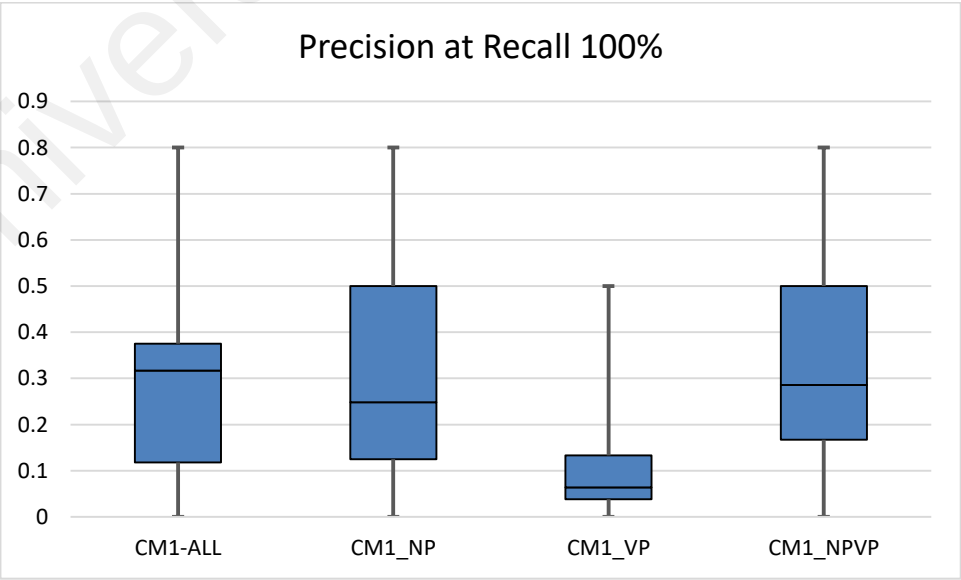


Figure 5.4 Precision at recall 100%_CM1

5.2.2 Result of MODIS Dataset

For the second dataset MODIS, high-level requirements are also traced to low-level requirements. The results obtained from adopting these strategies to VSM are highlighted in Figure 5.5, 5.6, 5.7, and 5.8 and also presented in appendix D.

Average precision boxplot on Figure 5.5 illustrates that NP achieved higher average precision than indexing NPVP and all terms. This is because 50% of average precision value above 0.5. Whereas 50% of indexing all terms scored 0.416 or above. VP achieved the lowest average precision value, 75% scored 0.16 or less.

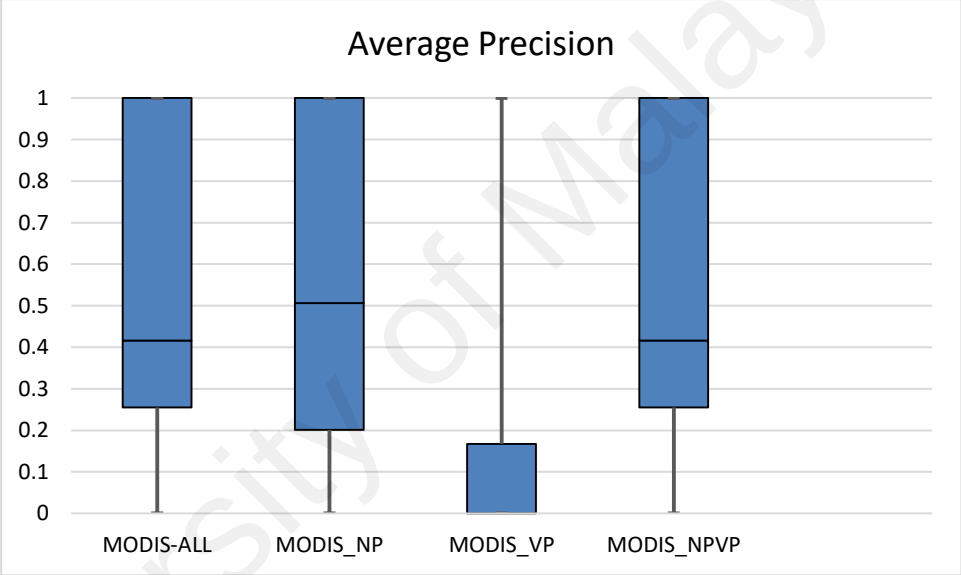


Figure 5.5 Average precision_ MODIS

Recall boxplot on Figure 5.6 illustrates that all strategies achieved high recall except VP. All terms, NP, and NPVP strategies achieved the higher recall than VP because 75% of each strategies scored 1, while VP strategy obtained the lowest recall because 75% was below 0.5.

