USING NATURAL LANGUAGE PROCESSING FOR AUTOMATED CONSTRUCTION CONTRACT REVIEW DURING RISK ASSESSMENT AT THE BIDDING STAGE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

GÖRKEM EKEN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
CIVIL ENGINEERING

DECEMBER 2022

Approval of the thesis:

USING NATURAL LANGUAGE PROCESSING FOR AUTOMATED CONSTRUCTION CONTRACT REVIEW DURING RISK ASSESSMENT AT THE BIDDING STAGE

submitted by **GÖRKEM EKEN** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy** i**n Civil Engineering, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Erdem Canbay
Head of the Department, **Civil Engineering** _____

Prof. Dr. Mustafa Talat Birgönül
Supervisor, **Civil Engineering, METU** _____

Prof. Dr. İrem Dikmen Toker
Co-Supervisor, **Construction Man. And Eng., Reading Uni.** _____

Examining Committee Members:

Prof. Dr. Rifat Sönmez
Civil Engineering, METU _____

Prof. Dr. Mustafa Talat Birgönül
Civil Engineering, METU _____

Assist. Prof. Dr. Güzide Atasoy Özcan
Civil Engineering, METU _____

Assist. Prof. Dr. Gözde Bilgin
Civil Engineering, Başkent University _____

Assist. Prof. Dr. Hüseyin Erol
Civil Engineering, Hacettepe University _____

Date: 21.12.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Last name : Görkem Eken

Signature :

ABSTRACT

USING NATURAL LANGUAGE PROCESSING FOR AUTOMATED CONSTRUCTION CONTRACT REVIEW DURING RISK ASSESSMENT AT THE BIDDING STAGE

Eken, Görkem
Doctor of Philosophy, Civil Engineering
Supervisor : Prof. Dr. M. Talat Birgönül
Co-Supervisor: Prof. Dr. İrem Dikmen Toker

December 2022, 134 pages

Construction sector contains various risks, and construction projects are open to failure due to their nature and the involvement of multiple parties. Contracts are legal documents that are used to define the responsibilities of parties in a project. Risks that are taken by a party are highly related to their positions in the contracts. However, contracts are prepared by clients, and contractors generally do not have enough time to review their responsibilities before preparing their bids. Moreover, contracts may not always be clear in terms of all subjects. As a result, it may lead to ambiguities. Advances in information technology (IT) may provide solutions to the construction sector in this area. Natural Language Processing (NLP) focuses on using computers to understand, process, and manipulate natural language text to achieve a variety of objectives. NLP can be used to review contract documents within seconds depending on the volume of the documents and available processing power. In this study, FIDIC standard forms of contracts were selected and all sentences were labeled with sentence type and ownership in order to create a training dataset for machine learning applications. In addition to the training dataset, the test dataset was

created by using a contract of the real construction project. By using created datasets, 5 different machine learning algorithms were trained with different NLP techniques. The results of 12 machine learning models were evaluated, and the selected ones were combined by the ensemble method. In conclusion, sentence types in a FIDIC contract were categorized 89% accurately as heading, definition, obligation, risk, and right. Additionally, related parties for sentences that imply risk right and obligation were predicted 83% accurately. The proposed method can be used by contractors to quickly classify the contract text in order to identify the contractual risks required to decide risk premiums in the tender preparation phase.

Keywords: Construction Contract Review, Machine Learning, Natural Language Processing, Text Classification, Deep Learning

ÖZ

TEKLİF AŞAMASINDA RİSK DEĞERLENDİRMESİ YAPARKEN DOĞAL DİL İŞLEME KULLANARAK İNŞAAT SÖZLEŞMELERİNİN OTOMATİK OLARAK GÖZDEN GEÇİRİLMESİ

Eken, Görkem
Doktora, İnşaat Mühendisliği
Tez Yöneticisi: Prof. Dr. M. Talat Birgönül
Ortak Tez Yöneticisi: Prof. Dr. İrem Dikmen Toker

Aralık 2022, 134 sayfa

İnşaat sektörü çeşitli riskler içermesi ve inşaat projelerinin doğası ve birden fazla tarafın katılımı nedeniyle başarısızlığa açıktır. Sözleşmeler, tarafların sorumluluklarını tanımlamak için kullanılan yasal belgelerdir. Taraflarca üstlenilen riskler, sözleşmelerdeki pozisyonları ile son derece ilişkilidir ancak, sözleşmelerin işverenler tarafından hazırlanması nedeniyle yükleniciler, tekliflerini hazırlamadan önce sorumluluklarını gözden geçirmek için yeterli zamana sahip olmayabilirler. Bununla birlikte sözleşmeler her zaman sorumlulukların paylaşımı/tanımlanması konusunda net olmamakta ve bu durum sonuç olarak belirsizliklere yol açmaktadır. Bu kapsamda; bilgi teknolojilerindeki gelişmelerin, inşaat sektörüne bu hususta ki problemlerin çözümü açısından fayda sağlayabileceği düşünülmektedir. Doğal Dil İşleme (DDİ) ile belge hacmine ve mevcut donanıma bağlı olarak sözleşme belgelerini saniyeler içinde incelemek mümkün olabilecektir. Bu çalışmada, makine öğrenimi uygulamaları için bir eğitim veri seti oluşturmak amacıyla FIDIC standart sözleşme formları seçilmiş ve tüm cümleler cümle türü ve sahiplik ile etiketlenmiştir. Eğitim veri setine ek olarak gerçek bir projenin sözleşmesi kullanılarak test data seti

oluşturulmuştur. Oluşturulan veri seti ile 5 farklı makine öğrenme algoritması çeşitli doğal dil işleme teknikleriyle eğitilmiş, 12 makine öğrenmesi modelinin sonuçları değerlendirilmiş ve seçilen modeller topluluk öğrenmesi metodu kullanılarak birleştirilmiştir. Sonuç olarak, bir FIDIC sözleşmesinde yer alan cümle türleri; başlık, tanım, yükümlülük, risk ve hak sahipliği olarak %89 oranında doğru olarak sınıflandırılmış, ayrıca risk hak sahipliği, yükümlülük içeren cümleler için ilişkili taraflar %83 oranında doğru tahmin edilmiştir. Önerilen yöntem, müteahhitler tarafından ihale hazırlık aşamasında risk primlerine karar vermek için gerekli olan sözleşmesel risklerin belirlenmesi için kısa sürede sözleşme metnini sınıflandırmak için kullanılabilir.

Anahtar Kelimeler: İnşaat Sözleşmesi İnceleme, Makine Öğrenmesi, Doğal Dil İşleme, Metin Sınıflandırma, Derin Öğrenme

Dedicated to Poyraz

ACKNOWLEDGMENTS

I would like to express my sincerest thanks and regards to my advisors, Prof. Dr. M. Talat Birgönül and Prof. Dr. İrem Dikmen Toker, for their encouragement, unlimited support, guidance, and invaluable help throughout my study.

I would also like to express my great thanks to Scientific and Technological Research Council of Turkey (TUBITAK), which provided a scholarship to me throughout the accomplishment process of this dissertation.

I am sincerely thankful to my mother Fatma Eken and my father Ramazan Eken. I am very proud to be their son. I am also very thankful to my dear wife, Hazal Eken, for her priceless love and support.

I want to thank my friends Berk Demir, Görkem Doğan and Caner Utku for their invaluable help and great friendships.

I also would like to thank my Manager Mr. M. Fatih Koçak for allowing me to use academic leave and my colleagues for helping me in my absence in the workplace during this dissertation.

Finally, I would like to express my eternal gratitude to the founder of the Republic of Turkey: Mustafa Kemal Atatürk. He not only provided me with an independent research environment but also gave me courage, discipline, and faith.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	
ACC	Automated Compliance Checking
AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
BI&A	Business Intelligence and Analytics
BoW	Bag-of-Words
CNN	Convolutional Neural Networks
CRISP	Concept Relation Identification using Shallow Parsing
FIDIC	International Federation of Consulting Engineers
GATE	General Architecture for Text Engineering
IE	Information Extraction
ML	Machine Learning
NER	Named Entity Recognition
NL	Natural Language
NLARE	Natural Language Automatic Requirement Evaluator
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
POS	Part of Speech
RE	Requirement Engineering
RETA	Requirement Template Analyzer

RNN	Recurrent Neural Network
TC	Text Classification
TF-IDF	Term Frequency-Inverse Document Frequency

CHAPTER 1

INTRODUCTION

Brief information about the study is presented in this chapter. The research background section introduces the topic and current situation in the construction sector. In the next section, the aim of the study is presented, and the framework of the study is presented in the following section.

In the research methodology section, the steps followed throughout the study are given in detail. The outline of the study is presented in the organization of the thesis section.

1.1 Research Background

Construction projects are defined as high-risk prone projects mainly due to the involvement of multiple parties (El-Sayegh, 2008), limited project time and budget (Zeng et al., 2007), organizational and technological complexity and open production system to the environment (Taroun, 2014). Uncertain events and circumstances that affect project objectives, such as time and cost, can be considered as risks for projects (Dikmen et al., 2018). Project success depends highly on how risks are managed successfully. Three main steps must be implemented to achieve successful risk management, which can be listed as risk identification, risk assessment, and defining risk response strategies (Dikmen et al., 2007). Risks must be identified correctly to be successful in risk management.

The ideas of the owner are communicated to the contractors by the contract clauses (Mendis et al., 2015). A comprehensive contract between parties must specify an action for each party for every possible situation. If all potential sources of

disagreement are foreseen in the contract design phase, no conflict should arise. However, contracts do not specify entirely what the parties should do, and disputes occur regularly (Grant et al., 2012). Table 1.1, which is taken from a study conducted by Akintoye and MacLeod (1997), shows that contractual risk sources are the second important item that affects risk premiums. This implies that construction contracts must be reviewed thoroughly to be successful in risk management.

Table 1.1 Construction Sector Risk Premium Indexes provided by Akintoye and MacLeod (1997)

	Contractors	*Project Managers*	*All firms*
Financial	3.50	3.55	3.50
Contractual Arrangements	3.40	3.54	3.44
Market	2.90	3.00	2.95
Project	2.69	3.08	2.88
Construction	2.93	2.50	2.71
Company	2.50	2.58	2.56
Political, Social Economic	2.52	2.20	2.37
Environmental	2.33	1.69	2.05
Development in IT	1.89	1.71	1.81

Contracts consist of various textual documents, such as the general conditions of the contract and specifications, which are prepared by using natural languages (Al Qady & Kandil, 2010).

Terms in contract documents can lead to disagreements between contracting parties. Standard-form contracts have developed because of the increasing complexity of construction works and the difficulty in drafting tailored contract terms for each project. Using a standard form of contract brings various advantages. On the other hand, owners often change some parts of the standard forms of contracts to include specific requirements for a project (Rameezdeen & Rodrigo, 2014). Parties that prepare the first draft of contract documents try to transfer risks to other parties with deterrent provisions in order to avoid the claims (Mendis et al., 2013). The contractual positions and rights of the owner or contractor must be guaranteed based

on contractual facts to avoid claims and disputes emerging while the construction works are being executed. For this reason, contractual elements that may bring out risk and define rights or obligations need to be identified and responded to in advance (Lee et al., 2019). Another reason for claims can be stated as an inadequate definition of the scope of a contract and/or specification (Hayati et al., 2019). The language that transfers the idea from person to person is actually based on interpretation, and different interpretations change the meaning of the text. As stated by Chomsky (Noam, 1973), although the principles of the language are well-known, the "manner" in which the principles are evaluated is free and has infinite variations. Contracts will lead to disputes and litigation if the parties differ in interpretations of the applicable conditions, including required actions (Grant et al., 2014).

Traditionally organizations try to review construction contracts with the help of professionals who are knowledgeable in the area. This manual task is time-consuming, and clients are not giving sufficient time to review all documents during bidding periods in recent years (Lee et al., 2019). Although the time provided in the bidding stage is not sufficient to review the contract, it is essential because if a dispute occurs and the condition of the contract adversely affects the contractor, there is no way to avoid the risk. Therefore, contracts must be reviewed thoroughly at the bidding and contracting stage regarding inherent risk. Claims and disputes that cause significant financial losses and legal actions may arise if this is not performed (Lee et al., 2019).

Although legal professionals tried to asses risks in documents in detail, the possibility of error remains due to unidentified or misinterpreted risk elements. Therefore, there is a growing demand for intelligent systems that automatically analyze contracts to guarantee that clauses are accurately defined and categorized in contracts with minimal human intervention (Chakrabarti et al., 2018). An in-depth review of contracts cannot be made during the short bidding period, as contractors simultaneously review contract terms and technical documents such as design drawings and specifications (Lee et al., 2019). The time and cost required for manual evaluation of legal documents demonstrate the need to develop an Artificial

Intelligence (AI) based system that makes risk analysis of contracts fast, error-free, and person-independent; thus, the decision becomes more effortless (Chakrabarti et al., 2018).

As a result, contractual facts that may create risks must be identified in advance (Lee et al., 2019). Therefore, automated analysis of contracts can be a solution for the early identification of contractual risks. This can be a solution for eliminating mistakes made by human evaluators. Intelligent systems that are capable of automated analysis of text may provide correct interpretation with minimum intervention of humans in very short time periods.

As a result, this research focuses on investigating automated construction contract review methods by implementing artificial intelligence tools, namely Natural Language Processing (NLP) and Machine Learning (ML) in order to provide a method for contractors to be used in risk contractual risk identification for deciding risk premium in the bidding stage.

1.2 Research Objectives

The study aims to propose a process and provide its implementation results for automated construction contract document analysis that can be used for taking advantage of in the bidding stage.

With this perspective, targeted deliverables are;

- ML classification models that are developed with selected NLP methods and ML algorithms in order to be used in construction contract reviews,
- Domain-specific datasets to be used as input in the implementation of proposed models,
- Classification models' performance values that are obtained after evaluating models on developed datasets.

Thus, this study tried to find answers to the following questions;

1. How can a well-organized training dataset and test dataset be developed to train and test Machine Learning Models for classifying contract sentences according to risk, right and obligation with related parties.
2. How successfully can contract sentences be classified automatically according to predefined labels to improve the contract review process which is currently done manually?
3. How can machine learning models' performances be improved by implementing alternative and complementary approaches in classification model development processes?
4. How can machine learning models that are developed by using different natural language processing methods and machine learning algorithms be combined for improving classification performance?

1.3 Scope of the Study

The study is based on a literature review on applications of natural language processing and machine learning in text analysis. The scope of the study is defined as a construction contract review due to problems presented in the research background section. Literature review findings show that different approaches are available to analyze text in semantic and syntactic aspects. The rule-based analysis is based on defining rules to analyze texts. The rule-based approach uses experts' knowledge in an if-then-else form. In this approach domain knowledge of experts needs to be coded into the system. These systems are developed for predefined problems, and if the system encounters a problem that is not fit any rule, then the system cannot understand the problem. On the other hand, learning-based systems can provide more flexible solutions to problems. In artificial intelligence, learning-based systems are used to develop rules from inputs related to problems, and the system is able to improve and change these rules according to new inputs.

In this study, works related to both systems have been reviewed in the text analysis domain, and a learning-based method is proposed. The scope of the learning-based

contract review system for construction contracts has been limited to FIDIC contracts due to the required time needed to develop datasets. As a result, all efforts in this research are to develop a system that can categorize the sentence in FIDIC contracts in terms of sentence type and identifies the related parties for sentences that implies obligation, risk, and right. Additionally, investigating the performance of the proposed system by implementing the model in actual construction project contracts is included in the scope of the research.

1.4 Research Methodology

The research steps that are followed are visualized in Figure 1.1. A literature review on risk and construction contracts has been conducted, and it is found that there is a need for improvement in contract review prosses at the bidding stage to decrease the needed time and effort to analyze the documents. When the problem is defined clearly, it is considered that automated text analysis based on artificial intelligence has the potential to provide a solution to the problem. A literature review on text analysis shows that different approaches can be used to solve these types of problems. Rule-based and learning-based alternatives are compared based on advantages and disadvantages. When the availability of data, the complexity of the problem, and advances in the natural language processing-based machine learning models are considered, it is decided that the supervised machine learning approach is more promising compared to other approaches.

The automated construction contract document analysis model that is based on the selected method requires a dataset to develop models. In the literature, there are no available datasets that can be used directly to develop a supervised machine-learning classification model for the problem studied in this research. Analyzing FIDIC standard form of contacts is selected as the focus of this research. The selection of FIDIC contracts is based on the extensive use of FIDIC contracts on international construction works and the necessity to limit the required time to create datasets. Three FIDIC standard form of contracts which are Red Book, Yellow Book, and

Silver Book, were selected to create the Training Dataset. One actual construction project contract that has been prepared based on FIDIC Silver Book was selected to create Test Dataset. Textual data of FIDIC contracts have been preprocessed according to the findings of the literature review on NLP. Python libraries were used in the implementation of preprocessing methods on contracts.

After preprocessing the text data, duplicated sentences were removed from datasets, and sentences were labeled according to sentence types in the first round. Sentence types are defined according to the purpose of the sentence and implied risk, right, and obligation. Some of the text in contracts are added as a heading, and some of them provide definitions for terms used in contracts. These sentences are labeled as "Heading" and "Definition". Others were labeled as "Risk", "Right," and "Obligation" according to inherited meaning. Sentences that are labeled as "Risk", "Right," and "Obligation" have been re-investigated in order to define related parties. "Contractor", "Employer," and "Shared" labels have been appointed to these sentences by the researcher.

Completed Training Dataset and Test Dataset were discussed with Experts who are knowledgeable in the topic and work in a department of contracts. The validation study was conducted in 2 stages by dividing participants into a control group and a label group. After achieving satisfactory results from the validation study, Training Dataset and Test Dataset became finalized to train machine learning models and evaluate performances.

Before continuing to develop Machine Learning models, a literature review on performance evaluation has been made. It is found that there are various approaches to test the performance of models; however, it is decided that the most appropriate approaches for this research are calculating accuracy values and f1-scores.

After Training Dataset and Test Dataset are created and metrics to evaluate models are decided, machine learning models, which are used to be compared in this research for predicting sentence labels according to sentence type and a related party, are developed based on findings of literature review on Supervised Machine

Learning and Text Vectorization. A total of 6 Text Vectorization methods that are all based on Natural Language Processing and 5 Machine Learning algorithms which include commonly used algorithms and recently investigated deep learning-based algorithms, are combined, and 12 Machine Learning Models are decided to be used and evaluated in the scope of this research. 12 Models have been trained and tested by using the Training Dataset and Test Dataset developed in the previous steps of the research, and these models' performances are compared by using accuracy and f1 scores. These results are taken as step 1 results, and sentence type classification algorithms have been tried to be improved by implementing modification in the training process. This improvement step is named Step 2. In the last step, a literature review on ensemble methods showed that classification performance could be improved by combining individual models. The competitive voting method was implemented as an ensemble method, and the outcomes are named Step 3. Step 1, Step 2, and Step 3 results are compared at the end research by using accuracy and f1 scores.

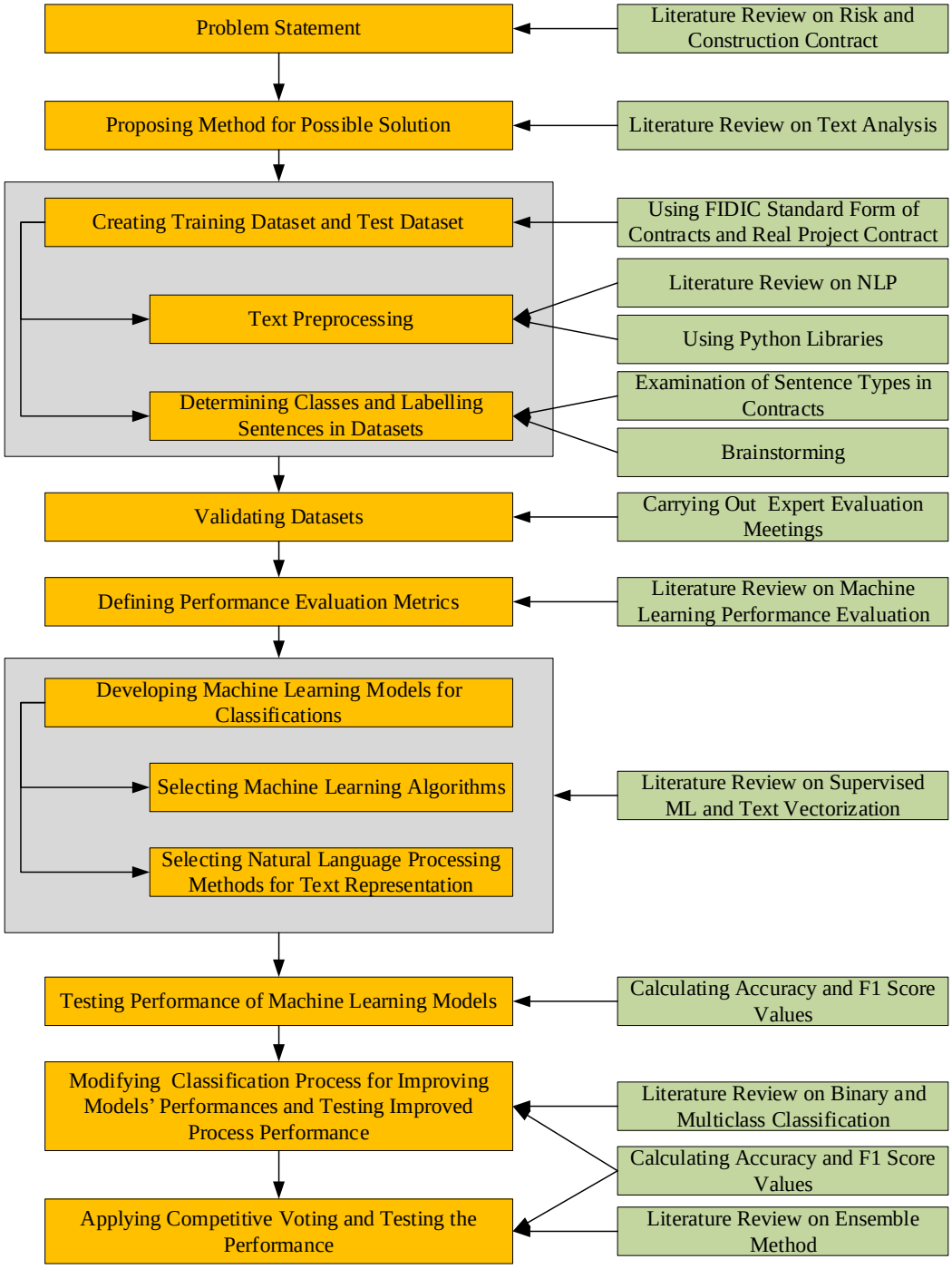

Figure 1.1. Steps Followed in This Study

1.5 Organization of the Thesis

This dissertation consists of 6 main chapters. The first chapter provides brief information about construction project risk, its relation to the construction contract, and the necessity to automate the contract review process. Research objectives, scope, and methodology are also presented in the first chapter.

The literature review chapter, which is Chapter 2, contains two main parts. A brief explanation of the terms Natural Language Processing and Machine Learning are given in the first section. Following the introduction of these terms, previous works that can be useful for automated contract review are presented in detail.

The third chapter presents efforts carried out for creating Training Dataset and Test Dataset.

Machine learning models, machine learning algorithms, and natural language processing tools that are used to develop machine learning models are introduced in Chapter 4. Implemented process model used in developing classification models is also presented in detail in Chapter 4.

Chapter 5 contains details of the results of all models developed in this research and applied improvements to increase these developed models' classification performance.

The dissertation is concluded in Chapter 6 by providing the contribution to the body of knowledge and limitations of the study. Additionally, future work recommendations that are considered to be beneficial to increase classification performance and usability of results in that topic are provided in Chapter 6.

CHAPTER 2

LITERATURE REVIEW

Business intelligence and analytics (BI&A) are getting attention from both academia and industry to solve up-to-date problems of organizations. BI&A is used in many areas like e-market intelligence, e-government, science & technology, smart healthcare, security, and safety. Different types of analytics are implemented for various applications. For example, e-government applications mainly use information integration, content and text analytics, government information semantic services and ontologies, social media monitoring and analysis, social network analysis, etc. Foundational technologies and emerging research in analytics can be grouped under five main headings as “Data Analytics”, “Text Analytics”, “Web Analytics”, “Network Analytics”, and “Mobile Analytics” (Chen et al., 2012). Text mining is a new era for predictive analytics and data mining (Kotu & Deshpande, 2015). Segel (2016) argues that “*If data is all Earth’s water, textual data is the part known as “the ocean.” Often said to compose 80 percent of all data, it’s everything we the human race know that we’ve bothered to write down.*” (Segel, 2016). This analogy explains why text analytics is an important research area. Text analytics with the synonym that is text mining are known names in commercial applications. The known name in the research field is “Natural Language Processing (NLP, aka computational linguistics)" (Segel, 2016).

In this chapter first NLP is introduced and then selected previous works related to NLP under requirement engineering, legal and construction industry domains are provided.

2.1 Natural Language Processing

Over the last decade, NLP has opened up research paths previously only imagined, in the process eliminating the need to painstakingly label words, phrases, and texts to calculate simple statistics such as word frequency or readability meticulously (Crossley et al., 2014). The main purpose of NLP is to understand and process natural language (NL), performing various tasks via computers. To achieve these goals, computer programs that understand and use NL like humans tried to be developed (Crossley, 2013). As a result, it can be stated that NLP is an AI subfield that concentrates on computer science and linguistics to develop methods to interact between computer and natural languages to achieve human-like language processing (Salama & El-Gohary, 2016). The famous examples of NLP applications are Apple's Siri, which is used in Apple products to get input from users in NL form, and Google Translate to automatically translate languages. Google's translation service is not a simple word-to-word replacement. Service uses NLP techniques to understand the grammar and context of the given article to translate it to the desired language with the correct syntax.

As seen in the studies presented next section, Machine Learning algorithm integration is not compulsory to process text with NLP; however, NLP can be used within ML classification models. The rule-based approach can also use to utilize NLP techniques. The rule-based approach uses predefined rules that accept NLP tool outputs to achieve desired results. On the other hand, the Machine Learning approach can also be used to extract rules from data sets. Three main types of Machine Learning approaches are available. The first is supervised Machine Learning, which takes the training data set as annotated. The supervised learning algorithm is based on building a mathematical model from the dataset that includes both the inputs and the desired outputs (*Machine Learning*, n.d.). This approach requires human involvement in the training phase. The second is the unsupervised approach, and algorithms learn without human involvement. Unsupervised learning algorithms need datasets that consist of only inputs. Algorithms are designed to determine the

relations in the data that can be used to group or cluster data points (*Machine Learning*, n.d.). The third is the reinforcement learning which is based on is training an agent in an undetermined environment for a specific task. As a result, different alternatives are available for different types of applications. In the next section, NLP and ML-related works are presented, and these concepts are tried to be further explained through the example works. Best of our knowledge reinforcement learning application is not available in literature related to contract review.

2.2 Previous Works with NLP

In this section, previous works that tried to solve domain-specific problems or improve the efficiency of processes by using NLP are presented. Previous works are presented under three domains, which are construction, legal, and requirement engineering. However, presented works under these domains are considered to be interrelated. For example, a study presented in the legal domain that focuses on element extraction from the contract can be a solution for the construction domain since the companies in the construction sector also need to deal with contracts to be successful in the business. Other than these three domains, NLP is used in various areas. For example, Fazlic et al. (2019) focus on providing medical recommendations to doctors by analyzing medical guidelines with the help of NLP. It can be considered an advanced search mechanism that gets a search topic as input from doctors in NL form and provides information from medical guidelines for diagnosis and treatment. Another work, which was conducted in discourse processing, provides a tool named Simple NLP. The tool is able to calculate linguistic features such as the number of word types, letters per word, the incidence of negations, and so on to predict essay scores for student works. (Crossley et al., 2014). Yet, as stated, the main focus is works conducted in construction, legal, and requirement engineering domains, which especially focuses on text analytics to improve the efficiency of the document analysis process in this study.

First, works that fall into the requirement engineering domain will be presented. After that, papers on works conducted under the legal domain will be given. Lastly, works that are focused on solving problems specific to the construction domain will be given.

2.2.1 Requirement Engineering Domain

Works that are presented in this section were conducted under the requirement engineering domain. According to Nuseibeh and Easterbrook (2000), the best definition of requirement engineering (RE) is done by Zave (1997), which defines RE as "*Requirements engineering is the branch of software engineering concerned with the real-world goals for functions of and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families*" (Zave, 1997). Goals in RE are related to customer needs. Thus, RE can be expressed as pinpointing and comprehending customer requirements. (Zait & Zarour, 2018). Software requirements are commonly expressed in the NL form. NL provides some advantages, such as ease of understanding and learning for stakeholders, because it requires little training (Arora et al., 2015; Robeer et al., 2016), and it can be used for any problem since it is universal. (Arora et al., 2015). So, NL exists in most of the requirements documents (Yang, Willis, et al., 2010). However, NL-based communication in RE suffers from drawbacks that can be listed as ambiguity and difficulty in making a holistic view (Robeer et al., 2016). Ambiguity and not restricting NL may make requirement documents hard to analyze automatically. In addition to free NL text, templates can also be used for information exchange in RE (Arora et al., 2015). Templates can be expressed as a restricted form of NL. That means that if templates do not be used properly, RE using a template may also suffer from drawbacks related to NL.

Three problems in RE are tried to be solved by using NLP in the presented works. The first and most important problem is ambiguity in RE. Some of the works

presented here tried to identify ambiguity problems in documents with the help of NLP (Huertas & Juárez-Ramírez, 2012; Rosadini et al., 2017; Yang, Willis, et al., 2010; Zait & Zarour, 2018). Another research is trying to identify key entities and relationships from user stories automatically with the help of NLP (Robeer et al., 2016). The last presented work that is conducted under RE is related to automated checking of document conformance that is prepared by template (Arora et al., 2015).

2.2.1.1 Ambiguity Related Studies

Ambiguity arises when a term can be understood differently by readers (Yang, Willis, et al., 2010). Ambiguity is deemed harmful for RE. Ambiguity in the requirement texts can lead to a weakness in analyzing customers' needs. This may also result in project failure (Zait & Zarour, 2018).

