UNIVERSITY OF TARTU

Faculty of Science and Technology

Institute of Computer Science

Olumide Olugbenga Oluyide

Causality Management and Analysis in Requirement Manuscript for Software Designs

Master's Thesis (30 ECTS)

Supervisor: Ishaya Peni Gambo, PhD

Tartu, 2023

Causality Management and Analysis in Requirement Manuscript for Software Designs

Abstract:

For software design tasks involving natural language, the results of a causal investigation provide valuable and robust semantic information, especially for identifying key variables during product (software) design and product optimization. As the interest in analytical data science shifts from correlations to a better understanding of causality, there is an equal task focused on the accuracy of extracting causality from textual artifacts to aid requirement engineering (RE) based decisions. This thesis focuses on identifying, extracting, and classifying causal phrases using word and sentence labeling based on the Bi-directional Encoder Representations from Transformers (BERT) deep learning language model and five machine learning models. The aim is to understand the form and degree of causality based on their impact and prevalence in RE practice. Methodologically, our analysis is centered around RE practice, and we considered 12,438 sentences extracted from 50 requirement engineering manuscripts (REM) for training our machine models. Our research reports that causal expressions constitute about 32% of sentences from REM. We applied four evaluation metrics, namely recall, accuracy, precision, and F1, to assess our machine models' performance and accuracy to ensure the results' conformity with our study goal. Further, we computed the highest model accuracy to be 85%, attributed to Naive Bayes. Finally, we noted that the applicability and relevance of our causal analytic framework is relevant to practitioners for different functionalities, such as generating test cases for requirement engineers and software developers and product performance auditing for management stakeholders.

Keywords: software, causality management, causality extraction, causal effects, machine learning, deep learning, requirement engineering, natural language processing

CERCS: P170 Computer science, numerical analysis, systems, control

Kausaalsuse juhtimine ja analüüs nõuete juhendites tarkvara disainideks.

Lühikokkuvõte:

Tarkvara disainide ülesanneteks, mis sisaldavad loomulikku keelt, kausaalsuse uuringu tulemused tagavad väärtuslikku ja kõikehõlmavat semantilist informatsiooni, eriti peamiste muutujate identifitseerimiseks toote tarkvara disaini ja optimeerimise jooksul. Kuna huvi analüütilise andmeteaduse vastu nihkub korrelatsioonitest kausaalsuse parema mõistmisele, on olemas võrdne ülesanne keskendunud kausaalsuse väljavõtte täpsusele tekstilistest esemetest, et abistada nõuete inseneerimise põhinevaid otsusi.Käesolev magistritöö keskendub kausaalsete fraaside identifitseerimisele, väljavõttele ja rühmitamisele kasutades sõna ja lause märgistamist, mis põhineb transformerite kahesuunaliste kodeerijatele (BERT) sügava õppimise loomuliku keele mudelil ning viiel masinõppe mudelil. Eesmärgiks on mõista kausaalsuse vormi ja astet, mis põhinevad oma mõjul ja levimisel nõuete inseneerimise praktikal. Metodoloogiliselt, meie analüüs keskendub nõuete inseneerimise praktikale, arvestati 12 438 lausega välja võtnud 50 nõuete inseneerimise juhenditest (REM), et harjutada meie masinate mudeleid. Meie uuring teatab, et kausaalsed väljendid moodustavad umbes 32% REM lausetest. Me rakendasime nelja hindamismõõdikut nimelt meenutamine, täpsus, kordustäpsus ja F1, et hinnata meie masina mudelite jõudlust ja täpsust, et tagada tulemuste vastavust meie uuringu eesmärgile. Lisaks, arvutasime kõrgeima mudeli täpsus on 85%, mis omistati Naive Bayes. Lõpuks, peab märkama, et kohaldatavus ja kausaalsuse analüütilise raamistiku asjakohasus sobiks erinevatele praktikutele, näiteks, nõuete inseneridele ja tarkvaraarendajatele, et genereerida testjuhtumeid. Samuti juhtkonna sidusrühmad võiksid kasutada seda, et tulemusauditi paremaks teha.

Võtmesõnad: tarkvara, kausaalsuse juhtimine, kausaalsuse väljavõtte, kausaalsed tagajärjed, masin, õppimine, sügav õppimine, nõuete inseneerimine, loomuliku keele

tootmine.

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine

Contents

List of Figures

List of Tables

1 Introduction

Sentences incorporating causal relations, such as "*A push notification shall be displayed if the system has completely processed the data*" describes the desired action of the system. Interestingly, causal expressions are inherent in most requirement engineering manuscripts (REM), such as software requirement specification (SRS) manuscripts. Acknowledging the full scope of use and detecting and efficiently extracting the causal relationships has significant application potential, particularly in requirements engineering (RE) practice. Some of the applications include generating automated test cases and enhancing reasoning regarding inter-requirement components [2, 3]. Extracting causality from text artifacts is regarded as a relation-based extraction problem which is primarily centered on identifying causal phrases containing cause and effect

In RE, functional requirements describe a system from three major standpoints; First is the input that is to be processed by the system, second is the action undertaken by the system in processing the input, and lastly, the processed information that is given as output by the system. These three procedures usually follow a causal pattern between their links, such as if x then y. Retrieving causal relations from natural language texts facilitates different analytical activities and is already applied for information retrieval and prediction tasks, as highlighted in Chapter 2 of this thesis.

However, two key factors are responsible for the challenges encountered in extracting causal relations in RE artifacts. Firstly, although the use of natural language (NL) in RE tends to foster easy re-usability for further theoretical implementation, but unregulated natural language is still extensively used in RE. As a result, extracting the key and relevant information from requirement artifacts becomes complicated, primarily attributed to the underlying ambiguity and complexity inherent in NL (textual) representation. Secondly, causal relations are in different formats, e.g., implicit or explicit (i.e., clearly presenting the cause and effect) and unmarked or marked (when a cue expression describes a causal

relation). Thus, there could be difficulty in identifying and extracting cause as well as effect. Notably, many existing approaches for extracting cause and effect from NL-based requirement artifacts still fall short of effective and trustworthy use, especially in practice, due to performance factors.

Making remarks to this observation, we assert that an innovative, trustworthy, and high performance-based approach to detect and extract causal relations is required for effective consideration and wide acceptability of causality application in RE. Therefore, in this study, we are concerned with using a variety of machine learning algorithms (described ion Chapter 3) and deep learning framework[1] to explore the REMs that are extracted from different sources (data described in Section 3.6 of Chapter 3). By so doing, our experiment is able to investigate and categorize the REMs into different causality types. In addition, the detection of causality by the machine models implemented in this study assisted us in measuring the impact and prevalence of causality, particularly in RE studies, as it applies to its applicability to practitioners in the field of software engineering. Finally, we used standard evaluation techniques (recall, accuracy, precision, and F1 described in Chapter 3) to assess the results of our machine models and the deep learning model. By using the evaluation techniques, we can compare the performance of the computational frameworks implemented in this study and arrive at a conclusion based on applicability to the software RE practice.

1.1 Overview and Categorization of Causality

The notion of causality can be introduced as a relation between two proportional variables referred to as antecedent h_1 and consequence h_2, which could be expressed as h_1 causes h_2 [4]. However, natural language-based texts have many correlations, which appear in different forms [3, 4]. Extracting causality in natural language texts could be difficult because of factors of different human and grammatical-interpretation-based errors. It

[1] https://en.wikipedia.org/wiki/Deep_learning

is worth noting that causality has no boundary regarding its identification in a sentence, as it can occur in a single, multi-line, simple, and complex sentence formation [5]. To properly convey the notion of causality, we must define the two keywords that underpin the notion of causality: cause and effect.

A ***cause*** according to the Macmillan dictionary[2] is simply defined as "*the ability to make something happen*", while an ***effect*** is defined as *a change produced in a thing or object*. Below, we present examples of the different forms of causalities identified in sentences, which exemplify the variation in language constructs.

(i) **Single cause to single effect:** This type of causal relation exists in sentence(s) in which there is only one cause and one effect. For example, if we examine the statement *the app crashed because of the bugs which were not fixed*, it is clear that there exists one cause, i.e., the "*bugs*", and one effect, i.e., the "*crash*". More clearly, the position of the cause and effect in a sentence could be interchanged, i.e., the cause can precede the effect and vise versa. In the above example, the effect preceded the cause, while in the example below, the cause preceded the effect, i.e., *"the unfixed bugs triggered the app*[3] *to crash"*.

(ii) **Multiple cause to multiple effects:** This type of causal relation exists where there are more than one cause and more than one effect in a sentence. Similarly, the position of the cause and effect in the sentence is not static as the cause can either precede the effect and vise versa. Below is an example of multiple cause multiple effect "*the unfixed bugs, as well as application programming interface (API) communication protocol, triggered the freezing of the app which finally resulted in the crashing of the app*".

(iii) **Causal to effect Loop:** This type of causal relation exists in a sentence where the cause resulted in a series of recurring effects. For example, in the sentence

[2]https://www.macmillandictionary.com/
[3]an executable software product

"*the absence of the software testers to evaluate product upgrade resulted in long duration of product-acceptance stage, thereby causing reduction in customers patronage and has resulted in low revenue for the company*". In this example, the single cause is identified as "*absence of the software tester*", and this led to a loop of effect identified "*delayed product-acceptance stage*", "*reduction of customers*", and "*reduction in company revenue*". Therefore, the effect loop will continue as long as the software testers are unavailable to test the product's upgrade, which leads to customers getting unsatisfied with the existing product functionality, which leads to reduced customer patronage, which finally leads to a decrease in the revenue of the company. This example can also be called a single cause to multiple effect causal relation.

1.2 Quality Measure of Requirement Engineering Manuscripts (REM)

Low quality of REM could have severe impact during and after software development life cycle, especially for robust or large-requirement-based projects [6]. Deficiencies such as ambiguities or insufficient requirements in REM can be computationally costly in terms of project duration and resources (e.g., computational and financial). Software developers (referred to as practitioners in this study) should ensure quality requirements to have a rich and balanced project life cycle and final deliverable.

Some of the challenges require adequate awareness of domain knowledge because, without a proper understanding of the domain knowledge, it could be difficult to determine whether a requirements manuscript is complete and is good enough to be used as an implementation road-map during software development process and acceptance stage of the software product. For example, the word *sufficient* in a typical statement such as: "*sufficient transition should be allowed between order interface and checkout interface*" could have different connotation (in terms of motive-of-use rather than grammatical meaning) to different requirement evaluator. To one evaluator, it could imply duration,

while to another evaluator, it could imply quantity. The primary aim of ensuring quality in REM is to prevent ambiguity and misinterpretation of concepts but foster clarity in defining REM statements.

Quality control of REM primarily relies extensively on review-loop to discover any hidden or obvious inadequacies. However, reviews of REM must involve the key, and applicable stakeholders who will personally examine and identify the uncertainties in the manuscripts [7]. The examination task could be difficult and rigorous to undertake, especially if no *defacto* or *dejure* standard is put in place as a standard. However, the reviewers should have in-depth domain knowledge and evaluation expertise for the requirement examination task. By so doing, analysis of causalities in REM will be uncomplicated, which thus ensures maximum benefit that could be derived from such practice.

1.3 Research Problem Statement

Causality in REM appears in various forms (such as implicit or explicit, marked or unmarked, ambiguous or unambiguous cue expressions [8]), and different techniques and frameworks have been implemented for extracting causalities in REM. One of the challenges we identified is the low performance threshold of the existing causality extraction techniques (details given in Chapter 2). Some studies used hand-coded rule-based techniques (e.g., Christopher *et al.* in [9], and Jose et al. in [10]), which implies that their framework is restricted to the defined linguistic pattern rules and not dynamic to fit the constantly emerging advancement in causality studies, especially, as it relates to the field of RE. In some other studies, they considered a computational framework that is based on cue phrases to establish the probability of causal relations from sentences (e.g., Sanda *et al.* in [11] and Chang *et al.* in [12]). The problem with this technique is that using cue phrase functions at the word level may not be sufficient to establish a robust causality structure. Therefore valuable details relating to the cause and effect of causality

may not be adequately captured, causing less applicability to practitioners' use cases.

Further studies were made to improve the cue phrases that are dependent on word level. They considered using cues based on phrase level using a bi-directional Long short-term memory framework (e.g., Zhaoning *et al.* in [13] and Rupsa *et al.* in [14]). The challenge with this approach is that the performance of the model is based on the domain of the corpora used for training. Hence, its application is domain-based, and the model will be re-trained for its use per domain, especially for RE tasks. Therefore, our study addresses these challenges by considering the extraction of a complete and embedded causal relation in order to find a robust application in dependency detection and test case generation in RE practice. To achieve this, we implemented an approach that is based on machine learning and deep learning frameworks *(details provided in Chapter 3)*. By using these two techniques, we are able to evaluate and compare their performance as it relates to addressing our study goals.

1.4 Research Justification

Our research aims to strengthen the RE practice, which in turn has a ripple effect on software products. Practitioners, who in the sense of our study refer to the requirement engineers, software developers, and managerial stakeholders, should be able to estimate factors that improve or reduce the quality of the software product that they administer. Therefore, to achieve this, one of the vital procedure is for practitioners to carefully analyze the software requirement manuscript of their software product with regard to the current deliverables of their application. By so doing, factors influencing the overall capability of their application are uncovered. With this knowledge, practitioners could formulate necessary activities that lead to restructuring and re-strategizing RE techniques to maximize the impact of their software product on end users. In addition, the impact of the updated restructuring can be measured with regard to previous RE techniques, leading to improved productivity of the practitioner and a better version of the software

product. Hence, our study is in the right direction, as we present applicable techniques via computational frameworks that are capable of harnessing the realization of these goals.

1.5 Research Scope

Identifying causal information in textual data is a critical language processing task with various forms of applications [13, 15]. Causal information is present in most manuscripts, such as documentation, articles, and social contents [16]. This implies that causality management is a branch of scientific study that should be given utmost attention because of its application to different domains with regard to RE. However, in this study, we are concerned about managing causality as it applies to the specifics of software requirement engineering. To this end, we provide computational techniques to detect causalities in REM, estimate the degree of prevalence in the REM, and evaluate the overall impact of causality in the deliverable of software products. These three variables explain the notion of causality management that we considered in this study. Further, our study is not concerned about the technology or framework that should be applied to achieve the solution recommendation from our causality management framework. Rather, we are more particular about investigating how causal inference affects the software product and also guide practitioners to instigate necessary actions in their RE practice.

1.6 Research Aim and Objectives

The aim of this study is to provide a computational technique that is based on machine learning and deep learning framework to identify causality in REM, justify the existence of causality in REM, and evaluate the impact of causality in REM. In essence, this study will benefit the RE structure of practitioners involved in software development projects of different level of magnitude. Further, the applicability of our findings will help to produce better version of software products with improved functionalities. More

clearly, the following are the objectives of this study; (i) We retrieve REM (dataset) from different domains (as shown in Table 2) to software engineering domain to enable broad and robustness of our study findings to the field of RE in software development. Secondly, we examine the dataset by performing exploratory data analysis to bring the dataset to a standard required to fit our computational models. Thirdly, we simulate our computational models by consolidating different parts and configuring the parameters for optimal result. Fourthly, we presented our model result, explain our findings and evaluate our computational frameworks to check if our study goal is satisfied, especially with regards to the influence of causality on REM of software products to the practitioners.

1.7 Research Questions

Our study goal is to identify and analyze the prevalence of causality in REM which will help solidify the strength of REM for software development task, which is dependent on the scope, complexity level, and structure of causality (described in Section 3.1 of Chapter 3. Therefore, for us to fulfill our study goal, we considered the following research questions, the answers to which influence our research direction.