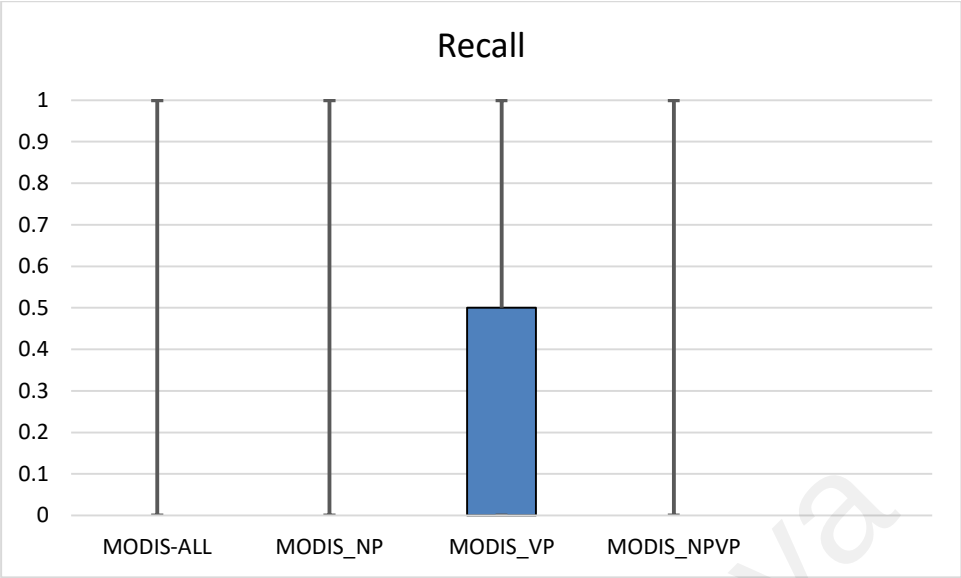


Figure 5.4 Recall_ MODIS

On precision boxplots in Figure 5.7, NP strategy did better because 50% scored 0.029 or above, whereas 50% of NPVP and all terms strategies scored 0.27 or above. Indexing VP achieved 25% of precision value above 0.15 but 50% scored 0.

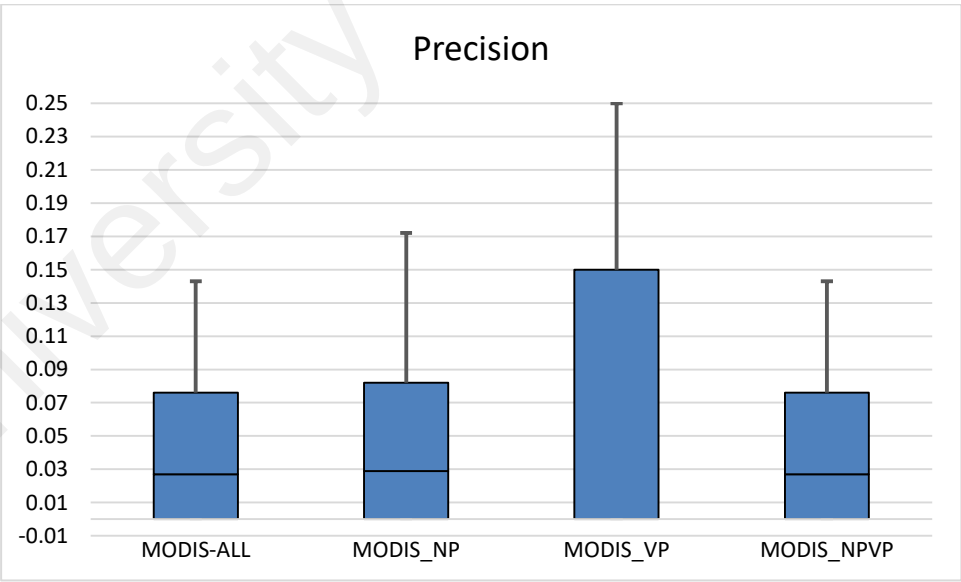


Figure 5.5 Precision_ MODIS

Figure 5.8 demonstrates precision when recall is 100%. The highest value obtained by NP because 50% above 0.107, whereas 50% of All terms and NPVP strategies

scored 0.091 or above, in addition NP is more consistent than All terms and NPVP. VP strategy achieved the lowest precision at recall 100%.