The study conducted by Zait and Zarour (2018) classifies ambiguity into six categories: lexical ambiguity, syntactic ambiguity, semantic ambiguity, pragmatic ambiguity, language errors, and vagueness. They are trying to identify lexical and semantic ambiguity. Lexical ambiguity is caused due to words with multiple meanings. This can be homonymy or polysemy ambiguity. On the other hand, semantic ambiguity can be thought of as scope ambiguity. As an example, the "All lights have a switch" sentence can be interpreted as all bulbs controlled by only one switch or each bulb controlled by exclusive switches (Zait & Zarour, 2018).

To detect and solve lexical and semantic ambiguity, Zait and Zarour (2018) employ three steps: preprocessing, ambiguity detection, and disambiguation. The text preprocessing step includes Part of Speech (POS) tagging and requirement normalizing. POS tagging is the process of assigning labels to each word to indicate the part of speech, tense, singularity, etc. (Lexical Computing, n.d.). Stanford parser is used in the research conducted by Zait and Zarour (2018). Stanford parser is a software package that analyzes the structure of sentences grammatically. (Klein et

al., n.d.). After POS tagging, requirements are normalized by dividing sentences according to the predefined algorithms for sentences connected by connectives.

Normalized requirements are taken as input, and each word is checked to find ambiguous words by BabelNet. BabelNet is a lexical database that stores several different word meanings to determine context knowledge of requirements (Zait & Zarour, 2018). If an interpretation of a word is more than one meaning having the same POS, the word is marked as lexical ambiguity. Words are also compared with the blacklist that contains word indicators such as all, any, few, little, many, much, several, and some. If a word in a sentence is matched with words in the blacklist, the sentence is labeled as semantic ambiguity.

Work conducted by Yang et al. (2010) focuses on the automatic detection of nocuous ambiguities in documents that are prepared by natural language. Nocuous ambiguity is defined as uncertainties that may lead to misunderstandings for different readers. The definition is made by Yang, de Roeck, et al. (2010) as nocuous and innocuous ambiguities. Innocuous ambiguity is not defined as a problem because all readers interpret it mostly the same. The research focuses on the automatic identification of nocuous coordination ambiguities (Yang, Willis, et al., 2010). Coordination ambiguity is a type of syntactic ambiguity that occurs when a given phrase has multiple grammatical structures, where each one provides different meanings (Zait & Zarour, 2018).

Similar to the work of Zait and Zarour (2018), Yang et al. (2010) implement preprocessing to prepare a document for analysis. In this step, various NLP tools, which are sentence splitting, POS tagging, shallow parsing, word occurrence, and word distribution, are used. Sentence splitting is used to define the start and end of any sentence in the text and label them with the sentence numbers before starting to analyze. Stanford parser is used for POS tag and shallow parsing (Yang, Willis, et al., 2010). The POS tag is already explained. However, shallow parsing is different from the POS tag. Parsing can provide information about word relations in sentences.

Word co-occurrence and distribution are statistical information that is computed based on the words' positions in sentences (Yang, Willis, et al., 2010).

The ambiguous instance detection step is finding sentences that contain coordination ambiguities according to predefined coordination construction patterns. POS tag of the sentence is used in this step. Yang et al. (2010) have prepared and presented the dataset, which describes 138 coordination instances. This data set is labeled with human judges. A key concept is training a classifier model for ambiguity pinpointing. An ML algorithm (LogitBoost) is used to build the classifier model (Yang, Willis, et al., 2010).

Another work on automatic requirement analysis was conducted by Huertas and Juárez-Ramírez (2012). Their work proposes a model called “Natural Language Automatic Requirement Evaluator” (NLARE). NLARE uses a rule set and regular expressions to solve NL-based RE problems. The architecture of NLARE is given in Figure 2.1.

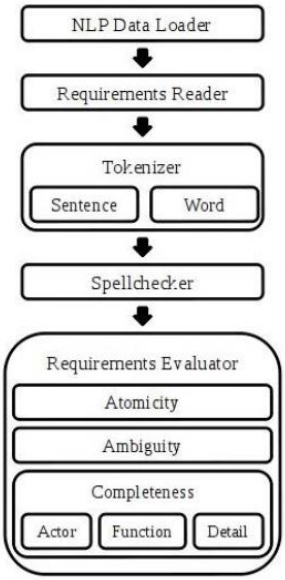

Figure 2.1 NLARE Tool Architecture (Huertas & Juárez-Ramírez, 2012)

NLP data loader is a part of a tool that stores tokenizers, chunkers, taggers, and corpora to evaluate text. The requirements reader is part of the tool takes analyzed

requirement documents. The tokenizer preprocessing step divides loaded documents into sentences and words. After that, each word is checked to be sure they are spelled correctly with PyEnchant library, a spelling check library for Python. If a word in a sentence has more than one meaning, the sentence is labeled with ambiguity. If a word is a conjunction between two sets of a word, the sentence is labeled with a lack of atomicity. Researchers propose that each requirement has to have an actor, function, and detail. If one of them is missing function is labeled as incomplete (Huertas & Juárez-Ramírez, 2012).

POS Tagger, named Maxent POS Tagger (Toutanova & Manning, 2000), is used in work to detect ambiguity and atomicity. Completeness is accomplished with RegeX, which is a grammar-based chunk parser, to define actor, detail, and function (Huertas & Juárez-Ramírez, 2012).

Another automated ambiguity detection in requirement engineering that is reviewed is conducted by Rosadini et al. (2017). This work's domain differs from the others because, in others, the main focus was software development. In a broader aspect, all of them try to determine ambiguity in requirement documents; however, this work focuses explicitly on the requirement documents of railway signaling manufacturers. GATE is used as an NLP tool in this study (Rosadini et al., 2017). General Architecture for Text Engineering (GATE) is a Java-based tool initially developed at the University of Sheffield. Scientists, companies, and academicians utilize it for NLP tasks, such as information extraction (*General Architecture for Text Engineering*, n.d.). Five features of GATE, which can be listed as Tokenization, POS Tagging, Shallow Parsing, Gazetteer, and JAPE Rule, are used by Rosadini et al. (2017) in their research to identify ambiguity in railway signaling. The first three features are explained in previous works. Gazetteer is used to search occurrences of words in a predefined list to determine the existence of ambiguity. By using JAPE, rules can be defined according to tokens and other features of the text. Rules are written for each ambiguity class to be able to determine them. In their research, ten ambiguity classes are determined. These can be listed as; anaphoric ambiguity, coordination ambiguity, vague terms, modal adverbs, passive voice, excessive

length, missing condition, a missing unit of measurement, missing reference, and undefined term (Rosadini et al., 2017).

Their work depends on hard-coded rules and not the ML approach to determine ambiguity. However, they used an extensive data set that humans annotated to test their approach. As a result, they state that the total performance of the proposed system is 85.6% (Rosadini et al., 2017).

2.2.1.2 Template Check Related Studies

The previous section focuses on conducted studies that directly determine ambiguity in a requirement document prepared using unrestricted NL. Differently, the work conducted by Arora et al. (2015) focuses on eliminating sources of ambiguity from requirement documents that are prepared by using templates. They use NLP to be sure each sentence complies with the template structure. Their approach accepts that requirement documents prepared according to templates are enough to eliminate ambiguity. Templates limit the syntactic structure of requirement documents by using several predefined spaces (Arora et al., 2015). Two templates, which are Rupp's template (Pohl & Rupp, 2011) and The EARS template (Mavin et al., 2009), are used in their research to analyze the conformance of requirements by using NLP.

The article attempts to provide a solution for automatically determining the conformance of text to templates. GATE, which is also used in work conducted by Rosadini et al. (2017), is used as an NLP platform to develop a tool that is named Requirement Template Analyzer (RETA). RETA is designed to help reviewers to check the conformance of text in terms of selected templates and define problematic syntactic structures in requirements statements (Arora et al., 2015). The major NLP technique employed in RETA is text chunking. Text chunking specifies sentence elements (chunks). Chunks, which are mostly noun phrases (NPs) and verb phrases (VPs), are able to provide abstraction on the natural language required level for distinguishing template spaces in order to perform conformance checks on templates.

2.2.1.3 Conceptual Model Creation Related Studies

The work conducted by Robeer et al. (2016) in requirement engineering focuses on creating models from text-based user stories. They are proposing 11 heuristic rules that can be used in extracting conceptual models from stories. The Visual Narrator tool that was developed based on the 11 heuristics rules enables the automated extraction of conceptual models from stories. It is developed in Python environment by using Spacy (Honnibal et al., 2020), which is an NLP toolkit similar to the one provided by Stanford. The algorithm proposed by Li et al. (2003) is used in performing verb extraction. An example model created by the Visual Narrator is given in Figure 2.2 (Robeer et al., 2016).

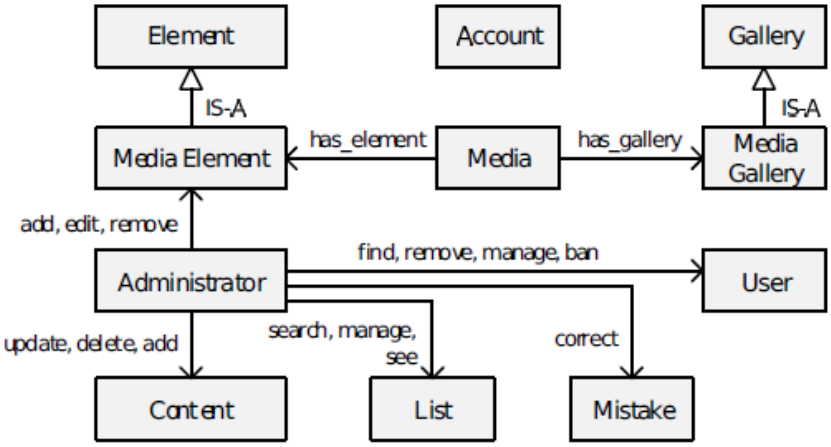

Figure 2.2 Example Conceptual Model created by Visual Narrator (Robeer et al., 2016)

2.2.1.4 Summary of Works Conducted Under Requirement Engineering

Works conducted under requirement engineering are given in previous sections. The ambiguity concept is explained in order to clarify its importance in NL-based documents. NL may suffer from ambiguity, and information exchange by using NL may become problematic. Different methodologies are provided for the

identification of different types of ambiguities. These methodologies employ different NLP techniques and tools, which are tried to be explained their usage of them in the related works. POS tagging, which is an important tool for NLP, is explained in this section. Shallow parsing, another important tool used by computers to understand NL, is also given in this section. These tools are used for understanding the semantics of the sentence. Other techniques, such as tokenizer, sentence splitter, and lexical database, are also mentioned. Two works use the ML approach to define ambiguity, and others use the rule-based approach. The ML approach used in the presented works is supervised machine learning. Unlike the unsupervised approach, the supervised approach needs human intervention to understand a dataset's logic. In general, datasets are divided into training data sets and testing data sets. The supervised approach needs to label datasets to train ML algorithms.

Another topic that tried to be achieved by NLP is automated template check. The logic behind using templates is to eliminate the risk of ambiguity. However, the conformance of prepared documents according to a selected template is another issue that can create ambiguity. The presented work is tried to achieve automated template checks by using NLP tools. The rule-based approach is used in this work to determine the conformance of the text to templates. The last work presented is trying to create automated conceptual models from user stories. The Spacy toolkit is used in this work. This work is also based on predefined rules.

2.2.2 Legal Domain

Works that are presented in this section were conducted under the legal domain. The legal text refers to several types of text written for various purposes related to the regulation, including law, judgments, contracts, request for proposals, disclaimers, etc. Presented studies in this section are mainly related to contract analysis with NLP because even if the contract does not refer directly to a construction contract, it is considered that they may provide information to investigate construction contracts as well. Presented works classified under three categories as; risk check (Chakrabarti

et al., 2018), text classification (Galser et al., 2018; Mok & Mok, 2019), contract element extraction (Chalkidis et al., 2017; Chalkidis & Androutsopoulos, 2017). First, work related to risk check is presented; after that, contract element extraction-related works are explained. Text classification-related works will follow up these two categories before concluding this section.

2.2.2.1 Risk Checking Related Study

A contract determines the scope of work, required activities, and responsibilities. Reviewing contracts and exploring the risks is crucial for any company/corporation/organization. Reviewing process and risk analysis can be done through a basic search using keywords; however, this is not sufficient for understanding the context of the different items (Chakrabarti et al., 2018).

Chakrabarti et al. (2018) propose a framework named "risk-o-meter" to solve limitations in identifying risk-prone paragraphs in contractual risk assessment. Risk-o-meter is developed by using machine learning algorithms and natural language processing techniques. Their approach also aims to associate risk-prone paragraphs with predefined risk categories like liability, indemnity, and confidentially.

The process flow diagram of risk-o-meter taken from Chakrabarti et al. (2018) is given in Figure 2.3. Chakrabarti et al. (2018) offer a continuous learning approach. Phrases taken from contractual/legal documents are labeled according to predefined risk categories to develop a training dataset. Vector representation of phrases, which is named paragraph vector by Chakrabarti et al. (2018), are used as text representations to be able to determine syntactic and semantic relations between terms in the text and the topic. The vectorization process is applied to each input in the training dataset. ML model, which is trained by using the training dataset, is used to calculate paragraph vectors for new inputs. An unsupervised machine learning model is used for creating paragraph vectors that can be defined as context-based word embedding. The paragraph vector method is similar to the word vector model.

However, words are vectorized exclusively in the word vector method. Word embedding is a technique used in NLP applications. It is a method for language modeling and feature engineering technique where words are represented as n-dimensional vectors. Word embedding identifies similarities between words based on their occurrence in large training data. Various methods are developed to train and use word embeddings, such as Word2vec from Google, GloVe from Stanford University, and fastText from Facebook Artificial Intelligence Research (*Word Embedding*, n.d.). On the other hand, in the given research in this section, all words in phrases are represented in one vector in the paragraph vector method (Chakrabarti et al., 2018).

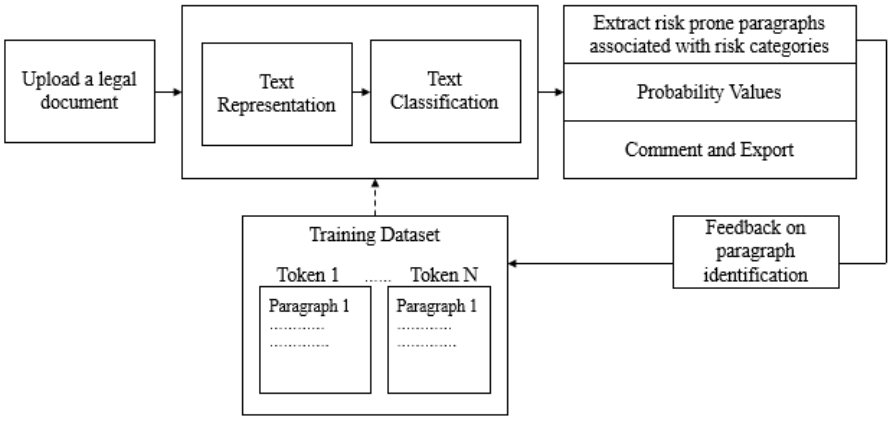

Figure 2.3 Process Flow Diagram for Risk-o-meter (Chakrabarti et al., 2018)

In the text classification step, supervised ML algorithms are used to train ML models. New clauses that the model could not identify can be manually integrated into the training data and rerun the training step to improve performance (Chakrabarti et al., 2018). As a result, it can be stated that NLP techniques and machine learning algorithms are used in developing models to classify the text based on NL in their research.

2.2.2.2 Contract Element Extraction Related Studies

Contract element extraction is another research focus accomplished with NLP under the legal domain. Two interrelated works (Chalkidis et al., 2017; Chalkidis & Androutsopoulos, 2017) are given to explain possible usage.

Contracts are legal texts describing the conditions of agreements between parties. Contracts need to be monitored in areas like payment times and validity for successful completion. Manually tracking these features, which can be called elements of contracts, is costly and time-consuming. Automatic extractions like validity dates, payment dates, and parties are considered beneficial in the process (Chalkidis et al., 2017; Chalkidis & Androutsopoulos, 2017).

The study conducted by Chalkidis et al. (2017) focuses on automated contract element extraction. 3500 contracts were manually labeled according to 11 types of contract elements, which are contract title, start date, contracting parties, clause headings, effective date, legislation references, contract period, termination date, contract value, governing law, and jurisdiction (Chalkidis et al., 2017).

In addition to the labeled dataset, which contains 3500 contracts, it is stated that 750,000 unlabeled contracts are used for developing word embedding in their research. ML models are developed using Support Vector Machine and Logistic Regression algorithms, and word embedding is used for vectorizing inputs. After that, trained models are used to classify and determine contract elements (Chalkidis et al., 2017).

The study conducted by Chalkidis & Androutsopoulos, (2017) uses the same data set and word embedding to investigate the performance of BiLSTM, which is a deep-learning approach. When results in both studies are compared, BILSTM with Conditional Random Field provides the highest performance with a 0.88 f1 score value (Chalkidis et al., 2017; Chalkidis & Androutsopoulos, 2017).

2.2.2.3 Classification of Legal Text Related Studies

Two works, which focus on the automatic classification of legal text, are presented in this section. The first study was conducted by Mok and Mok (2019). Their study focuses on classifying sentences taken from contract court decisions. They propose using an ontology to classify court decisions related to contract breaches. They argue that NLP and ML can be used in ontology engineering to extract facts, rules, and reasoning from court decisions (Mok & Mok, 2019). Contract fact, contract law, contract holding, contract issue, and contract reasoning are defined as critical sentences that can be extracted from court decisions. Spacy (Honnibal et al., 2020) is selected as a natural language library for parsing the decisions. Logistic Regression supervised ML algorithm is implemented by using the scikit-learn library (Pedregosa et al., 2011) for text classification in their research. Three court decisions, which contain 529 sentences, are used to create a dataset. 370 sentences are used in the training set by labeling them manually. 159 sentences are used in the test set. They provided results for various modifications that were made in their models and compared results. As a result, they obtained nearly 55% correct guesses for 159 sentences included in the test dataset (Mok & Mok, 2019).

Another study conducted by Galser et al. (2018) states that the analysis of contracts is a significant study topic because it can provide practical knowledge like rights and obligations. They classify sentences in legal rental contracts, which are written in the German Language. They propose a taxonomy containing nine classes: duty, indemnity, permission, prohibition, objection, continuation consequence, definition, and reference. They are using a machine-learning approach for classification. Extra Trees Classifier and Support Vector Machine Algorithms are employed in their research. Word embedding, Bag of Words, and TFIDF are selected NLP methods to vectorize texts to make them understandable by the computer. Two legal experts manually classified 913 sentences, which were taken from the German Civil Code and rental contracts, according to the given taxonomy to create training and test datasets. The models are trained using the dataset, which contains sentences from the

German Civil code. The models are tested by using the dataset that includes 312 sentences taken from rental contracts. The best result was achieved with 0.72 f1 score by the model that was developed by using Extra Trees Classifier as algorithm and Bag of Words as vectorizer.

2.2.2.4 Summary of Works Conducted Under Legal Domain

Works presented under the legal domain are related to the automated review of contracts. Studies are not focusing primarily on construction contracts. However, they focus on risk checks of a contract, contract element extraction, and clause classification, such as risk-prone paragraph identification in contracts by using NLP and ML. These can be implemented in construction contracts to solve problems faced in the sector. The supervised machine learning technique is used for classifying text. Different ML algorithms, such as Logistic regression, Vector Machines, and Naïve Bayes, are proposed to extract information and classify text. The literature review on the legal domain shows that contract element extraction can also be automated by using NLP and ML. As a result, this section provides information about supervised ML and its possible usage in text mining. This section also presents the idea that NLP can be used to automate the analysis of contracts. Unlike the previous section, the works presented in this section use the ML approach rather than the rule-based approach.

2.2.3 Construction Industry Domain

NLP-related studies that are conducted under the construction domain are given under four headings in this section. First, compliance checking and keyword extraction-related works are presented. After that, responsibility extraction works will be given, and text classification work is following it. Under the third and fourth headings, text classification and the contract risk checking related studies are presented, respectively. All works focus on using NLP in construction documents

(especially contracts) in order to automate document analysis. In the end, the presented studies are summarized.

2.2.3.1 Compliance Checking and Keyword Extraction Related Works

Zhang and El-Gohary (2016) propose an NLP-based information extraction system for automated compliance checking (ACC) to be used in construction regulatory documents. ACC is considered important because manual checking compliance with regulatory documents is time-consuming, costly, and error-prone. Using ACC have the potential to reduce the required time and cost, as well as eliminate errors faced in manual processes (Eastman et al., 2009; Tan et al., 2010; J. Zhang & El-Gohary, 2016).

Zhang and El-Gohary (2016) utilize NLP techniques to automate document analysis and a rule-based approach to extract requirements from documents. Pattern matching is used to identify phrases that are wanted to be extracted based on predefined rules. The semantic and syntactic features of the texts are analyzed to recognize patterns in a text. Tokenization, sentence splitting, morphological analysis, POS tagging, and phrase structure analysis are the employed NLP techniques to identify syntactic features of texts. An ontology is used to capture semantic features based on domain knowledge (J. Zhang & El-Gohary, 2016). Soysal et al. (2010) also underline semantic feature identification based on ontology in information extraction (IE) has the potential to increase performance due to consideration of domain-specific terms and meanings.

The proposed methodology for IE by J. Zhang & El-Gohary (2016) comprises seven phases: Information representation, preprocessing, feature generation, target information analysis, development of information extraction rules, extraction execution, and evaluation. Preprocessing phase has three steps: Tokenization, sentence splitting, and morphological analysis. ANNIE, which is distributed by the Java-based NLP tool GATE, is used for tokenization and sentence-splitting steps of

preprocessing. Additionally, GATE's built-in morphological analyzer is also used in preprocessing phase. Set of features that describe the text are generated by using NLP, especially by the part of speech tagging-based method in the feature generation step. ANNIE is implemented to create part of speech tags. The built-in ontology editor of the GATE is used for ontology-based semantic analysis to represent construction domain-specific meanings of words. The development of IE Rules is a manual rule development step to be used by computers in the IE process. JAPE rule tool, which is a GATE feature, is used in coding the rules (J. Zhang & El-Gohary, 2016). This rule-based system's performance was evaluated using 144 inputs grouped by experts, and a 0.91 performance value in terms of f1 score was achieved in the evaluation step. 0.91 F1 score is a promising result; however, it is needed to be stated that the system depends on a rules-based approach with manual identification of rules according to document context, and different documents may give different results.

Keyword determination from construction documents for information acquisition of international construction is done by Moon et al. (2018). Six websites that contain 25143 documents are used as data sources. Web crawling is the proposed method to collect data. Keywords are extracted in 3 steps from documents. The first two steps are performed by implementing the NLP techniques, which are part of speech tagging and term frequency calculation. According to results gathered in the term frequency calculation step, manual filtering is done. The result of the filter is used for keyword extraction from documents. After that, the word cloud method is used to visualize search results. These documents are collected under country names. Users can select a country from the map, and related documents are presented with the country’s word cloud.

2.2.3.2 Responsibility Extraction Related Works

Al Qady & Kandil (2010, 2009) present a tool for automatic semantic information extraction from contract documents using an NLP tool called "*Concept Relation*

Identification using Shallow Parsing (CRISP)" (Al Qady & Kandil, 2010, 2009). According to them, document management processes employed by construction companies in contract management can be improved with the proposed tool.

Responsibilities in the contract clause, which is taken from The American Institute of Architects (AIA) contract, "*The Owner shall furnish surveys describing physical characteristics, legal limitations and utility locations for the site of the Project, and a legal description of the site. The Contractor shall be entitled to rely on the accuracy of information furnished by the Owner but shall exercise proper precautions relating to the safe performance of the Work."* can be extracted as given in Table 2.1 by using the proposed model (Al Qady & Kandil, 2010, 2009).

Table 2.1 CRISP Output of Sample Clause Taken From AIA (Al Qady & Kandil, 2009)

	Active Concept	Relation	Passive Concept
1	the owner	shall furnish	surveys
2	surveys	describing	physical characteristics
3	the contractor	shall be entitled to rely	on the accuracy of information
4	the accuracy of information	furnished	by the Owner
5	the owner	shall exercise	proper precautions

CRISP contains four main steps. The input file preparation step is a manual process to improve the system's performance. Section numbers in the document must be labeled with brackets to get section numbers with extracted concepts. After each section number, the end term is added to identify the boundaries of each section. Numbers in the list are also removed manually. Shallow parsing is used in the second step to identify nouns, verbs, and prepositional parts of sentences. Sudance, developed by the School of Computing at the University of Utah (Riloff & Phillips, 2004), is used as the shallow parser in their research. The rules-based approach is proposed to extract responsibilities based on the output of the shallow parsing step (Al Qady & Kandil, 2010).

Six provisions from AIA Document A201-1997 were chosen for the evaluation set. The gold standard created during expert meetings has been used for evaluating CRISP performance. The achieved f1 score of the gold standard is 0.76. On the other hand, the f1 score of CRISP is 0.42 (Al Qady & Kandil, 2009).

2.2.3.3 Text Classification Related Works

Text classification (TC) is a subfield of information retrieval, which is the automated process of retrieving documents or information within documents from a large collection of data. The first study focuses on developing a semantic text classification method using NLP and ML. In the approach proposed by Salama and El-Gohary (2016), TC is utilized for classifying parts of the text, like contract clauses, into predefined classes to facilitate information extraction. They are proposing 14 labels to classify construction text. These can be listed as energy management, security, safety and health, environmental, labor relations, change management, claims and disputes, scope, risk management, general, quality, contracting, time, and cost (Salama & El-Gohary, 2016).

In their application, they have been focused on classifying text as environmental and non-environmental. Three different Machine Learning algorithms, which are Naïve Bayes, Support Vector Machine and Maximum Entropy are used and performances are compared. The text classification methodology proposed by (Salama & El-Gohary, 2016) is given in Figure 2.4. 330 clauses, which are taken from standard form of contracts, books and real construction project contracts, are used to create training dataset and test dataset.

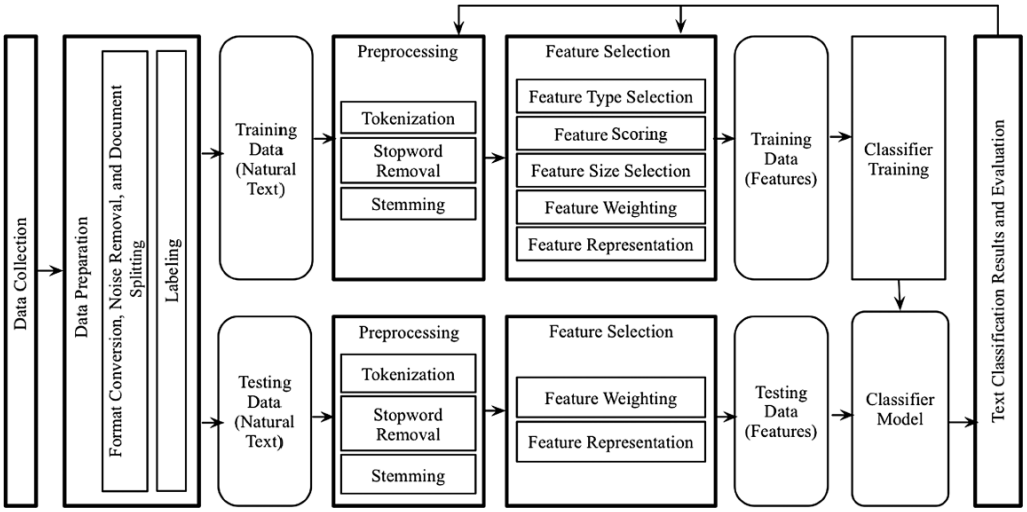

Figure 2.4 Text Classification Methodology Proposed by Salama & El-Gohary (2016)

Data preparation is a manual step in which documents are converted to TEXT files (.txt). Additionally, each clause is manually labeled as environmental or non-environmental in this step. Tokenization, stop-word removal, and stemming are NLP techniques that were implemented in preprocessing step. Java-coded algorithms are used in the feature selection step. (Salama & El-Gohary, 2016) Bag of Words and Term Frequency Inverse Document Frequency (TF-IDF) are used methods in the vectorization of the text. The best model that gives the 0.97 f1 score is developed by using SVM and Bag of Words.

Candaş & Tokdemir (2022) argues that the construction contract review process needs the participation of various departments of a company. So they are proposing the classification of contract clauses according to related departments. They employ the supervised machine learning approach to automatically define related departments that clauses are needed to be sent for their review. They used FIDIC Pink Book to create a dataset. Each clause in FIDIC Pink Book is labeled with their related departments as contractual, technical, planning, operational, and financial. A multilabel classification approach is implemented, and some clauses are labeled with more than one category. After preprocessing the text with some NLP techniques, such as stopword removal and stemming, the sentences are vectorized using the term

frequency-inverse document frequency. Gaussian naïve bayes, multinomial naïve bayes; support vector machine; decision tree classifier, multilayer perceptron and logistic regression are the implemented machine learning algorithms in their research. The best algorithm in terms of accuracy is the support vector machine with a 0.649 accuracy level. On the other hand, in terms of the f1-score, the best model is the decision tree classifier with a 0.75 f1 score.

2.2.3.4 Contract Risk Check Related Study

The work conducted by Lee et al. (2019) focuses on identifying poisonous clauses in construction contracts by using NLP. They define poisonous clauses as modified sentences that disadvantage the contractors when compared to the original text in FIDIC Red Book and Pink Book. It can be stated that their focus is finding modified clauses from FIDIC. Unmodified clauses in FIDIC are accepted as risk-free sentences. They are proposing rule-based semantic IE for identifying poisonous clauses. The process of risk identification is given in Figure 2.5. Sentence elements are extracted with SyntacNet, which was developed by Google, and is used for performing parsing.