- **RQ1:** To what extent is causality present in REM?
- **RQ2:** How complex is it to identify and establish causality in REM?
- **RQ3:** What is the level of causal ambiguity in requirements?
- **RQ4:** How frequent does causal relations (such as *enable, prevent*, and *cause*) appear when analyzing requirement manuscript?
- **RQ5:** What is the degree of influence of temporal relations (such as *during, overlap*, and *before*) in REM?

The answers to these research questions are elaborated in our result analysis presented in Chapters 4 and specifically presented in Section 4.3. Notably, answering RQ2 and RQ3

entails addressing some other sub-questions which are key to understand the direction, impact, and application of this study to the requirement engineers and other practitioners in the field of software development.

1.8 Research Methodology Overview

In this section, we present the overview of our research methodology by describing the research instruments, computational tools, evaluation techniques, and relevance of our findings. To begin with, we retrieved our dataset from Kaggle[4], a publicly available data repository, which contains over 675 total collections of REM in which 50 random REM were extracted (statistics of dataset is presented in Table 2). Due to the unstructured and unorganized format of the dataset (primarily made up of natural language composition), we performed the necessary exploratory data analysis task such as data cleaning to bring the dataset to an acceptable format required of the machine models that were implemented in this study (details of the exploratory data analysis is presented in Section 3.6.2 of Chapter 3). Further, an annotation procedure was performed on the dataset by seven individuals (who are data scientists and friends to the researcher) chosen from three Universities (two individuals from University of Tartu (UT)[5] in Estonia, three individuals from University of Witwatersrand (WITs)[6] in South Africa, and two individuals from Obafemi Awolowo University (OAU)[7] in Nigeria. The goal of the annotators is to distribute the 12,438 extracted dataset into different categories based on the guideline that we provided.

Next, We applied two computational frameworks namely machine learning algorithm and deep learning technique for investigating the occurrence and impact of causality in REM. The machine learning algorithms were trained on the aforementioned extracted

[4]https://www.kaggle.com/code/kmader/quickdraw-with-wavenet-classifier
[5]https://ut.ee/en
[6]https://wits.ac.za
[7]https://oauife.edu.ng

dataset and are configured with the optimal hyperparameters (presented in Table 9). The machine learning algorithms have been proven by researchers to handle classification task efficiently (e.g. as implemented by Patel *et al.* [17] and Nigam *et al.* [18]. Hence, we are sure about their capability for causality prediction task based on their learning experience from their training process. The machine learning algorithms explored in this study are listed as follows: Support Vector Machine (SVM), Naive Bayes (NB), Decision Tree (DT), Random Forest (RF), K-Nearest Neighbor (KNN), and Adaptive Boosting -AdaBost (AB).

In addition, we applied the deep learning framework (described in Section 3.10 of Chapter 3) configured with optimal hyperparameters (presented in Table 9) for causality classification task. Both the machine learning models and deep learning models used for the prediction task were all evaluated using the standard computational evaluator techniques namely: recall, accuracy, precision, and F1 score. By using different different machine learning and deep learning algorithms, we are able to identify the best performing model and rank their performance based on our evaluation techniques. The evaluation result of the computational models implemented in this study is presented in Chapter 4. The details of the machine learning algorithm, deep learning framework, annotation procedures as well as their evaluation strategies are presented in Chapter 3. In Chapter 4, we present and discuss the results obtained from the implemented models and how they were able to meet our research goal and answer our research questions. The implementation and research procedures followed in this study is described by Figure 5 in Chapter 3.

1.9 Research Contribution

For every notable research, contribution to the body of knowledge is a vital part of the research that cannot be overemphasize. Our study agree with this notion. One of the major challenges relating to causality extraction is the ability to identify if causality exists in a

requirement engineering manuscript. This implies that presence of causality determines which sentence will be extracted, hence, sentences where causality is not identified will be not be considered from the corpus. Another key challenge in causality extraction is to have a rich technical and theoretical *know-how* and understanding pertaining to extracting the identified causality. To address these concerns, it is important to understand the scope, complexity level, and structure of causality in RE practice. Accurate insights about the distribution of causality is required before implementing workable and applicable frameworks for detecting and extracting causal relations. Therefore, this study aim to close these gaps and make applicable contributions to the domain of knowledge in the following ways:

(i) **Identifying Causalities:** Our study presents an analytical description of the scope, complexity level and structure of causality. The analytical description is based on 12,438 sentence retrieved from 50 REM with 18 distinct domains detail is presented in Table 2. Based on our analysis, we assert that the common form of causality in REM is the marked and explicit type. In addition, we also noted that 23% of the sentence has information relating the predicted system behaviour. Therefore, we are encouraged about the relevance of this study to causality management in REM.

(ii) **Causality Extraction Technique:** In this study, we implemented a robust framework that is capable of extracting causalities from REM. Our model was trained using the dataset described in Section 3.6. Next, we performed the model evaluation where we achieved F1 score of 30%. When we compared our framework with existing causality detection models (which majorly depends on cue expressions or un-optimized machine learning models), our framework yielded 20% performance improvement in recall and 22% in precision when compared vis-a-vis existing causality extraction models. Further details relating to the evaluation of the existing causality extraction models is presented in Chapter 2.

(iii) **Causality Impact on REM:** We presented the results of a investigative scenario by examining the relationship between causal occurrence and its influence on implementation pipeline as highlighted on the REM. This analysis is not just illustrating a possible application of causality awareness technique, but additionally validated the impact of causality on RE to practitioners, particularly to the software developers and the SRS evaluators.

(iv) **Support for Framework Reproducibility and Open Dataset:** We provided the cloud-hosted uniform resource locator (URL) path which contains the implementation details of this study to foster replication of the implementation framework. In addition, we presented the URL to the datasets that was used for training and evaluating the model. The path for both the implementation source code, annotation guide and the dataset is presented in of Chapter 5.

1.10 Thesis Structure

The remainder of this thesis is structured as follows. Chapter 1 presents an introductory layout and summary of the different parts of this study. In Chapter 2, we highlighted the computational frameworks, strengths, and limitations of existing studies in this research field. Chapter 3 discusses the research instrument and computational tools that were applied to meet our research objectives. Further, the analysis of our findings is presented in Chapter 4, where we analyzed and evaluated our findings vis-a-vis the study goal. The discussion of the result analysis of our study findings is presented in Chapter 5, while Chapter 6 highlights the observed threats to validity of our study. Finally, Chapter 7 concludes this study by giving a concise summary of our research, the study's contribution to the body of knowledge, and recommendations for further studies relating to our research.

1.11 Definition of Terms

Definition of Terms:

Practitioners: these are individuals that are in charge of the requirement management and product (software) design

Software Requirement Specification: refers to the descriptive features of a software

Requirement Engineering Manuscript: refers to the document containing the specification and configuration details of a software product

2 Literature Review

In Chapter 1, we introduced this research by presenting various aspects and components required for consolidating this research study. This chapter will provide an overview of existing studies related to our research. Our overview will capture the implementation framework, the focus of the study, and the gaps that we identified in the study as it relates to causality management in requirement engineering (RE) practice. This review procedure will also allow us to compare the relevance of our study with regard to the existing research that has been carried out. Identifying the occurrence and impact of causality in requirement engineering manuscripts (REM) is one of the significant factors for determining the efficiency and effectiveness of a good RE practice, especially for software developers and software requirement specification (SRS) evaluators. Hence, reviewing related studies is essential for a progressive RE practice, which was considered in this study. Our review captured the application of causality both within and outside the RE practice described below.

2.1 Causality Studies outside the Field of Requirement Engineering

The first study that we evaluated and which was also one of the early applications to causality outside the RE field is the study of Roxana et al. [19] and Hideki *et al.* [20] where they implemented a lexicon-based pattern that operates in a single or multiple sentences. Their synthetic pattern consists of two noun phrases (NP) which are linked together with a single causative verb (VP), i.e., <NP_i VP NP_j>. They developed their pattern by exploring the WordNet[8] dictionary to search for NP that are connected with VP (or *cause-to* relationship) that are clearly defined in the WordNet corpus. They labelled the VP connecting the NP as causative verbs. Based on the extracted VP, which was consolidated with other WordNet features, they were able to develop lexicon-based rules

[8]https://en.wikipedia.org/wiki/WordNet

that are capable of detecting causality in studies relating to question and answer. A significant limitation of their study is that their computational framework may not be able to capture new inventions of NP and VP. Hence, the relevance of their approach may be limited to being applied to novel question and answering-based systems.

Another application of causality studies outside the RE field is the research undertaken by Choi *et al.* [21], which was similar to the study of Marcu *et al.* and Kamal *cite23* expand on the work of Roxana et al. [19] and Hideki *et al.* [20] by taking the probabilities of the cue phrases and conceptual pairs as additional factors for identifying causalities in a question and answer-based system. Their application focused primarily on the reason why questions are expanded further in a question-and-answer-based system by exploring the probabilities of the cue phrases and conceptual pairs not only at a word level (as considered by Roxana et al. [19] and Hideki *et al.* [20]) but also on a sentence level. By using this, they were able to expand on the result of Roxana et al. and Hideki *et al.* for further usage and application to their field of reference.

Furthermore, another reference to one of the early methods of causal derivation and usage beyond the RE field is the work of Kentaro *et al.* [22], where they explored the field of communication and media by extracting causal relation from newspaper articles. They primarily considered using cue phrases to extract causal relations from their dataset. This kind of study classifies causality-based individual perception of a cue phrase in the sentence by majorly identifying a cause and the effect. So, identifying causality in this sense will consider four factors: means, precondition, cause, and effect. We noted that using only cue phrases to extract causality may not be sufficient for robust analysis, especially if there are complex and ambiguous words (or expressions) in the corpus being considered. This notion was evidenced by Jinghang *et al.* [23] in their research, where they acknowledge the limitation of the cue-phrase-based causality extraction technique by examining and summarizing causality in the dataset.

It is interesting also to note that apart from the field of software engineering, causality

has also been studied in the medical field. This was evidenced by Christopher *et al.* [24] in their study, where they implemented a graphical-based pattern to extract causal information from a medical database containing health diagnoses of individuals. In their study, they structured causality attributes and roles using a template built on three-layer computational architecture for manually detecting patterns graphically. They reported that a significant limitation of their study is the inability of their causality-detection framework to be dynamic to trend updated reporting medical information. Hence, they have to code new structures for every new trend published in the medical line. A similar study to that of Christopher *et al.* [24] is that of Manabu *et al.* [25] where they explored tweets from twitter[9] (a social network platform) database by using dependency tree and part-of-speech (PoS) tags to retrieve causal relations that is manually coded. Their implementation was built to function on the word level, not the sentence level. They reported that their study goal was to identify why users tweet more about some topics or hashtags than others.

Another interesting application of causality is in the field of economics, where they were able to use causality to detect the relations between factors, items, and policy that influences the market structure and market value of goods and services, as evidenced by the work of Kentaro *et al.* [26]. Their causality identification framework was built on analyzing the semantic relation in noun phrases. They reported how causality study provided insight into market research analysis for economic advantage by examining factors that lead to economic spikes or recession. Finally, we would like to note that causality has a wide range of applications beyond RE application *(as proven by the studies mentioned earlier)*, as it can be applied in most fields of study. Knowing what to do and the "*how*" is essential in causality study, and this largely depends on the insights drawn from such research.

[9]https://twitter.com/

2.2 Causality Studies in the Field of Requirement Engineering

There has not been much study of causality in the field of RE, but the interest is growing and promising in the nearest future. One of the reasons for the lack of interest in studying causality in RE, especially in natural language requirements, can be attributed to a poor understanding of the concept due to the cumbersome tasks involved in detecting reliable causal relations [27]. As observed by Frattini *et al.*, [27], the causal relations express powerful and meaningful (semantic) information, which, when extracted, can be used to generate more test cases to aid the software design and testing process.

However, one of the key motivations for studying causality is to be aware of some of the benefits of causality and how it can influence the goal of a RE practice. Below, we highlighted some existing studies in this line of research as well as presented an overview of their implementation and findings. First, based on the insight of Nabiha *et al.* [28], causality detection in natural language manuscripts can be categorized into two forms, namely: (i) rule-based; and (ii) feature-based method. The rule-based approach is based on a defined pattern formulated by the researcher based on their domain knowledge, research expertise, or intuition. An example of the rule-based approach is the work of Joanne *et al.* [29] and Phillip *et al.* [30]. One of the major disadvantages of the rule-based approach is the low performance threshold which is attributed to the inability of the defined rules to be applied to extended and novel innovations. Hence, the rule-based is lowlily applied in RE practice.

Similarly, in contrast to the rule-based approach, some studies applied the concept of the feature-based method by using the technique of transfer learning inherent in Bidirectional Encoder Representations from Transformers models (BERT)[10]. An example of the feature-based technique is the research conducted by Manolis *et al.* [31] where they applied a transfer learning technique by using Embeddings from Language Model

[10]https://en.wikipedia.org/wiki/BERT_(language$_model$)

(ELMO)[11] and BERT model and were able to overcome the challenge of identifying causal inference using a bi-directional gated recurrent units (GRU) configured with self-attention. They used a publicly available dataset for their experiment. They reported that transfer learning could perform well only in a small amount of dataset. At the same time, GRU configured with self-attention performed well with large datasets during the causality extraction task.

Another interesting study which explores causality was carried out by Frattini *et al.* [27], where they developed CiRA for detecting causality in software requirement documents. One of the major limitations of their study approach is that they generalized their results and its practical use based on the outcome of one dataset (from a single domain). Relating to their study findings, the impact of causality in one domain could be totally different from its impact in another domain, especially considering the market size of the product's end users. Another limitation to the study of Frattini *et al.* [27] is the duration of dataset that was used for their modeling process. They only considered six (6) years (2010 to 2015), which may not be sufficient to adequately capture model requirement for an optimal output. Converse to this, our study explored dataset for eighteen (18) years (2005 to 2022) in order to have a robust understanding and development of causality over different years. The findings from the long range of the datasets allowed us to study and analyze how causality in software designs evolve from year to year and from one decade to another. Empirically, we did not generalize the impact of our causality tool to one domain

Reklos *et al.* [32] explored a different approach in their study as they tackled the entity identification sub-tasks that are inherent in causal relations during the RE process. Therefore, they implemented an ontological BERT model to detect entities involved in causal relations. They trained their model and validated it against a novel dataset where they achieved an F1-score of 0.8. Theirs was instrumental in entity detection

[11] https://en.wikipedia.org/wiki/Elmo

in causal relations, and with their framework, they could extract causal relations from new collections of the dataset for a RE task. Gupta *et al.* [33] also conducted research by implementing a domain-based BERT model to identify whether causality exists in a financial-based manuscript. Their study was used to identify and predict probable risk and its cause in developing a financial analysis framework. They noted that one of the major challenges in their study is the imbalanced dataset used for training their BERT model. However, they used techniques like data augmentation, under sampling, and cost-sensitive learning to train their model, and they attained an F1-score of 96.9 during their model evaluation. Lastly, we examined the study of Kern *et al.* [34], where they performed an investigation on the mobility dynamics during corona virus-19 disease (COVID-19). They developed a computational framework for their investigation to evaluate the factors influencing the acceptability of government policies in society toward mitigating the spread of COVID-19. They explored the tweets from Sweden and United Kingdom (UK) as their data source and case study. Their implementation focus was based on mediating causal scenarios by using natural language processing techniques (*specifically, sentiment analysis and Latent Dirichlet allocation (LDA) topic modeling were used*) consolidated with social science and economic methods. Based on the tweets extracted, they conducted numerous causal inference investigations by segmenting the impact of government regulation on society with regard to mobility during COVID-19. Based on the causal-impact experiment, their framework predicted that mobility dynamics during COVID-19 is not totally influenced by government policy during the COVID-19 pandemic. Secondly, their implementation presented to them that texts gleaned from social media could be trusted for examining the concern of society at large. Differently, we identified some notable gaps by carefully examining these existing works, especially concerning RE practice.