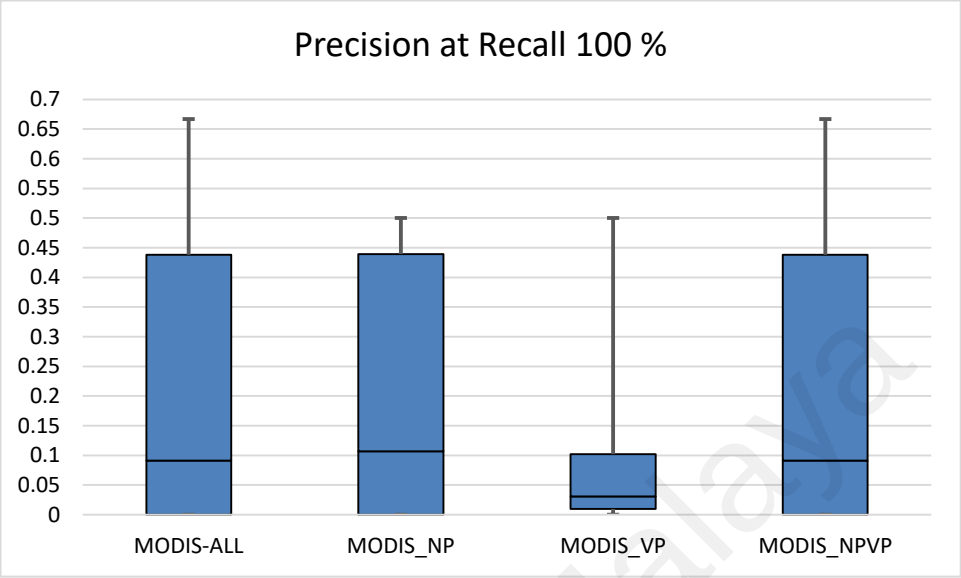


Figure 5.6 Precision at recall 100%_MODIS

5.2.3 Result of PINE Dataset

PINE is the third dataset used in this research. Traceability links are created between requirements and use cases artifacts. The results are presented in appendix E.

Figure 5.9 demonstrates average precision boxplot. It shows that both NPVP and all term strategies achieved high average precision followed by NP. Roughly there is no difference between indexing All terms or indexing NPVP, both of them almost have the same median and variability. But when compare NP and VP, it is obvious that NP performed better and more consistent than VP, because 50% of average precision when using NP scored 0.6 and above while average precision value by VP scored 0.32 or more.

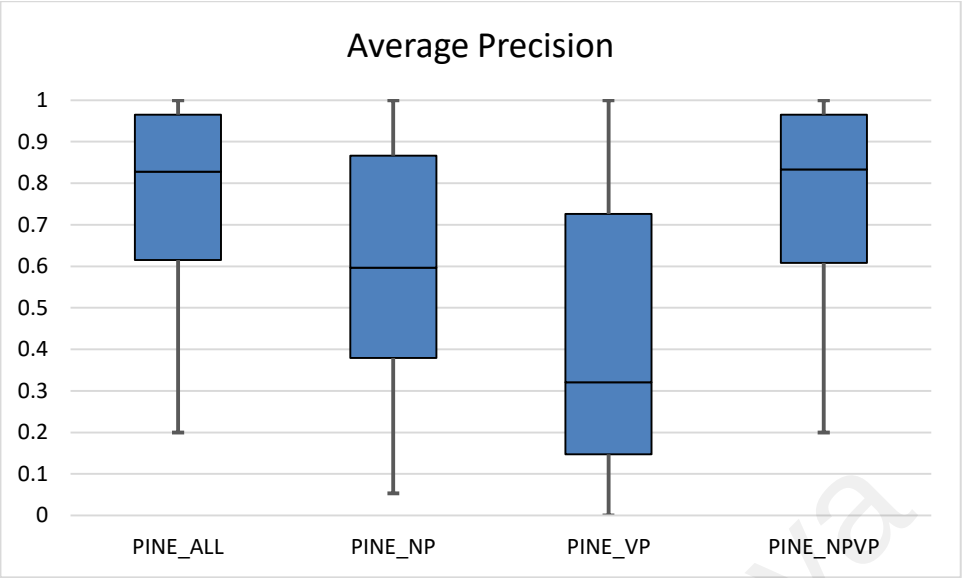


Figure 5.7 Average precision_ PINE

Figure 5.10 demonstrates recall boxplots. Also NPVP and all terms strategies obtained the highest value followed by NP. NP surpass VP because 75% of NP obtained above 0.881 recall. Whereas 75% of VP scored 0.204 or above.

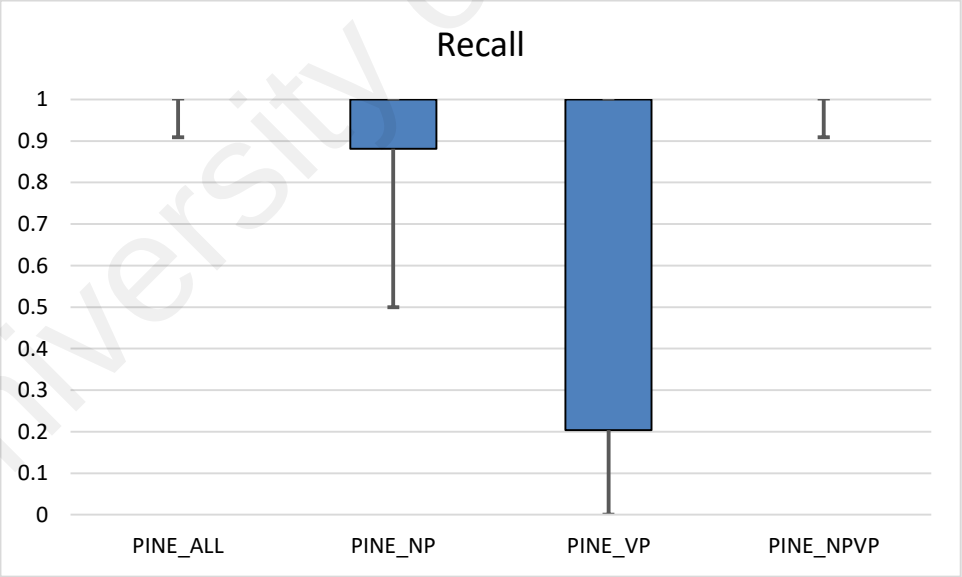


Figure 5.8 Recall_ PINE

Precision boxplots is shown in Figure 5.11. The highest precision is achieved by VP with 50% of precision above 0.2 but the spread is much larger than other strategies which means less consistent. NP achieved better precision than NPVP and all terms

strategies because 50% scored above 0.104 while 50% of all terms and NPVP scored above 0.098.