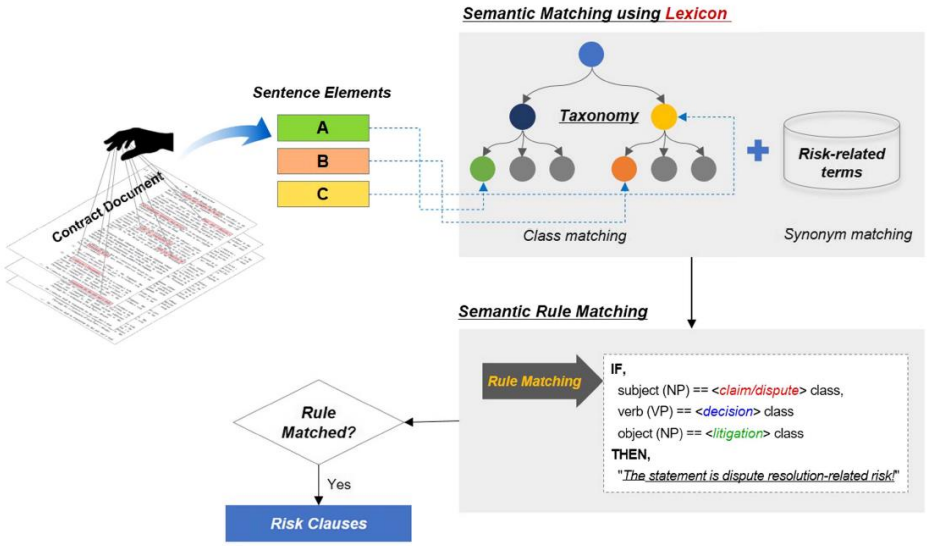

Figure 2.5 Risk Clause Identification Process Proposed by Lee et al. (2019)

The next phase is matching sentence elements with construction contract taxonomy. Together with taxonomy, a construction-contract lexicon was defined that contains 352 items. For example, the "right" term under Right and Responsibility is considered a synonym for duty, authority, entitlement, and so forth. After that, semantic rules, which are defined by researchers, are used to determine whether a sentence is risk related or not. Semantic rules are in the form of If, Then conditions. A gold standard has been created by domain experts, who are experienced in contract and claim management, to evaluate the performance of the proposed model. The comparison of proposed model outputs with the gold standard gives a 0.81 success rate in terms of the f1 score (Lee et al., 2019).

Lee et al. (2020), in their work, try to identify missing contract clauses that actually favor contractors. The study is based on the rule-based approach that is developed by considering the FIDIC standard form of contracts. They are making a definition for clause type as contractor friendly. According to subject-verb-object tuples extracted by natural language processing, sentences are analyzed and matched with their proposed rules. Performance of the proposed system compared with expert evaluation study and result calculated as 0.80 in terms of f1-score.

2.2.3.5 Ambiguity Identification Related Study

The work conducted by Candaş & Tokdemir (2022a) focuses on determining ambiguous clauses in a construction contract. They used FIDIC Silver Book to create a dataset. Researchers labeled each sentence in the contract as vague or not vague. The term frequency-inverse document frequency vectorization method is used for vectorizing text data after pre-processing step. They used both the supervised ML approach and the rule-based approach separately.

In the ML approach, naïve bayes algorithm, support vector machine classifier, decision tree algorithm, multi-layer perception algorithm, K-nearest neighbor, and random forest classifier are used, and the performances of each alternative are

compared. According to the results best algorithm is the random forest with a 0.80 accuracy level.

In the rule-based approach, they define seven words in base forms to be used in matching texts in the dataset. Selected terms are reason, appropriate, other, similar, suitable, necessary, and may. According to researchers, if these terms exist in the sentence, the sentence needs to be classified as vague. When their search-based rule is tested in their dataset, the accuracy of vague term identification is found as 0.89.

The rule-based approach has given more promising results when compared to the ML-based approach; however, this highly depends on the dataset structure and assumptions made in the research.

2.2.3.6 Summary of Works Conducted Under Construction Domain

Presented works in this section focus on solving problems or easing processes experienced in the construction domain with the help of NLP. Compliance checking is based on a rule-based approach and uses GATE and its built-in NLP toolkit ANNIE. Various NLP tools, such as part of speech tags, sentence splitting, and tokenization, are used to process construction documents. Automatically extracting responsible parties can decrease the time required to analyze construction contracts. Works presented under text classification and ambiguity identification topics use machine learning approaches. A supervised machine learning approach with different algorithms is implemented to classify contract clauses under environmental or non-environmental Various NLP tools are used to pre-process and vectorize the documents before classifying them through machine learning algorithms. Multilabel classification according to a related party is also presented in the text classification section. Risky clause identification is another topic presented in this section. The proposed methodology is rule-based, and it suggests using a taxonomy to identify risky clauses. Their approach does not contain risk identification. Their work tries to identify modified provisions when compared to two FIDIC standard forms of

contracts. In conclusion, they accept unmodified FIDIC clauses as not risky, which may not always be true. The proposed system uses the parser provided by Google. They also implemented a lexicon that was developed for construction to synonym match. Lastly, ambiguous clauses are tried to be determined with the ML and the rule-based approach in the same study.

As a result, the studies conducted in the construction domain use both rule-based and ML-based approaches to provide solutions for presented problems. Various NLP tools, such as POS tagging and parsing, are used as NLP tools. Studies which are presented here show that NLP opens the way to automated contract analysis for the construction sector.

2.3 Literature Review Conclusion

Natural language processing-based studies in text analytics are a subfield of artificial intelligence, and studies are conducted to analyze documents to facilitate processes. Two different primary approaches exist in the literature: machine learning and rule-based.

The rule-based approach has some drawbacks in terms of generalizability. ML-based approach suffers from the availability of labeled datasets to train ML models. While the rule-based approach for specific domains provides high performance, implementing proposed models for different models requires extra effort in terms of developing new rules. ML-based studies give lower performances when compared to rule-based but provide generalizable solutions.

In this research, ML-based approach is selected to create a system that classifies construction contract text to facilitate the contract review process. Other than the presented studies in literature, in this dissertation, recent advancements such as pre-trained word embeddings and deep learning algorithms for improving the classification performance of machine learning models trained based on a relatively small dataset are tried to be implemented. Additionally, the literature review shows

that the ensemble approach to combine different machine learning models to improve classification performance is not studied in the contract review domain. As a result, the study presented in this dissertation differs from previous studies in terms of generalizability and used methods in models based on current advancements in ML and NLP to obtain high performance in a relatively wide scope which is classification of construction contract text for using contractual risk identification in bidding stage.

CHAPTER 3

TEXT PREPARATION AND DATASET CREATION

This chapter presents the details of the processes employed to achieve the goal of classifying FIDIC-based construction contracts in terms of responsibility, risk, right, and related parties. Since the focus of this research is defined as classifying construction contracts that are drafted based on FIDIC contracts, three types of FIDIC contracts are used, which can be listed as FIDIC Red Book, FIDIC Yellow Book, and FIDIC Silver Book. In addition to unmodified FIDIC contracts, an actual construction project contract that is revised from FIDIC Silver Book for the project-specific requirements is used in this research. Text data of selected four contracts (namely three original FIDIC contracts and one actual construction project contract based on the FIDIC Silver Book) needed to be prepared to be classified with machine learning algorithms. Preparation processes include NLP techniques as well as conversions. The expected outcome at the end of this step of the research is creating an excel spreadsheet that contains sentences from selected contracts. With the help of NLP techniques, sentences with the same meaning but containing different words are matched in all texts, and only unique sentences are added to datasets.

After creating an excel spreadsheet that contains sentences in each row, sentences are labeled under two columns which are type of sentence and related party. The type of sentence has four inputs “Definition”, “Obligation”, “Risk”, and “Right”. Related party columns contain three inputs “Employer”, “Contractor”, and “Shared”. Each sentence in the dataset contains a label set given in Table 3.1. As seen in the table, if a sentence is labeled as “Heading” or “Definition”, a related party is not defined because these parts of the text or sentences are accepted as not containing any responsibility, risk, or right. Example sentences for the label sets and logic to be applied in defining labels are given in the following sections in detail, together with the method used to verify the integrity of all labels in datasets.

Table 3.1 Label Set Used to Create Datasets

Label Set	*Sentence Type*	*Related Party*
Label Set 1	Definition	-
Label Set 2	Heading	-
Label Set 3	Obligation	Employer
Label Set 4	Obligation	Contractor
Label Set 5	Obligation	Shared
Label Set 6	Risk	Employer
Label Set 7	Risk	Contractor
Label Set 8	Risk	Shared
Label Set 9	Right	Employer
Label Set 10	Right	Contractor
Label Set 11	Right	Shared

Conversion details of PDF files to txt files and text preprocess rules and NLP techniques that are implemented are given in detail in the following sections. Training and Test Data sets used to train various machine learning algorithms in the research are given in detail at the end of this chapter.

3.1 Text Preparation and NLP Processes

As described, three FIDIC standard forms of contracts and one actual construction contract, which is prepared based on FIDIC, are used in this research to create datasets. Since the research objective is to create a process that can be used to automatically identify terms in construction contracts in terms of risk, right, and obligation with ownership, an automated process for converting contract documents files to excel files is necessary. The manual creation of datasets from contract files was evaluated as unsuitable, so a process was defined, and a tool was created by using Python programming language according to this process. With the help of this

tool, both datasets, which are used to train machine learning models, and reviewed contracts, can be automatically processed with the same logic and assumptions. The process model is given in Figure 3.1. Each step in the process model is explained in detail following sections.

3.1.1 Converting Contract Documents from PDF files to TEXT files

At the first step [A0 in Figure 3.1] of the conversion process, contracts should be converted into a TEXT file from PDF, which is the most common file type. PDF files can be in two forms machine-readable, which can also be named searchable, or image form. If a PDF file is created by a scanner or from an image, computers cannot understand the text, and this type of PDF file is not included in the scope of this research since it is related to image processing. As a result, a contract needs to be in machine-readable format to convert into an excel file. There are solutions to convert non-machine-readable PDF files to machine-readable files, such as Optical Character Recognition; however, the accuracy of these solutions is highly dependent on the quality of the images. Therefore, the focus of this study has been limited to machine-readable PDF files.

Python library called pdfminer (Shinyama, 2019) is used in developed code to convert PDF files to TEXT files. The code output is a txt file with some extra characters, which need to be removed from the TEXT file before proceeding to other steps. The details of implemented hard-coded rules for removing extra characters from the output of conversation are given next section. If an analyzed contract is in a Word document (.doc or .docx format), it can be directly saved as a TEXT file and proceed to the next step by skipping the first step.

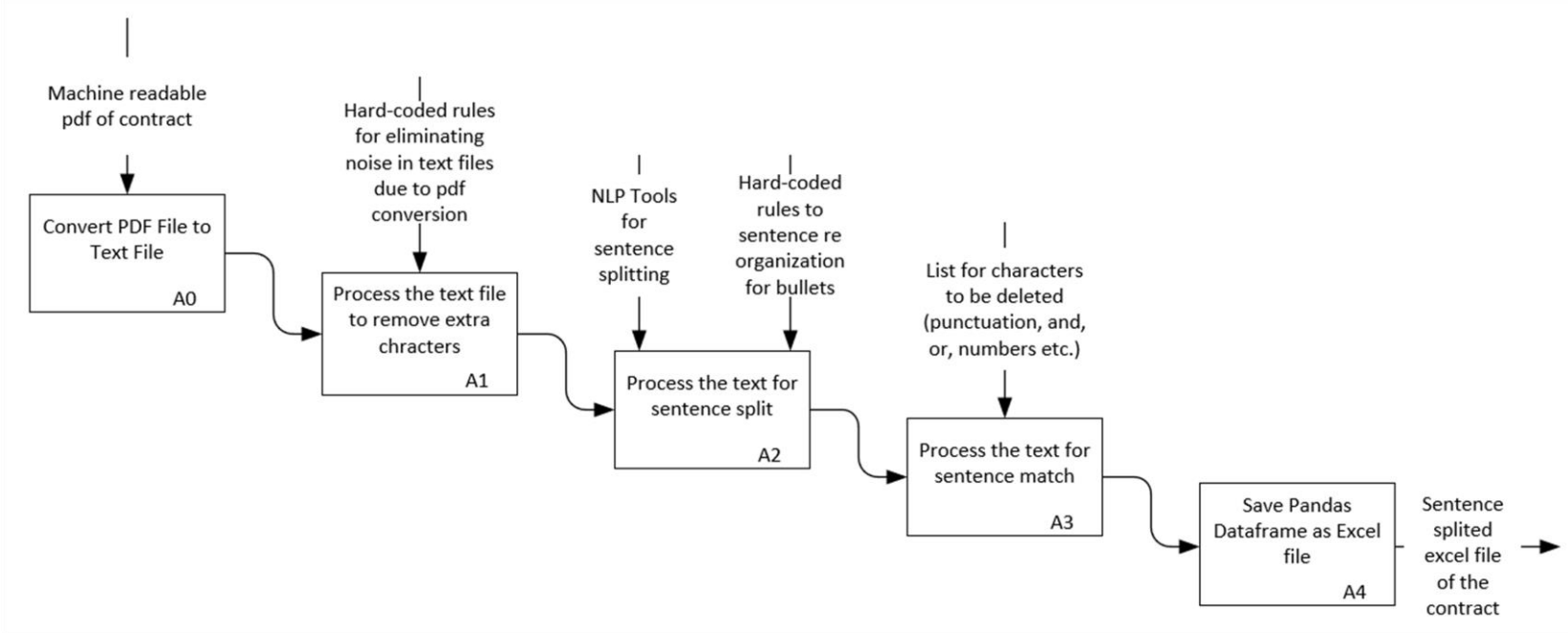

Figure 3.1. The Process Model for Converting Contract Documents to Excel File.

3.1.2 Removing Extra Characters from TEXT file.

Produced TEXT files need some arrangements to implement NLP techniques to extract sentences. The conversion process given in A0 step in Figure 3.1 creates new paragraphs for the headings after heading numbers. In addition, the sub-articles in the sentences also start new paragraphs. Page numbers are also considered characters that need to be removed because there are no contributions to classifying sentences. Lastly, watermarks in the PDF file are transferred to the TEXT file and need to be removed from the text. When the text was analyzed, it was determined that there exists an order in these errors. Rules to fix three issues in the TEXT file, which is the output of the A0, are given in Table 3.2. Implementation of rules is done by Python codes, and texts were cleaned from the extra characters that resulted from the conversion process and do not belong to the main text of contracts. The order of the implementation of rules is the same as ascending order of problem numbers which are given in Table 3.2.

After cleaning extra characters from the text with the provided four hard-coded rules, all paragraph characters are removed from the text if it is followed by another paragraph character. Created TEXT file after this removal is the output of the A1 step given in Figure 3.1.

As a result, the output of A1 is a TEXT file that includes only the main text body of contracts. However, most of the paragraph of text contains more than one sentence, and some sentences need to be split. A2 processes for splitting text into sentences and rearranging some sentences for labeling are given next section.

Table 3.2 Problems in Output of PDF Conversion and Rules to Eliminate Them

First Problem and Defined Rule to Solve	
Problem 1	Extra paragraph between heading text and heading number
Patterns in the text	There are always three successive new paragraphs between the heading number and heading text.
Rule to eliminate Problem 1	If there are successive three-paragraph characters, which are represented with \n, and the previous paragraph only contains numbers and dot characters, successive three-paragraph are replaced with space.
Second Problem and Defined Rule to Solve	
Problem 2	Sentences are split if they contain sub-numbers
Patterns in the text	There are always a maximum of 3 characters between parenthesis, and the previous paragraph ends with a colon or semicolon.
Rule to eliminate Problem 2	Paragraph characters in the pattern are replaced with space.
Third Problem and Defined Rule to Solve	
Problem 3	Page numbers of contracts create noise in the document
Patterns in the text	There are always two successive new paragraphs before and after a page number.
Rule to eliminate Problem 3	If two successive paragraph characters are followed by a paragraph that contains only numbers and numbers followed by two successive paragraph characters, all characters with page numbers are replaced with paragraph characters.
Fourth Problem and Defined Rule to Solve	
Problem 4	Characters of watermarks are transferred to a TEXT file as one character in each paragraph
Patterns in the text	A paragraph character follows each character in a watermark, and only one character exists in these paragraphs.
Rule to eliminate Problem 4	If there is only one character in a paragraph, the character is removed from the text.

3.1.3 Sentence Extraction from TEXT file

Up to the A2 step, the text of the contracts is converted from PDF file to txt file and cleaned from noises originating from the conversion process. Since the aim is to extract sentences to classify according to the labels defined, the text of reviewed contracts and train and test datasets needs to be split into sentences. This step is shown in Figure 3.1 as A2. The A2 step contains two stages. In the first stage, NLP techniques are used to sentence splitting, and Spacy Python library (Honnibal et al., 2020) was selected for this purpose. After sentence extraction with NLP, complicated sentences are rearranged, as proposed by Kim et al. (2020). Details of these two stages are given in the following sections.

3.1.3.1 Sentence splitting with NLP

Spacy (Honnibal et al., 2020) library has the ability to split text into sentences according to the selected configuration. The NLP pipeline was created by adding "sentencizer". The pipeline is configured to split texts into sentences by considering dots in the text. The output of this NLP pipeline is used as input in the next stage to reorganize complicated sentences.

3.1.3.2 Reorganization of complicated sentences.

Sentences in the FIDIC contracts can be long and complex. Some of the sentences imply various obligations, risks, and rights with numbered bullets. Kim et al. (2020) propose applying syntactic rules to transform complex sentences into simple sentences.

The syntactic rule in our research was developed based on bullet numbers and connector terms, and they can be seen in Table 3.3. Punctuations are also important indicators for finding complicated sentences; therefore, colons and semicolons were used to develop syntactic rules to simplify sentences. The number of sentences has

increased after implementing syntactic rules between 300 to 400 in each FIDIC contract used in this research.

Table 3.3 Bullets and Connectors Used to Develop Syntactic Rules

Category	**Characters to search**
First type bullets	(a), (b), (c), (d), (e), (f), (g), (h), (i), (j), (k), (l), (m), (n), (o), (p), (q), (r), (s), (t), (u), (v), (w), (x), (y), (z)
Second type bullets	(i), (ii), (iii), (iv), (v), (vi)
Connectors and punctuations	; or ; and ; and/or ; or/and ; :

The developed rule is implemented to text data according to the flow given in Figure 3.2. Complicated sentences, for which an example is given in Table 3.4, are converted into simple sentences as given in the same table. In that step, the plain text is also converted to Pandas (The Pandas Development Team, 2020) Data Frame by using Spacy (Honnibal et al., 2020) Sentencizer.

Table 3.4 Example of a Complex Sentence and Output of the Syntactic Rule

Original Clause	*"After receiving ********** agree or determine:* ***(a)** the additional payment ******** Contract Price; and/or* ***(b)** the extension (if any) of the ******* the Employer as the claiming Party),* *to which the claiming Party is entitled under the Contract."* (*FIDIC® Conditions of Contract for Construction for Building and Engineering Works Designed by the Employer*, 2017)
Returned Sentence 1	After receiving ********** agree or determine: **(a)** the additional payment ********** Contract Price; and/or to which the claiming Party is entitled under the Contract.
Returned Sentence 2	After receiving ********** agree or determine: **(b)** the extension (if any) of the ****** Employer as the claiming Party), to which the claiming Party is entitled under the Contract.

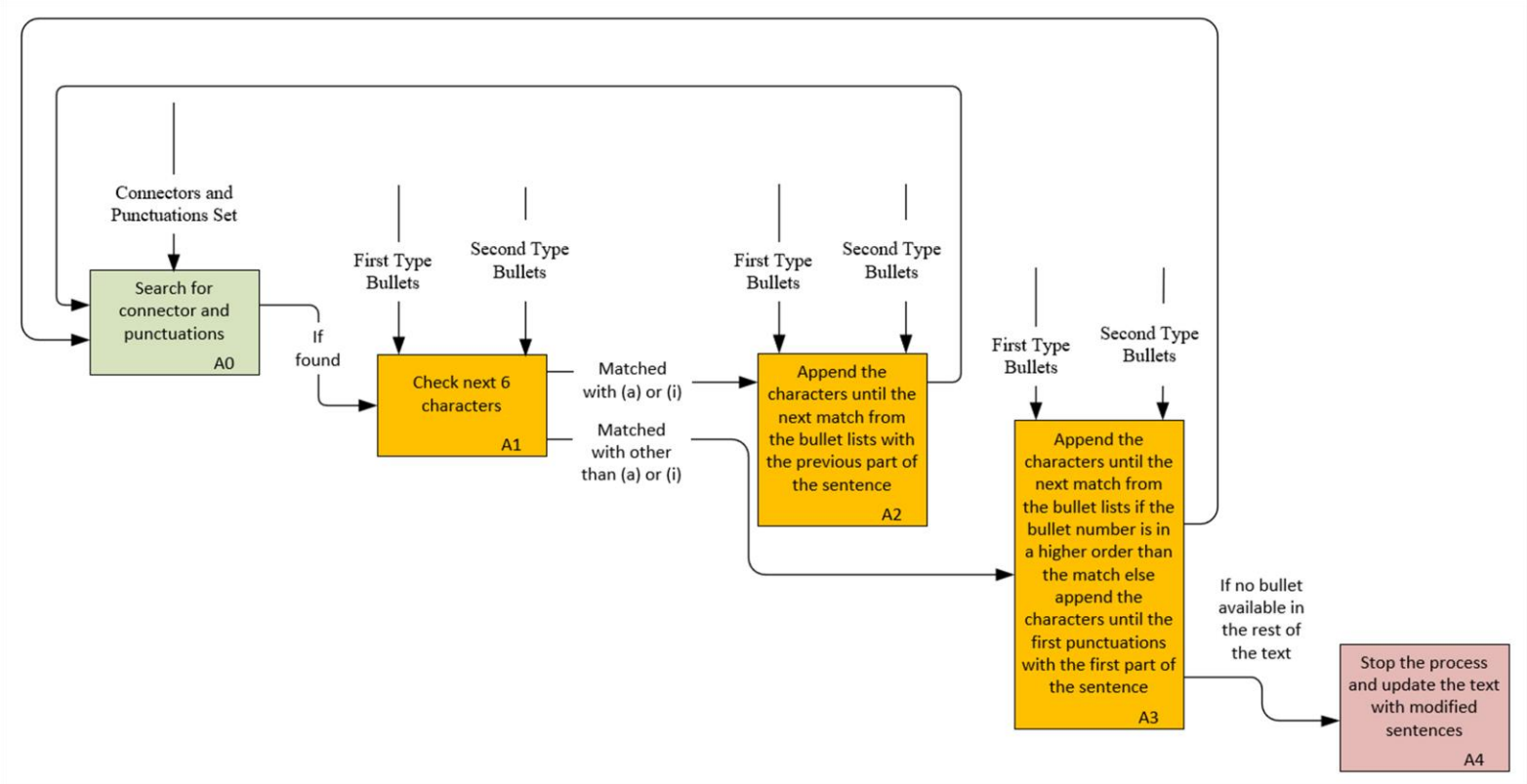

Figure 3.2. Syntactic Rules to Simplify Sentences

3.1.3.3 The Output of Sentence Extraction from TEXT file

As an output of step A2, which is given in Figure 3.1, four Pandas (The Pandas Development Team, 2020) Data Frames were created in Python environment, and each Data Frame has the number of sentences given in Table 3.5. Sentences in the Data Frames were preprocessed to define unique sentences for the purpose of creating the Train Data Set and the Test Data Set. Details are given in the next section.

Table 3.5 Output of Sentence Extraction from TEXT files

Contract	***Number of Sentences***
FIDIC Red Book	1791
FIDIC Silver Book	1726
FIDIC Yellow Book	1829
Actual Construction Project Contract	1305

3.1.4 Extracting Unique Sentences

At the end of step A2 in Figure 3.1, PDF files are converted into Excel files that contain sentences in each row. Unmodified FIDIC Books combined in one file and planned to be used as Train Dataset, and the actual construction project contract is planned to be used as Test Dataset, as shown in Figure 3.3.

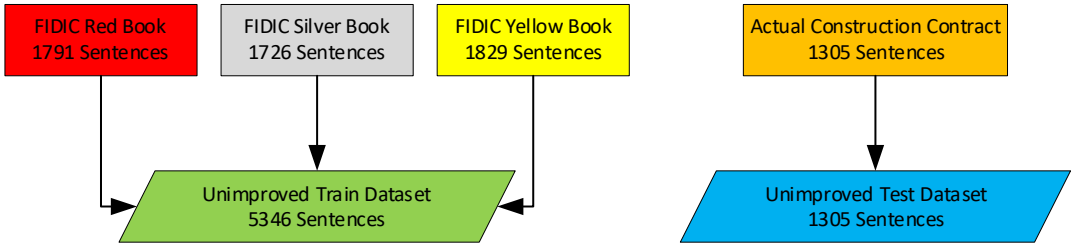

Figure 3.3. Content of Train and Test Datasets

However, when sentences were analyzed, it was noticed that the same sentences were repeated in the datasets. This repetition is much higher in the Training Dataset because FIDIC Books mostly have same provisions and definitions. As a result, deleting repeated sentences from datasets is considered logical before tagging them. This deletion also eliminates relabeling the same provisions with different labels due to human errors. Python code is planned to automate this process; however, exact matching is not enough to delete all repeated sentences because punctuations, some stop words and special characters such as parentheses or box brackets are not always the same in the texts, although the meaning of the sentences same. It is decided that before using an exact match algorithm to eliminate repeated sentences, some rules need to be implemented to preprocess the sentences in the A3 step, as shown in Figure 3.1.

So, the implementation of the rule set, which is given in Table 3.6, is decided. Briefly, some punctuations, special characters, some words, bullet indicators, numbers, and connectors were removed from sentences, and texts were converted to lowercase.

Table 3.6 Rule Set for Preprocessing to Sentences for Matching

Rule to implement	**Character Set**								
Removing punctuations	,	;	:	.	“	‘			
Removing special characters	(	)	[	]					
Removing bullet indicators	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
	(j)	(k)	(l)	(m)	(n)	(o)	(p)	(r)	(s)
	(t)	(u)	(v)	(w)	(x)	(y)	(z)	(ii)	(iii)
Removing numbers	All numbers and dots between them								
Removing connectors	and	or	and/or						
Removing some words	sub-clause	sub-paragraph							
Eliminating uppercase and lowercase mismatch	All characters (converted to lowercase)								

Implementation order of rules to create the comparison column is also important to get successful results. For example, if special characters are removed before bullet indicators, bullet indicators cannot be removed from text successfully. An order was defined for implementing the rules to create the comparison sentence column, as shown in Figure 3.4.

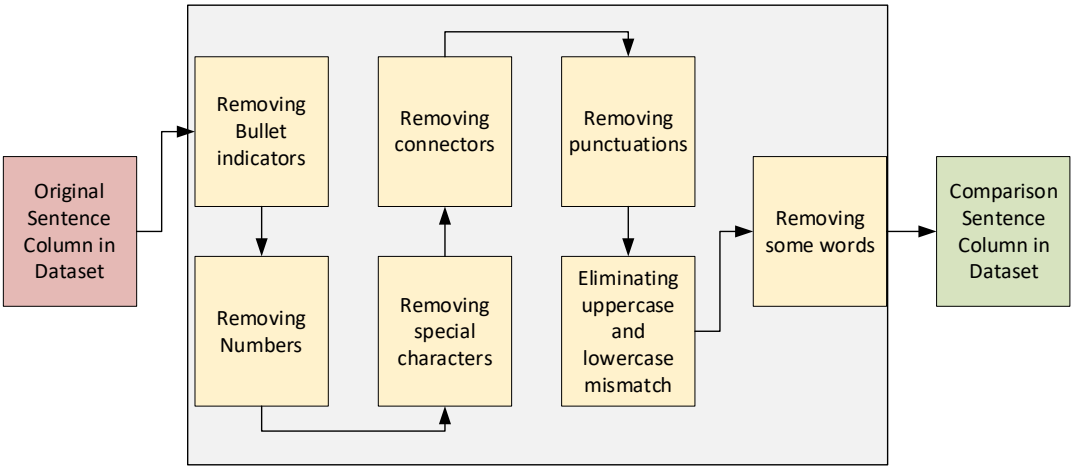

Figure 3.4. Implementation Order of Rules in Sentence Comparison

3.1.5 Final Training Dataset and Test Dataset Content

When the comparison sentence column was created, sentences matched within the datasets were removed. The final sentence list that has been used as Train Dataset and Test Dataset is finalized, and the output of this process is an excel file. A tool was created with a simple user interface in Python to simplify the process explained in this section. With this tool, users can analyze a machine-readable PDF file of FIDIC-based contracts and get an excel file ready to be processed.

The sentence number in datasets, which are created at the end of the A4 step of the process model given in Figure 3.1, is given in Table 3.7. As can be seen in the table, the number of sentences in the Training Dataset decreased from 5346 to 2268 after deleting repetitions. Similarly, the number of sentences in the Test Dataset decreased to 1217.

Table 3.7 Final Datasets and Change in Sentence Number

Dataset	***Number of sentences together with repetitions***	***Number of unique sentences***
Train Dataset	5346	2268
Test Dataset	1305	1217

Constructed excel files, whose part is shown in Figure 3.5, contain unique sentences, and sentences are needed to be labeled since these two files have been planned to be used as datasets in ML models. Details of the dataset labeling study conducted in this research are given in the following sections.

	A	B	C	D	E	F	I
1	index	Inname	Sentences	Cleaned Sentences	Comparison	rac	ontained_contr
52	350	371	If the Contract specifies th	If the Contract specifies that t	if the contract specifies that the	Red	Red
53	351	372	4.2 Performance Security	Performance Security	performance security	Red	RedSilverYellow
54	352	373	The Contractor shall obtair	The Contractor shall obtain (a	the contractor shall obtain at the	Red	RedSilverYellow

Figure 3.5. Screenshot of Excel File that Contains Unique Sentences

3.2 Dataset Labelling for Supervised Machine Learning

The previous section presents details of processes that are implemented to extract articles sentence by sentence from contracts. The output of this step is Excel files that contain article sentences in each row. Since this research aims to create machine learning classification models to analyze construction contracts and categorize articles, sentences in the Excel file is needed to be labeled with a label set illustrated in Figure 3.6. Sentences are labeled in terms of sentence type and related party.

Details of the logic of labeling and validation of suitability are given following sections.