2.3 Gaps Identified in Existing Studies

By carefully investigating existing studies in this line of research, we identified the present state of research gaps which are highlighted as follows:

* **Extraction Methodology:** One of the major gaps we identified was the strategy undertaken to extract the *causes* and *effect*. For instance, given the requirement (R1) defined as follows: *if* X is *true* and Y is *false*, then Z must occur. Based on this, some studies (e.g., in [11] and [12] derived their *cause* and *effect* on the *word* level of sentence structure, i.e., cause(c1): X, cause(c2): Y, effect(e1): Z. Using this direct relation implies that important information about the causal relation is omitted, i.e., the conditions of X, Y, and Z are disregarded. Similarly, based on R1, other studies (e.g in [13] and [14]) derived their *cause* and *effect* on the *phrase* level of sentence structure, i.e., cause(c1): X is *true*, cause(c2): Y is *false*, effect(e1): Z must occur. However, the cause-and-effect fragment should be further decomposed beyond the word and phrasal level to ensure an extensive use case. To achieve this, (i) the cause and effect should be segmented into variable and condition such that cause_variable(*i*): X, cause_condition(*i*): is *true*. (ii) there must be a proper understanding of the relation between the cause and effect because R1 states that X and Y must occur before C occurs.

* **Manual Extraction of Features:** Extracting useful features manually (e.g., in [15], and [2]) from raw text is another gap that we identified. An automated extraction technique will be more advantageous than manual feature extraction, especially when dealing with complex, unstructured, and large corpora. In addition, the manual extraction through defined rules enables limited features extraction because the defined linguistic pattern is hand-coded, thus, resulting in bounded extraction results (e.g., the work of Christopher *et al.* in [9], and Jose *et al.* in [10]).

* **Dataset Domain:** In some existing studies (e.g., in [35, 16, 36]), they explored datasets outside the software engineering domain (such as the news from Cable News Network - CNN [37])). Thus, their framework may not be efficiently functional and applicable for software requirement engineering purposes because some vocabularies and grammatical constructs are only applicable in the field of RE. Hence, building a framework on a generalized dataset could lead to less accuracy when applied in the RE field. We took this into consideration by ensuring that our datasets were retrieved from sources that are related to the field of RE.

* **Implementation Link:** Result reproducibility link for implementing most of the existing studies were not specified (e.g., as seen in Cristina *et al.* [10], Roxana *et al.* [19], and Christopher *et al.* [9]). Hence, their framework could only be applied with much re-implementation effort. We ensured that we provided the link to the source code and dataset used in this study as presented in Section 5.5 of Chapter 5.

Overall, in this chapter, we highlighted the key findings that were identified in existing and related studies. In addition, we identified the computation framework that was explored by the researchers as well as the limitations of their study. The next chapter (3) describes the computational instruments used in this study and the approach by which they were applied to answer our research question.

3 Methodology

This chapter discusses in detail the implementation tools explored to achieve our study goals. We begin by describing the different components of the computational framework we explored and their configuration modalities. Finally, this chapter concludes by describing the evaluation metrics used to assess the strength and capabilities of our computational frameworks, especially as it relates to the objective of our study. Before we describe the implementation tools explored, we presented an overview of causality management as it relates to RE practice in the following section. This will assist in understanding the "*why*" behind our research and the rationale behind the research question highlighted in Section 1.7. After establishing the notion of causality management, we then describe the implementation tools and computational instruments explored in this study.

3.1 Causality Management in Requirement Engineering Practice

Causality study has received significant attention in several fields, including psychology [38] and health [25] (others presented in Chapter 2). As described in Section 1.1, the overview of causality has been presented, especially concerning the requirement engineering practice. Hence, in this section, we shall extend the overview of our causality definition in a broader term by describing the required aspects of causality study.

3.1.1 Causality

Causality refers to a relation that exists between two or more events in which one of the events is the causing event (*cause*), while the other represents the effect. Generally, an event is regarded as a scenario (such as a state or process) that occurs in a definite period or instantaneously [39]. It is worth noting that the relation that exists between a cause and an effect is hypothetical, meaning if a cause c_i does not exist, then an effect

e_i is not likely to occur [38, 39]. Therefore, this implies that an effect only happens when the cause happens; else, the relation could be regarded as a constrained relation, implying that they are not causal. The relation could be represented with boolean algebra notations and defined as an equivalence that exists between a cause c_i and an effect e_i, i.e., $c_i \iff e_i$. The notation implies that an *effect* is *true* as long as the *cause* remains *true*. Similarly, an *effect* remains *false* as long as the *cause* is *false*.

The logical equivalence representation of causality is not an absolute description of the form of the relation, particularly regarding its chronological sequence. One of the challenges of formal representation of causal relation *w.r.t* notation and ambiguity, identified in their interpretation, is considered. We simply refer to logical simplicity to mean causal relation. In this study, we categorized the causal relation into three forms, namely: cause, enable, and prevent relation.

* **Cause Relation:** This type of relation is a direct relation between two events where an event c_i causes the effect for another event e_i, i.e., c_i causes e_i ($c_i \iff e_i$); meaning if c_i occurs, then, e_i also occurs. For instance, in the statement "*If a user inputs the wrong password, then, a pop-up error window shall be displayed*". In this example, the error window will only be displayed or triggered whenever a user supplies the wrong password into the system.

* **Enable Relation:** In this type of causal relation, if an event c_i does not occur, then another event e_i cannot occur. Simply stating, c_i enables e_i, i.e. $\neg c_i \iff \neg e_i$. For instance, in the statement "*given that you are a registered student, you can have access to the University library*", the valid student status in the institution is the determining factor that *enables* an individual has access to the University library and its facilities.

* **Prevent Relation:** This type of causal relation between two or more events describes a scenario where the occurrence of an event c_i *prevents* the occurrence

of another event c_j, i.e., $c_i \iff \neq c_j$. For instance, in the statement "*data redundancy ensures that failure of one component does not imply absolute data loss*", data redundancy *prevents* complete data loss in case of failure of one system component.

3.1.2 Sequential Order of Cause and Effect Phrase

In any given sentence or expression that is verified to contain *cause* and *effect*, there are three major orders in which the *cause* and *effect* could be related [39, 27]. In the first order, the *cause* happens before the *effect* (e.g., a system user enters the wrong pin before an error window is triggered). In the second order, both the *cause* and *effect* intersect each other (e.g., a fire burns down a building - a fire is burning and a building is being consumed), while for the third order, the *cause* and *effect* happen at the same time (e.g., a student has access to a University library as long as he is legitimately registered with an institution).

3.1.3 Structure of Causality

According to Blanco *et al.*, there are three structures in which causality could be categorized, namely: (i) implicit and explicit causal relation, (ii) unmarked and marked causality; and (iii) unambiguous and ambiguous cue causality. Cue refers to a linguistic terminology associated with causality study [12, 40] and is defined as an expression that links one event to the other with a defined or non-defined relation [12]. In practice, cue has been regarded as a strong lexical indicator for identifying causal relations in statements [40].

* **Implicit and Explicit Causality:** In an *implicit* causal relation, the effect clause is not clearly stated in the sentence. In REM, implicit causal relations could be difficult to process and interpret, thereby leading to ambiguity due to their indeterminate form. For instance, a statement that reads "*a parent computational*

process stops a child computational process" could be termed *implicit* owing to the fact that cause and effect are not properly defined. On the other hand, an *explicit* causal relation refers to a statement that contains causal details of cause and effect. For instance, a statement such as "*a wrong password triggers the error window*" has a cause (wrong password) and an effect (error window).

* **Marked and Unmarked Causality:** For a statement to be regarded as a *marked* causality, then a cue expression must be present to indicate the causal relation. For instance, the statement "*if the wrong password is supplied, then the error alert window should be triggered*" is regarded as a marked causal relation because of the cue phrase "*if*". A cue expression may not be explicitly stated for an unmarked causal relation. For instance, a statement that reads "*the user cannot open the folder because he has no admin right*" is an unmarked causal relation because of the absence of the cue expression.

* **Ambiguous and unambiguous Cue Causality:** As mentioned earlier in *marked* causality, the presence of some cue expression makes it easier to determine if a statement contains causality or not. However, some cue phrases such as "*since*" could indicate that causality exists in a sentence and, depending on the context being used, could also mean time limitation. Therefore, we refer to such cue expressions as *ambiguous*. Conversely, some cue expressions such as "*because*" that mainly indicate causality in statements are regarded as *unambiguous* cue causality.

3.2 Estimating the Complexity of Causal Relation

Some of the examples presented in this thesis contain a simple level of causality, usually containing both cause and effect. However, because of the introduction of complex computational systems, a complex level of causality is observed where a statement could

contain multiple causes and effects. Therefore, a conjunction $c_i \wedge c_j \wedge c_k ... \iff e_i$ or a disjunction $c_i \vee c_j \vee c_k ... \iff e_i$, or a combination of both conjunction and disjunction could be used to connect the causal relation together (e.g., $c_i \wedge c_j \vee c_k ... \iff e_i$). Furthermore, the components of causal relations can be found in many sentences, which presents a notable challenge for causality extraction because it broadens the scope of causality identification beyond a single statement. Hence, in this study, *n*-sentence causality is also considered where $n \in [1, 2]$. We noted that the complexity of causality increases with an increase in *n*, especially for a relation that connects multiple causal instances together. For example, if an effect of relation r_i indicate a cause in separate relation r_j, then the causal relation where r_j is dependent on r_i is regarded as event link i.e. $r_i : c_i \iff e_i \; and \; r_i : e_i \iff e_j$.

3.3 Importance of Causality in Requirement Engineering Practice

The main focus of this study is to explore how causality in REM can improve the RE practice of the software development process by assisting the practitioners in gleaning applicable insight for an optimal product in the RE phase. Therefore, some of the key benefits of causality study to the RE practice in software engineering are highlighted below.

(i) Causality management helps practitioners to identify the rationale behind some certain conditions and outcomes during the software evaluation

(ii) The descriptive nature of causality management helps to evaluate the relationship between different variables or parameters that constitute a system, especially where the independent variable has occurred.

(iii) Causality study in RE practice helps to understand the modalities of each process or software development phase. Hence, this understanding helps practitioners to resolve theoretical and computational issues that may be encountered at any stage

of the software development life cycle (SDLC) and also helps to optimize product development strategies.

(iv) Many iterative RE process can be implemented and used in different contexts because there is an informed understanding of different components of the systems, making it easy to configure any part of the system component to achieve an excellent software product.

(v) In order to achieve an effective software product, causality study can be instrumental to practitioners to estimate the importance and relevance of a particular process or event during product (software) implementation, testing, and acceptance stage.

(vi) Causal management helps estimate a variable's historical effect on another variable during system design and implementation. By so doing, recurrent design and implementation errors would be reduced to a minimal level.

3.4 Modeling Causality Categorization

To answer our research question and to ensure that our implementation process aligns with our study goal, we must formulate categories of causality based on the definitions presented in Section 3.1.3. Making reference to Stubbs *et al.* [41], the initial stage of category formation for a NL-based task is to define a theoretical and hypothetical standard for each category. Therefore, relating this notion to our study, we define a functional model K with vocabulary V and the relation R_0 between each term and their interpretation J. In essence, the category annotation formation task is modeled as follows:

* V = {sentence is "causal" or "not causal"}

* R_0 = {sentence is "causal" | "not causal"}

* J = {causal definition:: regard a sentence to be causal if there is a relation between two or more events, i.e

¬ causal implies a non-causal sentence if and only if it has an independent event state}

Based on the theoretical standard evaluated for the categorization and annotation task, the following are the nine categories that was formulated in this study, and they are: causality, marked, explicit, unit cause, unit effect, single sentence, event link, temporal feature, and relationship feature[12]. Notably, because of the inter-domain unbalanced dataset, a general rule-of-thumb was formulated for each of the nine categories, which states that "In a domain Q, sentences originating from other domain other than Q has similar distribution of values as the domain Q". This notion was also helpful and relevant in our category annotation task.

3.5 Research Design

For a good research practice, it is imperative to formulate a good implementation design. Hence, in this section, we discuss the dataset (i.e., the source, form, and exploratory analysis performed) and the implementation tools considered in this study.

3.6 Dataset Exploration

To fully meet the objectives of this study, it is imperative that we explore datasets that can fit our machine models. In this section, we present the description of the dataset explored in this study, such as the source and exploratory analysis, as well as how we fit them into our machine learning and deep learning models.

3.6.1 Data Source

For us to assert the option regarding the degree to which causality is considered in REM for practical application, it is expedient that a large collection of REM is extracted. Therefore, we extracted our dataset based on two key factors: (i) the dataset will not be centered around a single domain, as multiple domains of interest will be considered. (ii)

[12]Definition details can be accessed here

the dataset will be REM that has practical applicability (i.e., either currently in use or has been used in the past). Using these criteria to extract our dataset affords us to have a generalized and extended notion of causality in RE practice. Our extensive collection of the dataset was extracted from the collection on kaggle[13] containing 675 requirement documents. We randomly selected 50 documents from the 675 collections spanning different domains for our analysis. The 50 documents extracted has 12,438 sentences with statistics presented in Table 2 (and features presented in Table 1) showing the domain and year distribution. In addition, the percentage of each domain with regards to the overall dataset is represented in Figure 2. The pre-processing steps and exploratory analysis performed on the dataset is presented in the section below.

Table 1. Dataset Feature

Feature	**Explanation**	**Data Type**
Id	A unique identifier of a requirement record	Numeric
Description	An explanation of a record	Text
Log	Past record of state changes	Classification List
Creation Date	Date of document's inception	Datetime

3.6.2 Exploratory Data Analysis

Our dataset contains RE-sentences that originate from different sources and contains different degree of error. Training the machine models on an incomplete sentence and structure-based phrase (such as heading) is not a good practice as we may not get the best model performance. Therefore, to tune the dataset to conform with training our machine models, we engaged the following data analysis procedures:

(i) We retrieved the raw texts from the documents by applying the *getPage()* and *extractText()* python function. Further, the extracted texts are put into structure

[13]https://www.kaggle.com/code/kmader/quickdraw-with-wavenet-classifier

Table 2. Statistics of Dataset Showing Domain and Year Distribution

Domain Distribution	Sentence Count	Year Distribution	Document Count
Aeronautical	214	**2005**	3
Banking	1082	**2006**	3
Defence	405	**2007**	1
Smart Society	721	**2008**	2
Healthcare	1064	**2009**	4
Insurance	385	**2010**	4
Science	602	**2011**	3
Astronomy	407	**2012**	2
Agriculture	1305	**2013**	2
e-society	497	**2014**	1
e-library	136	**2015**	2
e-analytic	669	**2016**	4
Automotive	648	**2017**	6
Communications	1832	**2018**	1
Energy	1014	**2019**	2
Sustainability	201	**2020**	5
Infrastructure	795	**2021**	2
Regulatory	461	**2022**	3
Total:	12,438	**Total**	50

using python's dataframe object. In addition, we removed both the trailing and leading white spaces.

(ii) We filter lines that contain irrelevant contents or satisfy at least one of the following conditions:

(a) If the text begins with the either *chapter, figure* or *table*, and/or contains a page character.

(b) The text has less than 45 characters and does not end with a "!", "." or "?".

(c) The text contains consecutive characters with no grammatical or contextual meaning. For example, characters such as "..., *???, aaaaa*", and "***".

(iii) Enumerate symbols such as "(I) and (a)" are removed.