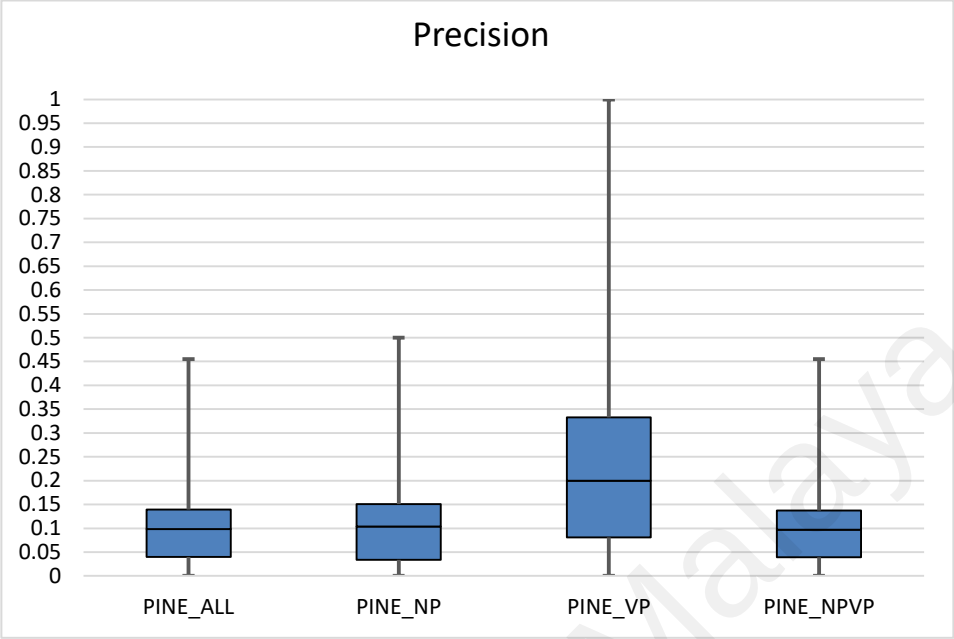


Figure 5.9 Precision _ PINE

The precision when recall is 100% is described in Figure 5.12. The boxplots illustrate that indexing all terms and NPVP strategies obtained the same. NP achieved higher precision at recall 100% than VP. 50% of NP obtained above 0.25 while 50% of VP obtained above 0.2.

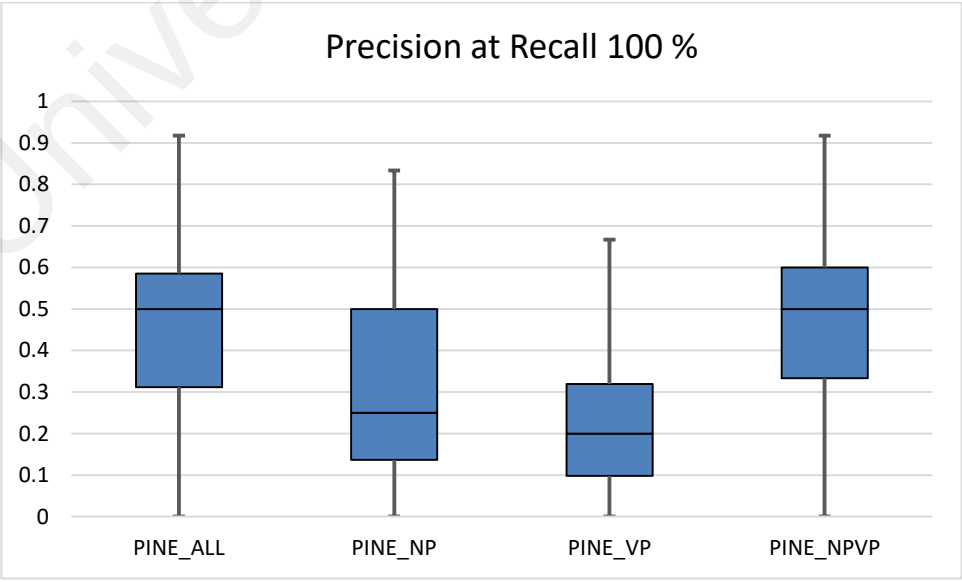


Figure 5.10 Precision at recall 100%_ PINE

5.3 Result Summary and Discussion

The previous subsection reports and analyse all results obtained by different indexing strategies used in this research (All terms, NP, VP, NPVP) for CM1, MODIS, and PINE datasets. The comparison and discussion about the results obtained from many evaluation metrics is described in this subsection, to validate the method of indexing noun phrases to enhance the performance of IR-based traceability recovery.