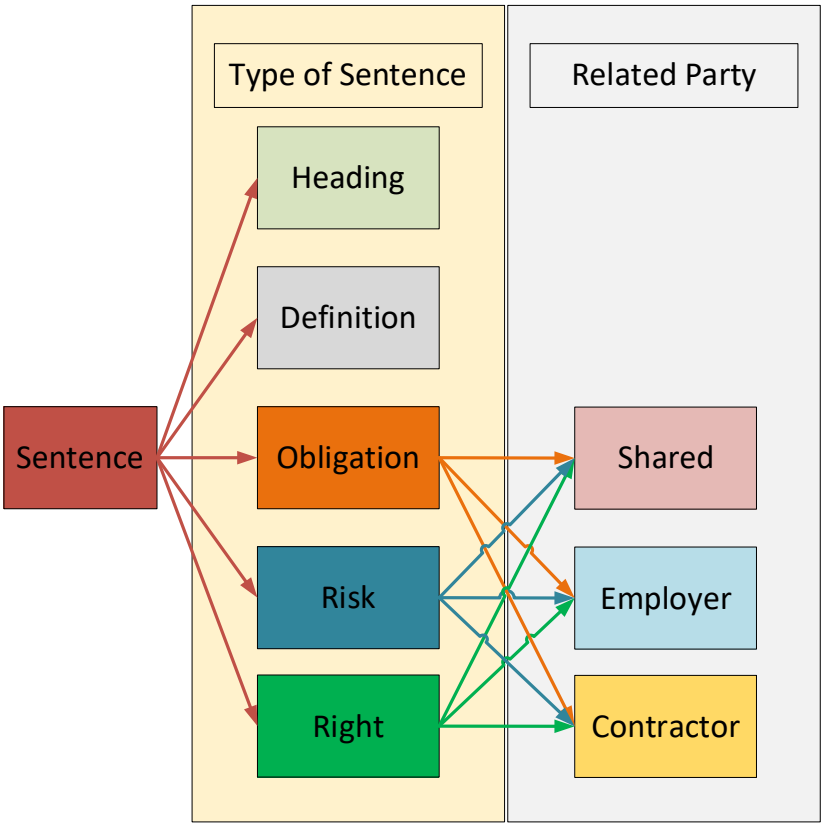

Figure 3.6. Dataset Labels Used to Create Supervised Machine Learning Models

3.2.1 Type of Sentence

Contracts have different types of sentences in terms of purpose. Some parts of the text are used to define the heading of the sections, and some of them are used to precisely define contract terms. On the other hand, contracts mostly consist of articles to define parties' obligations and rights in addition to assigning risk. With this perspective, it is decided to label sentences in the training and test datasets in terms of sentence type in five categories, as can be seen in Figure 3.6. 2268 sentences in Training Dataset and 1217 sentences in Test Dataset are analyzed to label them as “Heading”, “Definition”, “Obligation”, “Risk” and “Right”.

The categorical distribution of labeling in terms of sentence type of the datasets was given in Table 3.8.

Table 3.8 The Categorical Distribution of Dataset in terms of Sentence Type

Sentence type category	Number of sentences	
	Training Dataset	**Test Dataset**
Heading	228	205
Definition	178	91
Obligation	1033	565
Risk	488	242
Right	341	114
Total number of sentences	**2268**	**1217**

The labeling procedure is very time-consuming, so labels were assigned by the author of this dissertation. However, as explained in section 3.2.4, assigned labels were validated throughout the study with the participation of experts.

Before the validation study, sentences that were labeled as “Heading” and “Definition” were excluded, and the number of label ratios of “Obligation”, “Risk” and “Right” were calculated. It is found that ratios are very similar in Training Dataset and Test Dataset, as shown in Figure 3.7, and it is concluded that the labeling process is consistent.

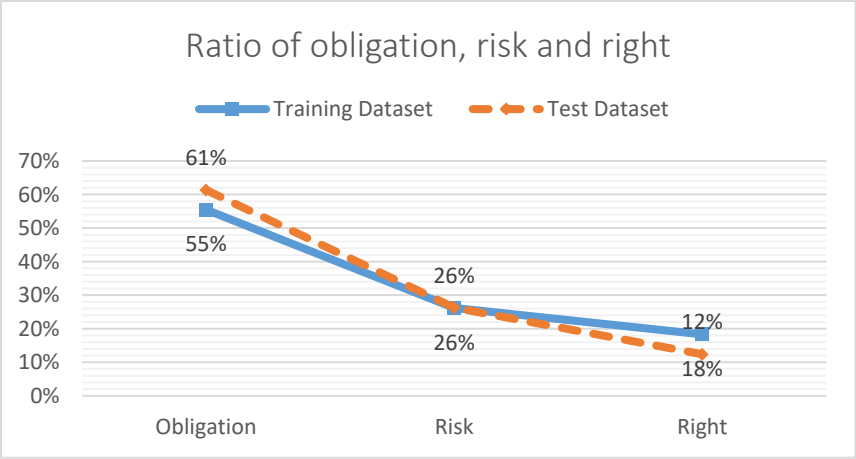

Figure 3.7. Comparison of Training Dataset and Test Dataset Label Ratios in terms of Obligation, Risk, and Right

3.2.2 Related Party

Since the “Headings” and “Definitions” do not imply any responsibility to any party, after the sentences were labeled in terms of types, only sentences that were labeled as “Obligation”, “Risk” and “Right” were labeled in the Related Party classification.

In this section, sentences were analyzed to define related parties according to implied risk, right, or obligation. Three categories were defined as “Shared”, “Contractor” or “Employer”. Labeling a sentence as “Employer” means that the sentence implies a risk for the employer, the right of the employer or states an obligation for the employer. The same is valid for contractors also. On the other hand, the “Shared” label means that sentence implies risk, right, or obligation to both parties. As a result, it needs to be stated that sentence type labels and related party labels are interrelated.

1862 sentences in Training Dataset and 921 sentences in Test Dataset were analyzed and labeled in terms of the related party category, and the total numbers are given in Table 3.9.

Table 3.9 The Categorical Distribution of Dataset in terms of Related Party

Related party category	**Number of sentences**	
	Training Dataset	**Test Dataset**
Shared	269	118
Contractor	1044	617
Employer	549	186
Total number of sentences	**1862**	**921**

Similar to the study conducted in sentence type category, before the validation, the number of label ratios of “Shared”, “Contractor” and “Employer” were calculated. It was found that ratios were very similar in Training Dataset and Test Dataset, as shown in Figure 3.8. It was concluded that the labeling process is consistent.

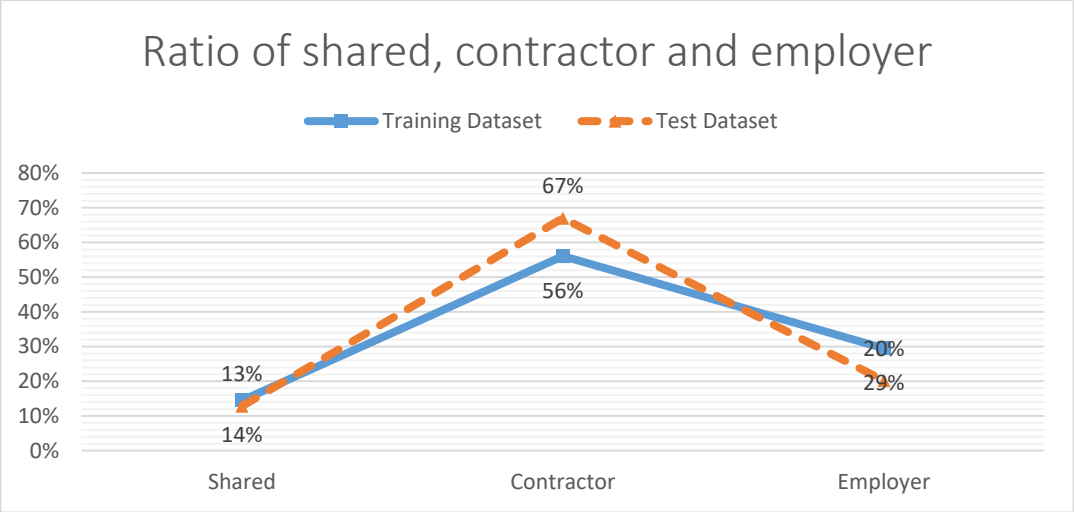

Figure 3.8. Comparison of Training Dataset and Test Dataset Label Ratios in terms of Shared, Contractor and Employer

3.2.3 The Output of the Labelling Process

The result of labeling sentences in datasets is given in Table 3.10. As a result, sentences are grouped under eleven categories.

Sentence distribution labeled with “Obligation – Shared”, “Obligation – Contractor”, “Obligation – Employer”, “Right – Shared”, “Right – Contractor”, “Right – Employer”, “Risk – Shared”, “Risk – Contractor”, and “Risk – Employer” analyzed to check the consistency. It was assumed that since the test dataset was developed from an actual construction project contract that is prepared based on the FIDIC contract, similar ratios need to be found when the Training and the Test Datasets were compared.

The total number of sentences related to Risk, Right and Obligation were used to calculate ratios for the categories mentioned. The result of this comparison is given in Figure 3.9. As can be seen in the figure, there are some minor differences. Datasets were analyzed to find the reasons for the difference. 2 reasons were found for this difference. The first reason is that the training dataset is the combination of three

FIDIC standard forms of contracts, and their allocations related to risk, right and obligation also have minor differences. The second reason is modifications in the actual project contract. Obviously, contracts for construction projects are mostly prepared by employers, which increases the obligations to contractors, as seen in the comparison. The same situation is valid for the contract used to create Test Dataset.

Table 3.10 The Complete Categorical Distribution of Datasets

No	Combined categories		Number of sentences	
			Training Dataset	Test Dataset
1	Heading	-	228	205
2	Definition	-	178	91
	Total of Heading and Definition		**406**	**269**
3	Obligation	Shared	142	81
4	Obligation	Contractor	624	401
5	Obligation	Employer	267	83
	Total of Obligation		**1033**	**565**
6	Right	Shared	55	14
7	Right	Contractor	135	39
8	Right	Employer	151	61
	Total of Right		**341**	**114**
9	Risk	Shared	72	23
10	Risk	Contractor	285	177
11	Risk	Employer	131	42
	Total of Risk		**488**	**242**
	Total of Risk, Right and Obligation		**1862**	**921**
	Total number of sentences		**2268**	**1217**

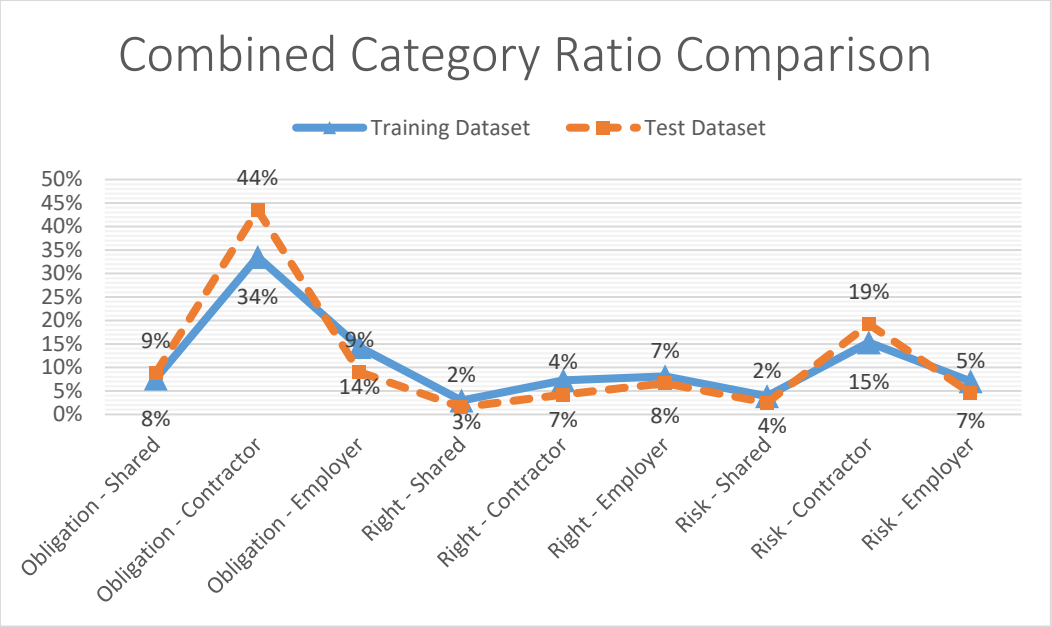

Figure 3.9. Comparison of Training Dataset and Test Dataset Label Ratios

Obligations of Employer in the training dataset were found that they were transferred to contractors in Test Dataset. Similarly, shared risks and employer risks were also transferred to contractors in Test Dataset. As a result, it was concluded that the labeling process is consistent between the datasets. However, it is also acknowledged that this comparison does not validate the logic employed to label sentences. The validation study was conducted, as explained in the next section, to be sure that employed logic is valid for professionals who are experts in contracts.

3.2.4 Validation of Dataset Labels

As explained in the previous sections in the Training and Test Datasets, 3485 sentences were labeled. 2783 of the 3485 sentences which were labeled as Risk, Right, and Obligation are subjected to validation study. Headings and Definitions were excluded from the validation study since these sentences can be identified without any doubt.

Validation of the labels has been done through expert review meetings. Six people attended these meetings. 10% of the sentences from each label set were selected randomly as validation subsets because of the required time to label sentences. Participant profiles and methodology are given in the following sections.

3.2.4.1 Validation Study Participant Profile

All six participants were working in departments of contract. As presented in Table 3.11, one of them has a Ph.D. degree, and three of them have an M.Sc. degree. Half of them have more than 10 years of work experience.

Table 3.11 Validation Study Participant Profile

Participant	*Education*	*Experience*	*Position*
Participant 1	M.Sc.	16-20	Chief Contract Manager
Participant 2	M.Sc.	10-15	Chief Contracting Officer
Participant 3	Ph.D.	10-15	Senior Contract Specialist
Participant 4	B.Sc.	5-10	Senior Contract Specialist
Participant 5	B.Sc.	0-5	Contract Specialist
Participant 6	M.Sc.	0-5	Assistant Contract Specialist

3.2.4.2 Methodology to Validate Dataset Labels

The validation study is conducted according to the workflow presented in Figure 3.10. In this section methodology that employed validate labels are presented in detail. The results of the validation study are presented in the next section.

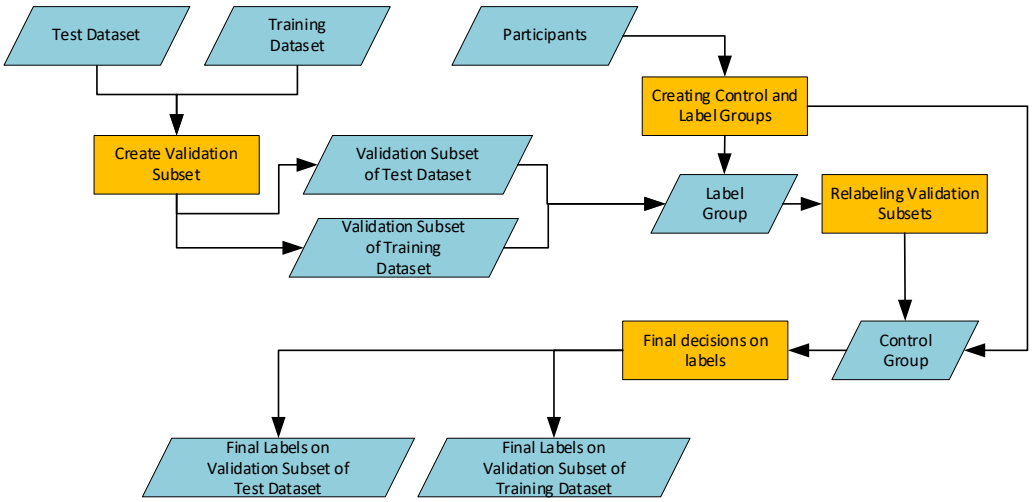

Figure 3.10. Validation Study Workflow

As mentioned before, subsets were created from datasets in order to be used in the validation study. Subsets were created by randomly selecting 10% of the sentences from each label category. The validation subset contains 280 sentences in total, 185 sentences taken from Training Dataset and 95 sentences taken from Test Dataset. The distribution of sentences according to labels is given in Table 3.12.

Six participants were divided into two groups, as label group and control group. Participant 1, Participant 2, and Participant 3 are in the control group due to their experiences. Participant 4, Participant 5 and Participant 6 were selected as Label Group.

Expert meetings were held separately for the two groups by providing different levels of detail about the datasets.

Validation subsets were presented to the label group by removing labels, and the group was asked to label sentences according to predefined categories. All three participants in the label group participated in the same meeting, and labels were given in consensus.

The complete validation subset with the results of the Label Group was presented to the control group in a different meeting. In this meeting, the control group evaluated the labels given by the label group and researcher together. They could also change

the label completely or select one of the given attained labels. Similar to the previous meeting, Participants in the control group attended the same meeting, and decisions were given with consensus.

As a result, 280 sentences in the datasets were labeled three times by the researcher, label group, and control group. The results of this study are given in the next section.

Table 3.12 Number of Sentences in Validation Subsets

No	Labels		Number of sentences		Number of sentences	
			Training	Test	Subset of Training	Subset of Test
3	Obligation	Shared	142	81	14	9
4	Obligation	Contractor	624	401	62	40
5	Obligation	Employer	267	83	26	9
6	Right	Shared	55	14	6	2
7	Right	Contractor	135	39	13	4
8	Right	Employer	151	61	15	6
9	Risk	Shared	72	23	8	3
10	Risk	Contractor	285	177	28	17
11	Risk	Employer	131	42	13	5
		Total	**1862**	**921**	**185**	**95**

3.2.4.3 Results of the Validation Study

Expert meetings' results are presented in Table 3.13. When results are compared with labels determined by the researcher, it is found that 8 sentences out of 280 sentences are labeled differently by the control group.

Table 3.13 Results of Expert Meetings

Label		**Training Dataset Subset**			**Test Dataset Subset**		
		Researcher	Group Label	Group Control	Researcher	Group Label	Group Control
Obligation	Shared	**14**	12	**13**	**9**	11	**10**
Obligation	Contractor	**62**	59	**61**	**40**	39	**39**
Obligation	Employer	26	27	26	9	8	9
Right	Shared	**6**	6	**7**	2	3	2
Right	Contractor	**13**	16	**14**	4	3	4
Right	Employer	**15**	12	**14**	6	5	6
Risk	Shared	8	9	8	3	2	3
Risk	Contractor	**28**	33	**29**	17	19	17
Risk	Employer	13	11	13	5	5	5

6 of the 8 differences are in the Training Dataset subset. Training Dataset subset contains 185 sentences. When the percentage of the difference is calculated, it is found that the difference is limited to 3%. In order to visualize the difference, Figure 3.11 is presented.

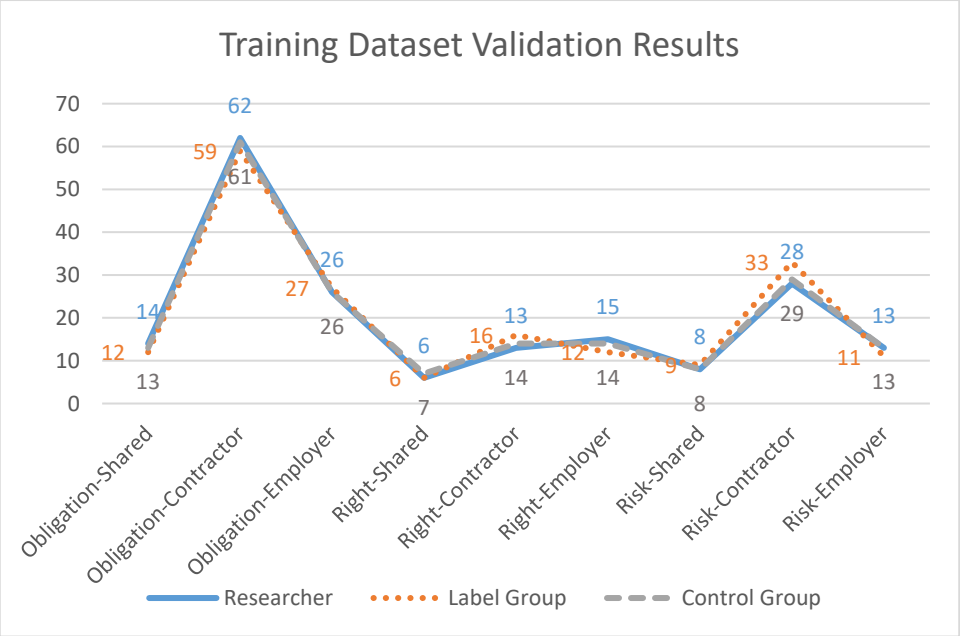

Figure 3.11. Validation Results of Training Dataset Subset

2 of the 8 differences are in the Test Dataset Subset, and the deviation percentage in Test Dataset is limited to 2%. A complete comparison of the Validation study related to Test Dataset Subset is given in Figure 3.12.

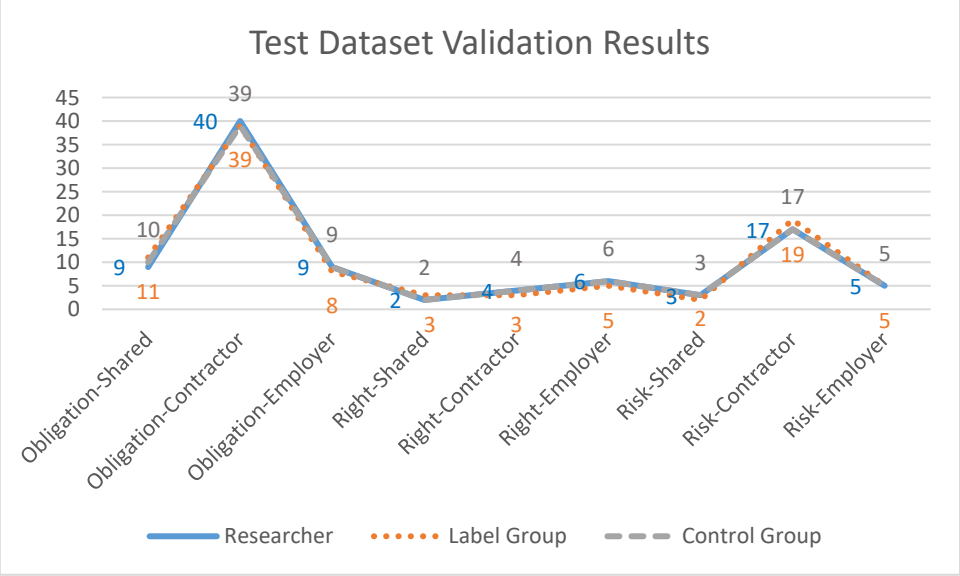

Figure 3.12. Validation Results of Test Dataset Subset

The total deviation percentage in 280 sentences is calculated as %2.8. It is considered that difference is acceptable because the labeling process is based on personal experiences and knowledge. It is acknowledged that labels in datasets can be modified with a comprehensive study with the participation of experts, but this requires more than 160 hours of study when considering that the Validation study takes 16 hours for each group. In the scope of this research, the developed Training Dataset and Test Dataset have been accepted as decently accurate to develop machine learning models to analyze the FIDIC-based construction contracts.

The following section presents selected machine learning algorithms, developed models, and employed processes in detail.

CHAPTER 4

CLASSIFICATION PROCESS, DATA CLEANING AND MACHINE LEARNING MODELS

In this chapter, the first section gives details of selected NLP techniques used for data cleaning and feature engineering to build a classification model. The second section gives details of machine learning algorithms. The third section presents the 12 machine learning models that were trained in this study. In the last section, the employed classification model training process is presented. The classification process has two main steps to develop Machine Learning based text classification model. The next chapter gives training results of Machine Learning Models and performance comparisons.

4.1 Data Cleaning and Feature Engineering with NLP

Data cleaning and feature engineering are important steps to develop solid machine learning models when dealing with text data.

Noise, which can be defined as extra characters that do not change the meaning of the text in the perspective of machine learning, such as punctuations, different cases, and stop words, needs to be cleaned before feeding the text data into the machine learning model since all these noises are not used by machines meaningfully. These steps are defined as text preprocessing in ML and NLP literature.

In literature, various text preprocessing techniques are defined. These can be listed as expanding contractions, lowering cases, removing punctuations, removing digits, removing stop words, lemmatization, and removing extra spaces. These are commonly used techniques in NLP literature. Implementations of text preprocessing to Training Dataset and Test Dataset are given in section 4.1.1 Preprocessing Text Data in detail.

Feature engineering is defined as processing the raw data for extracting suitable information that can be used by machine learning models. Extracted information, which is named as features, are inputs for machine learning models. It is a critical step in developing an ML model. When dealing with text data, feature engineering mainly refers to transforming text data into numerical representation to make computers able to process it. In summary, the goal is to represent sentences in datasets in a computer-friendly way.

There are different options to represent text data in numerical form. Parsing, Part of Speech (POS) tagging, and Named Entity Recognition (NER) focus on grammatical features of text data. Parsing is breaking a sentence into smaller chunks to understand the syntactic structure of sentences. POS tagging is related to labeling the corresponding part of speech to each word in a text, such as a noun, verb, etc. Named Entity Recognition is the process of extracting proper names from texts that represent real-world objects such as people, locations, and organizations.

Parsing is commonly used in systems that deal with grammatical features, such as correction systems. An example of a grammar correction system can be given as Grammarly software. POS tagging is a useful tool for developing chatbots, information retrieval, etc. Named Entity Recognition is used if the objective is information extraction from a large corpus.

In this research, the objective is to classify sentences in predefined labels so that statistical methods and advanced word representation methods are used. Statistical methods employed in this research are Bag-of-words (BoW), which is based on word occurrence, and term frequency-inverse document frequency (TF-IDF), which is based on a calculation related to the number of times a word appears. Employed advanced method is word embedding, and different versions of it are used in this research, such as Spacy, Glove, and Bert pre-trained embeddings. Details of text vectorization in the scope of this research are given in section 4.1.2 Vectorization of Text.

4.1.1 Preprocessing Text Data

In the previous section, expanding contractions, lowering cases, removing punctuations, removing digits, removing stop words, lemmatization, and removing extra spaces are mentioned as commonly used text-cleaning techniques in text preprocessing. Techniques are explained in the following sections in detail.

4.1.1.1 Expanding contractions

Contraction is used in English in both writing and speaking. Contraction is a short form of a word. An example of a common contraction in English is that “I have” can also be written as “I’ve”. Since the data in this research is derived from the FIDIC contracts and an actual construction project, which are legal documents, there are no problems related to contractions in Training Dataset and Test Dataset. Therefore, this step is excluded from this research.

4.1.1.2 Lowering Cases

Text data needs to be in the same case to be interpreted the same by some of the algorithms due to being handled differently in lowercase and uppercase. All text data in Training Dataset and Test Dataset are converted to lowercase. This step was handled by using Natural Language Tool Kit (NLTK) (Steven Bird, Ewan Klein, 2009) in Python.

4.1.1.3 Removing Punctuations

Another text cleaning step is removing punctuations from text data. Parenthesis and box brackets are the main problems in the datasets used in this research. To simplify the text data, 32 main punctuations, which are given in Table 4.1, are replaced with space in all sentences by using the regular expression library in Python.

Table 4.1 Punctuations List

~	!	“	#	$	%	&	'	(	)	*	+	,	-	.	/
:	;	<	=	>	?	@	[	\	]	^	_	`	{	|	}

4.1.1.4 Removing Digits

In the literature, it is stated that removing digits from text data can increase the total performance of machine learning models when compared to models trained with text data that contains digits. However, text data in this research is taken from contracts, and digits are important indicators for understanding risk, obligation, and rights contained in a text. As a result, to eliminate the drawbacks of mismatching due to different digit values, all digits are replaced with the same number in all sentences in the datasets using Python's regular expression library.

4.1.1.5 Removing Stop Words

In the text, some of the words are commonly used, such as “they”, “there”, “this”, and “where”; however, these words do not provide specific information that can be used in machine learning models to understand the text. As a result, the literature suggests removing stop words from text data before feature engineering. NLTK (Steven Bird, Ewan Klein, 2009) includes a stop word list, and in this research, NLTK with the predefined stop word list has been used to remove stop words from sentences.

4.1.1.6 Lemmatization

Words can be inflected in sentences in English. As an example, “pay” be inflected as “pay”, “paid”, “pays” or “paying” in sentences. Lemmatization focuses on

converting inflected words to the base form that can appear in the dictionary. This research uses NLTK to convert words in sentences into lemma form.

4.1.1.7 Removing Extra Spaces

Text preprocessing steps that are mentioned in previous sections may result in extra spaces in the texts. So these extra spaces are removed from sentences by using the regular expression library in Python.

4.1.1.8 Employed Preprocessing Steps

In this research, X values, which are sentences taken from contracts, in Training Dataset and Test Dataset were subjected to preprocessing steps in the order given in Figure 4.1. Datasets are uploaded from excel files into Pandas Data Frames (The Pandas Development Team, 2020). Starting from the first row in the X values column, all characters in the sentence are converted to lowercase. After that, punctuations are removed from the text, and by using the predefined stop word list, the sentence is simplified by removing stop words. As explained in previous sections, digits are replaced with one. Before removing extra spaces in the sentence, words are converted to lemma form in step 5. The preprocessed sentence is saved into a new column in the data frame. After, the same process is repeated for the following X Value in the data set until all X values are preprocessed in Dataset. At the end of preprocessing, Data Frame that contain modified X values in a new column is exported to an excel file for being subject to vectorization steps given in the next section.