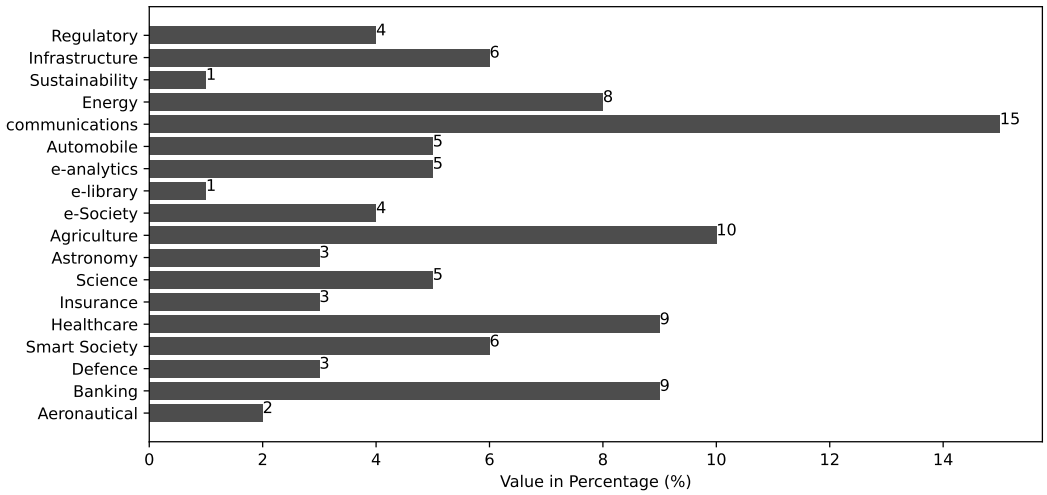

Figure 1. Data Distribution Percentage

(iv) We constructed paragraphs by removing empty lines and combining successive lines of text. Next, we divided every paragraph into sentences by using the following rules:

(a) Every line of text that ends with the characters "!", "." or "?" is considered a sentence.

(b) Every line of text ending with notations such as "*e.g., i.e.,* " is combined with the successive line.

3.7 Implementation Tools

In this study, we explored the machine learning (ML) classification framework as well as the deep learning framework as our implementation tools. Specifically, Naive Bayes (NB), K-Nearest Neighbor (KNN), Ada Boost (AB), Support Vector Machine (SVM), and Random Forest (RF) are the ML classification algorithms that was explored. In addition, to have an optimal performance of the algorithms, we applied a 10-fold cross validation

to harness the selection of optimal hyperparameters for the best model performance. In Table 9, we highlighted the hyperparameter configurations used for training the ML models. By applying five ML classification models, we were able to compare the results obtained from these models and identify the best performing model based on evaluation criteria. In addition, we applied BERT (Bidirectional Encoder Representations from Transformers) model as the deep learning (DL) model for training the dataset for identifying the form of causality in the REM that was explored. This enabled us to extend, evaluate and compare the results of the ML and DL models as our study is concerned with obtaining an optimal standard of computational result. In the sub-sections below, details about the ML (NB, KNN, AB, SVM, and RF) and DL (BERT) models explored in this study are presented.

3.8 Machine Learning Models

Machine learning is an engineering technique where computer systems observe patterns and learn from past occurrences [42]. The learning function of a system represents its ability to improve on a task T_i based on an experience E_i, with respect to a performance metric P_i to present intelligent predictions about the observed pattern [42, 43]. An example of a task can be software sales prediction, a crime rate forecast, or anything else measured using prediction accuracy like the F1-score or root mean square error and the experience denoted by a historical dataset [43]. More specifically, the ML field focuses on answering two key questions: (i) How can computer systems be engineered and configured to optimize their functionality through experience? and (2) What are the fundamental computational and statistical information rules governing the learning system? An analytical study of the ML approach is required to answer these questions. The sub-sections below presents the machine learning frameworks explored in this study. We used grid search (a 10-fold cross validation) technique to identify the optimal hyperparameters for each machine classifier by fitting the model on different

hyperparameter combinations where the best hyperparameter combination is selected. The hyperparameter configuration of the machine learning algorithms is presented in Table 9 of Chapter 4. Additionally, for word embedding of our dataset, we explored the Bag-of-word (BoW) technique, with details presented below.

3.9 Word Embedding Using Bag-of-Word

A bag-of-words model, known as BoW, is a means of extracting text features for modeling tasks. A BoW is a text representation that clearly defines the appearance of words in a document based on the glossary of known words [44]. The term bag of words relates to the fact that any information on the structure or sequence of words in the document is removed. However, the model is solely concerned with whether recognized terms appear in the sentence or corpus and not the position it appear. It has been applied to different natural language tasks over the years (as evidenced by Miller *et al.* [45]). The steps undertaken in implementing BoW in NLP tasks involve data collection, vocabulary designing, and creating document vectors. In this study, we used BoW for our word embedding task for the natural language models by using the *CountVectorizer()* in-built python function.

3.9.1 Naive Bayes

The Naive Bayes (NB) algorithm is a type of ML classification framework that is based on Bayes' principle, which proposes that "*each class feature contributes independently and equally to a particular target class*" [46]. The primary purpose of the classification task is to identify the optimal mapping between a given piece of data and a collection of classes within a certain area of investigation. In order to make the mapping computable probabilistic-wise, given mathematical procedures are engaged to change joint probabilities into multiplier operations of conditional and prior probabilities.

As a ML approach, it transforms a simple division into a long sequence of numerators

divided by another long sequence of denominators. The mathematical transformation in NB computation is required because conditional and prior probabilities are less difficult to summarize from the dataset. This is done by estimating the number of instances that satisfies a given condition. To express the operation of the NB classifier, if we consider our classification problem to consist of g attributes and h classifications, then NB will compute the mapping accuracy of the attributes $(R_1, R_2, R_3, ..., R_n)$ and classification set $(S_1, S_2, S_3, ..., S_n)$. Therefore, the accuracy of fitting the vector $R_1 = w_1, R_2 = w_2, R_3 = w_3, ..., R_n = w_n$ into the classification S_i is estimated by Bayesian supposition for computing the maximum probability as represented in the equation below:

$$
\begin{aligned}
&P(S_i|(R_1 = w_1 \cap R_2 = w_2 \cap \ldots \cap R_n = w_n)) \\
&= \frac{P(S_i \cap (R_1 = w_1 \cap R_2 = w_2 \cap \ldots \cap R_n = w_n))}{P(R_1 = w_1 \cap R_2 = w_2 \cap \cap R_n = w_n)} \\
&= \frac{P((R_1 = w_1 \cap R_2 = w_2 \cap \ldots \cap R_n = w_n) \cap S_i)}{P(R_1 = w_1 \cap R_2 = w_2 \cap \cap R_n = w_n)} \\
&= \frac{P((R_1 = w_1 \cap R_2 = w_2 \cap \ldots \cap R_n = w_n)|S_i) \times P(S_i)}{P(R_1 = w_1 \cap R_2 = w_2 \cap \cap R_n = w_n)} \\
&where\ 1 \leq i \leq h
\end{aligned}
$$

Following the computation of this probabilistic series, next, the fitness variable between a vector and the possible classification will be quantitatively defined, which determines the class an attribute is allocated.

3.9.2 K-Nearest Neighbor

The k-Nearest Neighbors (KNN) algorithm is a supervised, non-parametric learning classifier that employs proximity to predict, classify or predict the grouping of an observation. The algorithm primarily functions on the notion that similar observations can be found close to one another. For our classification task, a class label is assigned

based on the most frequent label around an observation x_i. KNN retains its training dataset vis-a-vis undergoing training process, which causes it to consume large memory space. The primary objective of the KNN is to identify the nearest observations of a given query point for us to assign a class label to such observation. Therefore, for us to achieve this, we need to define the distance metrics by estimating the closest observation to a given query observation. The distance metrics assist us in formulating decision boundaries that divide the query observation into various regions. The distance metrics considered in this study is the euclidean distance which is limited to vectors that are real-valued. By applying Equation 1, KNN estimates a parallel line between the query observations (U_i) and other observation (V_i) in the dataset.

$$d(u, v) = \sqrt{\sum_{i=1}^{N} (U_i - V_i)^2} \tag{1}$$

In the KNN algorithm, the value of K defines how many neighbor will be evaluated to determine the classification of an observation point. For example, if K=1, the observation in this scenario will be allocated to the same class as the nearest neighbor. Defining the value of K in KNN significantly determines model output, as the value of K can influence underfitting or overfitting of the model. However, low values of K could imply high variance with low bias, while a large value of K could result to low variance with high bias. Based on Taunk *et al.* [47], it is recommended that odd number is assigned for K to prevent classification ties [47]. In addition, data with high noise or variance will likely perform better with a high value of K. Cross validation technique using grid search is used to select the best value of K.

3.9.3 Adaptive Boosting

The Adaptive Boosting (AdaBoost) framework is a known approach for constructing an ensemble classifier by choosing weak component classifiers. The AdaBoost framework

creates a strong classifier from multiple weak classifiers by discovering combination of weak classifiers with weight adjustment via a recurrent process, with the original training dataset remaining unchanged [48]. The core idea of AdaBoost is to establish the weights of classifiers and train the data in each iteration to ensure accurate predictions of new observations. Before we implemented AdaBoost, two key actions were engaged to ensure accurate prediction: (i) we proactively trained the classifier on different weighted training instances. (ii) For each iteration, we ensured adequate model fit for the dataset, and this was done by minimizing the training error. The structural implementation idea of AdaBoost in our study is supported by Figure 2. In this study, we followed seven key steps during the implementation process of AdaBoost. They are highlighted below:

(i) We assigned equal weight to all our observation

(ii) We built the model on the training dataset, which is (75%) of the total dataset, while 25% of the dataset was used for testing the model

(iii) We made prediction on the whole dataset with the model developed in (ii)

(iv) We computed model error by comparing the predicted and actual values

(v) When we are developing the succeeding model, we allocated higher weights to data points that the preceding model does not correctly predict

(vi) We estimated the weight using the error value, which implies that an observation with higher error implies higher weight allocation to such observation

(vii) We repeated the process (i) - (vi) until we achieved a constant error function or when we attained the maximum number of an estimator

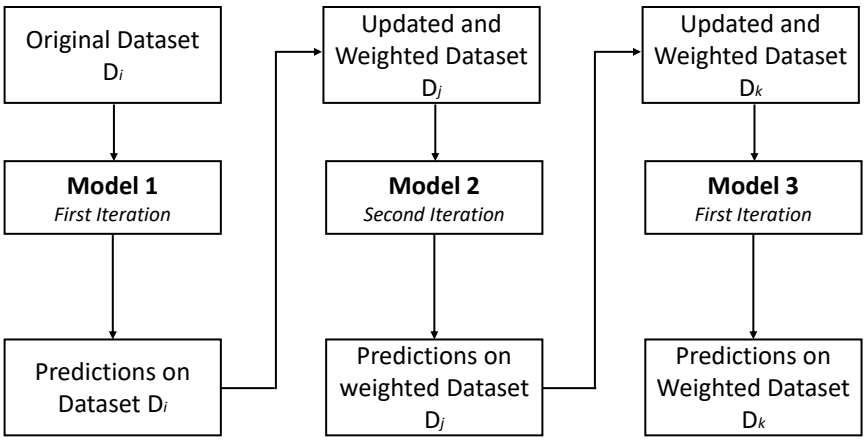

Figure 2. Implementation Procedure of AdaBoost Machine Algorithm

3.9.4 Support Vector Machine

Support Vector Machines (SVM) are supervised machine learning algorithms that have demonstrated success in classification (using pattern recognition) and related tasks over the years of implementation [49]. In the past, using SVMs for computational tasks was enabled by factors such as (i) Whilst other machine learning techniques, such as Artificial Neural Network (ANN), converges to local minima, SVM solutions converge to global optima. (ii) Rather than minimizing the training error, SVM will minimize the upper limit of its model's performance, which thus enables efficient and good generalization results. (iii) SVM contains a convex optimization feature to attain optimal results [50], and (iv) The prediction result of SVM is less influenced by outliers in the dataset, which enables its high degree of efficiency in dealing with high dimensional dataset [49, 50]. SVM functions by partitioning data into two separate groups using a hyperplane, as shown in Figure 3. The data points in the training set can be classified as belonging to either of the two partitions. Specifically, SVM splits data points into classes to construct an optimum hyperplane that maximizes the distance between the data points and the hyperplane.

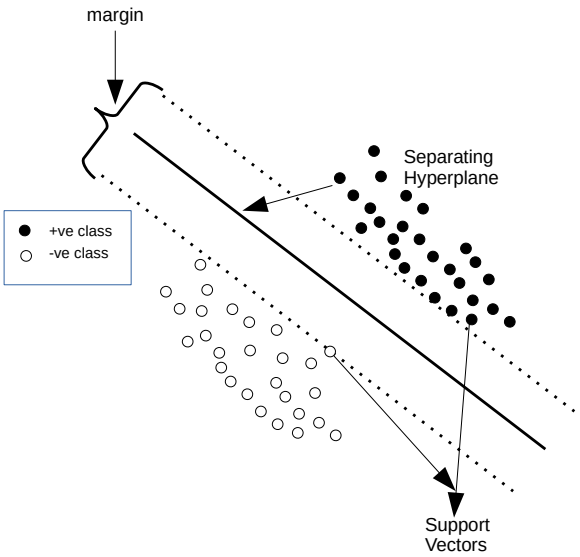

Figure 3. Schematic diagram of Support Vector Machine

3.9.5 Random Forest

Random forest (RF) is one of the recent machine models that have been applied for classification analysis [51]. RF addresses classification problems by generating multiple decision tree branches during model training and uses an internal metric to determine the value of best tree branch. We are implementing an optimized RF in our research experiment as one of the ML models resulted in high performance. In Equation 2, we present the mathematical expression responsible for the formation and operation of the RF machine learning algorithm.

$$y(x_i) = \frac{1}{H} \sum_{n=1}^{H} T_n x_i \tag{2}$$

where H represents the number of trees in the forest, and T_n indicates individual trees in the forest (computational field).

3.10 Deep Learning Technique

Relating to the emergence of Deep Learning (DL) in recent decade, an increasing number of research activities are employing DL models to solve problems involving natural language. Our study explored the Bidirectional Encoder Representations from Transformers (BERT) model, which represents a branch of deep learning framework, especially for NLP tasks. The operation of BERT is notable and has been applied to different NLP-based research, such as the study of Devlin *et al.* [52] in which it was considered for solutions to question and NER (i.e., named entity recognition). BERT model is architectured on a 12 layer transformer-based ML encoding framework for natural language processing (NLP) pre-training [53]. It is intended to pre-train deep bidirectional representations from an unlabeled dataset by relying on both left, and right context simultaneously in all levels [52]. As a result, the pre-trained BERT model could be fine-tuned with just one extra output layer in its implementation structure to provide cutting-edge model results. In this study, we employ BERT's fine-tuning approach to assess its ability to identify causality in REM.

For our implementation, first, the token for each sentence is generated because BERT necessitates input sequence length (usually, the maximum acceptable length is 512). Therefore, we applied padding tokens to sentences that are less than the fixed length to bring all the sentences to a uniform length. In addition to the padding tokens, classification tokens were added to give additional details (of the sentence) as model input. The classification token was configured and represented as the sentence sequence's first token to evaluate the sentence as valid model input. Next, the whole sentence token was transmitted to the single layer feed forward neural network configured with softmax (activation layer) for calculating the probability of determining causality in a sentence. In our implementation, we configured the BERT model in three varieties, and we evaluated their performance. The three forms are Base-BERT, PoS-BERT, and Dep-BERT.