Analyzing these results highlighted the following points:

a. For all datasets, VP achieved the lowest value for average precision, recall, and precision at recall 100%. Except for precision, as VP obtained the highest precision value. This is due to the fact that indexing VP only eliminates most of false positives (incorrect links) but at the same time many correct links are missing and it is less consistent.
b. There is no difference between NPVP and all terms in MODIS and also in PINE.
c. NPVP and all terms obtained the highest recall for all datasets followed by NP.
d. NP achieved higher precision for all datasets than all terms and NPVP.
e. For MODIS dataset NP achieved highest average precision, precision, recall, and precision at recall 100%.
f. NP performed better than VP in average precision, precision, recall, and precision at recall 100% metrics for PINE.

Table 5.1 summarizes the results of all datasets obtained from the comparative experiment by calculating the average for all metrics.

Table 5.1 Result Summary for All Datasets

Datasets	Indexing Strategies	Metrics		
		Recall	Precision	Precision at Recall 100%
CM1	All terms	0.95	0.061	0.322
	NP	0.95	0.073	0.335
	VP	0.55	0.166	0.147
	NPVP	0.95	0.046	0.35
MODIS	All terms	0.8	0.049	0.239
	NP	0.8	0.056	0.209
	VP	0.3	0.08	0.128
	NPVP	0.8	0.049	0.239
PINE	All terms	0.98	0.146	0.463
	NP	0.88	0.158	0.344
	VP	0.64	0.322	0.257
	NPVP	0.98	0.146	0.47

Absolutely indexing all terms will obtain the highest recall, but on the other hand, it will also retrieve undesirable incorrect links because of a large amount of noise that should be filtered out.

As verb has a functional description and a connection role, verb phrases have been indexed in this research. It is noticeable that indexing VP always obtains the lowest results for all metrics for all datasets. Although, in PINE dataset, the number of verbs is greater than nouns, indexing VP still cannot achieve high recall due to that it has considerably negative effect on recall.

On the other hand, some studies mentioned that both verbs and nouns are important and have to be considered, as nouns have a semantic role and verbs have a connection and descriptive role. Therefore, NPVP is indexed in this research to be compared with other strategies. Indexing NPVP achieved good results in some cases and sometimes there

was no difference between indexing NPVP and indexing all terms. Therefore, it has negative effect on precision due to unwanted terms and false positive links.

The observation from results is that indexing noun phrases only from the software artifacts always achieves the highest performance for almost all metrics for CM1 and MODIS data sets. For PINE dataset indexing NP achieved better than indexing VP. It has never obtained the lowest result in this experiment. This is due to the fact that nouns have a semantic role and carry more information value than other terms. Furthermore, noun phrases give better description for the document content than single terms. In addition, indexing noun phrases filter out unwanted terms and increase the overall accuracy for VSM.

CHAPTER 6: CONCLUSION

6.1 Fulfillment of Research Objectives

The intention of this research was to fulfil the three objectives described in Section 1.3 of Chapter 1. The achievement for each objective is described below:

- **Objective 1:** To investigate IR methods for traceability recovery.

The first objective is achieved by doing the literature review which is highlighted in Chapter 2. Requirements traceability its significance, issues, and creation were studied. Furthermore, information retrieval methods, their categories also studied. In addition to traceability recovery, IR-based traceability recovery process, enhancement strategies for IR-based traceability, performance metrics for IR-Based traceability recovery and tradeoff between recall and precision which all are investigated to accomplish the first objective. Finally, as a result VSM and NP indexing are selected enhance the performance of IR-based traceability recovery.

- **Objective 2:** To propose an IR-based method that achieves high performance (as high recall and precision as possible) in traceability recovery.

An IR-based method for traceability recovery is proposed to fulfil the second objective; indexing NP from software artifact is chosen to augment VSM. The proposed method consists of two phases each phase consists of many steps. Phase 1 is extract phrases from the artifacts, it performed using OpenNLP and GATE, and the steps are: artifact preprocessing, POS tagging, chunking, and then export phrases. Phase 2 is the traceability recovery process which is done using TraceLab. It consists of two steps: text preprocessing step to perform stemming, remove stop words and non-characters, the second step is calculating the similarities using VSM. This method is proposed to achieve high performance (as high recall and precision as possible) in traceability recovery.

- **Objective 3:** To empirically validate the proposed method through an experimental analysis to demonstrate its ability to improve the performance (as high recall and precision as possible) in traceability recovery.

To achieve the third research objective the proposed method is evaluated through a comparative experiment to compare the performance of indexing NP with other indexing strategies. The artifacts datasets are collected from CM1, MODIS, and PINE datasets to conduct the experiment. The traceability links are recovered using different indexing strategies which are: All terms, NP, VP, and NPVP. The recall, precision and Average Precision evaluation metrics are calculated to compare between the indexing NP and other indexing strategies. The result shows that indexing NP is tends to outperform other indexing strategies in improving the performance (as high recall and precision as possible) in traceability recovery.