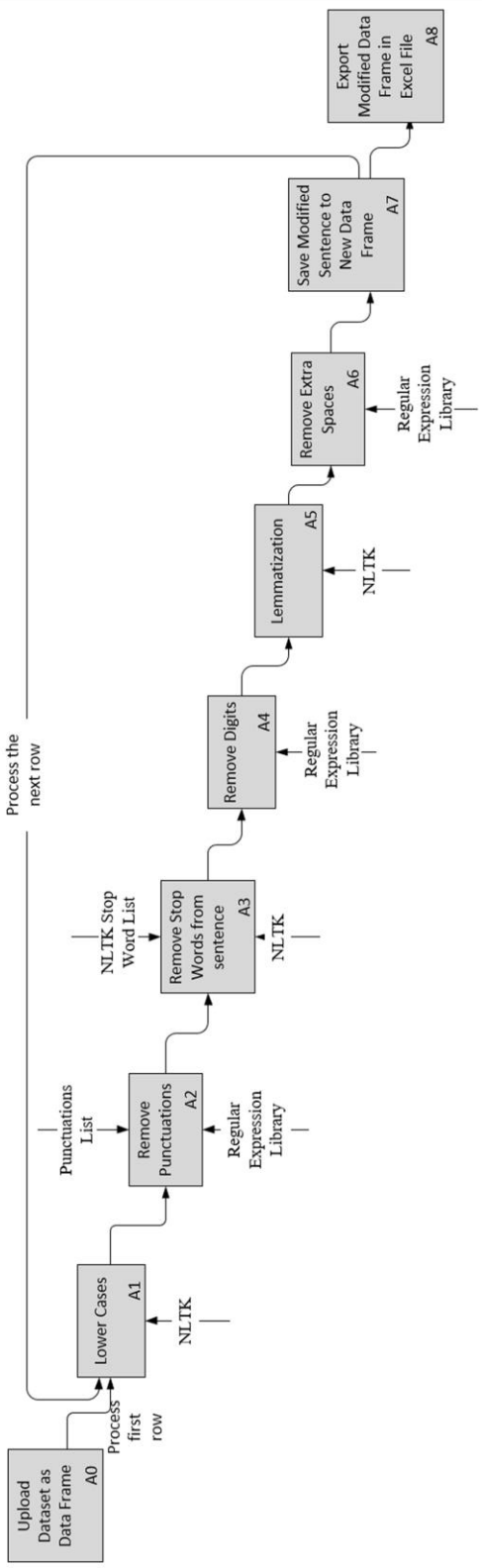

Figure 4.1. Preprocessing Steps

4.1.2 Vectorization of Text

Supervised ML models are needed inputs called X values and outputs called Y values to be trained. As explained, in NLP, inputs are pieces of text. Y values are categories that are defined and described in Chapter 3. In this research, X values are sentences taken from construction contracts; however, these X values are unsuitable for use directly in ML model training. In the previous section, X values are simplified by cleaning steps to focus only on essential parts of the texts; however, these cleaned X values still cannot be processed by computers. When dealing with text data, feature engineering mainly refers to transforming text data into numerical representation to make computers able to process it. In summary, this section aims to present the path followed to represent sentences in datasets in a computer-friendly way.

In this research, three types of word representation methods were used mainly. These can be listed as Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and Word Vectors. In the word vector method, four different sub-alternatives were evaluated. These four alternatives were training a custom word embedding from the datasets, Spacy (Honnibal et al., 2020) pre-trained word embedding, Glove (Pennington et al., 2014) pre-trained word embedding, and Bert (Devlin et al., 2019) pre-trained word embedding. Five machine learning algorithms were used to develop machine learning models in this research. These machine learning algorithms were combined with 6 word representation methods mentioned, and 12 machine learning models were developed in total.

Details of these 5 machine learning algorithms are given in 4.2 Machine Learning. Also, machine learning models, which are combinations of 5 machine learning algorithms, and 6 word representation methods, are given 4.3 Implemented Machine Learning Models.

In the following sections, details of employed vectorization alternatives, which are BoW, TF-IDF, Spacy pre-trained word vector, Glove pre-trained word embedding,

Bert pre-trained word embedding, and training a custom word embedding by using Keras (Chollet & others, 2015) are explained in details.

At the end of this step, sentences in Training Dataset and Test Dataset have been converted into a form that can be used in machine learning applications as inputs, namely X values, as shown in Figure 4.2.

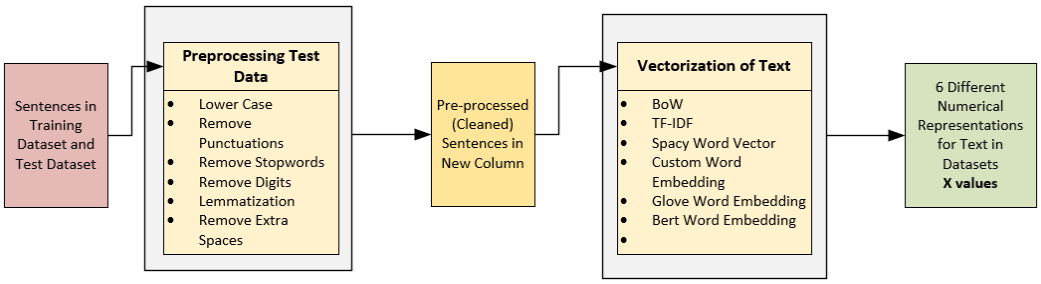

Figure 4.2. Conversion Steps of Sentences in Datasets to Numerical Representation

4.1.2.1 Bag of Words (BoW)

BoW is used to represent the text according to the number of appearances of a word by using a fixed-length vector, which is created from a vocabulary. A simple explanation is made by Qader et al. (2019) by using two texts. Texts given in their research are similar to “Berk loves to go to theatre. Going to theatre is one of favorites of Poyraz.” for the first text and “Berk also wants to go to watch curling or to go to trekking.” for the second text.

Vocabulary that is created from these two texts can be represented as {“Berk”, “loves”, “to”, “go”, “theatre”, “going”, “is”, “one”, “of”, “the”, “favorites”, “Poyraz”, “also”, “wants”, “watch”, “curling”, “or”, “trekking”}

The length of the vector for BoW created from this vocabulary is equal to the number of words in the vocabulary, which is 18. Sentences are represented by using this vector with word frequencies.

The first text with word frequency is vectorized as;

[1, 1, 3, 1, 2, 1, 1, 1, 2, 1, 1, 1, 0, 0, 0, 0, 0, 0]

When the vector of the first sentence is decoded result;

[1x Berk, 1x loves, 3x to, 1x go, 2x theatre, 1x going, 1x is, 1x one, 2x of, 1x the, 1x favorites, 1x Poyraz, 0x also, 0x wants, 0x watch, 0x curling, 0x or, 0x trekking]

The second text with word frequency is vectorized as;

[1, 0, 4, 2, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1].

When the vector of the second sentence is decoded result;

[1x Berk, 0x loves, 4x to, 2x go, 0x theatre, 0x going, 0x is, 0x one, 0x of, 0x the, 0x favorites, 0x Poyraz, 1x also, 1x wants, 1x watch, 1x curling, 1x or, 1x trekking]

BoW logic has been implemented to represent all sentences in the dataset as vectors by using scikit-learn (Pedregosa et al., 2011) Python library. The CountVectorizer function was used to create vocabulary from Training Dataset and convert sentences to vectors. The Result vocabulary contains 1680 sentences, and each sentence is represented by a vector that lengths 1680.

4.1.2.2 Term Frequency-Inverse Document Frequency (TF-IDF)

TF IDF is a method to calculate numerical statistics for defining how a word is important in a text. Similar to BoW, a vocabulary is created from the text. Term Frequencies and Inverse Document Frequencies are calculated, and the multiplication of these two is TF-IDF. TF-IDF calculation is demonstrated by using the example derived from the work of Qader et al. (2019) that is presented in the BoW section.

Term Frequency is calculated based on how many times a word appears in a text. This frequency is divided by the total number of words in the text, as given in Equation 1. The result is taken as term frequency, and implementation to sample texts is given in Table 4.2.

$$TF_{w,t} = \frac{n_{w,t}}{number\ of\ words\ in\ the\ text} \qquad (1)$$

Table 4.2 Term Frequency Calculation Example

Term	***Text 1***	***Text 2***	***TF Text 1***	***TF Text 2***
Berk	1	1	1/16	1/13
loves	1	0	1/16	0
to	3	4	3/16	4/13
go	1	2	1/16	2/13
theatre	2	0	2/16	0
going	1	0	1/16	0
is	1	0	1/16	0
one	1	0	1/16	0
of	2	0	2/16	0
the	1	0	1/16	0
favorites	1	0	1/16	0
Poyraz	1	0	1/16	0
also	0	1	0	1/13
wants	0	1	0	1/13
watch	0	1	0	1/13
curling	0	1	0	1/13
or	0	1	0	1/13
trekking	0	1	0	1/13

To understand the importance of a word, the IDF value is calculated based on equation 2.

$$IDF_w = \log \frac{number\ of\ text}{number\ of\ text\ with\ word\ "w"} \qquad (2)$$

If a word frequently appears in texts, its importance decreases. IDF calculation for sample sentences is given in Table 4.3;

As can be seen in the table, IDF values are calculated as 0 for the words “Berk”, “to” and “go” since these words appear in both sentences.

Table 4.3 Inverse Document Frequency Calculation Example

Term	***Text 1***	***Text 2***	***IDF***
Berk	1	1	0
loves	1	0	0.3
to	3	4	0
go	1	2	0
theatre	2	0	0.3
going	1	0	0.3
is	1	0	0.3
one	1	0	0.3
of	2	0	0.3
the	1	0	0.3
favorites	1	0	0.3
Poyraz	1	0	0.3
also	0	1	0.3
wants	0	1	0.3
watch	0	1	0.3
curling	0	1	0.3
or	0	1	0.3
trekking	0	1	0.3

TF-IDF is the multiplication of TF and IDF values calculated for a word, as given in Equation 3. TF-IDF calculations are given in Table 4.4 for sample texts. As seen in the table, although the word “Berk” appears in the texts, TF-IDF values in both sentences are 0 since the IDF value equals 0.

$$(TF - IDF)_{w,s} = TF_{w,t} * IDF_w \qquad (3)$$

As a result, TF-IDF columns in Table 4.4 are the vectorized form of text that can be used as input in machine learning algorithms.

By using TF-IDF, the first text is vectorized as shown in column TF-IDF Text 1 of Table 4.4;

[0, 0.01875, 0, 0, 0.0375, 0.01875, 0.01875, 0.01875, 0.0375, 0.01875, 0.01875, 0.01875, 0, 0, 0, 0, 0, 0]

By using TF-IDF, the second text is vectorized as shown in column TF-IDF Text 2 of Table 4.4;

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.02308, 0.02308, 0.02308, 0.02308, 0.02308, 0.02308].

TF-IDF logic has been used to represent all sentences in the dataset by using scikit-learn (Pedregosa et al., 2011) Python library. The TfidfVectorizer function is used to create vocabulary from Training Dataset and to convert sentences into vectors. The resulting vocabulary contains 1680 sentences, and each sentence is represented by a vector that lengths 1680.

BoW and TF-IDF are useful methods to convert text to vectors that can be understood and processed by computers. However, the context of the words is still missing part of them. To account for the similarity between words, word embedding techniques are used. Word embeddings that are used in this research are given following sections.

Table 4.4 TF-IDF Calculation Example

Term	*TF Text 1*	*TF Text 2*	*IDF*	*TF-IDF Text 1*	*TF-IDF Text 2*
Berk	0.06	0.08	0	0	0
loves	0.06	0.00	0.3	0.01875	0
to	0.19	0.31	0	0	0
go	0.06	0.15	0	0	0
theatre	0.13	0.00	0.3	0.0375	0
going	0.06	0.00	0.3	0.01875	0
is	0.06	0.00	0.3	0.01875	0
one	0.06	0.00	0.3	0.01875	0
of	0.13	0.00	0.3	0.0375	0
the	0.06	0.00	0.3	0.01875	0
favorites	0.06	0.00	0.3	0.01875	0
Poyraz	0.06	0.00	0.3	0.01875	0
also	0.00	0.08	0.3	0	0.02308
wants	0.00	0.08	0.3	0	0.02308
watch	0.00	0.08	0.3	0	0.02308
curling	0.00	0.08	0.3	0	0.02308
or	0.00	0.08	0.3	0	0.02308
trekking	0.00	0.08	0.3	0	0.02308

4.1.2.3 Custom Word Embedding

Word Embeddings are a neural natural language model initially presented in 1986 (Landthaler et al., 2016). Word embedding is developed to represent words in n-dimensional space that allows for defining similarities between them. It is the collective name for techniques in NLP, where words or expressions are represented by vectors. Word embedding aims to quantify and categorize semantic similarities between words based on their distributional properties in large samples of language

data. Various research groups developed pre-trained word embedding models such as Word2vec (Mikolov et al., 2013) by Google, GloVe (Pennington et al., 2014) by Stanford University, fastText (Bojanowski et al., 2016) by Facebook Artificial Intelligence Research, Bert (Devlin et al., 2019) by Google.

In this research, three pre-trained word embeddings were used in addition to custom word embedding that is trained based on Training Dataset. Pre-trained word embeddings are explained in the following sections.

Keras (Chollet & others, 2015) Python Library was used to develop custom word embedding to be used in this research. All text in Training Dataset was used as input to train word embedding. As the vector size, 200 was selected, and this word embedding was used as input in Recurrent Neural Network (RNN) whose details are given in section 4.2.

As a result, word embedding was developed for words that exist in the vocabulary of Training Dataset. Each word was represented by a word vector whose length was equal to 200.

4.1.2.4 Spacy Pre-trained Word Embedding

Spacy (Honnibal et al., 2020) is an open-source Python library that provides language models in 23 languages. This research focuses on English contract documents; therefore, the most extensive language model for English is used to transform sentences into vectors. “en_core_web_lg” (Explosion, 2022) is an English language model that contains vocabulary and pre-trained vectors. This language model was developed from web sources that consist of blogs, news, and comments. This model includes 514.000 unique vectors, and each vector has 300 dimensions.

Each word in Training Dataset and Test Dataset has been converted into vectors, which have 300 dimensions, by using the selected Spacy pre-trained word vector in Python. These vectors were used as input in 3 ML models that were developed with 3 different ML algorithms.

4.1.2.5 Glove Word Embedding

Glove is developed by using an unsupervised learning algorithm to represent words in vectors (Pennington et al., 2014). Glove research has been conducted at Stanford University, and researchers provide 4 different pre-trained word embedding models that are trained over different corpus. The most lightweight model was trained over Wikipedia and Gigaword's fifth edition. This model contains 6 billion tokens and 400 thousand vocabulary size 300 dimensions vectors. The most extensive model is trained over Twitter data that contain 2 billion tweets, 27 billion tokens, 1.2 million vocabulary size, and 200 dimensions vectors.

The Wikipedia model is used in this research because it is more suitable since the language used in the dataset is more suitable for the research conducted on construction contracts.

Each word in Training Dataset and Test Dataset has been converted into vectors by using the selected Wikipedia Glove model. Each vector has 300 dimensions. These vectors are used as input in the RNN algorithm.

4.1.2.6 Bert Word Embedding

Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) is developed by Google. This model pre-trained over unlabeled book corpus and Wikipedia, which contains 3300 million words in total. It has been used in the Google search engine since 2020.

BERT is an in-depth bidirectional unsupervised language representation that is pre-trained using only a plain text sentence. Models developed before BERT, like GloVe, generate only one vector for each word, while BERT also considers the context in which the word is used. For example, considering the context of the word “running”, two separate vectors should be created for its use in the sentences like "He is running a company" and "He is running a marathon". BERT provides two different vectors

for use in these two sentences, while Glove provides the same vector in both contexts. (“BERT (Language Model),” 2022)

BERT vocabulary size is 30522, and a 768 dimensions vector represents each word. Each word in Training Dataset and Test Dataset has been converted into vectors by using the selected BERT Base model. These vectors are used as input in the BERT algorithm.

4.2 Machine Learning Algorithms

Building computers that automatically evolve through experience is the question that the Machine Learning (ML) topic investigates. This topic lies between the intersection of computer science and statistic and is located at the center of artificial intelligence. Data-driven machine learning methods expedite the improvements in evidence-based decision-making systems in many areas (Jordan & Mitchell, 2015).

Computer algorithms and analytics are used to create predictive models to solve real-life problems in ML. In order to create a successful prediction of the future, ML needs to access structured or unstructured data to learn from them. The basic representation of the machine learning process is given in Figure 4.3.

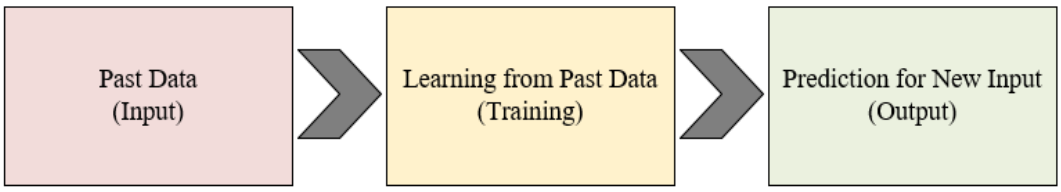

Figure 4.3. Basic Representation of Machine Learning

Machine learning methods can be classified into 3 main categories according to historical data that can be used in algorithms. These categories are Supervised Learning, Unsupervised Learning, and Reinforcement Learning.

Supervised Learning: As the name implies, in supervised learning, supervision is provided in the training of machine learning algorithms by feeding labeled input according to expectations. Supervised learning future outcomes predicted based on labeled past data. For example, pictures of fruits are labeled with the names, and a trained machine learning algorithm with this dataset can predict the name of the fruit in new pictures. Examples of supervised learning algorithms can be listed as Support Vector Machines, Logistic Regression, Decision Trees, Nearest Neighbor, Naïve Bayes, Discriminant Analysis, and Neural Networks, as presented in Figure 4.4.

Unsupervised Learning: Unlabeled data is used as input in unsupervised learning, and algorithms define the pattern in the data. Unsupervised learning can be used to categorize the data according to hidden features that exist in input data. For example, pictures containing bus, car, and truck images can be categorized into 3 main classes by unsupervised learning. Examples of unsupervised learning algorithms can be listed as K-means, Hierarchical, Hidden Markow Model, and Neural Networks, given in Figure 4.4.

Reinforcement Learning: Reinforcement Learning is based on training an agent in an undetermined environment for a specific task. The feedback from the environment to an agent is given as a reward. The agent tries to maximize the reward taken from the environment. An example of reinforcement learning implemented in real life is autonomous cars. Trafic is an open environment that has unlimited scenarios. Collecting all scenarios as input data to train models is not possible. So algorithms that are used in autonomous cars use reinforcement learning to make optimal decisions while cruising according to predefined tasks, such as not hitting any other object. Examples of reinforcement learning algorithms can be listed as Markov Decision Process, Q-Learning, and Q-Learning with Neurol Networks (Deep Q-Learning), as presented in Figure 4.4.

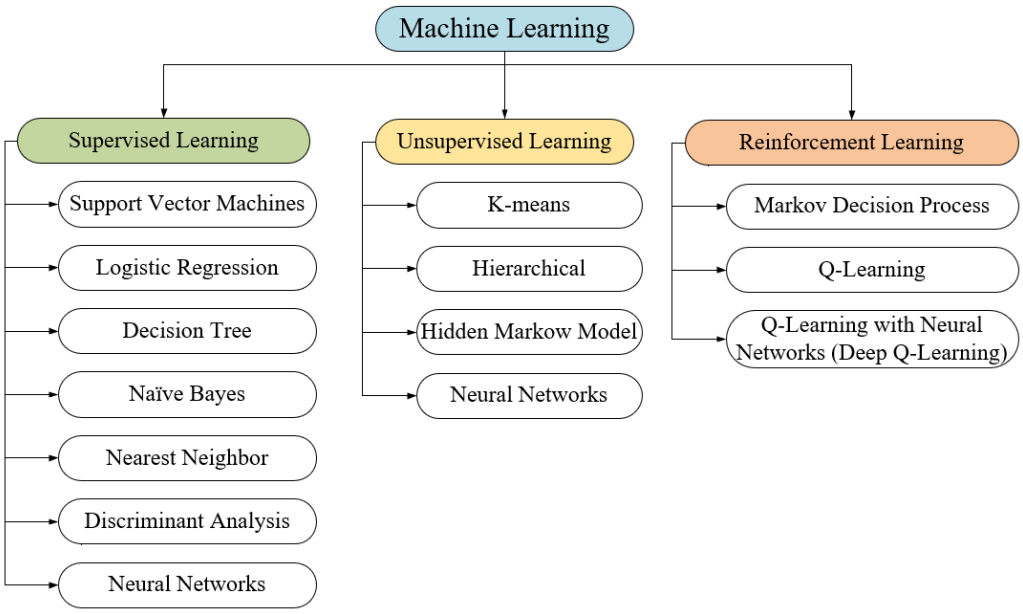

Figure 4.4. Main Machine Learning Methods

In this research supervised learning method is the suitable approach to achieve the needed outcome, which is classifying sentences in contracts in predefined classes. As given in Chapter 3, labeled datasets were created to train and test machine learning models. Support Vector Machine, Logistic Regression, and Decision Tree algorithms are selected to be implemented in this study according to preliminary trials' performance results.

In addition to these, neural networks, which are also named deep learning, are selected to be implemented. Deep learning is part of machine learning which is developed based on neural networks. As seen in the previous section, deep learning algorithms can be used in all three machine learning methods.

Several deep learning algorithms are available, but the most known are Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). For tasks involving sequential input, such as language, it is better to use RNNs when compared to CNN. (LeCun et al., 2015).

In addition to RNN and CNN, Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) is a recent improvement in deep learning

algorithms for natural language processing. BERT models are trained over millions of textual data and have the potential to improve prediction performances in relatively small datasets with the help of pre-trained language models.

Selected Deep Learning algorithms are Recurrent Neural Network (RNN) and Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) as Deep Learning algorithms in this study.

In conclusion, 5 machine learning algorithms are included, 3 are statistical methods, and 2 are deep learning methods. As explained in the previous section, textual data is converted to numerical forms with vectors to be made textual data understandable by the computer. Selected machine learning algorithms are matched with 6 text vectorization alternatives by comparing the complexity of the algorithm and vectorization method. Implemented machine learning models are presented in the following section.

4.3 Implemented Machine Learning Models

Machine learning algorithms and vectorization techniques that are used to develop machine learning models in this research are presented in previous sections. By using these algorithms and techniques, 12 machine learning models, which are listed in Table 4.5, were developed and implemented in the scope of research by varying the vectorization technique and machine learning algorithms. Three machine learning models were based on deep learning algorithms, three machine learning models were based on regression analysis algorithm, three models were based on support vector machine algorithm, and three models were based on decision tree algorithm.

Logistic Regression: Classification issues are resolved via logistic regression. In contrast to linear regression, which predicts a continuous outcome, it achieves this by forecasting categorical outcomes.

Parameters used in logistic regression is given below.

[penalty='l2', dual=False, tol=0.0001, C=1000.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=100, solver='lbfgs', max_iter=100, multi_class='ovr', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None]

Support Vector Machine: A supervised machine learning approach called Support Vector Machine (SVM) is used for both classification and regression. Finding a hyperplane in an N-dimensional space that clearly classifies the data points is the goal of the SVM method. The number of features determines the hyperplane's size.

Parameters used in support vector machine is given below.

[C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=-1, decision_function_shape='ovr', break_ties=False, random_state=100]

Decision Tree: Both classification and regression issues can be resolved using decision trees. Each leaf node of the decision tree corresponds to a class label, and the interior nodes of the tree are used to represent the attributes in order to answer the problem. The decision tree can be used to represent any boolean function on discrete attributes.

Parameters used in decision tree is given below.

[criterion='gini', splitter='best', max_depth=3, min_samples_split=2, min_samples_leaf=5, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, class_weight=None, ccp_alpha=0.0]

Recurrent Neural Network: An artificial neural network that employs sequential data or time series data is known as a recurrent neural network (RNN). For ordinal or temporal issues, such as language translation, natural language processing (NLP), speech recognition, and image captioning, these deep learning methods are frequently applied. Recurrent neural networks (RNNs) use training data to learn, just like feedforward and convolutional neural networks (CNNs) do. They stand out due

to their memory, which allows them to affect the current input and output by using data from previous inputs.

Parameters used in RNN is given below.

model = Sequential()

model.add(Embedding(MAX_NB_WORDS,EMBEDDING_DIM, input_length=X.shape[1]))

model.add(SpatialDropout1D(0.2))

model.add(LSTM(100, dropout=0.2, recurrent_dropout=0.2))

model.add(Dense(13, activation='softmax'))

model.compile(loss='categorical_crossentropy',optimizer='adam', metrics=['accuracy'])

epochs = 5

batch_size = 64

BERT: At its core, BERT is a transformer language model with self-attention heads and a variable number of encoder layers. Language modeling (15% of tokens were hidden, and BERT was trained to infer them from context) and next sentence prediction were the two tasks that BERT had been pretrained on (BERT was trained to predict if a chosen next sentence was probable or not given the first sentence). BERT gains knowledge of word contextual embeddings as a result of training. BERT can be fine-tuned with fewer resources on smaller datasets after pretraining, which requires expensive computational resources, to maximize its performance on certain tasks.

Parameters used in Ktrain Python Library for BERT is given below.

(X_train, y_train), (X_test, y_test), preproc = text.texts_from_df(train_df = data_set_filtered_to_OwnerShip, text_column='Cleaned_Sentences', label_columns='OwnerSimplified',maxlen=315, preprocess_mode="bert")

model = text.text_classifier(name='bert', train_data=(X_train, y_train), preproc=preproc)learner = ktrain.get_learner(model=model, train_data=(X_train, y_train), val_data=(X_test, y_test), batch_size=10)

learner.fit_onecycle(lr = 2e-4, epochs=4)

Logistic Regression, Support Vector Machine, and Decision Tree algorithms were matched with Bag of Words, Term Frequency Inverse Document Frequency, and Spacy Word Embedding vectorization alternatives. 9 models were created from these algorithms and vectorization methods. RNN was matched with Custom Word Embedding and Glove Word Embedding vectorization alternatives. 2 models were created from Recurrent Neural Network to implement in this study. BERT algorithm is matched with BERT Word Embedding.

Details of models that were trained and compared in this research are given in Table 4.5 with included machine learning algorithm and text vectorization technique.

Table 4.5 Implemented Machine Learning Models

Model No	***Vectorization Technique***	***Machine Learning Algorithm***
Model 1	Bag of Words	Logistic Regression
Model 2	Bag of Words	Support Vector Machine
Model 3	Bag of Words	Decision Tree
Model 4	TF-IDF	Logistic Regression
Model 5	TF-IDF	Support Vector Machine
Model 6	TF-IDF	Decision Tree
Model 7	Spacy Word Embedding	Logistic Regression
Model 8	Spacy Word Embedding	Support Vector Machine
Model 9	Spacy Word Embedding	Decision Tree
Model 10	Keras Custom Word Embedding	Recurrent Neural Network
Model 11	Glove Embedding	Recurrent Neural Network
Model 12	BERT Word Embedding	BERT

4.4 Machine Learning Classification Model Training Process

The process that followed to train the machine learning models is presented in Figure 4.5. In the first step, Training Dataset and Test Dataset, whose details are given in Chapter 3, were used as inputs.

Training Dataset, which is developed from FIDIC contracts, was used to train the classification models. In order to be used with ML algorithms, text data needs to be vectorized and preprocessed to improve the model classification performance. These requirements were met with various NLP techniques, which are explained in the data cleaning and feature engineering step. In this research, Test Split is 10% of the Training Dataset. The reason to use a Test split in addition to Test Dataset was to calculate the classification performance in Training Dataset, which includes unmodified sentences from FIDIC contracts.

Train Split and Test Split have X and Y values. X values are vectorized sentences in the data cleaning and feature engineering step, and Y values are defined labels in the dataset creation step. Train Split and Test Split are used to train the machine learning model. The ML model selection depends on the NLP technique used in the data-cleaning feature engineering step.

Train Dataset is used to train defined machine learning models. When a model is trained, it is tested with Test Split. The Confusion Matrix and Classification Report are created by using made predictions for X values and predefined Y values in Test Split to determine the internal performance of the developed classification model.

Test Dataset, which was developed from an actual construction project contract, was not used to train any of the models. Sentences in Test Dataset were processed with selected NLP techniques to clean data and extract features to use in classification models. Following this step, Text Dataset X Values were fed into the models, and Y values were predicted with developed classification models. As explained, Test Dataset also has labels which are named Test Dataset Y values in this process. Confusion Matrix and Classification Report were created by using Predicted Y

Values and Test Dataset Y Values to determine the developed classification model performance.

Internal test results and performance on the actual project were compared to determine the deviation in model accuracy. As can be seen in the results given in Chapter 5, internal test results and Test Dataset deviates in some models. This was caused because of the vectorization of text data and the selected machine learning algorithm. This can be considered as a validation step to selected classification models for the following steps of the research that have similar performances in each dataset.

The result of this process gives the performance of each individual model, whose details are presented in 4.3 Implemented Machine Learning Models.

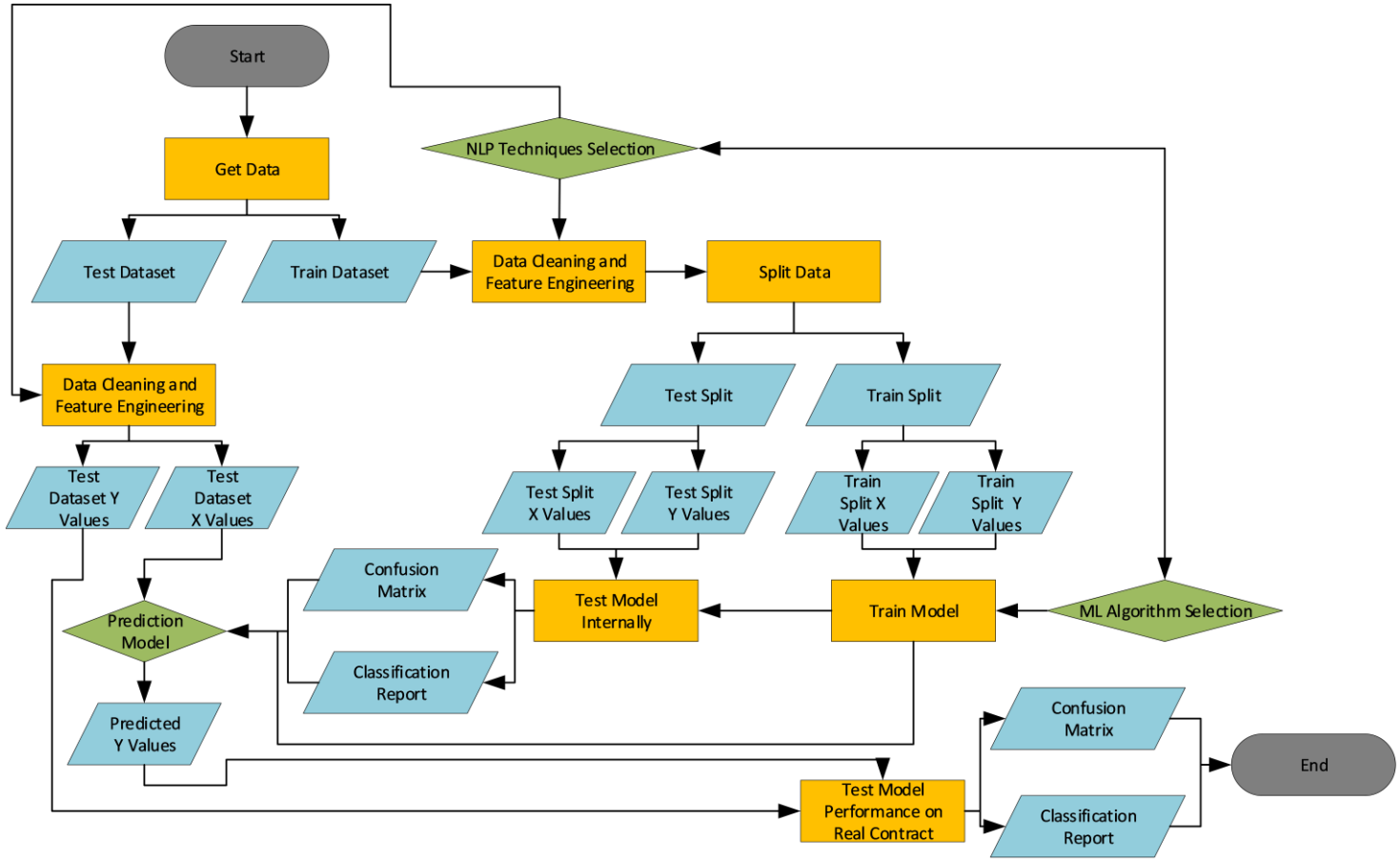

Figure 4.5. Process Model of Classification Model Development

CHAPTER 5

CLASSIFICATION RESULTS AND ENSEMBLE METHOD

In this chapter, first of all, the method for evaluating machine learning models' performance is presented. After explaining the evaluation method, the classification results of 12 machine learning models are presented. Results are presented for both sentence type classification and ownership classification. These results are named "Step 1" results.

