8-23-2021 11:00 AM

Automatic extraction of requirements-related information from regulatory documents cited in the project contract

Sara Fotouhi, *The University of Western Ontario*

Supervisor: Madhavji, Nazim, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Science degree in Computer Science

Abstract

[Context and motivation] Project contracts for building a system contain a large number of cross-references to regulatory documents such as environmental regulations, quality standards, and regulatory "codes". The system being developed must comply with regulatory requirements in such documents. Thus, a domain expert needs to read and interpret the relevant regulatory documents. [**Problem]** This can be an arduous and time-consuming task in large projects because the relevant regulatory requirements may be scattered across numerous regulatory documents. [**Principal idea and novelty]** The text prior to or following an external cross-reference in a contract contains information that can assist in automatically locating relevant information from the target regulatory documents. This study used dependency parsing, Part of Speech tagging and Regular Expression to extract the Target Phrase, which is the text referencing more elaborate content in the cited external document, and the target position, which is the location of the referenced text within the external document. The study then conducted a search operation using Elasticsearch and query DSL to retrieve relevant information from the cited legal documents and standards. **[Research Contribution**] This thesis describes a software solution that, to our knowledge, for the first time automatically extracts requirement-related information from external documents cross-referenced in the contract. **[Conclusion]** The final output displays the relevant text, the content of relevant pages and the page number for a corresponding regulatory requirement ordered by relevance score. For Target Phrase extraction, we obtained Precision = 0.81, Recall = 0.98 and F-measure = 0.89. We obtained Precision = 1 and Recall = 1 in target position extraction. Automatically extracting the relevant information from disparate sources will save an enormous amount of time and reduce the workload for requirement analysts and domain experts.

Keywords

External cross-references, Regulatory compliance, Elasticsearch, Dependency parsing, information extractions

Summary for Lay Audience

Project contracts for building a system contain a large number of internal and external cross-references to regulatory documents such as environmental regulations, quality standards, and regulatory "codes". External Cross-references are citations that refer to a fragment of text within an external legal document. The system being developed must comply with these regulations to avoid defective or sub-standard systems, customer dissatisfaction and potential penalties for violating the law. Thus, domain experts and requirement analysts need to read and interpret the relevant regulatory documents, but this can be an arduous and time-consuming task in large projects because the relevant regulatory requirements may be scattered across numerous regulatory documents. The text prior to or following an external cross-reference in a contract contains information that can assist in automatically locating relevant information from the target regulatory documents. This study is the first to extract the Target Phrase, which is the text referencing more elaborate content in the cited external document, and the target position, which is the location of the referenced text within the external document, to automatically find relevant information from the target regulatory documents in the contract. In this study, such keywords and key phrases were extracted automatically using the dependency structure of the sentences, and after that, a search operation was used to search the Target Phrase within the text of those documents to find any possible matches. Automatically extracting the relevant information from disparate sources will save an enormous amount of time and reduce workloads for domain experts. This method will make the work of domain experts more efficient.

Acknowledgement

This thesis would not have been possible without the inspiration and support of a number of wonderful individuals.

I owe my deepest gratitude to my supervisor Professor Nazim Madhavji. Without his enthusiasm, encouragement, and continuous support, this thesis would hardly have been completed.

I would like to thank the Department of Computer Science at the University of Western Ontario for education, infrastructure and scholarship support.

My deepest appreciation goes to my Mom and Dad for their never-ending love, sacrifices and encouragement. I would not be able to pass this experience without their support.

My special thank goes to the love of my life, Arash, for being part of this journey and for his support and encouragement.

Finally, I would like to thank my little son Ryan, who joined us when I was writing my thesis, for giving me unlimited happiness and pleasure.

Glossary of Terms

Cross-references	Cross-reference is defined as a citation that refers to a fragment of text in the same legal document or a fragment of text in an external document. (Maxwell et al., 2012).
DC (Direct Cue) reference	Format of cross-reference expressions that contains a reporting phrase followed by a reference (Rahmani et al., 2020).
Reporting Phrase	Reporting phrases are specific phrases that are often used for referring to a cross-reference, such as in accordance with, comply with, etc. (Rahmani et al., 2020)
Target Position	Place of referenced text within the external documents, which provide additional information to existing regulatory requirements. For example, part 2.5.6.
Target Phrase	A phrase in regulatory requirements which is elaborated by the content of the cited external document, which provides additional information and more details about the Target Phrase.
nsubj	Nominal subject, syntactic subject of a clause
nsubjpass	passive nominal subject, syntactic subject of a passive clause
dobj	Direct object, (accusative) object of the verb

Table of Contents

List of Tables

List of Figures

Chapter 1

1 Introduction

This research aims to propose an automated solution to efficiently support requirement analysts with requirements-related information extraction from numerous external documents cited in project contracts. The following subsections describe the topic's context and motivation for studying this subject, the problem description, the summary of the proposed solution, and the key contributions. Finally, the thesis structure will be represented.