6.2 Strengths and Contribution

Requirement traceability is known as a software quality measure, as it aims to improve validity, verification, and reusability. Traceability recovery when performed manually is a tiresome and time-consuming process. Consequently, multiple methods of information retrieval were used to recover traceability links automatically between various software artifacts. However, the performance of the IR methods is adversely impacted because of the shortcomings of the software engineer and the IR techniques such as for VSM, it returns low precision due to the fact that many important and correct links are missed while unuseful incorrect links are retrieved (false positives). This research proposed a method that can enhance the performance of IR-based traceability recovery, by achieving high recall and precision as possible. This is due to filtering out false positives and retrieve the relevant and correct links. The major contribution of this research are stated in the following points:

- Proposed a method to enhance the performance of VSM for traceability recovery.

- Assist the software engineers (analysts) in the process of traceability links creation by improving the quality of retrieved links and also reduce time and efforts consumed by the analyst to filter out unwanted links.
- Increase the analyst's confidence in IR method, and also the industrial practitioners to adopt IR-based traceability tool.
- Provide a comparative analysis for using noun phrases, verb phrases, and a combination of noun and verb phrases to enhance IR-based traceability recovery. The result of this experiment can help in understanding and selecting suitable approach.

6.3 Limitations

The limitations that encountered this research is that some of the datasets which have been used in the study of Capobianco et al. 2013, cannot be used in this research because it is in Italian. In addition, the accuracy of POS tagger and chunker may affect the result, the verbosity of each artifact also has an important role in result improvement.

6.4 Future work

Working in this research has revealed many research interests that can be conducted in future work, such as the following:

- Replicating the experiment using different datasets.
- Replicating the experiment using different software artifacts such as methods, source code, test cases, and UML diagram.
- Replicating the experiment using different IR methods.
- Experiment the performance of these indexing strategies in supporting other software engineering tasks.

6.5 Summary

This research proposed a method to enhance the IR-based traceability recovery. A comparative experiment has been conducted to analyze the effectiveness of various indexing strategies including NP, VP, NPVP, and all terms, in enhancing the performance of the IR-based traceability Recovery.

The performance of these strategies in terms of recall and precision has been compared. The results demonstrated that indexing NP tends to outperform indexing VP, NPVP, and all terms. Furthermore, using a certain type of phrases helps to filter out noise and obtain better precision, however, these strategies have effect on recall as some links may be lost.

REFERENCES

Abebe, S. L., & Tonella, P. (2010, June). Natural language parsing of program element names for concept extraction. In *2010 IEEE 18th International Conference on Program Comprehension* (pp. 156-159). IEEE

Ali, N., Cai, H., Hamou-Lhadj, A., & Hassine, J. (2019). Exploiting Parts-of-Speech for effective automated requirements traceability. *Information and Software Technology*, *106*, 126-141.

Al-Saati, N., & Abdul-Jaleel, R. (2015). Requirement Tracing Using Term Extraction. *arXiv preprint arXiv:*1506.08789.

Arunthavanathan, A., Shanmugathasan, S., Ratnavel, S., Thiyagarajah, V., Perera, I., Meedeniya, D., & Balasubramaniam, D. (2016, April). Support for traceability management of software artefacts using Natural Language Processing. In *2016 Moratuwa Engineering Research Conference (MERCon)* (pp. 18-23). IEEE.

Baeza-Yates, R., Ribeiro-Neto, B. (1999). Modern Information Retrieval. *Addison-Wesley, Reading, MA.*

Bavota, G., De Lucia, A., Oliveto, R., & Tortora, G. (2014). Enhancing software artefact traceability recovery processes with link count information. *Information and Software Technology*, 56(2), 163-182.

Borg, M. (2016). Advancing Trace Recovery Evaluation-Applied Information Retrieval in a Software Engineering Context. *arXiv preprint arXiv:1602.07633*.

Borg, M., Runeson, P., & Ardö, A. (2014). Recovering from a decade: a systematic mapping of information retrieval approaches to software traceability. *Empirical Software Engineering, 19(6),* 1565-1616.

Borg, M., Runeson, P., & Brodén, L. (2012). Evaluation of Traceability Recovery in Context: A Taxonomy for Information Retrieval Tools. In T. Baldassarre, M. Genero, E. Mendes, & M. Piattini (Eds.), *16th International Conference on Evaluation & Assessment in Software Engineering (EASE 2012)* (pp. 111-120). IEEE - Institute of Electrical and Electronics Engineers Inc

Brodén, L. (2011). Requirements Traceability Recovery: A Study of Available Tools. *Department of Computer Science, Faculty of Engineering, LTH, Lund University.*

Capobianco, G., Lucia, A. D., Oliveto, R., Panichella, A., & Panichella, S. (2013). Improving IR-based traceability recovery via noun-based indexing of software artifacts. *Journal of Software: Evolution and Process, 25(7)*, 743-762.

Chen, X., Hosking, J., & Grundy, J. (2011, May). A combination approach for enhancing automated traceability (NIER track). *In Proceedings of the 33rd International Conference on Software Engineering (pp. 912-915)*. ACM.

Chikh, A., & Aldayel, M. (2012, December). A new traceable software requirements specification based on IEEE 830. In *2012 International Conference on Computer Systems and Industrial Informatics (ICCSII),* (pp. 1-6). IEEE.