After Step 1 results are presented, the method that was implemented to improve machine learning models' classifivation performance is explained. Results for each step are presented before providing complete results of the improved process. The end results in this step are named "Step 2" results.

In the last step, the best models with the highest accuracy and f1 scores were used to create voting models. The voting model is presented in this section, and results are given by naming them as "Step 3" results. The flow between Step 1, Step 2, and Step 3 is represented in Figure 5.1.

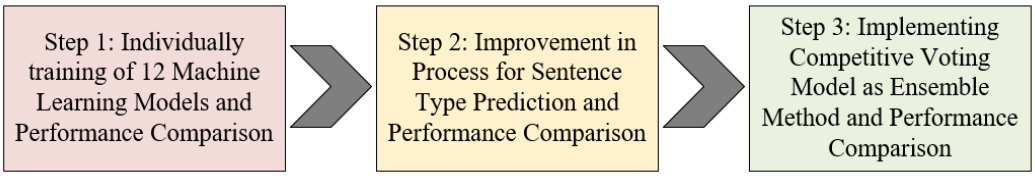

Figure 5.1. Relation Between Step 1, Step 2, and Step 3

In the last part of this chapter best results on each step are presented with an in-depth comparison.

5.1 Evaluation Method

The classification model's performance can be evaluated using four parameters presented in Table 5.1 (Sokolova & Lapalme, 2009). These four parameters are True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). These terms will be explained over binary classification using two groups which are Class A and Class B, in terms of the Class A perspective.

True Positive is the number of classified examples correctly. It means that an example that belongs to Class A is correctly identified as Class A.

False Positive is the number of incorrectly classified examples that do not belong to the class. It means that an example that belongs to Class B is classified incorrectly as Class A.

True Negative is the number of correctly classified examples that do not belong to the class. It means that an example that belongs to Class B is correctly identified as Class B.

False Negative is the number of incorrectly classified examples that actually belong to the class. It means that an example that belongs to Class A is incorrectly identified as Class B.

Table 5.1 Confusion Matrix for Binary Classification

	Actual Positive	***Actual Negative***
Predicted Positive	True Positive	False Positive
Predicted Negative	False Negative	True Negative

In multi-class classification, which is used in this research to classify sentences in terms of types under five categories, the confusion matrix can be exemplified as given in Table 5.2.

Table 5.2 Confusion Matrix for Multi-Class Classification

	Class A	***Class B***	***Class C***
Class A	Cell 1	Cell 2	Cell 3
Class B	Cell 4	Cell 5	Cell 6
Class C	Cell 7	Cell 8	Cell 9

In multi-class classification, TP, FN, FP, and TN calculation example for Class A, Class B, and Class C are given in Table 5.3. In terms of Class A, similar to binary classification, the number of examples that belong to Class A and are correctly identified as Class A is TP which is equal to Cell 1. The number of examples that belong to Class B or Class C and are incorrectly classified as Class A is FP which is equal to the sum of Cell 4 and Cell 7. The number of examples that belong to Class B or Class C and are not classified as Class A is TN which is equal to the sum of Cell 5, Cell 6, Cell 8, and Cell 9. The number of examples that belong to Class A and are incorrectly classified as Class B or Class C is FN, which is equal to the sum of Cell 2 and Cell 3.

Table 5.3 TP, FP, TN, FN Calculation Example on Multi-Class Classification

	Class A	***Class B***	***Class C***
True Positive (TP)	Cell 1	Cell 5	Cell 9
False Positive (FP)	Cell 4 + Cell 7	Cell 2 + Cell 8	Cell 3 + Cell 6
True Negative (TN)	Cell 5 + Cell 6 + Cell 8 + Cell 9	Cell 1 + Cell 3 + Cell 7 + Cell 9	Cell 1 + Cell 2 + Cell 4 + Cell 5
False Negative (FN)	Cell 2 + Cell 3	Cell 4 + Cell 6	Cell 7 + Cell 8

The performance of a machine learning model is commonly measured by six metrics (Sokolova & Lapalme, 2009) which are accuracy, precision, recall, f1 score, specificity, and area under curve (AUC).

- **Accuracy:** Indication of the overall performance of the model.
- **Precision**: Focuses on measuring the ratio of true positive over total positive predictions. It is a well-tested method to calculate how good a model is in positive identification.
- **Recall:** Measures the ratio of true positives over actual positives. With this method model's capture rate on positives can be calculated.
- **F1 score**: Harmonic mean of precision and recall (Taha & Hanbury, 2015), and it focuses on the balance between TP over predicted positives and TP over actual positives.
- **Specificity:** Opposite of the recall, which focuses on calculating the effectiveness of the model for finding negative labels.
- **Areas Under Curve (AUC):** Calculating model effectiveness on false classification avoidance.

All these six metrics are calculated by using TP, FN, FP, and TN values with the given formulas in Table 5.4.

Table 5.4 Performance Metrics and Calculation Formulas

Metric	***Calculation Formula***
Accuracy	$\frac{TP + TN}{TP + FN + FP + TN}$
Precision	$\frac{TP}{TP + FP}$
Recall	$\frac{TP}{TP + FN}$
F1 Score	$2x\frac{Precision * Recall}{Precision + Recall} = \frac{2xTP}{2xTP + FP + FN}$
Specificity	$\frac{TN}{FP + TN}$
AUC	$\frac{1}{2}X(\frac{TP}{TP + FN} + \frac{TN}{FP + TN})$

The most suitable performance metric for this study was considered to be the accuracy. However, accuracy measurements might not perform as expected if the dataset has uneven distribution throughout different classes. This problem was investigated by using an example confusion matrix given in Table 5.5.

Table 5.5 Example Imbalanced Confusion Matrix

	Actual Positive	*Actual Negative*
Predicted Positive	2 (TP)	1 (FP)
Predicted Negative	3 (FN)	400 (TN)

406 predicted data is presented in the example confusion matrix, and among these data, 5 of 406 is actual positive while 401 is actual negative. When accuracy is calculated;

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} = \frac{2 + 400}{2 + 3 + 1 + 400} = \frac{402}{406} = 0.99$$

If only the accuracy value is considered model seems to be very successful; however, when the f1 score is calculated;

$$F1\ Score = \frac{2xTP}{2xTP + FP + FN} = \frac{2x2}{2x2 + 1 + 3} = \frac{4}{8} = 0.5$$

it is found that this model is not successful in terms of F1 Score compared to Accuracy.

As a result, in this research, the performance of the developed models was measured by calculating Accuracy and f1 Score. In addition to these two metrics, Precision and Recall values were also calculated and presented following sections to analyze the f1 score more accurately.

5.2 Step 1 Classification Results

As mentioned, all 12 models used in this study were trained using the Training Dataset. The training dataset was divided into 2 (0.1 test size and 42 random state) Training Split and Test Split, to evaluate model performance internally. Additionally, model performance was assessed by Test Dataset, which was developed by using an actual construction project contract.

Precision, recall, f1 score, and accuracy values for each model were calculated, and the best models were selected based on f1 score and accuracy. Results are presented for sentence type classification and related party classification separately.

Model 12 (BERT Word Embedding and BERT Machine Learning Algorithm) is the best model for both sentence type classification and related party classification. The sentence type classification performance of Model 12 is 0.82 according to accuracy and 0.79 according to f1 score on Test Dataset. On the other hand, the related party classification performance of Model 12 is 0.80 in terms of accuracy and 0.73 in terms of f1 score. Details of other models' results are presented in the following sections

5.2.1 Type of Sentence Classification

All results for sentence type classification are given in Table 5.6. When the results are analyzed, the worst-performing model is Model 9, which uses the Decision Tree algorithm. Model 9 gives the lowest accuracy and f1-score for both Test Split and Test Dataset. It gives a 0.38 f1 score, 0.55 accuracy value for Test Split, and 0.41 f1 score, 0.62 accuracy value for Test Dataset. Model 2 and Model 4 Internal Test results give higher than 0.80 for f1 score and accuracy; however, these values decrease to 0.73 for f1 score and 0.76 and 0.75 for accuracy respectively, in Test Dataset classification performance.

Other than the differences observed in Model 2 and Model 4, the difference between Internal Test Results and Test Dataset Results is not higher than 0.06, which

indicates consistent behavior among different test conditions. The performances of the models are compared based on Test Dataset Results. The most successful models are Model 5 and Model 12, whose accuracy results are higher than 0.80 and f1 scores 0.78 and 0.79 respectively. Visualization of Test Dataset results is given in Figure 5.2. As a result, sentences can be predicted by using Bert Model with 82 percent accuracy at the end of Step 1. 0.82 accuracy value and 0.79 f1 scores are the benchmark points that tried to be improved for sentence type classification in the following steps.

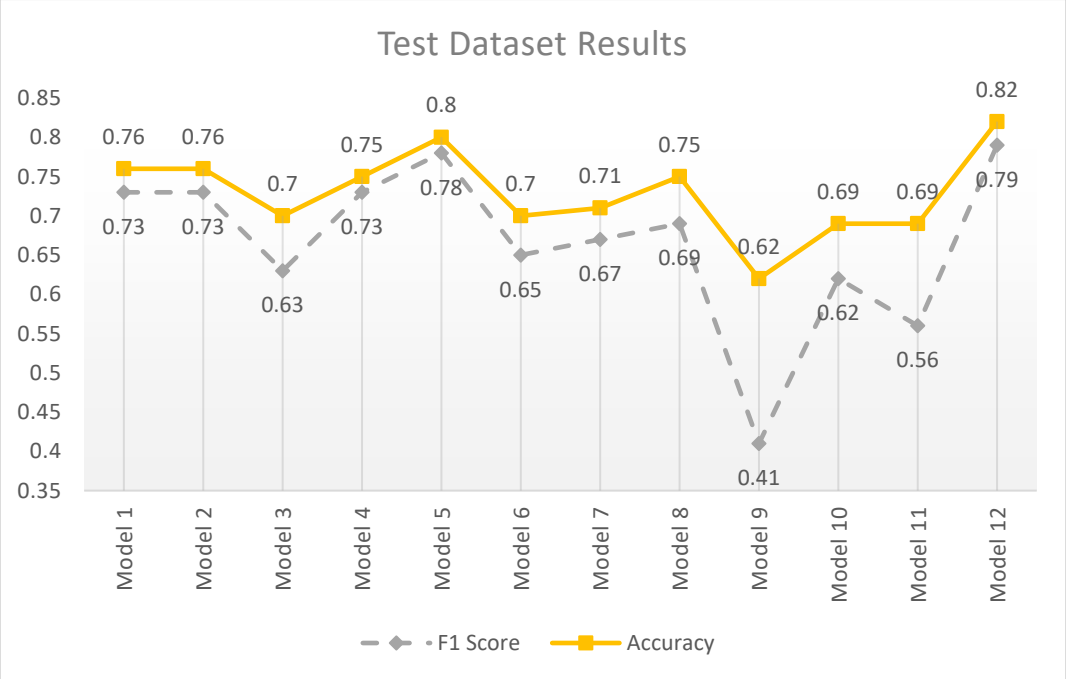

Figure 5.2. Comparison of Performance Results for Sentence Type Classification in Step 1

Table 5.6 Step 1 Result for 12 ML Models on Sentence Type Classification

			Internal Test Results				Test Dataset Results			
Model No	**Vectorization**	**ML Algorithm**	**Precision**	**Recall**	**f1-score**	**Accuracy**	**Precision**	**Recall**	**f1-score**	**Accuracy**
1	BAG OF WORDS	Logistic Regression	0.76	0.78	0.77	0.79	0.72	0.76	0.73	0.76
2	BAG OF WORDS	Support Vector Machine	0.82	0.82	0.82	0.81	0.73	0.75	0.73	0.76
3	BAG OF WORDS	Decision Tree	0.75	0.65	0.69	0.69	0.70	0.66	0.63	0.70
4	TF-IDF	Logistic Regression	0.83	0.84	0.83	0.83	0.74	0.75	0.73	0.75
5	TF-IDF	Support Vector Machine	0.83	0.79	0.81	0.81	0.79	0.77	0.78	0.80
6	TF-IDF	Decision Tree	0.75	0.63	0.65	0.69	0.70	0.66	0.65	0.70
7	Spacy	Logistic Regression	0.73	0.70	0.71	0.71	0.67	0.67	0.67	0.71
8	Spacy	Support Vector Machine	0.74	0.66	0.69	0.72	0.70	0.68	0.69	0.75
9	Spacy	Decision Tree	0.38	0.41	0.38	0.55	0.40	0.44	0.41	0.62
10	Keras Embedding	RNN	0.71	0.57	0.58	0.66	0.69	0.60	0.62	0.69
11	Glove Embedding	RNN	0.66	0.60	0.59	0.68	0.62	0.56	0.56	0.69
12	BERT Embedding	BERT	0.81	0.78	0.80	0.81	0.79	0.80	0.79	0.82

5.2.2 Related Party Classification

All results for related party classification are given in Table 5.7. When the results are analyzed, similar to sentence type classification, the model that has the worst performance is Model 9, which uses the Decision Tree algorithm. Model 9 gives the lowest accuracy and f1-score for both Test Split and Test Dataset, which are 0.52 f1 score, 0.61 accuracy value and 0.49 f1 score, 0.61 accuracy value, respectively. Model 1, Model 2, Model 4, Model 5, Model 10, and Model 11 Internal Test results give higher than 0.80 for f1 score and Accuracy; however, these values decrease to zero point sixties for f1 score and zero point seventies for accuracy. Upon further investigation, the reason for the differences in performance for Training Dataset and Test Dataset was found to be the different representation of parties in these datasets. In that point, Model 12, developed by using Bert Embedding and Bert Algorithm, shows the importance of its context-based prediction feature. As shown in Table 5.7, the best model is Model 12, with 0.80 accuracy value and 0.73 f1-score.

Visualization of Test Dataset results is given in Figure 5.3. As a result, related parties can be predicted by using Bert Model with 80% accuracy at the end of Step 1. 0.80 accuracy value and 0.73 f1-score are the benchmark points that tried to be improved for related party classifications in Step 3.

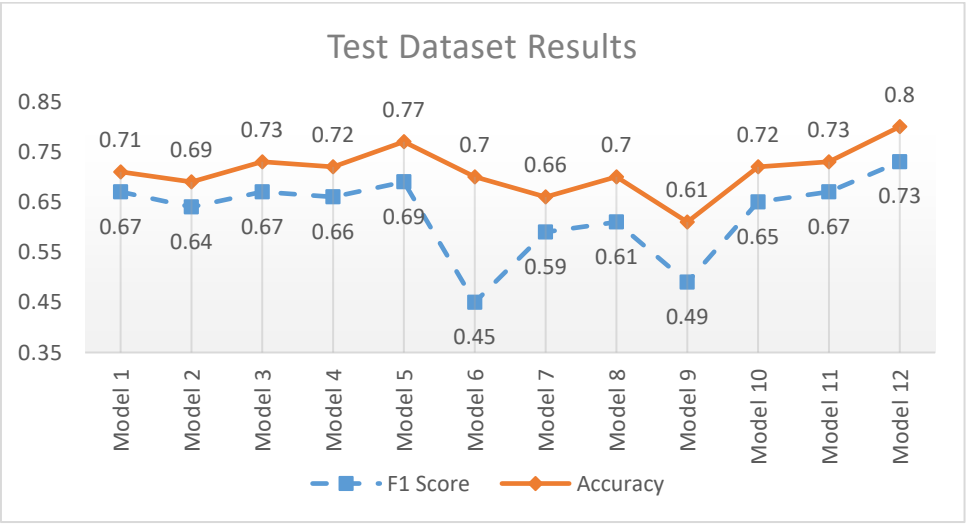

Figure 5.3. Comparison of Results for Related Party Classification in Step 1

Table 5.7 Step 1 Result for 12 ML Models on Related Party Classification

			Internal Test Results				Test Dataset Results			
Model No	**Text Vectorization**	**ML Algorithm**	**Precision**	**Recall**	**f1-score**	**Accuracy**	**Precision**	**Recall**	**f1-score**	**Accuracy**
1	BAG OF WORDS	Logistic Regression	0.82	0.80	0.81	0.81	0.66	0.68	0.67	0.71
2	BAG OF WORDS	Support Vector Machine	0.85	0.84	0.84	0.84	0.64	0.66	0.64	0.69
3	BAG OF WORDS	Decision Tree	0.75	0.56	0.55	0.69	0.66	0.68	0.67	0.72
4	TF-IDF	Logistic Regression	0.84	0.85	0.84	0.84	0.65	0.68	0.66	0.72
5	TF-IDF	Support Vector Machine	0.85	0.80	0.82	0.82	0.72	0.67	0.69	0.77
6	TF-IDF	Decision Tree	0.74	0.55	0.54	0.67	0.61	0.45	0.45	0.70
7	Spacy	Logistic Regression	0.75	0.73	0.74	0.76	0.58	0.61	0.59	0.66
8	Spacy	Support Vector Machine	0.74	0.71	0.72	0.75	0.63	0.60	0.61	0.70
9	Spacy	Decision Tree	0.59	0.50	0.52	0.61	0.55	0.47	0.49	0.61
10	Keras Embedding	RNN	0.84	0.81	0.82	0.84	0.64	0.67	0.65	0.72
11	Glove Embedding	RNN	0.82	0.78	0.79	0.81	0.72	0.68	0.67	0.73
12	BERT Embedding	BERT	0.88	0.84	0.86	0.85	0.79	0.69	0.73	0.80

5.3 Step 2 Improvement in Process for Sentence Type Classification

The classification of sentence type is a multi-class label classification problem with 5 classes which are Heading, Definition, Obligation Risk, and Right. The multi-class process can be converted to a binary classification problem by implementing “one vs rest” method. In the “one vs rest” method, the dataset is relabeled as 2 classes. In the sentence type classification problem of this research one vs rest method is implemented by labeling groups, which are given in Table 5.8, according to the process shown in Figure 5.4. In Training Dataset and Test Dataset, four new columns were created. Sentence Types were re-grouped under Heading and Clause labels in the Label Group 1 column. If a sentence in the dataset is labeled as Definition, Obligation, Risk, or Right, it is labeled as Clause in the Label Group 1 column. In Label Group 2 column, sentences that are labeled as Clause in Label Group 1 are labeled with Definition or Other. In this group, Other labels are given for sentences that are actually labeled as Obligation, Risk, or Right. In Label Group 3, sentences that are labeled as Other in Label Group 2 are labeled with Obligation and Other labels. Other label in Label Group 3 is given for sentences that are actually labeled as Risk or Right. Label Group 4 contains only 2 labels, so sentences that are labeled as Other in Label Group 3 are labeled with their actual labels.

Four label groups were used in sentence type classification in the order shown in Figure 5.4.

Table 5.8 Defined Label Groups in Process Improvement

Groups	***Label 1***	***Label 2***	***Actual Labels included in Label 2***
Label Group 1	Heading	Clause	[Definition, Obligation, Risk, Right]
Label Group 2	Definition	Other	[Obligation, Risk, Right]
Label Group 3	Obligation	Other	[Risk, Right]
Label Group 4	Risk	Right	[Right]

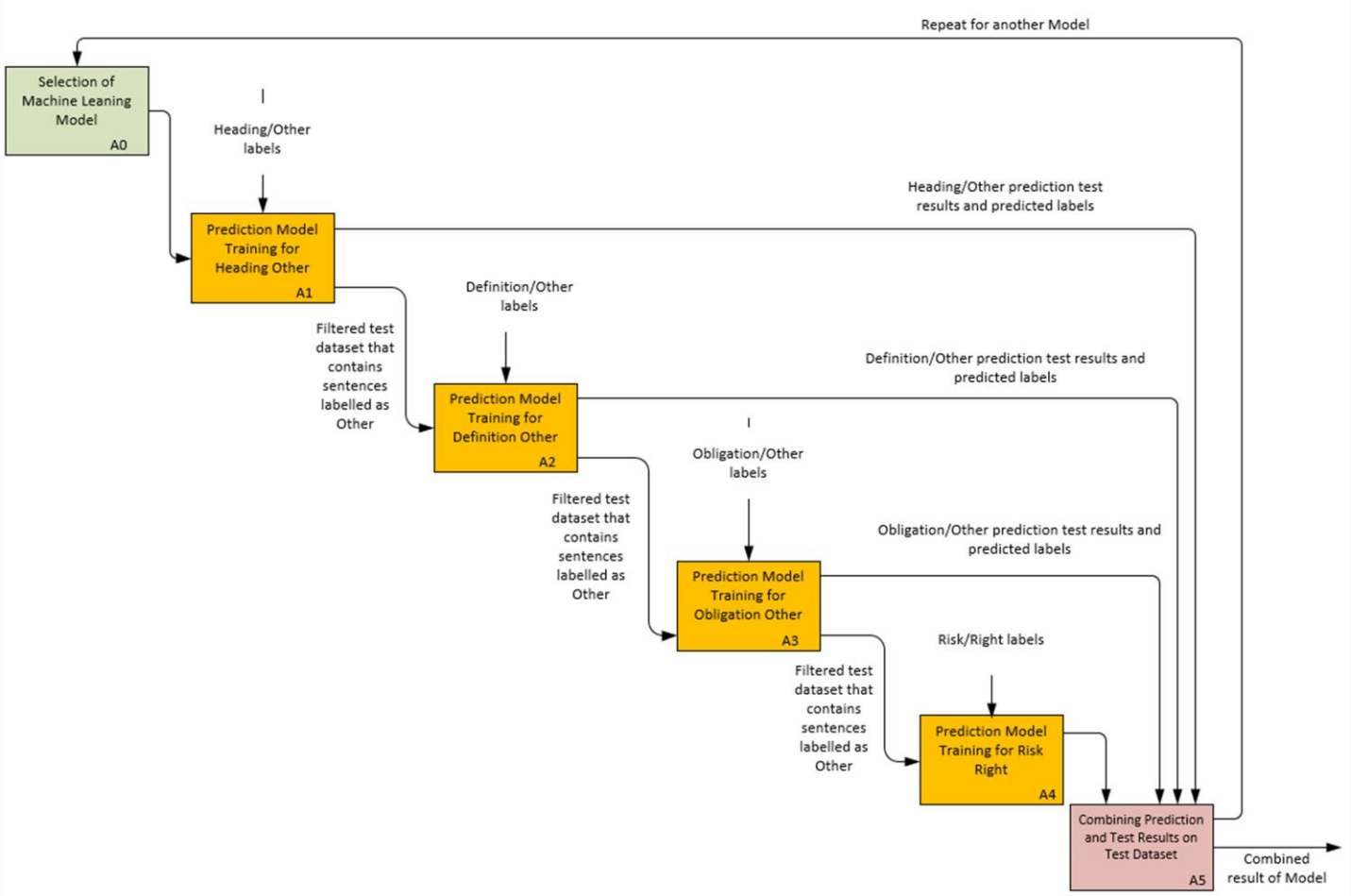

Figure 5.4. Conversion Process for Multi-Class Classification to Binary Classification in Sentence Type Label

The improvement process is implemented in all 12 Machine Learning models, which their results are presented in 5.2.1 Type of Sentence Type of Sentence Classification. At the A0 step shown in Figure 5.4, ML model was selected to implement the improvement process defined in this section.

Models were trained in A1, A2, A3, and A4 steps are shown in Figure 5.4 according to the logic given in Figure 5.5. In the A1 step, Test Dataset and Training Dataset X values are vectorized form of sentences as given in 4.1.2 Vectorization of Text section. Training Dataset and Test Dataset Y values are labels assigned in Label Group 1.

Train Dataset was divided into 2 as Test Split and Train Split. Train Split was used to train the selected Machine Learning Model. Classification performance was assessed internally by using Test Split. Test Dataset was used to assess the classification performance of the selected Machine Learning Model for Label Group 1 in an actual construction project contract.

The same process explained in the A1 step is repeated for Label Group 2 in the A2 step, Label Group 3 in the A3 step, and Label Group 4 in the A4 step. Precision, recall, f1 score, and accuracy values are given in the 5.4 Individual Label Group Results in Step 2 Improved Process section for each label group.

In the A5 step, predicted values were combined in a column to asses classification performance at the end of Step 2. This was done by combining the prediction columns created in A1, A2, A3, and A4 steps. The created column was compared with the Test Dataset Sentence Type label column, and precision, recall, f1 score, and accuracy values were calculated.

Complete results for all 12 Machine Learning models are presented in 5.5 Complete Results After Process Improvements.

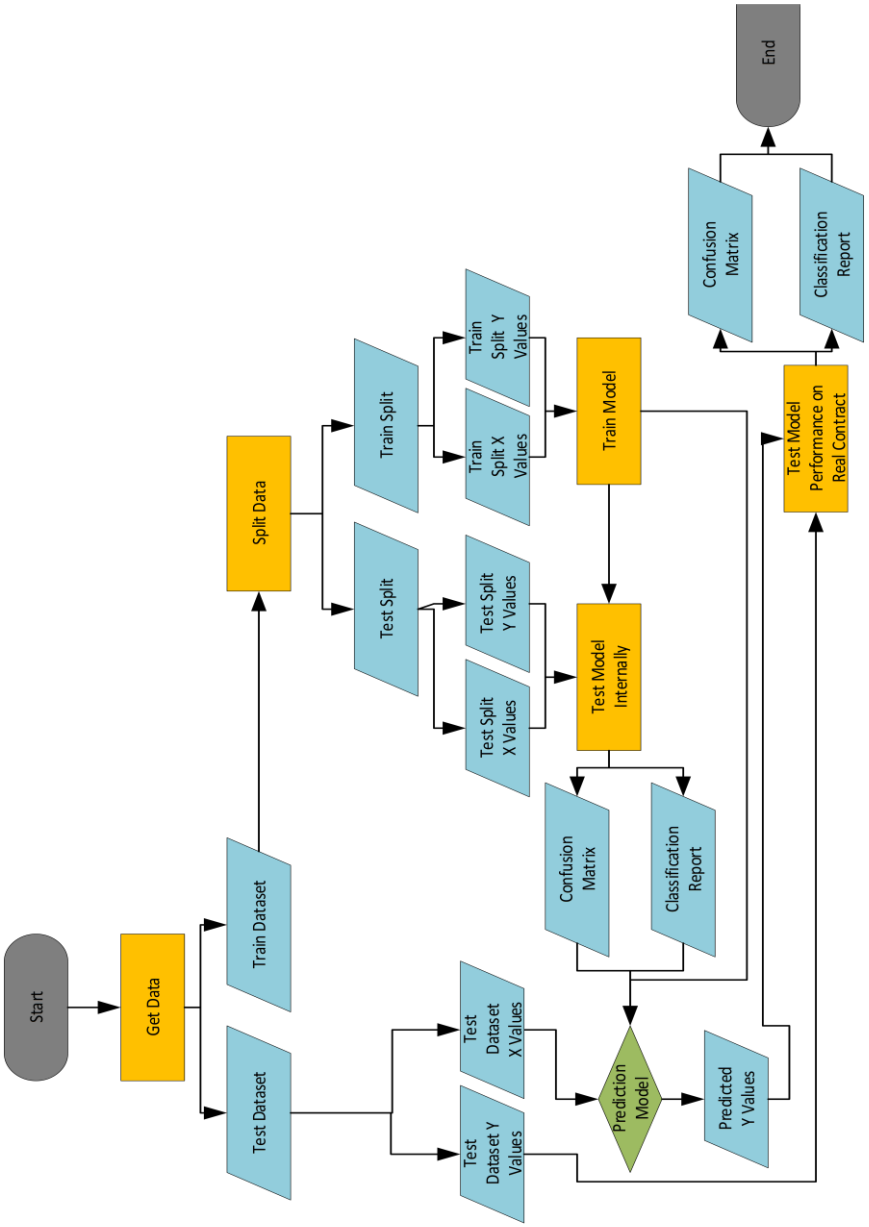

Figure 5.5. Implemented Process in A1, A2, A3, A4 steps given in Figure 5.4

5.4 Individual Label Group Results in Step 2 Improved Process

Performance metrics for each label group were calculated for all 12 Machine Learning models. Individual results show that the ability to determine Heading is 100% accurate. Performance on identifying Definitions was 98% accurate, but

performance decreases to 0.84 in defining Obligations and 0.79 in Risk and Right identification.

5.4.1 Classification of Heading Performances

Headings can be predicted successfully by all 12 Machine Learning Models, as given in Table 5.9. The best model is Bert Model, which identifies the Headings 100% in both internal test results and Test Dataset results. The accuracy level is higher than 0.99 for nine models, as shown in the figure. The worst model is Model 9, similar to the results given in Step 1.