In the Base-BERT architecture, we tokenize sentences as described above and feed them into the classifier. To estimate the length of the input sequence, we performed an analysis on our dataset, and we decided to use 128 as the fixed length of our token to satisfy the minimum acceptable length required by the BERT algorithm. Secondly, for the PoS-BERT architecture that was considered in this study, existing studies reported and proved that supplying precise previous representation of syntactic schemas to the model can increase its performance (Dinghan *et al.* [54]). Therefore, we considered this in our implementation by adding the associated part-of-speech (PoS) identifier to the tokens (this functionality was handled by the spaCy library[14]). An approach that was used for input sequence encoding and their respective identifier (tags) is by concatenating the tokens using PoS tags (i.e., hot encoded vector). But due to the high dimensional requirement of the BERT token [52], the effect of the PoS tag is minimal. Therefore, we concluded that syntactic knowledge would have a better influence, especially if the input (i.e., sentences) are annotated using PoS identifiers. Next, the annotated sentences are transmitted to the BERT model. The work of Stephan *et al.* [55] proves the validity of this method. In Figure 4, we presented the pictorial representation of our BERT PoS tag embedding into the input sequence. Owing to the extension of the input sequence, the fixed BERT model's length was adjusted accordingly to 368 as opposed to 128 token that we used in Base-BERT.

***Base*-BERT** – If the execution fails, display an error message

***PoS*-BERT** – If_PREP the_DET execution_NOUN fails_VERB ,_PUNCT display_VERB an_DET error_NOUN message_NOUN

***Dep*-BERT** – If_MARKER the_DET execution_SUBJ fails_NOUN ,_PUNCT display_ROOT an_DET error_COMPOUND message_SUBJ

Figure 4. Input sequence showing configuration of the BERT model. PoS identifiers are highlighted in red, while Dep identifiers are highlighted in green

[14]https://spacy.io/

It typically involves an adverbial phrase. Lastly, the Dep-BERT implementation was similar to the configuration of PoS-BERT. Still, in this case, we used dependency tags (Dep), which are linguistic features (such as grammatical and lexical class) instead of PoS tags *(as shown in Figure 4)*. Therefore, more details about the grammatical formation of the sentence is provided. In this Dep-BERT, we concluded that the model performance could be improved by adding the dependency features, as causal relation follows a grammatical structure, i.e., it typically involves an adverbial phrase. With this, the classifier would be able to learn certain grammatical patterns during the training process. In this architecture, we also used 368 as the fixed length of the input sentence tokens into the model.

3.10.1 Tokenization

In Natural Language Processing (NLP), tokenization is a typical task. It is a crucial component in both classic NLP approaches, such as advanced deep learning designs, and count vectorizer, such as transformers [53, 56]. Tokenization involves splitting a piece of text into smaller components known as tokens *(converted to bit representation)*. In this context, tokens can be letters, sub-words, or words. In this study, we used the sentence and word tokenizer function from the natural language tool kit (NLTK)[15]. Tokenization choices considerably impact future evaluation of the sentence (or word) that is splitted. Although tokenization is language sensitive and could be computationally costly, especially for big corpora, as considered in this study, we are able to manage and implement it adequately for training our language models.

3.11 Evaluation Metrics

Using human annotators for the dataset application of ML and DL models as classification algorithms equally requires an efficient evaluation technique to monitor their

[15]https://www.nltk.org/

performance. Therefore, in this study, we explored Cohen Kappa technique for evaluating the annotator's activity, while recall, accuracy, precision, and F1 score are used to evaluate the ML and DL classification model. Details of the evaluation frameworks are presented in the sections below.

First, we will define the hypothetical representation of the evaluation metrics which formulates their inference. They are: (i) True Positive (TP), (ii) True Negative (TN), (iii) False Positive (FP); and (iv) False Negative (FN). While TP is the situation where a classification model accurately predicts the positive class, TN is the situation where a classification model accurately predicts the negative class. Conversely, FP refers to an instance where a classification model incorrectly predicts positive class, and FN refers to an instance where a classification machine model incorrectly predicts the negative class from a set of observation [43].

3.11.1 Cohen Kappa Measurement

In the annotator task, the Cohen Kappa statistic is a measure of inter-rater agreement (or precision) when categorical data items are annotated [57]. In this study, we explored the Cohen Kappa statistic to assess the degree to which our annotators present the same predictions for each annotation category. Cohen Kappa was considered in this study because of its wide usage for annotation tasks and evaluating inter-rater's dependability in different research fields. For example, the Cohen Kappa measure was used by Blackman *et al.* [58] and Alyse *et al.*[59] in their study to compute inter-rater reliability for their annotation tasks. In Table 3, we present the Cohen Kappa's standard score rating and how the values are interpreted. Cohen Kappa's score is within the range of 0 to 1, with 0 showing less agreement between the annotators and 1 showing high agreement between the annotators. To implement Cohen Kappa score, we considered the *cohen_kappa_score()* inbuilt python function from the ***sklearn*** library to estimate the Cohen Kappa's value for our annotators, and the result obtained is presented in Table 4.

The mathematical formulae for estimating Cohen Kappa's measurement is presented by Equation 3.

Table 3. Cohen Kappa Interpretation of Score Distribution [1]

Cohen Kappa Value	Interpretation
0	No agreement
0.10 - 0.20	Little agreement
0.21 - 0.40	Moderate agreement
0.41 - 0.60	Average agreement
0.61 - 0.80	Considerable agreement
0.81 - 0.99	Significant agreement
1.0	Absolute agreement

$$C_K = \frac{X_0 - X_e}{1 - X_e} \tag{3}$$

Where X_0 represents the observed agreement between raters, while X_e is the theoretical probability of agreement. For example, consider two input sets containing the annotation from two different annotators A and B, such that A = (0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0) and B = (1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0). Then applying the *cohen_kappa_score(A,B)* function estimates the Cohen Kappa value to be 0.32. Hence, based on the interpretation highlighted in Table 3, the annotators have moderate level of agreement.

3.11.2 Recall

In ML classification task, recall refers to the number of correctly identified instances that are predicted correctly out of the total applicable instances [60]. It measures the ratio of identified TP from an observation. Mathematically, recall, otherwise referred to as the true positive rate is computed by applying Equation 4 below.

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

3.11.3 Accuracy

Accuracy is defined as the measure of model performance across all prediction class [60]. In essence, accuracy is the fraction of the correct model prediction, which is mathematically expressed as the ratio of total correct predictions to the total predictions as shown in Equation 5.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

3.11.4 Precision

Precision metrics in classification task refers to the ability of a classification model to correctly identify the suitable and relevant data points [43]. Mathematically, it is computed by dividing the summation of TP and FP by TP, as presented in Equation 6.

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

3.11.5 F1-Score

The F1-score in machine classification task is the harmonic mean (or the weighted average) of the recall (completeness) and precision (exactness) which measures the overall accuracy of a model prediction [60]. Equation 7 describes the mathematical formulae explored to compute the F1-score of the models explored in this study.

$$F1 - score = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{7}$$

In Figure 5, we summarized the implementation procedures that was explored for satisfying our study goal.

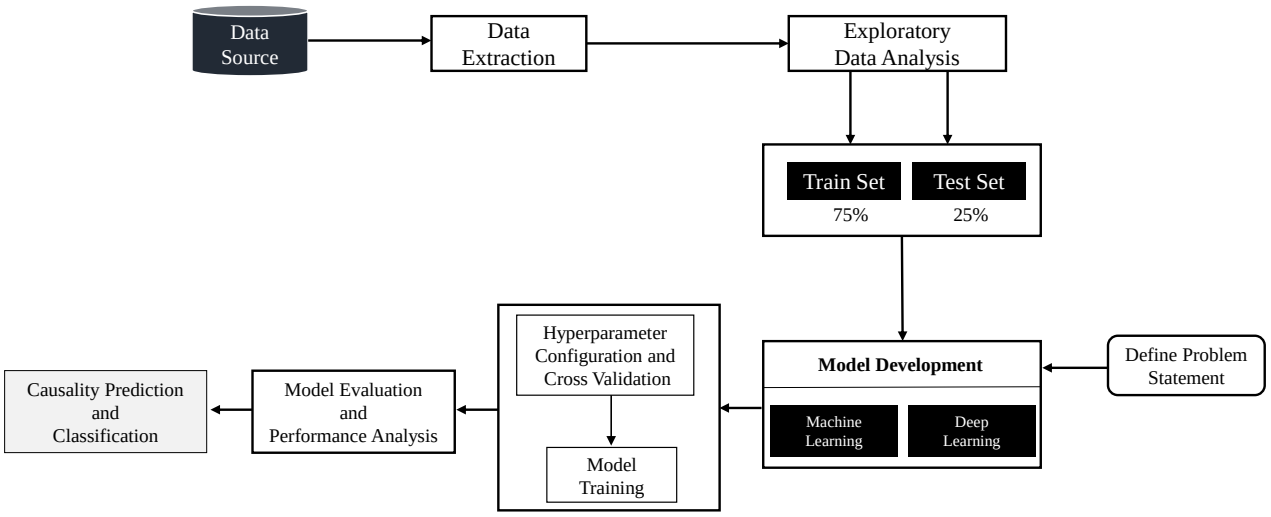

Figure 5. Summary of Implementation Procedure

3.12 Annotation of Dataset

The annotation process involves labeling the dataset by humans based on the directives and guides given to them. The annotation task is a major part of the study, as sentences were studied and labeled accordingly to their respective categories. To aid the annotation process, we developed a database of cue phrases.

3.12.1 Annotation Rules

Prior to the labeling procedure, we held a training with all annotators to establish a shared knowledge of both causality and the associated categories. The annotation task was performed by seven annotators, two from UT in Estonia, three from WITs University in South Africa, and two from OAU in Nigeria. The annotators are comprised of both honors and masters student in the computer science and information technology field.

These are the candidate that showed interest in the data annotation procedure as part of exercise to improve their research insight. The annotation task also allowed them to explore natural language analysis in research. Notably, we conducted three training session via google meet where the modus operandi and the rules governing the efficiency of the annotation task was discussed. A major instruction to the annotators is to adhere to the annotation rules and guides to maximize the result of the annotation process. The distribution of the 12,438 sentences are as follows: each of the annotators from UT, OAU, and WITS University were allocated 1000, 2000, and 2146 dataset, respectively. Although we specified an expected duration to complete the annotation task as 30 days, we got the last annotation batch from one of our annotators at WITS university after 47 days due to unforeseen logistic concerns.

One important guideline for annotators is to avoid labeling sentences as causal only based on cue phrases. This is so because this strategy is prone to an excessive number of False Positives. Therefore, we proposed that instead of focusing on the sentence's syntactic or lexical features, the annotation process must begin with a thorough comprehension of the phrase, especially on semantic level. For instance, one of the sentences reads *"if the dashboard's functionality is connected to the application programming interface (API), then, recent data from is supplied to the dashboard on a frequent basis"*. In this case, the presence of the cue phrase "*if*" does not imply a causal sentence; hence, annotators need a proper understanding of the sentence semantics. Another major instruction to the annotator was to assert if the *effect* is totally dependent on the *cause* before a sentence is referred to as causal. Thus, it is important that an artifact containing annotation guide is given to the annotators to ensure an efficient annotation process. The annotation guide can be found here, while the annotation result of this study is presented in Section 4.1 of Chapter 4.

3.13 Implementation Environment

The programming language used for modeling the machine learning and deep learning algorithms is python version 3.9.15[16], which was deployed via jupyter notebook[17]. All our codes were deployed using Linux operating system machine powered by a graphic processing unit (GPU) processor. On average, the execution time per code run is between thirty minutes because of the natural language being processed.

[16]https://www.python.org/
[17]https://jupyter.org/

4 Result Analysis

This study is concerned with causality management as it relates to requirement engineering (RE) and its applicability to practitioners, particularly by considering the causality's scope, complexity level, and structure. In Chapter 3, we described the machine algorithms used for the implementation phase and the dataset exploration. In this chapter, we present our study results, ranging from the data analysis to annotation guides and causality distribution by the machine algorithms. Lastly, we showed how we have been able to answer our research question based on our implementation analysis.

4.1 Annotation Assessment

The annotation guidelines are presented in Section 3.12, which describes the procedures taken for the annotation task. We considered evaluating the degree of agreement of the annotations to measure our annotation reliability. To distribute the dataset, a total of 12438 sentences was allocated to the annotators comprising at least 1000 distinct sentences and 1200 overlapping sentences. The overlapping sentence was used to evaluate the inter-annotator agreement by applying the Cohen Kappa (CK) measure described in Section 3.11.1 to measure the correlation of annotators in classifying sentences to the same category. In Table 4, we present the overview of the measure of inter-annotator agreement, accuracy, and confusion matrix based on each category. For the *causality* category, the annotator agreement was estimated based on the 1200 overlapping sentences because other categories depict the other causality types. Only sentences labeled as *causal* were used to estimate the inter-rater agreement. From our analysis and evaluation, we are able to ascertain the reliability of the agreement of our inter-rater annotations.

By observing all the categories based on Table 4, we noticed that an average of 79% was achieved as inter-annotator agreement, excluding the unit cause and unit effect, all the categories presented has an agreement of at least 81%. We concluded that the

low values of 77% for the two categories was attributed to the annotator's experience and intuition. For instance, an annotator A could decompose a sentence into different sub-causes and effects while another annotator B does not; hence, this influenced the difference in the inter-annotator's annotation agreement.

The Cohen Kappa's measure is particularly prominent for the event link and marked categories because, irrespective of the high degree of agreement of over 97.4%, CK resulted in a very low value, which implies a minimal degree of agreement. In addition, we observed that the mean value for all the categories is around 0.7, which suggests a close range of agreement. Hence, we concluded that our dataset is suitable and reliable for additional analysis as well as the development of our causality detection framework. In addition, this was evidenced by the descriptive analysis performed on the dataset to align with answering our research questions. Based on this, we formulated that, for every category, We evaluated the independent relationship between the distribution of requirements to a category and their connection with a specific domain. The detailed result obtained from the annotation process is presented in Figure 6, showing the category distribution and causality-type labeling. Further, in Tables, we present the percentage of cue phrases per domain and the most accurate and least accurate cue phrases. By so doing, we are able to evaluate the impact of these cue phrases on RE practice through the analysis of the REM.

Table 4. Statistics of Inter-annotator Labeling per Category

		Causality		**Marked**		**Explicit**		**Unit Cause**		**Unit Effect**		**Unit Sentence**		**Event Link**		**Average**
		0	1	0	1	0	1	0	1	0	1	0	1	0	1	
Confusion Matrix	0	2149	213	15	104	74	136	82	239	94	162	328	105	728	184	
	1	382	741	23	612	58	516	132	293	53	204	47	492	83	28	
Annotation Agreement		73.3%		84.5%		91.6%		83.3%		73.1%		97.4%		70.3%		80.5%
Kappa's Factor		0.64		0.71		0.68		0.59		0.48		0.76		0.64		0.64

In Figure 6, we show the annotation results obtained for all the causality categories. Further, we present the label distribution values in Table 5, showing the result obtained

from allocating the annotation into different causal groups for all domains considered. However, it is worth noting that our analysis covered all the functional and non-functional requirements because they are the ones relevant to this study. For example, some of the non-functional requirements identified in the study include expressions that focus on product objectives, content arrangement, etc.