Chowdhury, G. G. (2003). Natural language processing. *Annual review of information science and technology*, *37*(1), 51-89.

Cleland-Huang, J., Czauderna, A., & Hayes, J. H. (2013, July). Using tracelab to design, execute, and baseline empirical requirements engineering experiments. In *2013 21st IEEE International Requirements Engineering Conference (RE)* (pp. 338-339). IEEE.

Cleland-Huang, J., Gotel, O. C., Huffman Hayes, J., Mäder, P., & Zisman, A. (2014, May). Software traceability: trends and future directions. In *Proceedings of the on Future of Software Engineering* (pp. 55-69). ACM.

Cleland-Huang, J., Gotel, O., & Zisman, A. (2012). Software and systems traceability (Vol. 2, No. 3). Heidelberg: Springer

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Ursu, C., Dimitrov, M., ... & Shafirin, A. (2014). Developing Language Processing Components with GATE Version 8: (a User Guide). *University of Sheffield Department of Computer Science.*

Cunningham, H., Tablan, V., Roberts, A., & Bontcheva, K. (2013). Getting more out of biomedical documents with GATE's full lifecycle open source text analytics. *PLoS computational biology*, *9*(2), e1002854.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science,* 41(6), 391.

De Lucia, A., Marcus, A., Oliveto, R., & Poshyvanyk, D. (2012). Information retrieval methods for automated traceability recovery. *In Software and systems traceability (pp. 71-98). Springer, London.*

Diaz, D., Bavota, G., Marcus, A., Oliveto, R., Takahashi, S., & De Lucia, A. (2013, May). Using code ownership to improve ir-based traceability link recovery. *In 2013 21st International Conference on Program Comprehension (ICPC)* (pp. 123-132). IEEE.

Dit, B., Moritz, E., & Poshyvanyk, D. (2012, June). A tracelab-based solution for creating, conducting, and sharing feature location experiments. In *2012 20th IEEE International Conference on Program Comprehension (ICPC)* (pp. 203-208). IEEE.

Egyed, A., Graf, F., & Grünbacher, P. (2010). Effort and quality of recovering requirements-to-code traces: Two exploratory experiments. *In: IEEE International Conference on Requirements Engineering, pp. 221–230*. IEEE Computer Society, Los Alamitos, CA

Etzkorn, L. H., Bowen, L. L., & Davis, C. G. (1999). An approach to program understanding by natural language understanding. *Natural Language Engineering*, *5*(3), 219-236.

Gethers, M., Oliveto, R., Poshyvanyk, D., & De Lucia, A. (2011, September). On integrating orthogonal information retrieval methods to improve traceability recovery. In *2011 27th IEEE International Conference on Software Maintenance (ICSM),* (pp. 133-142). IEEE.

Gotel, O. C., & Finkelstein, C. W. (1994, April). An analysis of the requirements traceability problem. In *Proceedings of IEEE International Conference on Requirements Engineering* (pp. 94-101). IEEE.

Hayes, J. H., Dekhtyar, A., Larsen, J., & Guéhéneuc, Y. G. (2018). Effective use of analysts' effort in automated tracing. *Requirements Engineering*, *23*(1), 119-143.

Hayes, J. H., Dekhtyar, A. & Sundaram, S. K. (2005). Improving after-the-fact tracing and mapping: Supporting software quality predictions. *IEEE software*, (6), 30–37. IEEE.

Kang, N., van Mulligen, E. M., & Kors, J. A. (2011). Comparing and combining chunkers of biomedical text. *Journal of biomedical informatics*, *44*(2), 354-360.

Kchaou, D., Bouassida, N., Mefteh, M., & Ben-Abdallah, H. (2019). Recovering semantic traceability between requirements and design for change impact analysis. *Innovations in Systems and Software Engineering*, *15*(2), 101-115.

Keenan, E., Czauderna, A., Leach, G., Cleland-Huang, J., Shin, Y., Moritz, E., & Dekhtyar, A. (2012, June). Tracelab: An experimental workbench for equipping researchers to innovate, synthesize, and comparatively evaluate traceability solutions. In *2012 34th International Conference on Software Engineering (ICSE)* (pp. 1375-1378). IEEE.

Koehrsen, W. (2018). Beyond Accuracy: Precision and Recall. *Towards Data Science*. Retrieved from https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c

Kuang, H., Nie, J., Hu, H., Rempel, P., Lü, J., Egyed, A., & Mäder, P. (2017, February). Analyzing closeness of code dependencies for improving IR-based Traceability Recovery. *In 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering* (SANER) (pp. 68-78). IEEE.

Liddy, E.D. (2001). Natural Language Processing. *In Encyclopedia of Library and Information Science, 2nd Ed. NY*. Marcel Decker, Inc.

Li, Y., & Cleland-Huang, J. (2013, May). Ontology-based trace retrieval. In *2013 7th International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE)* (pp. 30-36). IEEE.

Mahmood, K., Takahashi, H., & Alobaidi, M. (2015, March). A semantic approach for traceability link recovery in aerospace requirements management system. In *2015 IEEE Twelfth International Symposium on Autonomous Decentralized Systems (ISADS),* (pp. 217-222). IEEE.