Table 5.9 Step 2 Result for Label Group 1

		Internal Test Results				Test Dataset Results			
Classification	ML Model No	Precision	Recall	f1-score	Accuracy	Precision	Recall	f1-score	Accuracy
Heading/Clause	1	1.00	1.00	1.00	1.00	0.97	0.99	0.98	0.99
Heading/Clause	2	1.00	1.00	1.00	1.00	0.98	0.99	0.99	0.99
Heading/Clause	3	0.99	0.93	0.96	0.98	0.98	0.95	0.96	0.98
Heading/Clause	4	0.97	0.99	0.98	0.99	0.99	0.97	0.98	0.99
Heading/Clause	5	0.99	0.98	0.99	0.99	0.99	0.97	0.98	0.99
Heading/Clause	6	0.99	0.93	0.96	0.98	0.98	0.95	0.96	0.98
Heading/Clause	7	1.00	0.98	0.99	1.00	0.98	0.97	0.98	0.99
Heading/Clause	8	0.99	0.97	0.98	0.99	0.99	0.98	0.98	0.99
Heading/Clause	9	0.87	0.87	0.87	0.95	0.89	0.92	0.90	0.94
Heading/Clause	10	0.99	0.93	0.96	0.99	0.99	0.96	0.97	0.99
Heading/Clause	11	1.00	0.97	0.98	0.99	0.99	0.98	0.99	0.99
Heading/Clause	12	0.97	0.10	0.99	1.00	0.99	1.00	0.99	1.00

5.4.2 Classification of Definition Performances

Sentences that belong to the definition group can be predicted over 90% accurately by all 12 Machine Learning Models, as given in Table 5.10. The best models are Bert model and Keras-RNN model, whose f1 scores and accuracies are 0.95 and 0.98, respectively. The worst model is Model 9, similar to the results given in Step 1.

Table 5.10 Step 2 Result for Label Group 2

Classification	ML Model No	Internal Test Results				Test Dataset Results			
		Precision	Recall	f1-score	Accuracy	Precision	Recall	f1-score	Accuracy
Definition/Other	1	0.90	0.91	0.91	0.97	0.88	0.90	0.89	0.96
Definition/Other	2	0.96	0.93	0.95	0.99	0.88	0.86	0.87	0.96
Definition/Other	3	0.97	0.85	0.90	0.97	0.97	0.94	0.95	0.97
Definition/Other	4	0.90	0.87	0.88	0.97	0.94	0.86	0.90	0.97
Definition/Other	5	0.93	0.89	0.91	0.97	0.96	0.85	0.90	0.97
Definition/Other	6	0.97	0.86	0.91	0.97	0.96	0.90	0.93	0.98
Definition/Other	7	0.93	0.91	0.92	0.98	0.79	0.79	0.79	0.93
Definition/Other	8	0.95	0.83	0.88	0.96	0.90	0.79	0.83	0.95
Definition/Other	9	0.76	0.60	0.64	0.90	0.70	0.62	0.65	0.91
Definition/Other	10	0.92	0.91	0.92	0.97	0.96	0.94	0.95	0.98
Definition/Other	11	0.99	0.83	0.89	0.97	0.95	0.88	0.91	0.97
Definition/Other	12	0.97	0.93	0.95	0.98	0.95	0.95	0.95	0.98

5.4.3 Classification of Obligation Performances

Sentences that were labeled as Obligation was most successfully predicted by Machine Learning Model 12 with 0.84 accuracy value and 0.84 f1 score value. The results of all 12 Machine Learning Models are given in Table 5.11. 8 models' accuracy and f1 score are below 0.80, which is the benchmark point defined in Step 1. Similar to previous results, the worst model is Model 9, with 0.66 f1-score and accuracy value.

Table 5.11 Step 2 Result for Label Group 3

Classification	ML Model No	Internal Test Results				Test Dataset Results			
		Precision	Recall	f1-score	Accuracy	Precision	Recall	f1-score	Accuracy
Obligation/Other	1	0.81	0.81	0.81	0.82	0.79	0.80	0.79	0.80
Obligation/Other	2	0.87	0.87	0.87	0.87	0.77	0.78	0.78	0.79
Obligation/Other	3	0.75	0.71	0.71	0.73	0.74	0.71	0.72	0.74
Obligation/Other	4	0.85	0.85	0.85	0.85	0.77	0.77	0.77	0.78
Obligation/Other	5	0.83	0.82	0.83	0.83	0.82	0.83	0.82	0.83
Obligation/Other	6	0.73	0.70	0.70	0.71	0.72	0.70	0.70	0.73
Obligation/Other	7	0.76	0.76	0.76	0.76	0.77	0.78	0.78	0.79
Obligation/Other	8	0.78	0.77	0.78	0.78	0.78	0.78	0.78	0.79
Obligation/Other	9	0.66	0.65	0.65	0.67	0.68	0.67	0.67	0.69
Obligation/Other	10	0.79	0.78	0.78	0.78	0.71	0.70	0.66	0.66
Obligation/Other	11	0.80	0.80	0.80	0.80	0.80	0.81	0.81	0.81
Obligation/Other	12	0.87	0.87	0.87	0.87	0.84	0.85	0.84	0.84

5.4.4 Classification of Risk and Right Performances

Individual results on Risk and Right classification are presented in Table 5.12. Sentences that were labeled as Risk or Right are predicted by Machine Learning Model 12 with 0.79 accuracy value and 0.77 f1 score value. Models 1,4,5 and 10 gave similar results to Model 12, which is the best model. However, the worst model is Model 9 again, with 0.66 accuracy and 0.52 f1 score values.

Table 5.12 Step 2 Result for Label Group 4

Classification	ML Model No	Internal Test Results				Test Dataset Results			
		Precision	Recall	f1-score	Accuracy	Precision	Recall	f1-score	Accuracy
Risk/Right	1	0.85	0.84	0.84	0.84	0.73	0.76	0.74	0.76
Risk/Right	2	0.83	0.84	0.83	0.83	0.72	0.76	0.72	0.73
Risk/Right	3	0.78	0.69	0.69	0.74	0.66	0.61	0.61	0.71
Risk/Right	4	0.80	0.80	0.80	0.81	0.73	0.76	0.74	0.76
Risk/Right	5	0.88	0.88	0.88	0.89	0.76	0.79	0.77	0.78
Risk/Right	6	0.67	0.64	0.64	0.69	0.65	0.61	0.61	0.70
Risk/Right	7	0.75	0.74	0.74	0.76	0.69	0.71	0.69	0.71
Risk/Right	8	0.73	0.73	0.73	0.75	0.71	0.73	0.72	0.74
Risk/Right	9	0.63	0.55	0.52	0.65	0.57	0.53	0.52	0.66
Risk/Right	10	0.73	0.69	0.70	0.73	0.74	0.71	0.72	0.77
Risk/Right	11	0.74	0.74	0.74	0.74	0.69	0.71	0.67	0.68
Risk/Right	12	0.80	0.78	0.79	0.81	0.76	0.78	0.77	0.79

5.5 Complete Results After Process Improvements

The output of the process given in Figure 5.4 is the combination of classification obtained for all label groups whose performance results are presented in Section 5.4. A new column is created in Pandas Data Frame (The Pandas Development Team, 2020) and named as Complete Result. The process of Complete Result Column creation is given in Figure 5.6. If ML model classification is Heading in the A1 step, it is copied directly to Complete Result Column. If the classification is Clause, the result of the A2 step is checked. If the result of the A2 step is Definition, the classification is copied to Complete Result Column. If the classification in the A2 step is Other, the result of the A3 step is checked. If it is Obligation, Complete Result Column filled as Obligation. If the A3 result is also Other classification in the A4 step is copied to Complete Result Column.

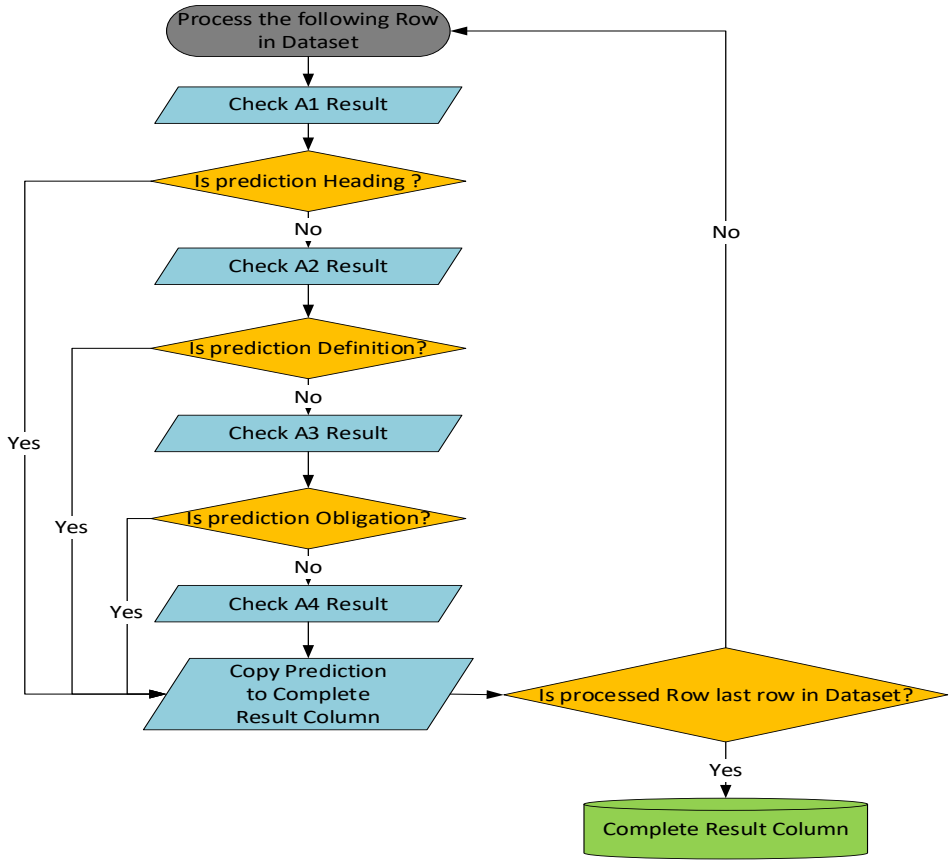

Figure 5.6. Complete Result Creation Logic in Step 2

Machine Learning Model performance values according to precision, recall, f1 score, and accuracy are presented in Table 5.13. Model 1, Model 5, Model 10, Model 11, and Model 12 accuracy values are greater than 0.80. Their f1 scores are also very close to or higher than the best model determined in Step 1. The model that has the worst performance is Model 9, which was developed by using the Decision Tree algorithm and Spacy word embedding.

Comparisons of Step 1 and Step 2 results are presented in Figure 5.8 in terms of accuracy and Figure 5.9 in terms of f1 score. Results show that modification in the process increases the performance of all 12 models; however, the best model is still Model 12. Accuracy in the best model is increased to 0.87 from 0.82. Similarly, f1 score increased by 0.04 points and reached to 0.83. As shown in the comparison figures, the most significant improvements are obtained in Model 10 and Model 11. Accuracy values and f1 scores are increased by more than 10% and 15%, respectively.

The best 5 models' performances are very close to each other, and they are considered to be usable in developing a voting classifier to increase accuracy and f1 score. Details of voting classifiers are presented in the following sections.

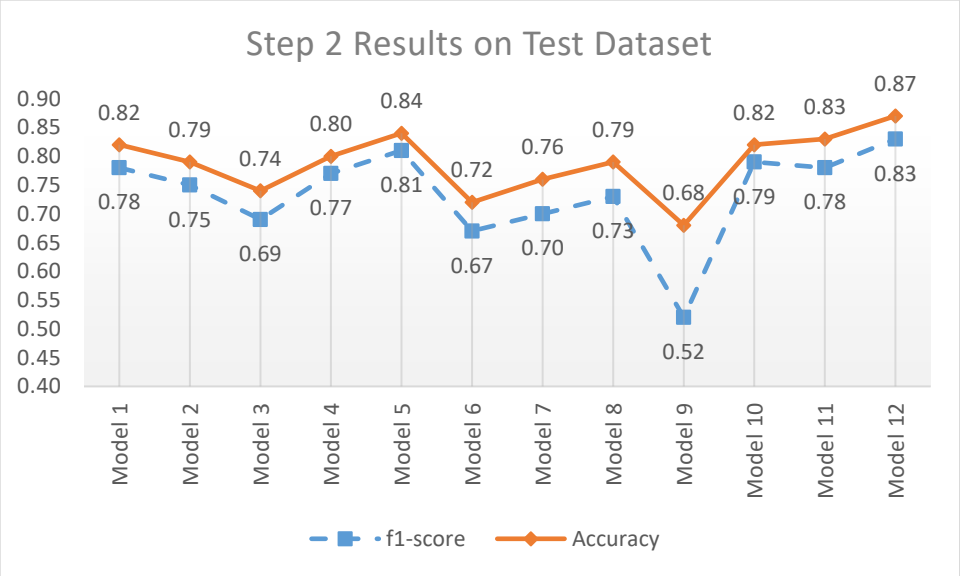

Figure 5.7. Models' Results Comparison for Sentence Type Classification in Step 2

Table 5.13 Step 2 Combined Performances of Machine Learning Models

				Test Dataset Results			
Classification	**ML Model No**	**Text Vectorization**	**ML Algorithm**	**Precision**	**Recall**	**f1-score**	**Accuracy**
Heading/Definition/Obligation/Risk/Right	1	BAG OF WORDS	Logistic Regression	0.77	0.79	0.78	0.82
Heading/Definition/Obligation/Risk/Right	2	BAG OF WORDS	Support Vector Machine	0.75	0.77	0.75	0.79
Heading/Definition/Obligation/Risk/Right	3	BAG OF WORDS	Decision Tree	0.72	0.68	0.69	0.74
Heading/Definition/Obligation/Risk/Right	4	TFIDF	Logistic Regression	0.78	0.77	0.77	0.80
Heading/Definition/Obligation/Risk/Right	5	TFIDF	Support Vector Machine	0.83	0.81	0.81	0.84
Heading/Definition/Obligation/Risk/Right	6	TFIDF	Decision Tree	0.70	0.66	0.67	0.72
Heading/Definition/Obligation/Risk/Right	7	Spacy	Logistic Regression	0.70	0.71	0.70	0.76
Heading/Definition/Obligation/Risk/Right	8	Spacy	Support Vector Machine	0.75	0.72	0.73	0.79
Heading/Definition/Obligation/Risk/Right	9	Spacy	Decision Tree	0.61	0.52	0.52	0.68
Heading/Definition/Obligation/Risk/Right	10	Keras Embedding	RNN	0.80	0.79	0.79	0.82
Heading/Definition/Obligation/Risk/Right	11	Glove Embedding	RNN	0.80	0.78	0.78	0.83
Heading/Definition/Obligation/Risk/Right	12	BERT Embedding	BERT	0.83	0.85	0.83	0.87

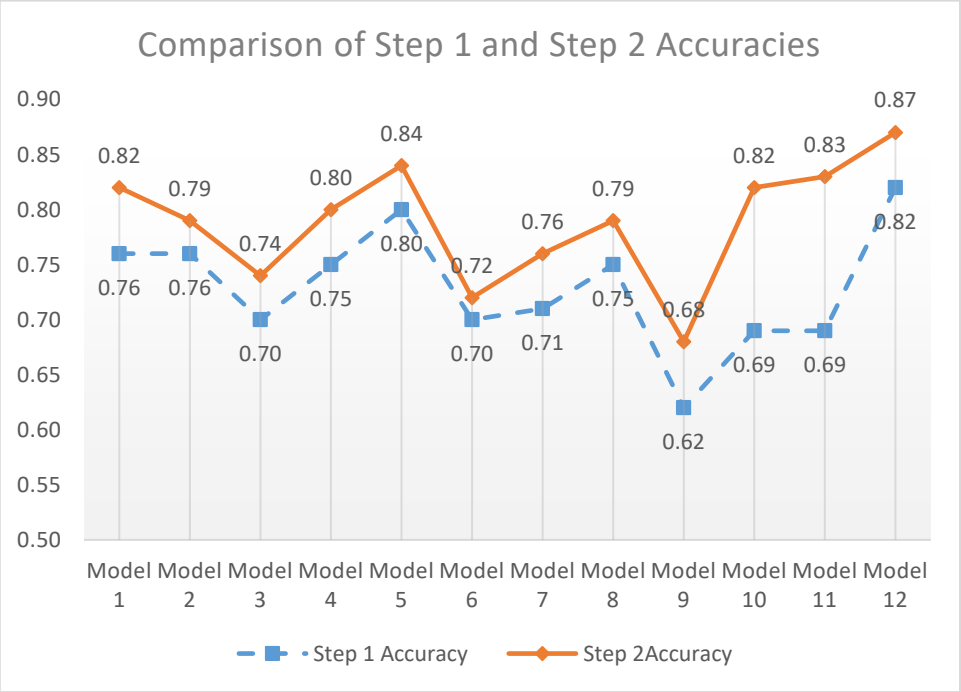

Figure 5.8. Comparison of Step 1 and Step 2 Accuracies

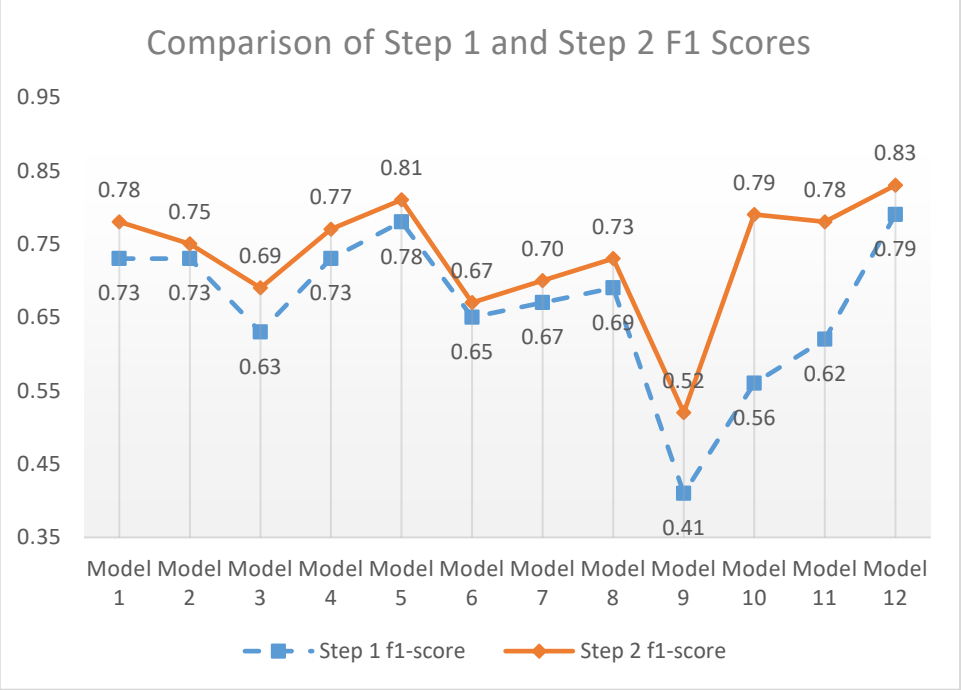

Figure 5.9. Comparison of Step 1 and Step 2 F1 Scores

5.6 Step 3 Appling Ensemble Method by Using Competitive Voting Classifier Model

Machine learning models can be combined to improve the performance of classification. The approach is named the ensemble method, and voting is the most popular and essential ensemble method (Zhou, 2012). Voting classifiers such as majority voting or weighted voting are developed for binary classification. In this research, clasification are tried to be made in multi-class labels; therefore voting algorithm needs to be modified. Zhang et al. (2020) propose Competitive Voting for multi-class classification. Their approach is based on combining the results of the top three machine learning models to increase the performance of classification.

In Competitive Voting;

- If all three models predict the same class, the final classification will be the same with all three
- If two of the models predict the same, but one of them predicts a different class, most predicted class is taken as the final classification,
- If all three predict different classes, the final classification is taken from the model that has the highest accuracy

The logic of competitive voting is presented in Figure 5.10. In the work of Zhang et al. (2020) competitive voting algorithm was implemented to solve 3 different problems with different class sizes. According to the result derived from their research, which is presented in Table 5.14, competitive voting increases the accuracy in all three classification problems.

Table 5.14 Competitive Performance in Work of Zhang et al. (2020)

	SVM	*RF*	*BP*	*KNN*	*SIMCA*	*Competitive Voting*
Four Class	90.35	88.58	77.62	79.10	81.60	**92.5**
Three Class	88.66	87.87	84.73	78.15	85.10	**95.04**
Two Class	97.95	96.90	97.20	90.69	97.38	**98.47**

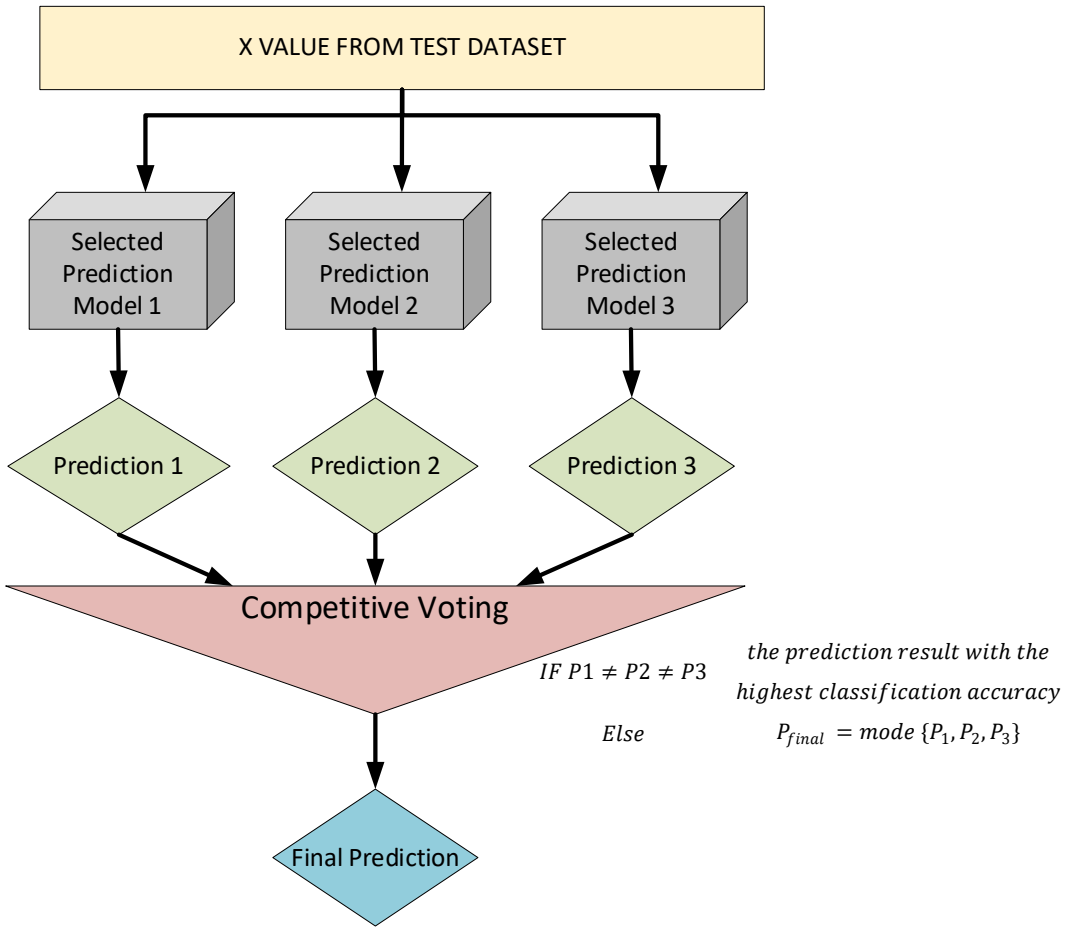

Figure 5.10. Competitive Voting

Related party classification performances obtained in Step 1 and type of sentence classification performances obtained in Step 2 are presented in Table 5.15. The best 3 models for both classifications are Model 5, Model 11, and Model 12. The logic presented in Figure 5.10 is implemented by using these three models.

Model 12 accuracies are 0.8 and 0.87 for related party classification and sentence type classification. Model 5 is the second best model in terms of accuracy and f1 score. The accuracy performances of model 5 are 0.70 and 0.84, respectively. The third best model is Model 11, with 0.73 and 0.83 accuracies. As mentioned, the best model is Model 12 in both classifications. Model 12 results are used as the final classification, as explained in the logic of competitive voting when all three models predict different classes.

Table 5.15 Related Party and Type of Sentence Classification Performances Before Implementing Competitive Voting

ML Model No		Text Vectorization	ML Algorithm	Test Dataset Results			
				Precision	Recall	f1-score	Accuracy
Shared/Employer/Contractor	1	BAG OF WORDS	Logistic Regression	0.66	0.68	0.67	0.71
	2	BAG OF WORDS	Support Vector Machine	0.64	0.66	0.64	0.69
	3	BAG OF WORDS	Decision Tree	0.66	0.68	0.67	0.72
	4	TFIDF	Logistic Regression	0.65	0.68	0.66	0.72
	5	TFIDF	Support Vector Machine	0.72	0.67	0.69	0.77
	6	TFIDF	Decision Tree	0.61	0.45	0.45	0.70
	7	Spacy	Logistic Regression	0.58	0.61	0.59	0.66
	8	Spacy	Support Vector Machine	0.63	0.60	0.61	0.70
	9	Spacy	Decision Tree	0.55	0.47	0.49	0.61
	10	Keras Embedding	RNN	0.64	0.67	0.65	0.72
	11	Glove Embedding	RNN	0.72	0.68	0.67	0.73
	12	Word Embedding	BERT	0.79	0.69	0.73	0.80
Heading/Definition/Obligation/Risk/Right	1	BAG OF WORDS	Logistic Regression	0.77	0.79	0.78	0.82
	2	BAG OF WORDS	Support Vector Machine	0.75	0.77	0.75	0.79
	3	BAG OF WORDS	Decision Tree	0.72	0.68	0.69	0.74
	4	TFIDF	Logistic Regression	0.78	0.77	0.77	0.80
	5	TFIDF	Support Vector Machine	0.83	0.81	0.81	0.84
	6	TFIDF	Decision Tree	0.70	0.66	0.67	0.72
	7	Spacy	Logistic Regression	0.70	0.71	0.70	0.76
	8	Spacy	Support Vector Machine	0.75	0.72	0.73	0.79
	9	Spacy	Decision Tree	0.61	0.52	0.52	0.68
	10	Keras Embedding	RNN	0.80	0.79	0.79	0.82
	11	Glove Embedding	RNN	0.80	0.78	0.78	0.83
	12	BERT Embedding	BERT	0.83	0.85	0.83	0.87

5.6.1 Result of Related Party Classification with Competitive Voting

Classification performance on Related Party (Shared/Employer/Contractor) is increased when competitive voting is implemented. As shown in Table 5.16 accuracy

value increased to 0.83, and f1 score increased to 0.76. Both performance metrics increased by 3% compared to the best model. At the end of Step 3 of this research, ownership of the risk, right and obligation can be predicted as 83% accurate.

Table 5.16 Used Models and Competitive Voting Performance on Related Party

Shared/Employer/Contractor			Test Dataset Results			
ML Model No	Text Vectorization	ML Algorithm	Precision	Recall	f1-score	Accuracy
5	TFIDF	Support Vector Machine	0.72	0.67	0.69	0.77
11	Glove Embedding	RNN	0.72	0.68	0.67	0.73
12	Word Embedding	BERT	0.79	0.69	0.73	0.80
13	Competitive	Voting	0.80	0.74	0.76	0.83

5.6.2 Result of Sentence Type Classification with Competitive Voting

Like related party result, classification performance on sentence type is increased when competitive voting is implemented. As shown in Table 5.17 accuracy value increased to 0.89, and f1 score increased to 0.86. The accuracy metric increased by 2% and f1 score metric increased by 3% compared to the best model. At the end of Step 3 of this research sentence that exists in a contract can be categorized as heading, definition obligation, risk, and right with 89% accuracy.

Table 5.17 Used Models and Competitive Voting Performance on Sentence Type

Heading/Definition/Obligation/Risk/Right			Test Dataset Results			
ML Model No	Text Vectorization	ML Algorithm	Precision	Recall	f1-score	Accuracy
5	TFIDF	Support Vector Machine	0.83	0.81	0.81	0.84
11	Glove Embedding	RNN	0.80	0.78	0.78	0.83
12	BERT Embedding	BERT	0.83	0.85	0.83	0.87
13	Competitive	Voting	0.87	0.85	0.86	0.89

5.7 Comparison of Best Results on Steps

As presented in this chapter, 12 ML models were developed to classify sentences in a contract regarding their type and related parties. Classification performance for sentence type is increased in 3 steps. In the first step, the machine learning models were directly trained with a multi-label dataset and tested with Test Dataset. In the second step, the multi-class classification problem was converted to a binary classification problem with provided logic, and the classification accuracy increased by 5%. In Step 3, Competitive Voting logic was used to combine the 3 best models obtained in Step 2. The accuracy in Step 3 was increased to 0.89, which is 2% more when compared to the Step 2 result and 7% more when compared to the Step 1 result. Results obtained in each step are presented in Table 5.18.

Table 5.18 Sentence Type Classification Performance Results of Step 1, Step 2, and Step 3

Heading/Definition/Obligation/Risk/Right				**Test Dataset Results**			
Step	**ML Model No**	**Text Vectorization**	**ML Algorithm**	**Precision**	**Recall**	**f1-score**	**Accuracy**
1	12	BERT Embedding	BERT	0.79	0.80	0.79	0.82
2	12	BERT Embedding	BERT	0.83	0.85	0.83	0.87
3	13	Competitive	Voting	0.87	0.85	0.86	0.89

To visualize the improvement in the classification performance of sentence type, the comparison of obtained results in each step is presented in Figure 5.11.