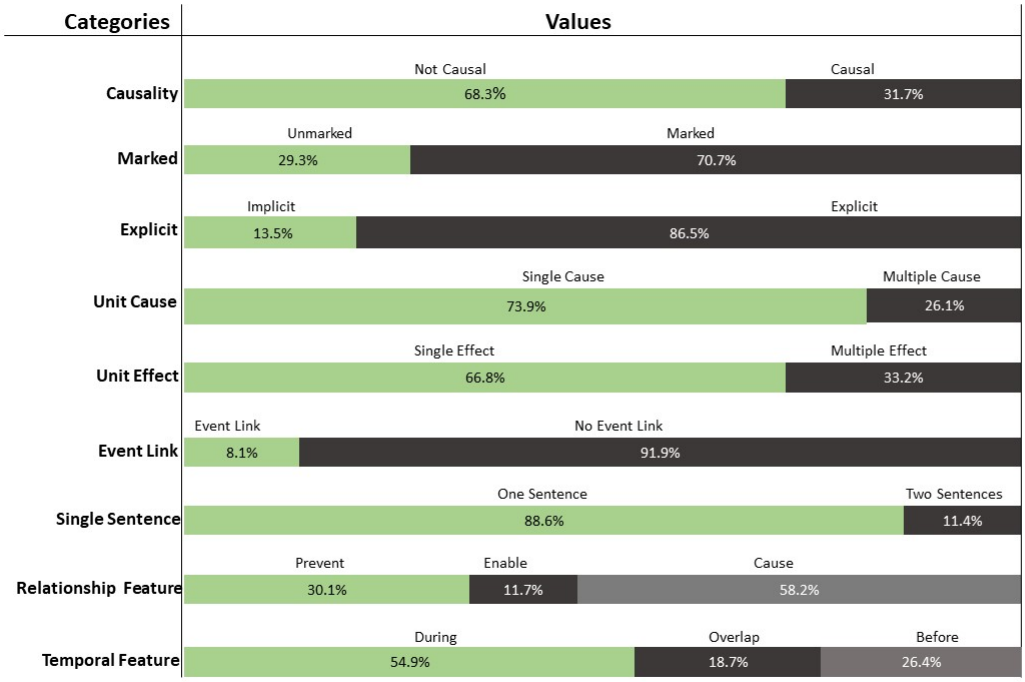

Figure 6. Category Distribution of Annotation Result

From our implementation analysis, we deduced and concluded that using only the cue phrases as a key (or only) determinant to detect and classify a sentence as causal or non-causal is not sufficient. Using the five machine learning language (explored in this thesis) and the variation of BERT models showed the strength of applying a ML-based model for detecting and extracting causality in REM based on their performance, as evidenced in the results presented in Tables 8, 4, 9, and 5.

Based on the evaluation of the ML models used for classification, the performance ranking of the models showed that Naive Bayes (NB) was the best performing model with

an accuracy of 84%. Similarly, Dep-BERT performed best from the variation of BERT models considered in this study with an accuracy of 89%. Based on this, it is clear that the overall best performing model is the DL approach. Interestingly, all three variants of the BERT model performed excellently with an f1 score greater than 78% (i.e., f1 > 75%) for both causal and non-causal sentences. Interestingly, we observed that adding syntactic details to the input sequence does not improve model performance. Conversely, one of the BERT variations, i.e., PoS-BERT, performed low when the syntactic information was added with an accuracy of 73%, which differs by 3% when compared with base-BERT. However, there was an increase in performance for Dep-BERT owing to the highest f1 and accuracy score of 86% for the causal and non-causal class.

Furthermore, when compared with the machine learning algorithms, Dep-BERT has an improvement of about 7.1% increase in recall and about 12.4% increase in precision. Interestingly, Dep-BERT outperforms both Base-BERT and PoS-BERT across all the evaluation metrics with a small difference. Based on this, we noted that Base-BERT has an understanding of the deep learning language, which can be attributed to the rigorous pretraining process, which could be configured to detect causality without many input configurations. Lastly, it is worth noting that applying dependency tags *(showing the relation between words in a sentence)* has a relatively small but significant effect on the overall performance of the model. In essence, our findings have been able to uncover the key features and concept of causality in the RE practice. The scope, complexity, and structure of causality in REM, which determines its impact on RE practice, have been studied and analyzed. Hence, we can make a remark that causal analysis in RE contributes immensely to the quality of the computational product in view (based on domain and architecture type).

We extracted the causal cue expression from our annotation analysis and categorized them into their grammatical class as shown in Table 6. Next, we present the cue phrases in Table 7 to indicate their focus as a *prevent, cause,* or *enable* relationship. Furthermore,

Table 5. Label Distribution for all Domain Categories

Domain	Causal Sentence		Marked Type		Explicit Type		Event Link		Unit Cause		Unit Effect		Single Sentence		Relationship Feature			Temporal Feature		
	0	1	0	1	0	1	0	1	0	1	0	1	0	1	*During*	*Overlap*	*Before*	*Prevent*	*Enable*	*Cause*
Aeronautical	4302	2942	3034	2124	1329	2029	452	844	853	2421	1853	1841	3011	926	2935	542	309	859	223	359
Banking	593	224	622	202	1196	231	194	257	179	232	641	233	192	204	132	257	480	502	244	294
Defence	231	39	149	243	92	140	148	242	49	84	34	29	139	209	159	23	342	14	203	33
Smart Society	494	353	1034	259	1395	2395	454	492	135	249	13	244	133	259	958	583	529	593	94	243
Healthcare	852	243	602	423	135	243	164	204	104	244	301	432	931	250	1503	295	405	103	253	35
Insurance	43	52	31	42	31	32	15	42	291	24	41	292	39	249	493	493	35	53	24	34
Science	141	242	140	352	124	42	51	203	13	229	301	246	129	292	145	204	340	14	42	30
Astronomy	981	424	501	402	149	235	149	252	591	202	21	233	491	243	921	462	23	142	262	34
Agriculture	35	252	442	240	953	499	394	295	935	395	593	335	939	24	133	249	349	320	953	599
e-Society	734	302	421	258	142	240	1305	302	124	203	944	492	941	592	332	293	593	542	93	35
e-library	198	242	4321	294	195	242	492	753	492	404	492	853	294	492	439	402	440	402	742	294
e-analytics	942	422	1940	2244	1420	374	150	229	1128	232	402	240	393	2424	187	442	403	401	3904	310
Automobile	531	374	551	202	192	493	123	402	301	402	945	294	1404	495	423	492	33	441	302	303
Telecommunications	294	104	492	394	492	829	413	273	194	205	994	243	402	204	141	294	342	141	132	248
Energy	83	53	21	52	21	20	12	28	41	22	41	92	14	24	31	32	203	14	25	31
Sustainability	637	326	312	202	491	23	131	482	134	253	172	283	140	254	139	930	301	136	827	362
Infrastructure	834	427	724	232	1042	20	162	393	115	263	401	22	122	302	149	273	349	142	382	203
Regulatory	32	12	21	32	81	16	13	23	41	90	31	62	31	12	10	32	11	9	3	13
Total	**11957**	**7033**	**15358**	**8197**	**9480**	**8103**	**4822**	**5716**	**5720**	**6154**	**8220**	**6056**	**9745**	**7455**	**9230**	**6298**	**5487**	**4828**	**8708**	**3460**

to evaluate the degree of ambiguity of a cue phrase J, a binary classification task was formulated regarding all the extracted sentences as the sample space. To compute the precision of a cue phrase J in the sample space, we considered a conditional probability (as shown in Equation 8) of a causal sentence as the presence of cue phrase J, thus indicating the degree of ambiguity of the cue phrase. If the distribution that a sentence is causal is given as C_i, and D_i represents the distribution that a sentence contains cue phrase J, then the conditional probability is expressed as:

$$p(C_i \mid D_i) = \frac{p(C_i) \cap p(D_i)}{p(D_i)} \tag{8}$$

The precision with high values signifies a cue phrase that is not ambiguous. That is, the sentence is regarded as causal owing to the identified cue phrase. Conversely, a low precision value signifies an ambiguous cue phrase. From the analysis presented in Table 6, it is clear that various cue expressions could signify causality in REM. Interestingly, we observed that cue phrases such as *therefore, if* and *because* has a very high precision with minimum value of 86%. However, we equally observed that some cue phrases indicating causality in sentences also find expression in some non-causal scenarios, especially for cue phrases that are pronouns and verbs. But in general, the presence of verbs and pronouns as cue phrases in a sentence does not imply a causal sentence.

Table 6. Cue Phrases used for Identifying Causality Based on Grammar Features

Grammatical Group	**Cue Phrase**	**Causal**	**Non Causal**	**Precision**
Adverbs	As a result	16	9	0.64
	Due to	93	28	0.77
	Since	60	27	0.69
	Whenever	11	2	**0.85**

Continued on next page

Table 6 – *Continued from previous page*

Grammatical Group	**Cue Phrase**	**Causal**	**Non Causal**	**Precision**
	Then	121	80	0.60
	Therefore	72	17	**0.81**
	Consequently	6	10	0.34
	Rather	36	53	0.40
	To this (or that) end	18	2	**0.90**
	Thus	56	7	**0.89**
	Where	223	124	0.64
	When	343	68	**0.83**
	Based on this	4	2	0.67
	Hence	31	10	0.76
Conjunctions	In order to	164	24	**0.87**
	As long as	32	2	**0.94**
	Unless	12	3	**0.80**
	Except	4	6	0.4
	Because	82	12	**0.87**
	Once	58	25	0.69
	So that	184	43	**0.81**
	If	497	142	0.78
	As	597	12	**0.98**
	But	203	318	0.39
Prepositions	After	234	86	0.73
	During	439	106	**0.80**
	With	690	1230	0.36

Continued on next page

Table 6 – *Continued from previous page*

Grammatical Group	**Cue Phrase**	**Causal**	**Non Causal**	**Precision**
	As far as	6	50	0.10
	Around	219	103	0.68
	Upon	402	324	0.55
	Through	193	493	0.28
	Upon	35	58	0.38
	In response to	13	18	0.42
	In case of	725	638	0.53
	Until	69	32	0.68
	As part of	49	44	0.52
	From	869	1432	0.37
	Based on	51	165	0.23
	In both cases	4	1	0.80
	Before	72	31	0.69
	In the event of	26	9	0.74
	According to	44	53	0.45
	For	2672	3046	0.47
Adjectives	Prior to	181	29	**0.86**
	Given	78	146	0.34
	Only	176	101	0.64
	Imperative	6	11	0.35
	Necessary	34	16	0.68
Pronouns	Who	48	126	0.28
	Which	388	916	0.29

Continued on next page

Table 6 – *Continued from previous page*

Grammatical Group	**Cue Phrase**	**Causal**	**Non Causal**	**Precision**
	Whose	49	23	0.68
	That	933	173	**0.89**

Table 7. Cue Phrases used for Identifying Causality in REM Based on Causal Features

Causal Feature	**Cue Phrase**	**Causal**	**Non Causal**	**Precision**
Cause	Imply	28	49	0.36
	Result	149	93	0.62
	Lead	16	3	0.84
	Force	201	153	0.57
	Reduce	37	83	0.31
	Attain	8	45	0.15
	Perform	382	724	0.35
	Create	84	210	0.29
	Eliminate	12	84	0.13
	Impose	27	82	0.25
	Imply	91	219	0.29
	Trigger	492	239	0.67
	Maximize	399	639	0.38
	Minimize	628	493	0.56
	Degrade	372	38	0.91
Enable	Necessitate	14	4	0.78
	Achieve	389	183	0.68

Continued on next page

Table 7 – *Continued from previous page*

Causal Feature	**Cue Phrase**	**Causal**	**Non Causal**	**Precision**
	Rely on	18	22	0.45
	Need	198	262	0.43
	Allow	284	192	0.59
	Depend on	429	320	0.57
	Meet	73	69	0.51
	Provide	193	421	0.31
	Support	938	369	0.72
	Ensure	634	227	0.74
	Permit	320	294	0.52
	Enhance	92	39	0.70
	Enable	328	284	0.53
	Require	931	729	0.56
	Get	614	409	0.60
Prevent	Avoid	74	183	0.29
	Hinder	14	1	0.93
	Mitigate	19	104	0.15
	Prevent	249	14	0.95

Figure 7 illustrates the causality distribution in sentences across different domains (highlighted in Table 5). We ensured that a minimum of 120 sentences were extracted from each domain for a valid implementation analysis. Empirically, we observed that the percentage of a causal sentence in the REM is between 1.3% to 19.6% across all domains, which shows the importance and relevance of causality to all domains of interest but with a different statistical level of significance. The domain with the lowest and highest

percentage of causal relations based on all the sentences extracted from the 50 REM are infrastructure and science, respectively. Based on these findings, we can conclude that, to a certain degree, the distribution of causal values for all the categories is not dependent on the domain but on the influence of requirement goals, practitioner interest, and requirement developers. This is evidenced by the information provided in Table 8 showing the accuracy of the cue phrases across all the domains from the dataset.

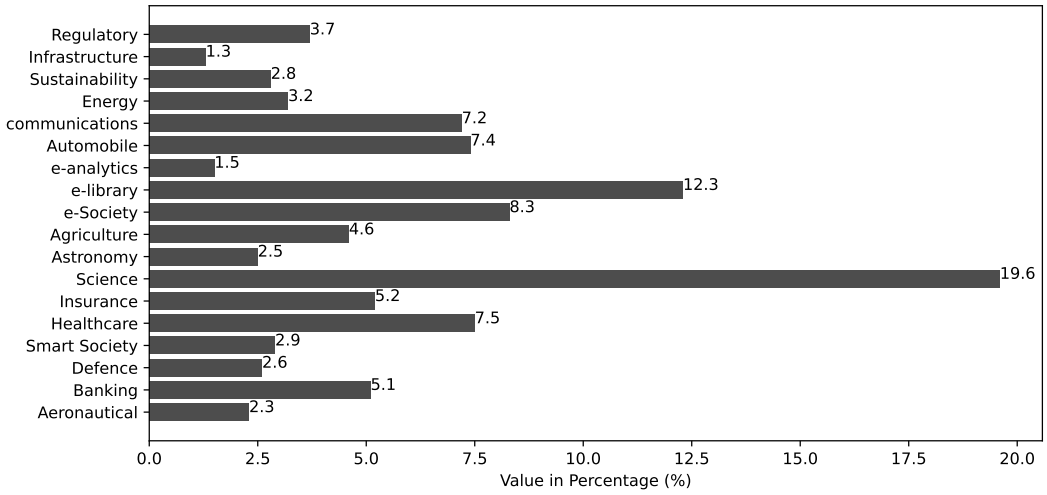

Figure 7. Causality Distribution Across Domain

Based on the information provided in Table 8, the cue phrase *"if, when, where"* was observed to be the three most used cue phrases across all the domains. The percentage distribution presented in Table 8 aligns with the evaluation reported in Tables 6 and 7, which also support our analysis that distribution of cue phrases is not associated with any domain. Evaluating the percentage of cue phrases for each dataset domain as presented in Table 8 helps us to aggregate our findings regarding identifying causal sentences in REM. Particularly by analyzing the impact of the cue phrases in causality detection in REM.

Table 8. Percentage of Cue Phrases per Dataset Domain

Domain	Most Recurring	2nd Most Recurring	3rd Most Recurring
Aeronautical	therefore (11.4%)	if (5.2%)	in order to (4.8%)
Banking	during (9.1%)	so that (5.6%)	when (2.4%)
Defence	to ensure (13.6%)	if (4.8%)	for (2.7%)
Smart City	result to (12.4%)	lead to (32.9%)	enables (22.5%)
Healthcare	allows (16.1%)	when (8.4%)	as far as (2.4%)
Insurance	increase (11.7%)	while (10.3%)	to provide (3.7%)
Science	based on (8.2%)	if (3.2%)	through (2.5%)
Astronomy	enables (14.5%)	allows (12.2%)	ensure (5.7%)
Agriculture	will require (6.2%)	therefore (5.2%)	if (2.1%)
e-society	allows (8.4%)	during (4.7%)	when (2.8%)
e-library	if (4.2%)	requires (3.2%)	where (2.3%)
e-analytics	therefore (12.4%)	if(32.9%)	in order to (22.5%)
Automobile	cause (5.3%)	reduce (3.4%)	since (2.9%)
Telecommunication	through (5.4%)	unless (4.3%)	allows (3.8%)
Energy	where (15.8%)	enable (4.3%)	within (3.1%)
Sustainability	during (9.5%)	when (7.3%)	once (5.2%)
Infrastructure	to get (7.2%)	which (6.9%)	so that (2.7%)
Regulatory	when (12.4%)	during (32.9%)	through (22.5%)

4.2 Hyperparameter Selection and Configuration

In order to ensure the optimal performance of our language and machine learning models, we performed a tuning operation on all the models by implementing a 10-fold cross validation technique. This method evaluated the search space for each model and presented the parameter combinations with the best model result based on the training dataset. Hence, each parameter value was noted and used to configure the model for the final classification task. The optimal parameter values with the best performance for all the models considered in this study is presented in Table 9.