Mahmoud, A., & Niu, N. (2015). On the role of semantics in automated requirements tracing. *Requirements Engineering*, *20*(3), 281-300.

Nyamisa, M., Mwangi, W., & Cheruiyot, W. (2017). A Survey of Information Retrieval Techniques. *Advances in Networks, 5(2), 40. Retrieved from* http://www.sciencepublishinggroup.com/journal/paperinfo?journalid=131&doi=10.11648/j.net.20170502.12

Panichella, A., De Lucia, A., & Zaidman, A. (2015, May). Adaptive user feedback for ir-based traceability recovery. *In Proceedings of the 8th International Symposium on Software and Systems Traceability* (pp. 15-21). IEEE Press.

Panichella, A., Dit, B., Oliveto, R., Di Penta, M., Poshyvanyk, D., & De Lucia, A. (2016, March). Parameterizing and assembling IR-based solutions for SE tasks using genetic algorithms. *In 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER),* (Vol. 1, pp. 314-325). IEEE.

Panichella, A., McMillan, C., Moritz, E., Palmieri, D., Oliveto, R., Poshyvanyk, D., & De Lucia, A. (2013, March). When and how using structural information to improve ir-based traceability recovery. In *2013 17th European Conference on Software Maintenance and Reengineering (CSMR),* (pp. 199-208). IEEE.

Pinheiro, F. A. (2004). Requirements traceability. *In Perspectives on software requirements* (pp. 91-113). Springer US.

Rago, A., Marcos, C., & Diaz-Pace, J. (2017). Using semantic roles to improve text classification in the requirements domain. *Language Resources and Evaluation*, 1-37.

Saiedian, H., Kannenberg, A., & Morozov, S. (2013). A streamlined, cost-effective database approach to manage requirements traceability. *Software Quality Journal*, *21*(1), 23-38.

Sayyad Shirabad, J. and Menzies, T.J. (2005) The PROMISE Repository of Software Engineering Databases. *School of Information Technology and Engineering, University of Ottawa, Canada . Available:* http://promise.site.uottawa.ca/SERepository

Shin, Y., Hayes, J. H., & Cleland-Huang, J. (2015). Guidelines for Benchmarking Automated Software Traceability Techniques. *Proceedings - 2015 IEEE/ACM 8th International Symposium on Software and Systems Traceability, SST 2015*, 61–67.

Sultanov, H., & Hayes, J. H. (2010, September). Application of swarm techniques to requirements engineering: Requirements tracing. In *2010 18th IEEE International Requirements Engineering Conference* (pp. 211-220). IEEE.

Sundaram, S. K. (2007). Requirements Tracing Using Information Retrieval. *University of Kentucky.*

Sundaram, S. K., Hayes, J. H., & Dekhtyar, A. (2005, May). Baselines in requirements tracing. In *ACM SIGSOFT Software Engineering Notes* (Vol. 30, No. 4, pp. 1-6). ACM.

Sundaram, S. K., Hayes, J. H., Dekhtyar, A., & Holbrook, E. A. (2010). Assessing traceability of software engineering artifacts. *Requirements engineering, 15(3)*, 313-335.

Thijs, B., Glänzel, W., & Meyer, M. S. (2015, June). Using noun phrases extraction for the improvement of hybrid clustering with text-and citation-based components. The example of "Information Systems Research". In *Proc. of the Workshop Mining Scientific Papers: Computational Linguistics and Bibliometrics, 15th International Society of Scientometrics and Informetrics Conference (ISSI), Istanbul, Turkey* (Vol. 1384, pp. 28-33).

Wang, X., Xue, X., & Chu, S. (2016). Towards Supporting Feature Location Using Syntactic Analysis. *Journal of Information Hiding and Multimedia Signal Processing, 7(1),* 115-126.

Winkler, S., & Pilgrim, J. (2010). A survey of traceability in requirements engineering and model-driven development. *Software and Systems Modeling (SoSyM)*, *9*(4), 529-565.

Zhang E., Zhang Y. (2009) Average Precision. In: LIU L., ÖZSU M.T. (*eds) Encyclopedia of Database Systems*. Springer, Boston, MA

Zhao W, Zhang L, Liu Y, Luo J, Sun JS (2003). Understanding how the requirements are implemented in source code. In*: Proceedings of the 10th Asia-Pacific software engineering conference*, pp 68–77

Zhou, X., & Yu, H. (2007, March). A clustering-based approach for tracing object-oriented design to requirement. In *International Conference on Fundamental Approaches to Software Engineering* (pp. 412-422). Springer, Berlin, Heidelberg.

Zou, X., Settimi, R., & Cleland-Huang, J. (2010). Improving automated requirements trace retrieval: a study of term-based enhancement methods. *Empirical Software Engineering*, *15*(2), 119-146.

Zou, X., Settimi, R., & Cleland-Huang, J. (2006, September). Phrasing in dynamic requirements trace retrieval. In *30th Annual International Computer Software and Applications Conference (COMPSAC'06)* (Vol. 1, pp. 265-272). IEEE.