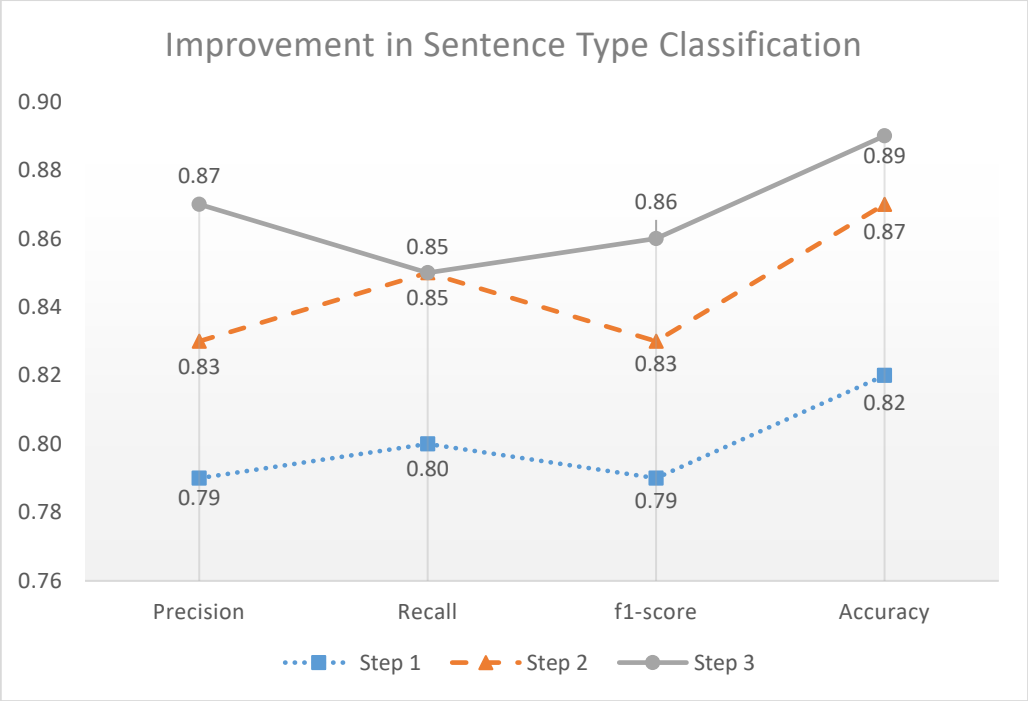

Figure 5.11. Visualization of Sentence Type Classification Improvements in Each Step

Classification performance for the related party is increased in Step 3 when compared to Step 1. In the first step, machine learning models were directly trained with multi-label datasets and tested with Test Dataset. In the third step of this research, Competitive Voting logic was used to combine the 3 best models obtained in Step 1 for related party classification. The accuracy value was increased to 0.83, which is 3% more when compared to the Step 1 result. Results obtained in each step are presented in Table 5.19.

Table 5.19 Related Party Classification Performance Results of Step 1 and Step 3

Shared/Employer/Contractor				**Test Dataset Results**			
Step	**ML Model No**	**Text Vectorization**	**ML Algorithm**	**Precision**	**Recall**	**f1-score**	**Accuracy**
1	12	BERT Embedding	BERT	0.79	0.69	0.73	0.80
3	13	Competitive	Voting	0.80	0.74	0.76	0.83

Improvement in the performance is visualized by providing related party classification precision, recall, f1 score, and accuracy values in Figure 5.12.

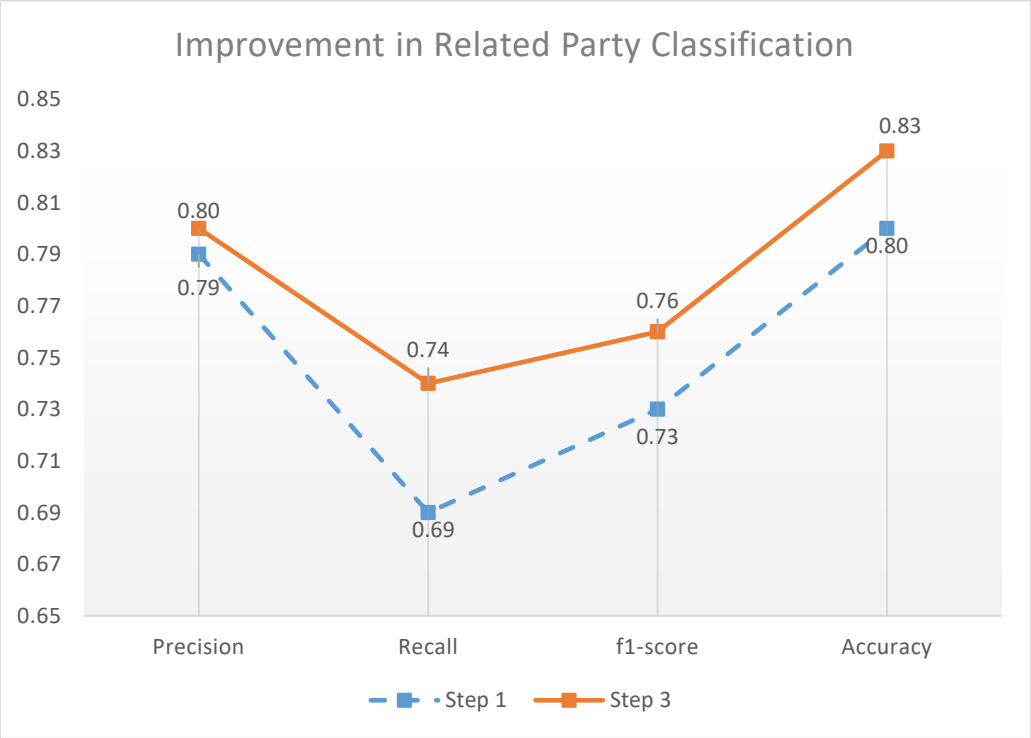

Figure 5.12. Visualization of Related Party Classification Improvements in Step 1 and Step 3

CHAPTER 6

CONCLUSIONS

This study investigates the possibility of the development of automated contract review in the construction sector to determine the risks, rights, and obligations assigned to contracting parties. Construction contracts are text-based documents, and contract makers may change contractual terms for their own sake. It is obvious that construction contract documents need to be analyzed before deciding to get a job and preparing proposals. Manual analysis of construction contract documents is time-consuming, costly, and error-prone. Recent advancements in natural language processing and machine learning have the potential to convert manual document analysis process that is currently employed in the sector to automated computer-based systems.

The literature review on NLP-based text analytics shows that ML-oriented construction contract review is not thoroughly investigated, and more research on this topic is needed to determine limitations as well as benefits that can be achieved from the topic.

Literature reviews on natural language processing and machine learning have directed this research focus to develop various machine learning models to compare classification performances. Latest advancements in natural language processing-based text analytics and deep learning algorithms were used in the development steps of models.

Since no construction contract dataset exists that can be used in a supervised machine learning training process, FIDIC books were analyzed and labeled to create datasets. Created datasets have been used to train and test selected machine-learning models. The results of the presented steps are found promising. The major findings taken in

this study and recommendations for further research together with the limitations of this research, are given in the following sections.

6.1 Major Findings

Although studies in the literature are focusing on mainly rule-based text classification approaches or machine learning approaches on a limited scope, this study shows that the implementation of current advancements in natural language processing and machine learning has the potential to develop a high-performing machine learning-based automated contract review model. Achieving 0.89 accuracy and 0.86 f1 scores with a relatively small training dataset for such a broad classification problem is very promising.

BERT algorithm that gives the best results in individual models proves that pre-trained models based on large datasets to create a model, which is focused on the classification of narrow scope, improve the classification performance. For computers to understand natural language like humans need large datasets; however, creating large enough datasets is not possible for each problem due to the unavailability of input in a domain. Pre-trained models provide an opportunity to combine domain-free extensive dataset information with domain-specific information to solve problems.

This study also shows that combining classifications made by various algorithms is important to get better results, like communities consider various ideas when making decisions. The competitive voting ensemble method, which is implemented in this research, increases classification performance by 2% without any improvement in the training dataset or algorithms. Research in the construction domain also needs to give importance to taking advantage of ensemble methods to create successful models, like importance given to comparing different types of methods and algorithms.

The current performance of the classification model shows that the automated construction contract review model is not ideal for considering as the sole method during the bidding stage by removing human-based analysis since the significance of contract review in the bid or not to bid decisions. However, even with the current performance of the proposed model, the outcome of this study provides valuable information that can be used in construction contract review, which is expected to decrease required time and errors due to overlooking.

This study provides a new approach for contractors to review construction contracts which will reduce employee workload and increase the quality of work in risk assessment at the bidding stage. To decide on risk premium, contractors can use the proposed approach to classify contract text in a short time in terms of risk inherent in the bid preparation step.

6.2 Limitations and Recommendations for Further Research

Dataset developed in this research is limited to FIDIC books. So developed classification model is tested through the actual construction project contract prepared based on FIDIC. Performance on construction contracts that are prepared based on different drafts was not investigated. In order to create a generalized classificaiton model, the dataset is needed to be expanded with different types of standard forms of contracts in forthcoming studies.

12 ML models based on 5 machine learning algorithms and 6 vectorization methods were trained in the scope of this research; however, it must be noted that other alternatives need to be evaluated for both algorithm and vectorization sides for further research.

The study presented in this dissertation directly focuses on using the supervised machine learning method to classify contract text; however, it is known that the rule-based approach also provides promising results in well-defined texts, like contracts. It is considered that integrating a rule-based approach to the proposed model has the

potential to increase classification performance for some cases, and it needs to be investigated in detail.

Another point that is not included in this research is ambiguity in natural language. As presented in the literature review, ambiguity detection is also an important topic that various researchers focus on. Since the focused text data is FIDIC standard form of contracts, which is well defined in terms of responsibilities, in this research, it is considered that ambiguity is not an important concern point for this research. However, it is known that further research that focuses on generalizing the classification model needs to consider ambiguity in natural language. A parallel module must be integrated to define ambiguous sentences before being classified with machine learning models.

It is also known that the usability of the classification model in the construction sector depends on the appropriateness of labels in the training dataset. As presented, the dataset is not publicly available, and it is developed by the researcher according to personal risk perception. Dataset was validated through expert meetings by considering 10% of the datasets; however, to be used and get the full benefit of automated contract review in a construction company, it needs to be reviewed according to company risk perception.

REFERENCES

Akintoye, A. S., & MacLeod, M. J. (1997). Risk analysis and management in construction. *International Journal of Project Management*, *15*(1), 31–38. https://doi.org/10.1016/S0263-7863(96)00035-X

Al Qady, M., & Kandil, A. (2010). Concept Relation Extraction from Construction Documents Using Natural Language Processing. *Journal of Construction Engineering and Management*, *136*(3), 294–302. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000131

Al Qady, M., & Kandil, A. (2009). Techniques for Evaluating Automated Knowledge Acquisition from Contract Documents. *Construction Research Congress 2009*, 1479–1488. https://doi.org/10.1061/41020(339)150

Arora, C., Sabetzadeh, M., Briand, L., & Zimmer, F. (2015). Automated Checking of Conformance to Requirements Templates Using Natural Language Processing. *IEEE Transactions on Software Engineering*, *41*(10), 944–968. https://doi.org/10.1109/TSE.2015.2428709

BERT (language model). (2022). In *en.wikipedia.org*. https://en.wikipedia.org/wiki/BERT_(language_model)

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching Word Vectors with Subword Information. *ArXiv Preprint ArXiv:1607.04606*.

Candaş, A. B., & Tokdemir, O. B. (2022a). Automated Identification of Vagueness in the FIDIC Silver Book Conditions of Contract. *Journal of Construction Engineering and Management*, *148*(4). https://doi.org/10.1061/(ASCE)CO.1943-7862.0002254

Candaş, A. B., & Tokdemir, O. B. (2022b). Automating Coordination Efforts for Reviewing Construction Contracts with Multilabel Text Classification. *Journal of Construction Engineering and Management*, *148*(6).

https://doi.org/10.1061/(ASCE)CO.1943-7862.0002275

Chakrabarti, D., Patodia, N., Bhattacharya, U., Mitra, I., Roy, S., Mandi, J., Roy, N., & Nandy, P. (2018). Use of Artificial Intelligence to Analyse Risk in Legal Documents for a Better Decision Support. *TENCON 2018 - 2018 IEEE Region 10 Conference*, 0683–0688. https://doi.org/10.1109/TENCON.2018.8650382

Chalkidis, I., & Androutsopoulos, I. (2017). A Deep Learning Approach to Contract Element Extraction. In A. Wyner & G. Casini (Eds.), *Legal Knowledge and Information Systems* (pp. 155–164). IOS Pres. https://doi.org/10.3233/978-1-61499-838-9-155

Chalkidis, I., Androutsopoulos, I., & Michos, A. (2017). Extracting contract elements. *Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law - ICAIL '17*, 19–28. https://doi.org/10.1145/3086512.3086515

Chen, H., Chiang, R. H. L., & Storey, V. C. (2012). Business intelligence and analytics: From big data to big impact. *MIS Quarterly: Management Information Systems*, *36*(4), 1165–1188. https://misq.org/skin/frontend/default/misq/pdf/EdComments/V36I4_si_cheni ntroduction.pdf

Chollet, F., & others. (2015). *Keras*. GitHub.

Crossley, S. A. (2013). Advancing research in second language writing through computational tools and machine learning techniques: A research agenda. *Language Teaching*, *46*(2), 256–271. https://doi.org/10.1017/S0261444812000547

Crossley, S. A., Allen, L. K., Kyle, K., & McNamara, D. S. (2014). Analyzing Discourse Processing Using a Simple Natural Language Processing Tool. *Discourse Processes*, *51*(5–6), 511–534. https://doi.org/10.1080/0163853X.2014.910723

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North*, 4171–4186. https://doi.org/10.18653/v1/N19-1423

Dikmen, I., Birgonul, M. T., & Han, S. (2007). Using fuzzy risk assessment to rate cost overrun risk in international construction projects. *International Journal of Project Management*, *25*(5), 494–505. https://doi.org/10.1016/j.ijproman.2006.12.002

Dikmen, I., Budayan, C., Talat Birgonul, M., & Hayat, E. (2018). Effects of Risk Attitude and Controllability Assumption on Risk Ratings: Observational Study on International Construction Project Risk Assessment. *Journal of Management in Engineering*, *34*(6), 04018037. https://doi.org/10.1061/(ASCE)ME.1943-5479.0000643

Eastman, C., Lee, J., Jeong, Y., & Lee, J. (2009). Automatic rule-based checking of building designs. *Automation in Construction*, *18*(8), 1011–1033. https://doi.org/10.1016/j.autcon.2009.07.002

El-Sayegh, S. M. (2008). Risk assessment and allocation in the UAE construction industry. *International Journal of Project Management*, *26*(4), 431–438. https://doi.org/10.1016/j.ijproman.2007.07.004

Explosion. (2022). *en_core_web_lg-3.4.0*. https://github.com/explosion/spacy-models/releases/tag/en_core_web_lg-3.4.0

Fazlic, L. B., Hallawa, A., Schmeink, A., Peine, A., Martin, L., & Dartmann, G. (2019). A Novel NLP-FUZZY System Prototype for Information Extraction from Medical Guidelines. *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1025–1030. https://doi.org/10.23919/MIPRO.2019.8756929

FIDIC® Conditions of Contract for Construction for Building and Engineering Works Designed by the Employer (Second). (2017). FIDIC (International

Federation of Consulting Engineers).

Galser, I., Scepankova, E., & Matthes, F. (2018). Classifying Semantic Types of Legal Sentences: Portability of Machine Learning Models. In M. Palmirani (Ed.), *Legal Knowledge and Information Systems* (pp. 61–70). IOS Press. https://doi.org/10.3233/978-1-61499-935-5-61

General Architecture for Text Engineering. (n.d.). Wikipedia. Retrieved November 8, 2019, from en.wikipedia.org/wiki/General_Architecture_for_Text_Engineering

Grant, S., Kline, J. J., & Quiggin, J. (2012). Differential awareness, ambiguity, and incomplete contracts: A model of contractual disputes. *Journal of Economic Behavior & Organization*, *82*(2–3), 494–504. https://doi.org/10.1016/j.jebo.2012.02.021

Grant, S., Kline, J. J., & Quiggin, J. (2014). A matter of interpretation: Ambiguous contracts and liquidated damages. *Games and Economic Behavior*, *85*, 180–187. https://doi.org/10.1016/j.geb.2014.01.019

Hayati, K., Latief, Y., & Rarasati, A. D. (2019). Causes and Problem Identification in Construction Claim Management. *IOP Conference Series: Materials Science and Engineering*, *469*, 012082. https://doi.org/10.1088/1757-899X/469/1/012082

Honnibal, M., Montani, I., Van Landeghem, S., & Boyd, A. (2020). *spaCy: Industrial-strength Natural Language Processing in Python* (3.4). https://spacy.io/

Huertas, C., & Juárez-Ramírez, R. (2012). NLARE, a natural language processing tool for automatic requirements evaluation. *Proceedings of the CUBE International Information Technology Conference on - CUBE '12*, 371. https://doi.org/10.1145/2381716.2381786

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and

prospects. *Science*, *349*(6245), 255–260. https://doi.org/10.1126/science.aaa8415

Kim, Y., Lee, J., Lee, E.-B., & Lee, J.-H. (2020). Application of Natural Language Processing (NLP) and Text-Mining of Big-Data to Engineering-Procurement-Construction (EPC) Bid and Contract Documents. *2020 6th Conference on Data Science and Machine Learning Applications (CDMA)*, 123–128. https://doi.org/10.1109/CDMA47397.2020.00027

Klein, D., Manning, C., Levy, R., Manning, C., Grenager, T., Andrew, G., Marneffe, M.-C. de, MacCartney, B., Rafferty, A., Green, S., Tseng, H., Chang, P.-C., Maier, W., & Finkel, J. (n.d.). *The Stanford Parser: A statistical parser*. Stanford NLP Group. Retrieved November 7, 2019, from nlp.stanford.edu/software/lex-parser.shtml#About

Kotu, V., & Deshpande, B. (2015). Text Mining. In *Predictive Analytics and Data Mining* (pp. 275–303). Elsevier. https://doi.org/10.1016/B978-0-12-801460-8.00009-4

Landthaler, J., Waltl, B., Holl, P., & Matthes, F. (2016). Extending Full Text Search for Legal Document Collections using Word Embeddings. In F. Bex & S. Villata (Eds.), *Legal Knowledge and Information Systems* (pp. 73–82). IOS Press. https://doi.org/10.3233/978-1-61499-726-9-73

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lee, J., Ham, Y., Yi, J.-S., & Son, J. (2020). Effective Risk Positioning through Automated Identification of Missing Contract Conditions from the Contractor's Perspective Based on FIDIC Contract Cases. *Journal of Management in Engineering*, *36*(3). https://doi.org/10.1061/(ASCE)ME.1943-5479.0000757

Lee, J., Yi, J.-S., & Son, J. (2019). Development of Automatic-Extraction Model of Poisonous Clauses in International Construction Contracts Using Rule-Based NLP. *Journal of Computing in Civil Engineering*, *33*(3), 04019003.

https://doi.org/10.1061/(ASCE)CP.1943-5487.0000807

Lexical Computing. (n.d.). *POS tags*. Sketch Engine. Retrieved November 7, 2019, from https://www.sketchengine.eu/pos-tags/

Li, W., Zhang, X., Niu, C., Jiang, Y., & Srihari, R. (2003). An Expert Lexicon Approach to Identifying English Phrasal Verbs. *41st Annual Meeting of the Association for Computational Linguistics, 7-12 July*, 513–520.

Machine Learning. (n.d.). Wikipedia.

Mavin, A., Wilkinson, P., Harwood, A., & Novak, M. (2009). Easy Approach to Requirements Syntax (EARS). *2009 17th IEEE International Requirements Engineering Conference*, 317–322. https://doi.org/10.1109/RE.2009.9

Mendis, D., Hewage, K. N., & Wrzesniewski, J. (2013). Reduction of construction wastes by improving construction contract management: a multinational evaluation. *Waste Management & Research*, *31*(10), 1062–1069. https://doi.org/10.1177/0734242X13495724

Mendis, D., Hewage, K. N., & Wrzesniewski, J. (2015). Contractual obligations analysis for construction waste management in Canada. *Journal of Civil Engineering and Management*, *21*(7), 866–880. https://doi.org/10.3846/13923730.2014.893907

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*. arXiv. https://doi.org/10.48550/ARXIV.1301.3781

Mok, W. Y., & Mok, J. R. (2019). Classification of Breach of Contract Court Decision Sentences. In K. D. Ashley, K. Atkinson, L. K. Branting, E. Francescon, M. Grabmair, B. Waltl, V. R. Walker, & A. Z. Wyner (Eds.), *Proceedings of the ThirdWorkshop on Automated Semantic Analysis of Information in Legal Text, June 21* (p. p7). http://ceur-ws.org/Vol-2385/paper7.pdf

Moon, S., Shin, Y., Hwang, B.-G., & Chi, S. (2018). Document Management System Using Text Mining for Information Acquisition of International Construction. *KSCE Journal of Civil Engineering*, *22*(12), 4791–4798. https://doi.org/10.1007/s12205-018-1528-y

Noam, C. (1973). Language and Freedom. In *Reasons of State* (pp. 387–408). Pantheon Books.

Nuseibeh, B., & Easterbrook, S. (2000). Requirements engineering: a roadmap. *Proceedings of the Conference on The Future of Software Engineering - ICSE '00*, 35–46. https://doi.org/10.1145/336512.336523

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in {P}ython. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. https://doi.org/10.3115/v1/D14-1162

Pohl, K., & Rupp, C. (2011). *Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam - Foundation Level - IREB Compliant* (1st ed.). Rocky Nook.

Qader, W. A., Ameen, M. M., & Ahmed, B. I. (2019). An Overview of Bag of Words;Importance, Implementation, Applications, and Challenges. *2019 International Engineering Conference (IEC)*, 200–204. https://doi.org/10.1109/IEC47844.2019.8950616

Rameezdeen, R., & Rodrigo, A. (2014). Modifications to standard forms of contract: the impact on readability. *Construction Economics and Building*, *14*(2), 31–40. https://doi.org/10.5130/AJCEB.v14i2.3778

Riloff, E., & Phillips, W. (2004). *An introduction to the Sundance and AutoSlog system.* http://www.cs.utah.edu/~riloff/pdfs/officialsundance-%0Atr.pdf

Robeer, M., Lucassen, G., van der Werf, J. M. E. M., Dalpiaz, F., & Brinkkemper, S. (2016). Automated Extraction of Conceptual Models from User Stories via NLP. *2016 IEEE 24th International Requirements Engineering Conference (RE)*, 196–205. https://doi.org/10.1109/RE.2016.40

Rosadini, B., Ferrari, A., Gori, G., Fantechi, A., Gnesi, S., Trotta, I., & Bacherini, S. (2017). Using NLP to Detect Requirements Defects: An Industrial Experience in the Railway Domain. In P. Grünbacher & A. Perini (Eds.), *Requirements Engineering: Foundation for Software Quality* (pp. 344–360). Springer International Publishing.

Salama, D. M., & El-Gohary, N. M. (2016). Semantic Text Classification for Supporting Automated Compliance Checking in Construction. *Journal of Computing in Civil Engineering*, *30*(1), 04014106. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000301

Segel, E. (2016). Watson and the Jeopardy! Challenge. In *Predictive Analytics : The Power to Predict Who Will Click, Buy, Lie, or Die, Revised and Updated* (pp. 206–249). John Wiley & Sons, Inc. https://doi.org/10.1002/9781119172536.ch06

Shinyama, Y. (2019). *pdfminer* (No. 20191125). https://pypi.org/project/pdfminer/

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Soysal, E., Cicekli, I., & Baykal, N. (2010). Design and evaluation of an ontology based information extraction system for radiological reports. *Computers in Biology and Medicine*, *40*(11–12), 900–911. https://doi.org/10.1016/j.compbiomed.2010.10.002

Steven Bird, Ewan Klein, E. L. (2009). *Natural Language Processing with Python*. O'Reilly Media, Inc.

Taha, A. A., & Hanbury, A. (2015). Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC Medical Imaging*, *15*(1), 29. https://doi.org/10.1186/s12880-015-0068-x

Tan, X., Hammad, A., & Fazio, P. (2010). Automated Code Compliance Checking for Building Envelope Design. *Journal of Computing in Civil Engineering*, *24*(2), 203–211. https://doi.org/10.1061/(ASCE)0887-3801(2010)24:2(203)

Taroun, A. (2014). Towards a better modelling and assessment of construction risk: Insights from a literature review. *International Journal of Project Management*, *32*(1), 101–115. https://doi.org/10.1016/j.ijproman.2013.03.004

The Pandas Development Team. (2020). *pandas-dev/pandas: Pandas* (latest). Zenodo. https://doi.org/10.5281/zenodo.3509134

Toutanova, K., & Manning, C. D. (2000). Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics* -, *13*, 63–70. https://doi.org/10.3115/1117794.1117802

Word embedding. (n.d.). Wikipedia. Retrieved November 10, 2019, from https://en.wikipedia.org/wiki/Word_embedding

Yang, H., de Roeck, A., Willis, A., & Nuseibeh, B. (2010). A Methodology for Automatic Identification of Nocuous Ambiguity. *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, 1218–1226. https://www.aclweb.org/anthology/C10-1137

Yang, H., Willis, A., De Roeck, A., & Nuseibeh, B. (2010). Automatic detection of nocuous coordination ambiguities in natural language requirements.

Proceedings of the IEEE/ACM International Conference on Automated Software Engineering - ASE '10, 53. https://doi.org/10.1145/1858996.1859007

Zait, F., & Zarour, N. (2018). Addressing Lexical and Semantic Ambiguity in Natural Language Requirements. *2018 Fifth International Symposium on Innovation in Information and Communication Technology (ISIICT)*, 1–7. https://doi.org/10.1109/ISIICT.2018.8613726

Zave, P. (1997). Classification of research efforts in requirements engineering. *ACM Computing Surveys*, *29*(4), 315–321. https://doi.org/10.1145/267580.267581

Zeng, J., An, M., & Smith, N. J. (2007). Application of a fuzzy based decision making methodology to construction project risk assessment. *International Journal of Project Management*, *25*(6), 589–600. https://doi.org/10.1016/j.ijproman.2007.02.006

Zhang, J., & El-Gohary, N. M. (2016). Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking. *Journal of Computing in Civil Engineering*, *30*(2), 04015014. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000346

Zhang, Z., Liu, Y., Hu, Q., Zhang, Z., & Liu, Y. (2020). Competitive Voting-based Multi-class Prediction for Ore Selection. *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, 514–519. https://doi.org/10.1109/CASE48305.2020.9217017

Zhou, Z.-H. (2012). Combination Methods. In *Ensemble Methods* (pp. 67–97). Chapman and Hall/CRC. https://doi.org/10.1201/b12207

CURRICULUM VITAE

Surname, Name: Eken, Görkem

EDUCATION

Degree	Institution	Year of Graduation
MS	METU Civil Engineering	2017
BS	METU Civil Engineering	2014
High School	Kocaeli Körfez Fen Lisesi, Kocaeli	2009

FOREIGN LANGUAGES

Advanced English

PUBLICATIONS

1. Eken, G., Bilgin, G., Dikmen, I., & Birgonul, M. T. (2020). A lessons-learned tool for organizational learning in construction. Automation in Construction, 110, 102977. https://doi.org/10.1016/j.autcon.2019.102977

2. Bilgin, G., Eken, G., Ozyurt, B., Dikmen, I., Birgonul, M. T., & Ozorhon, B. (2019). Validation of Support Tools for Project Management: Case of COPPMAN. Proceedings of the Creative Construction Conference 2019, 513–522. https://doi.org/10.3311/CCC2019-071

3. Bilgin, G., Eken, G., Ozyurt, B., Dikmen, I., Birgonul, M. T., & Ozorhon, B. (2018). Construction Project Portfolio Management Tool (COPPMAN). COBRA 2018 Conference, 805–816. https://www.rics.org/globalassets/rics-

website/media/knowledge/research/conference-papers/construction-project-portfolio-management-tool-rics.pdf

4. Dikmen, I., Ozbakan, A. T., Yildiz, A. E., & Eken, G. (2018). An Experimental Study on Impact of Risk Data Visualization on Risk Evaluations. 5th International Project and Construction Management Conference (IPCMC 2018), 1419–1427.

5. Bilgin, G., Eken, G., Ozyurt, B., Dikmen, I., Birgonul, M. T., & Ozorhon, B. (2017). Handling Project Dependencies in Portfolio Management. Procedia Computer Science, 121, 356–363. https://doi.org/10.1016/j.procs.2017.11.048

6. Eken, G., Bilgin, G., Dikmen, I., & Birgonul, M. T. (2017). Knowledge-Based Portfolio Management: A Taxonomy for Lessons Learned. ISEC-9: Resilient Structures and Sustainable Construction, C–13. https://doi.org/10.14455/ISEC.res.2017.13

7. Bilgin, G., Eken, G., Dikmen, I., & Birgonul, M. T. (2015). A Relational Database for Construction Delay. The Eighth International Conference on Construction in the 21st Century (CITC-8), 286–293.

8. Eken, G., Bilgin, G., Dikmen, I., & Birgonul, M. T. (2015). A Lessons Learned Database Structure for Construction Companies. Procedia Engineering, 123, 135–144. https://doi.org/10.1016/j.proeng.2015.10.070

9. Bilgin, G., Eken, G., Özyurt, B., Dikmen, I., Birgönül, M.T., Özorhon, B., 2016. İnşaat Şirketleri için bir Proje Portföy Yönetim Aracı: COPPMAN, in: 4. Proje ve Yapım Yönetimi Kongresi. 3-5 November, Eskişehir, Turkey, pp. 233–245.

10. Özyurt, B., Bilgin, G., Eken, G., Dikmen, I., Birgönül, M.T., 2016. İnşaat Projelerinde Öğrenme: Benzerlik Değerlendirmesi için Kümeleme Analizi, in: 4. Proje ve Yapım Yönetimi Kongresi. 3-5 November, Eskişehir, Turkey, pp. 246–257.

11. Erol, H., Eken, G., Bilgin, G., Dikmen, I., Birgönül, M.T., 2014. Yalın İnşaat Uygulamalarının 4-Boyutlu Model ile Gösterimi, in: 3. Proje ve Yapım Yönetimi Kongresi. 6-8 November, Antalya, Turkey, pp. 351–362.