Table 9. Model Hyperparameter Selection

	Model	Optimal Hyperparameter	Precision	Recall	F1	Accuracy
Machine Learning Algorithm	NB	Kernel: rbf, Estimator: GaussianNB(), Verbose: 1, Number of Jobs: -1	0.75	0.62	0.82	0.85
	KNN	Leaf size: 40, Number of neighbor: 20, Distance method: Manhattan	0.84	0.71	0.77	0.84
	AB	embed: Learning rate: 1, Number of Estimator: 80, Base estimator: Decision tree classifier, Estimator weight: 100	0.62	0.71	0.88	0.60
	SVM	embed: Kernel: rbf, Gamma:100, Cache size:250, Shrinking: true, Verbose: false, Maximum Iteration: 1	0.87	0.75	0.68	0.81
	RF	Max features: 15, Max depth: 8, Min split: 4, estimator: 35, Bootstrapping: True, Min leaf: 14	0.81	0.71	0.69	0.79
Deep Learning Algorithm	Base-BERT	Learning: 2e-4, Batch size: 32, Optimizer: adam, Training epochs: 4, Maximun sequence length: 128, Random state split: 42	0.69	0.72	0.61	0.70
	PoS-BERT		0.80	0.72	0.83	0.74
	Dep-BERT		0.66	0.72	0.74	0.69

4.3 Answers to Our Research Questions

We present the descriptive summary of our research questions in this section. The answer to these research questions assisted us in unpacking the measures needed to achieve our study goal and giving direction to our research activity.

- **RQ1: To what extent is causality present in REM?**

  Making remarks to Figure 6, we can ascertain that causality does occur in REM to a notable extent. From our analysis, about 31% of the sentences are causal. Hence, we can aggregate our findings to prove that causality is a major linguistic component or REM because of its ratio in the total sentences explored for this study

- **RQ2: How complex is it to identify and establish causality in REM?**

  As illustrated by Figure 6, many causal relation in REM has only one *cause* (of about 14.3%) and about 85.7% multiple cause. The presence of some complex causal relation accounted for the high number of multiple causal relation in sentences extracted from REM. We also noted a similarity in the sentence's distribution of *effect* and cause. But a single effect has more occurrence than multiple effects, and this could be attributed to limited number of effects (between two and three) even for complex relations. Based on the analysis, many of the single sentences have causal relation. Conversely, approximately 22% of the sentences has *cause* and *effect* deployed in at least two sentences. Also, by evaluating sentences with *marked* type, we observed that cue phrases *before* and *thus* has the highest number of usage. For the category labeled as event link, we observed a very low occurrence in sentences extracted from the REM. Furthermore, our findings show that many of the causal sentences are made of a separate causal relation, and about 13% contains *event link* category.

- **RQ3: What is the level of causal ambiguity in requirements?**

  We noted that one (or more) cue phrases are present in many of the causal sentences as shown in Figure 6, which represents the clear existence of causal relation between some particular events. From the annotation task, we noted that about 18% of the sentences are regarded as an *unmarked* causal relation. Further, we

observed that about 13% of the causal sentences are *implicit*. Similarly, many of the sentences that are categorized as causal are *explicit*, implying that they contain adequate information relating to the cause and effect (as defined in Section 3.1).

- **RQ4: How frequent does causal relations (such as *enable, prevent*, and *cause*) appear when analyzing requirement manuscript?**
  From the REM, we observed that many of the causal sentences (about 49%) refer to *enable* causal relation between some activities. Notably, about 19% of the causal sentences identified are regarded to have a *prevent* causal relation. Further, 32% of the causal sentences contain *cause* relation from the REM explored in this study.

- **RQ5: What is the degree of influence of temporal relations (such as *during, overlap*, and *before*) in a REM?**
  Empirically, we identified an approximate occurrence of cause and effect in a *during* and *before* relation. The *before* relation constitutes the most re-occurring temporal relation from the REM that was considered in this study with a margin of about 7% when compared with the *during* relation. Then, we observed that only about 5% of the sentences contain an *overlap* relation.

Overall, we attributed the strength of the causality detection and classification models to be significant when dataset from multiple domain is considered as it aids the model training capabilities, hence, robust functionality and applicability of the models can be guaranteed. In the next chapter, further discussion about our results as well as the use cases of causality is presented.

5 Discussion of Result

We presented the results obtained from this study in Chapter 4 of this thesis and the impact of different components of the conflict detection and resolution framework. In this chapter, we will expand more on the results presented in Chapter 4.

5.1 Significance of Causality Detection and Extraction

Making remarks on our study findings, we were able to deduce the following regarding causality management; (i) Causality is dominance in REM, hence, its relevance in RE practice. Based on this, researchers are inspired to implement applicable and novel frameworks for detecting and extracting causal relations and causal requirement for different analytical and research purposes, as well as identifying the impact of causality in REM. Although causality may tend to be complex to identify in some cases, the complexity could still be managed efficiently based on the researcher's understanding of the research direction. However, suppose the causality detection and extraction approach is aimed at being applied in practice. In that case, a proper understanding of conditions powered by multiple cause and effects is required for justifiable applicability. To this end, high degree of lexical and grammatical knowledge should be attained. They include proper comprehension of grammar elements such as conjunctions, negations, and disjunctions, as it will be required effectively to manage cause and effect relations for practical application.

However, we noted that *event link* and *two-sentence-based* causality have rare occurrences in REM. Similarly, *explicit* causal relation dominates *implicit* causal relation reducing the causality detection and extraction process. Furthermore, we observed that by analyzing the precision variable, many of the cue expressions used are indefinite (claim also evidenced by Stouby *et al.* [61]). In general, causal detection and extraction demand a high level of understanding of grammatical structures, as cue phrases are

insufficient to determine causal sentences, especially as new terms surface in RE practice. Hence, it is imperative for researchers to understand grammar, syntax, and semantics accurately to ensure the reliability of findings.

5.2 Effect of Causal Relation in Engineering Practice

Leveraging on the assertion of our findings regarding the scope, complexity, and structure of causality, it interests us to substantiate the effect of causality in RE practice. Hence, our investigation involves identifying the effect of causality in RE features. Based on this, we regard the features to mean factors that are particular to each requirement, such as span time. Two main goals motivated us to engage in the exploratory procedure. First, we presented an independent case study of the causality identification technique (e.g., as presented in Section 5.3) with the methods described in Chapter 3. Based on this, we considered causal occurrence as part of the factors for determining requirement competence which is evaluated by the capability of the causal detection technique. Using a computation approach for detecting and extracting causality for NL requirements (as considered in this study) fosters causality eligibility as a significant part of determining requirement performance. Secondly, combining and analyzing evidences of the factor as mentioned earlier by evaluating the relationship between causality requirements and features of the requirement.

5.3 Use Case of Causality in Requirement Engineering

To demonstrate the applicability of this study in the requirement engineering practice, we present two key use case where our findings is relevant and applied. They are presented below:

1. Generating Test Cases: In software testing, generating test cases is one of the major tasks, and in most cases, it could be strenuous [62]. According to Garousi

et al. [63], the lack of appropriate computational tools is a major limitation to generating test cases by practitioners. Most test case derivation relies on formal or semi-formal representation of test cases, evidenced by the ambiguity of interpreting NL [64]. However, practitioners must be well informed about the formation and system behavior of the product requirement in view to support their argument. In essence, the causal relations that determine the combination of the cases (covering both positive and negative cases) must be understood.

For instance, if we consider this scenario: *if the user exceeds 21 years and has a proper driver's licence, then the system should not levy an extra charge during checkout*. We can extract two causes in this case: $C1$ - user is older than 21 years; and $C2$ - user provides valid driver's licence. This implies that effect depends on the cause, which are linked by a conjunction, which can be represented as $C1 \wedge C2 \iff e1$. In current practice, practitioners manually extract causal relations and identify the cause(s) and effect scenarios relevant to the test case. This practice is not overwhelming but could also imply low level of accuracy, especially for large and complex requirements, because, generally, in practice, the potential test cases identify grows by 2^n, where n represents the number of causes [64].

Therefore, for software testing and RE practice, extraction of test cases together with automated derivation of test cases needs to align to ensure an efficient proposition by practitioners. This is advantageous because the proposed system behavior is automatically interpreted, and there is an extended test scenario because they are determined and identified by heuristic factors. Therefore, there is a need for a high degree of precision for this use case because false positives could cause invalid test cases, and a low estimate of recall could lead to omitted test cases, which depicts inaccuracy. In essence, a thorough evaluation and monitoring of the recall and precision is important because, due to human factors, human error in

false positives is inevitable.

2. Detection of Dependencies in Requirements: The level of increase and complexity of modern computational systems equally implies complex requirements [63]. Also, managing relations that exist in complex requirements could be tedious due to the unrestricted use of NL in requirements, resulting in a faulty system design. Additionally, solid system requirement knowledge is necessary to articulate the resultant effect of the modular updates (or reconfiguration) on the entire system's functionality. Our study submits that automated extraction of causality from REM can assist semantic comparison by careful analysis of causal relations empirically. Therefore, we assert that Causality detection and analysis contribute largely to the detection of dependencies as well as their types in requirement engineering. This claim was also evidenced by the study of Anne *et al.* in [61]. To illustrate this claim, four major requirement categories are considered and are described below:

    * Redundancy in Requirements: This describes a situation where one or more requirement depicts similar behavior of the system. In this case, practitioners or requirement engineers should consider only one of the requirements and exclude the others to constrain the requirement collection. This could be denoted by: $c1 \iff e1$ is coherent to $c1 \iff e1$.

    * Inter-Requirement Dependency: This type of requirement describes how one requirement is linked with another requirement. For instance, if an effect in a causal relation $r1$ is a cause in another causal relation $r2$, then there is high probability that $r2$ is dependent on $r1$. In this requirement type, requirement traceability is easy for the practitioners, such that if there is change in $r1$, they could accurately predict the requirements that could be affected by this change. Hence, this notion boosts the efficiency of their RE practice during product design and product modification, i.e., $c1 \iff e1$ depends on

$e1 \iff e2$.

* Contrasting Requirements: In this requirement category, there is contradiction in causal relations; requirement content conflicts with each other. This is disadvantageous because it could result in instability in product design. In this requirement type, practitioners remediate this issue by proper investigation practice from end users, i.e., $c1 \iff e1$ contrast $c1 \iff \neg e1$.
* Requirement Modification: When the system functionality defined in a requirement is precisely defined in another requirement. For example, if a causal relation $r1$ defines a subset of another relation $r2$, then this type of requirement implies that $r1$ modifies $r2$. The computation formation is represented as: $C1 \iff e1$ modifies $C1 \wedge C2 \iff e1$.

5.4 Relevance of Causality Detection

The exploratory findings of this study *w.r.t* causality study show the degree of relevance of causality in RE practice. The small degree of feature correlation and their effect affirms that causality is not just significant but definitely a major variable during requirement feature analysis. This notion is important and relevant in practice, especially during product auditing and product performance feedback for practitioners. In general, proper investigation and insight into causality management can improve the efficiency and effectiveness of a product deliverable, especially when the behavioral analysis of the product's end user is of utmost importance to practitioners.

One of the key findings of this study is how we have been able to ascertain that there is the correlation between features of REM and the occurrence of causal relations. Based on these findings, it is worth noting that causality is a major component when evaluating the quality of requirements, especially in RE practice. From our analysis, we note that causality slightly correlates with the requirement's time. Hence, we can consider causality a key factor in estimating the requirement's life cycle. Notably, a firm

semantic structure of the causal relation could affect the degree of comprehension of a requirement, implying an easy representation and disintegration of complex scenarios into sub-components like test cases or programming code.

5.5 Reproducibility of Study Findings

To reproduce our study's framework and obtain similar results, our implementation details can be accessed by clicking here.

6 Threats to Validity

The threats to validity of our study is presented to demonstrate the area where computational and theoretical adjustments (and improvement) can be effected for an augmented result. Hence, we have highlighted three major threats to the validity of this study, they are:

* **Analytical Mapping Constraint:** To ascertain that the correlation of requirements from inception to completion and the presence of causality is not discomposed but causal, it is recommended that an extended qualitative assessment beyond the dataset explored should be performed. This could be achieved by considering the other supporting documents' requirement history of the product in view. In addition, this should be managed effectively because the analyzed data may have incomplete or incorrect information, which could result from low diligence or oversight during the requirements gathering process.

* **Data Annotation:** Our annotation process poses a threat to validity because many annotation tasks could be arbitrary due to different human factors, especially regarding representation and interpretation. For instance, implicit causality category could be difficult to manually identify due to its features as highlighted in Chapter 3. However, we performed the following activities to minimize the human-based annotation error. First, we performed a training (details in Section 3.12) for the annotators about the annotation structural modalities and guidelines. Secondly, we applied the Cohen kappa assessment technique (highlighted in Section 3.11.1 of Chapter 3) to evaluate and measure the degree of inter-rater annotation agreement. However, we noted that all excluding causality categories rely on associated sentence classification, which could affect the annotator's agreement for other categories. The computation of inter rater arrangement provides evidence to this

claim where all categories (excluding causality) are estimated *w.r.t* the identified causal sentence.

Hence, we conclude that the remaining categories are less relevant for identifying sentences that are not causal because they refer to the causal relation in the sentence. In addition, the coherence of the inter rater is not measured based on the same (or similar) domain. Hence, it could be difficult to estimate if some domains are responsible for the disagreement of notions among the annotators. As reported in Table 4, our result analysis was justifiable. Still, we recommend that a domain-based annotation should be encouraged, which would also improve the accuracy of the inter-rater agreement.

Additionally, considering only two consecutive sentences for manual causality detection could imply limited scope because the causal relation in more than two sentences would not be captured. Although, this effect may be minimal because one-sentence-based and two-sentenced-based causal relations could be used as leverage to determine causality in successive sentences. But for further studies, multiple-sentence-based (between two and five sentences) could be considered to have a more robust causal analysis.

* **Data Extraction:** In order for us to achieve general applicability to the RE practice irrespective of domain, the dataset from different domains are considered, as presented in Figure 2. Referencing Figure 5, the domains smart city, data analytics, and aerospace are the major sentence contributors with over 10, 000 sentences. All other domains also contribute sentences to the causality management models through machine models. This threat to validity was reduced by incorporating domain-specific analysis. To this end, we recommend that the dataset from more domains that could contribute meaningfully to the causal analysis technique should be considered for future studies.

7 Conclusion and Future Work

The system behavior is usually established with respect to causal relations in REM made up of natural language. If the causal information in REM are efficiently extracted, then downstream activities in product (software) development is enhanced, especially for those that rely on causal analysis (e.g., test case derivation and other requirement dependencies) [25, 27]. However, some contemporary causality analysis methods do not perform satisfactorily according to the findings and survey of Ruocheng *et al.* [65]. Therefore, in this study, we addressed the notable gaps identified in contemporary methods of causality analysis, both highlighted in Chapter 2 and identified by Ruocheng *et al.* [65].

We have used machine learning and deep learning models configured with optimal hyperparameters to perform our causality extraction and classification task. In this study, we noted that causality identification and extraction follow two steps. First, we need to assert if REM contains causal features. Second, we must identify and extract the causal relationships found in these REM. However, we may be unable to decode the degree of complexity and form of causality in the REM, which determines the best methodology for solving these issues.

This study focuses on how the prevalence of causal relations in natural language requirements affects important aspects of the REM. Having an empirical fact to evidence the prevalence of causality on the features identified in the REM (especially as it relates to scope, complexity level, and structure) supports the notion that the application of causality in natural language requirements contributes positively to the quality and standard of the requirement. Interestingly, determining a definite causality relation based on correlation assessment alone may not be feasible, but the exploratory and descriptive case study focuses on giving a better understanding of the strength of applying causal management as one of the reliable aspects of RE. In our empirical analysis of causality

management, the following under-listed features were selected to evaluate the influence of requirement features on the formation of requirement artifacts by practitioners.

* Modification Activity: which points to the number of configuration changes of the requirement

* Conclusive State: which refers to the final state of the requirement

* Span Time: this refers to the duration between the inception of the requirement till it is finalized and ready to guide the implementation phase of the product development

7.1 Contribution to Body of Knowledge

To show the relevance of our implementation analysis to RE practice, we closed the gaps identified in current research. We implemented a computational framework that can assist software design practitioners in improving deliverables through an analytical study of causality in RE practice. This is achieved particularly by considering the extent, form, and prevalence of causality in REM, as these factors largely influence the requirement specification of software products. Our research contributions are summarized as follows:

Contribution 1: We performed an exploratory case study where 12,438 sentences were extracted from 50 REM which originated from different domains (as presented in Table 2. The use cases showing the applicability of our study are also presented in Section 5.3. Our feasibility research found that causality is a regularly utilized language pattern to describe system functionalities, and it is most often used in a marked and explicit form.

Contribution 2: We are motivated to implement machine models capable of identifying and detecting causality in REM to address the challenges identified in the aforementioned use cases. Empirically, we evaluated our machine models with evaluation metrics with detailed results presented in Table 9. Overall, all our machine models performed

excellently after an adequate configuration process. Still, Naive Bayes was the best performing model as we achieved an accuracy of 0.88%, F1 score of 0.69, recall of 0.75, and precision of 0.86.

Contribution 3: We publicly disclosed the dataset and the source code used for our implementation analysis to foster result reproducibility, easy comprehension, and improvement of our study by other researchers who have interest in this research line. In addition, we noted that as a norm in the scientific community, giving access to the dataset and source code also demonstrates the credibility of our research. Hence, we provided the path to access this information by clicking this link here.

7.2 Direction for Further Studies

Further study could be performed to expand on the findings from our research. Below are some recommendations that could be considered for further study by researchers in this line of research.

1. The sentences extracted from the REM could be analyzed in a granular manner by grouping them accordingly to their respective class before they are modeled into the machine language. For instance, functional and non-functional-based requirements could improve insight into causality analysis in REM. They would be more effective for each category instead of their application to the general REM. This process would involve investigating the features that are particular to each category.

2. The process of extracting causal relations from causal sentences provides the required basis that can fit into different use cases and applications of causality. In this sense, further research could be performed by extensively considering the relationship between the requirement features vis-a-vis the presence and effect of causality in a broader notion. Therefore, the limitations in the current intra-

sentential approach that is considered in this study could be mitigated with a more robust computational framework by holistically exploring the relationship between requirement features and requirement dependencies.

References

[1] J. Sim and C. C. Wright, "The Kappa Statistic in Reliability Studies: Use, Interpretation, and Sample Size Requirements," *Physical Therapy*, vol. 85, no. 3, pp. 257–268, 03 2005. [Online]. Available: https://doi.org/10.1093/ptj/85.3.257

[2] C. S. G. Khoo, J. Kornfilt, R. N. Oddy, and S.-H. Myaeng, "Automatic extraction of cause-effect information from newspaper text without knowledge-based inferencing," *Literary and Linguistic Computing*, vol. 13, pp. 177–186, 2008.

[3] J. Fischbach, A. Vogelsang, D. Spies, A. Wehrle, M. Junker, and D. Freudenstein, "Specmate: Automated creation of test cases from acceptance criteria," *2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST)*, pp. 321–331, 2020.

[4] R. Girju and D. I. Moldovan, "Mining answers for causation questions," 2002.

[5] N. Asghar, "Automatic extraction of causal relations from natural language texts: A comprehensive survey," *ArXiv*, vol. abs/1605.07895, 2016.

[6] J. Mund, D. M. Fernandez, H. Femmer, and J. Eckhardt, "Does quality of requirements specifications matter? combined results of two empirical studies," in *2015 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE, 2015, pp. 1–10.

[7] F. Salger, "Requirements reviews revisited: Residual challenges and open research questions," in *2013 21st IEEE International Requirements Engineering Conference (RE)*, 2013, pp. 250–255.

[8] E. Blanco, N. Castell, and D. I. Moldovan, "Causal relation extraction," in *LREC*, 2008.

[9] C. S. G. Khoo, J. Kornfilt, R. N. Oddy, and S.-H. Myaeng, "Automatic extraction of cause-effect information from newspaper text without knowledge-based inferencing," *Literary and Linguistic Computing*, vol. 13, pp. 177–186, 2012.

[10] C. Puente and J. A. Olivas, "Analysis, detection and classification of certain conditional sentences in text documents," *Information Processing and Management Journal*, vol. 2, pp. 1097–1104, 2008.

[11] B. Rink and S. M. Harabagiu, "Utd: Classifying semantic relations by combining lexical and semantic resources," in **SEMEVAL*, 2010.

[12] D.-S. Chang and K.-S. Choi, "Causal relation extraction using cue phrase and lexical pair probabilities," in *IJCNLP*, 2012.

[13] Z. Li, Q. Li, X. Zou, and J. Ren, "Causality extraction based on self-attentive bilstm-crf with transferred embeddings," *ArXiv*, vol. abs/1904.07629, 2021.

[14] T. Dasgupta, R. Saha, L. Dey, and A. Naskar, "Automatic extraction of causal relations from text using linguistically informed deep neural networks," in *SIGDIAL Conference*, 2018.

[15] R. Narayanan, B. Liu, and A. N. Choudhary, "Sentiment analysis of conditional sentences," in *EMNLP*, 2009.

[16] T. N. de Silva, X. Zhibo, Z. Rui, and M. Kezhi, "Causal relation identification using convolutional neural networks and knowledge based features," *World Academy of Science, Engineering and Technology, International Journal of Mechanical and Mechatronics Engineering*, vol. 11, pp. 696–701, 2017.

[17] H. H. Patel and P. Prajapati, "Study and analysis of decision tree based classification algorithms," *International Journal of Computer Sciences and Engineering*, vol. 6, no. 10, pp. 74–78, 2018.

[18] K. Nigam, A. McCallum, S. Thrun, and T. M. Mitchell, "Text classification from labeled and unlabeled documents using em," *Machine Learning*, vol. 39, pp. 103–134, 2004.

[19] R. Girju, "Automatic detection of causal relations for question answering," in *ACL 2003*, 2003.

[20] R. Higashinaka and H. Isozaki, "Automatically acquiring causal expression patterns from relation-annotated corpora to improve question answering for why-questions," *ACM Trans. Asian Lang. Inf. Process.*, vol. 7, pp. 6:1–6:29, 2008.

[21] D.-S. Chang and K.-S. Choi, "Incremental cue phrase learning and bootstrapping method for causality extraction using cue phrase and word pair probabilities," *Inf. Process. Manag.*, vol. 42, pp. 662–678, 2006.

[22] T. Inui, K. Inui, and Y. Matsumoto, "Acquiring causal knowledge from text using the connective marker tame," *ACM Trans. Asian Lang. Inf. Process.*, vol. 4, pp. 435–474, 2005.

[23] J. Xu, W. Zuo, S. Liang, and X. Zuo, "A review of dataset and labeling methods for causality extraction," in *COLING*, 2020.

[24] C. S. G. Khoo, S. Chan, and Y. Niu, "Extracting causal knowledge from a medical database using graphical patterns," in *ACL*, 2008.

[25] S. Doan, E. W. Yang, S. Tilak, and M. Torii, "Using natural language processing to extract health-related causality from twitter messages," *2018 IEEE International Conference on Healthcare Informatics Workshop (ICHI-W)*, pp. 84–85, 2018.

[26] C. Hashimoto, K. Torisawa, J. Kloetzer, and J.-H. Oh, "Generating event causality hypotheses through semantic relations," in *AAAI*, 2015.

[27] J. Frattini, J. Fischbach, D. Méndez, M. Unterkalmsteiner, A. Vogelsang, and K. Wnuk, "Causality in requirements artifacts: prevalence, detection, and impact," *Requirements Engineering*, pp. 1–26, 2021.

[28] N. Asghar, "Automatic extraction of causal relations from natural language texts: A comprehensive survey," *ArXiv*, vol. abs/1605.07895, 2016.

[29] A. J. Viera and J. M. Garrett, "Understanding interobserver agreement: the kappa statistic." *Family medicine*, vol. 37, pp. 360–369, 2005.

[30] W. Phillip, "Representing causation," *Journal of experimental psychology. General*, vol. 136, pp. 82–111, 03 2007.

[31] M. Kyriakakis, I. Androutsopoulos, J. G. i Ametll'e, and A. Saudabayev, "Transfer learning for causal sentence detection," *ArXiv*, vol. abs/1906.07544, 2019.

[32] I. Reklos and A. Merono-Penuela, "Medicause: Causal relation modelling and extraction from medical publications," in *TEXT2KG/MK@ESWC*, 2022.

[33] S. Gupta, "Finlp at fincausal 2020 task 1: Mixture of berts for causal sentence identification in financial texts," in *FNP*, 2020.

[34] M. Jantscher and R. Kern, "Causal investigation of public opinion during the covid-19 pandemic via social media text," in *LREC*, 2022.

[35] X. Jin, X. Wang, X. Luo, S. Huang, and S. Gu, "Inter-sentence and implicit causality extraction from chinese corpus," *Advances in Knowledge Discovery and Data Mining*, vol. 12084, pp. 739 – 751, 2020.

[36] C. Kruengkrai, K. Torisawa, C. Hashimoto, J. Kloetzer, J.-H. Oh, and M. Tanaka, "Improving event causality recognition with multiple background knowledge sources using multi-column convolutional neural networks," in *AAAI*, 2017.

[37] C. N. Network, “Cnn site map - articles for year 2022,” https://edition.cnn.com/article/sitemap-2022.html, 2022.

[38] P. Wolff, G. Song, and D. M. Driscoll, “Models of causation and causal verbs,” 2002.

[39] N. Mostafazadeh, A. Grealish, N. Chambers, J. F. Allen, and L. Vanderwende, “Caters: Causal and temporal relation scheme for semantic annotation of event structures,” in *EVENTS@HLT-NAACL*, 2016.

[40] R. Girju and D. Moldovan, “Text mining for causal relations,” 10 2002.

[41] J. Pustejovsky and A. Stubbs, “Natural language annotation for machine learning - a guide to corpus-building for applications,” 2012.

[42] M. I. Jordan and T. Mitchell, “Machine learning: Trends, perspectives, and prospects,” *Science*, vol. 349, pp. 255 – 260, 2015.

[43] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT press, 2016.

[44] E. Rudkowsky, M. Haselmayer, M. Wastian, M. Jenny, Š. Emrich, and M. Sedlmair, “More than bags of words: Sentiment analysis with word embeddings,” *Communication Methods and Measures*, vol. 12, no. 2-3, pp. 140–157, 2018.

[45] Y. HaCohen-Kerner, D. Miller, and Y. Yigal, “The influence of preprocessing on text classification using a bag-of-words representation,” *PloS one*, vol. 15, no. 5, p. e0232525, 2020.

[46] D. Berrar, “Bayes’ theorem and naive bayes classifier,” *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, vol. 403, 2018.

[47] K. Taunk, S. De, S. Verma, and A. Swetapadma, "A brief review of nearest neighbor algorithm for learning and classification," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*. IEEE, 2019, pp. 1255–1260.

[48] R. E. Schapire, "Explaining adaboost," in *Empirical inference*. Springer, 2013, pp. 37–52.

[49] R. J. Gove and J. Faytong, "Machine learning and event-based software testing: Classifiers for identifying infeasible gui event sequences," *Adv. Comput.*, vol. 86, pp. 109–135, 2012.

[50] L. Cao, K. Chua, W. Chong, H. Lee, and Q. Gu, "A comparison of pca, kpca and ica for dimensionality reduction in support vector machine," *Neurocomputing*, vol. 55, pp. 321–336, 09 2003.

[51] L. Breiman, "Random forest machine learning technique," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[52] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *ArXiv*, vol. abs/1810.04805, 2019.

[53] F. Souza, R. Nogueira, and R. Lotufo, "Bertimbau: pretrained bert models for brazilian portuguese," in *Brazilian conference on intelligent systems*. Springer, 2020, pp. 403–417.

[54] D. Sundararaman, V. Subramanian, G. Wang, S. Si, D. Shen, D. Wang, and L. Carin, "Syntax-infused transformer and bert models for machine translation and natural language understanding," *ArXiv*, vol. abs/1911.06156, 2019.

[55] M. Fares, A. Kutuzov, S. Oepen, and E. Velldal, "Word vectors, reuse, and replicability: Towards a community repository of large-text resources," in *NODALIDA*, 2017.

[56] T. Hiraoka, H. Shindo, and Y. Matsumoto, "Stochastic tokenization with a language model for neural text classification," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 1620–1629.

[57] M. J. Warrens, "Five ways to look at cohen's kappa," *Journal of Psychology & Psychotherapy*, vol. 5, no. 4, p. 1, 2015.

[58] N. J.-M. Blackman and J. J. Koval, "Interval estimation for cohen's kappa as a measure of agreement," *Statistics in medicine*, vol. 19, no. 5, pp. 723–741, 2000.

[59] A. Davies, L. Wellard-Cole, A. Rangan, and M. Allman-Farinelli, "Validity of self-reported weight and height for bmi classification: A cross-sectional study among young adults," *Nutrition*, vol. 71, p. 110622, 2020.

[60] S. Moore, *Deep Learning for Computer Vision*. Packt Publishing Limited, 2018.

[61] y. Asa G. Dahlstedt and Anne Stouby Persson, "Requirements interdependencies: State of the art and future challenges."

[62] P. Kulkarni and Y. Joglekar, "Generating and analyzing test cases from software requirements using nlp and hadoop," 2014.

[63] V. Garousi, S. Bauer, and M. Felderer, "Nlp-assisted software testing: a systematic review," *ArXiv*, vol. abs/1806.00696, 2018.

[64] M. Kassab, C. Neill, and P. Laplante, "State of practice in requirements engineering: Contemporary data," *Innovations in Systems and Software Engineering: A NASA Journal*, 12 2014.

[65] R. Guo, L. Cheng, J. Li, P. R. Hahn, and H. Liu, “A survey of learning causality with data: Problems and methods,” *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–37, 2020.

I. Glossary

Acronym	Description
AB	Adaptive Boost
AI	Artificial Intelligence
BERT	Machine Learning
DL	Deep Learning
ELMO	Embedding from Language Model
GRU	Gated Recurrent Units
KNN	K-Nearest Neighbors
ML	Machine Learning
NL	Natural Language
RE	Requirements Engineering
REM	Requirement Engineering Manuscript
RF	Random Forest
SRS	Software Requirement Specification
SVM	Support Vector Machine
LDA	Latent Dirichlet allocation
SDLC	Software Development Life Cycle
NLTK	Natural Language Tool Kit
W.R.T	with respect to

II. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Olumide Olugbenga Oluyide**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Causality Management and Analysis in Requirement Manuscript for Software Designs,

   supervised by Ishaya Peni Gambo, PhD.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Olumide Olugbenga Oluyide

5/1/2023