1.1 Context and Motivation

Several different services are provided through developed software for industries working in areas such as government, finance and health (Kokaly et al., 2016; Nekvi & Madhavji, 2014). The software systems in these areas must be subject to relevant regulations and standards (Sannier et al., 2017). In different countries, these rules and standards are enacted for the organizations active in these different areas (Kiyavitskaya et al., 2008). Likewise, in software domains, the rules enacted usually include issues related to safety, privacy and security (Kokaly et al., 2016). One of the significant concerns of software development companies is the compliance of their products with the relevant regulations and standards (Adedjouma et al., 2014). Compliance is a very costly and complex task in software development, and companies do not have a choice to ignore it, and therefore compliance is an inevitable cost for them (Ingolfo et al., 2013). Of course, compared to compliance, the cost of non-compliance is much higher (Maxwell et al., 2012), which not only leads to a defective and substandard system but also leads to customer dissatisfaction as well as potential penalties for the violation of laws (Nekvi & Madhavji, 2014).

Some well-known laws in North America that directly affect software development practices include HIPPA (Health Insurance Portability and Accountability Act) and GLBA (Gramm–Leach–Bliley Act) (Hamdaqa & Hamou-Lhadj, 2011). In the United States, organizations have spent almost $ 17.6 billion in just a few years to develop health care systems compliant with HIPPA (Ingolfo et al., 2013). Although compliance means that

the operational system must comply with regulations and standards, it is clear that in most of these complex systems, software companies cannot wait until the end of the project, and in order to ensure the target system's compliance, at every stage of the project development, software companies need to ensure there is compliance. Requirement engineering is one of these stages that plays a key role in the system's compliance (Nekvi & Madhavji, 2014). In order to achieve compliance at the requirement level, it is necessary to ensure that the set of requirements comply with the standards and regulations (Ingolfo et al., 2013) and also to consider these relevant standards and regulations as important sources for the software requirements (Sannier et al., 2017).

An important issue in large software system projects is that such projects are contract-based (Daneva et al., 2014). In these contracts, in addition to the rights and liabilities of the parties to the contract, an agreement is made on specific measures to be acted upon before the systems' delivery. Among the things that lead to the complexity and high risk of such projects are the regulatory codes and standards, which are mentioned in the contract through cross-references (Berenbach et al., 2010). In order to achieve compliance at the requirements level, requirement engineers must, in addition to considering all the requirements in the contract, also elicit the requirements from the standards and regulations cited in the contract (Nekvi & Madhavji, 2014).

The presence of a large number of external cross-references and domain-specific terms in the contract text adds to the difficulty and complexity of the requirement analysis (Sannier et al., 2016). Therefore, for easier control and faster handling of the external cross-references, having a tool that provides faster and easier access to the information related to the requirements will be a great value for software companies and organizations (Adedjouma et al., 2014; Sannier et al., 2016).

1.2 Problem Description

Developing software that complies with the regulations and standards is a major challenge, as important information on regulatory requirements is scattered across different sources through cross-references, and the requirement analysts must investigate these regulations

and standards in a non-sequential manner, which can lead to the missing of important information that determines whether a system is compliant (Maxwell et al., 2012). Cross-references are used for a variety of reasons, such as defining and providing more details, as well as characterizing specific constraints or exceptions. (Sannier et al., 2017). If a single condition or exception is ignored in the relevant standards and regulations, it can lead to non-compliance, which will exert serious consequences (Adedjouma et al., 2014; Kiyavitskaya et al., 2008).

When analysts want to analyze the regulatory requirements in software systems, they also need to identify and track the cross-references and to take into account and analyze the information associated with them, which are scattered across various sections and pages of countless external documents (Adedjouma et al., 2014). However, usually, only a small part of the standards and regulations contain information related to regulatory requirements. Specifying specific sections of the standards and regulations that are related to a regulatory requirement in a large project can be difficult and error-prone, and time-consuming.

Several articles have referred to the need for tools to help the requirement engineers solve this problem and speed up the analysis of the requirements (Adedjouma et al., 2014; Hamdaqa & Hamou-Lhadj, 2011; Kiyavitskaya et al., 2008; Nekvi & Madhavji, 2014).

To date, there is no known method that can automatically provide access to information from external sources related to regulatory requirements.

Considering the importance of extracting the requirement-related information from many voluminous external documents, this thesis expands on previous studies and proposes an automated solution to help requirement engineers with information extraction from external documents to analyze external cross-references in contractual requirements efficiently.

Consider a sample page from our case study contract in Figure 1.1; this page contains twelve contractual requirements, two of which are regulatory requirements (Req. No. 1 and Req. No.7). In Req. No. 1, "plugboards" need to comply with ABCD. ABCD is more than

2000 pages and finding the relevant information about Req. No. 1 forces the requirement analyst to manually extract information for further analysis. Our solution to this problem is described in the following.

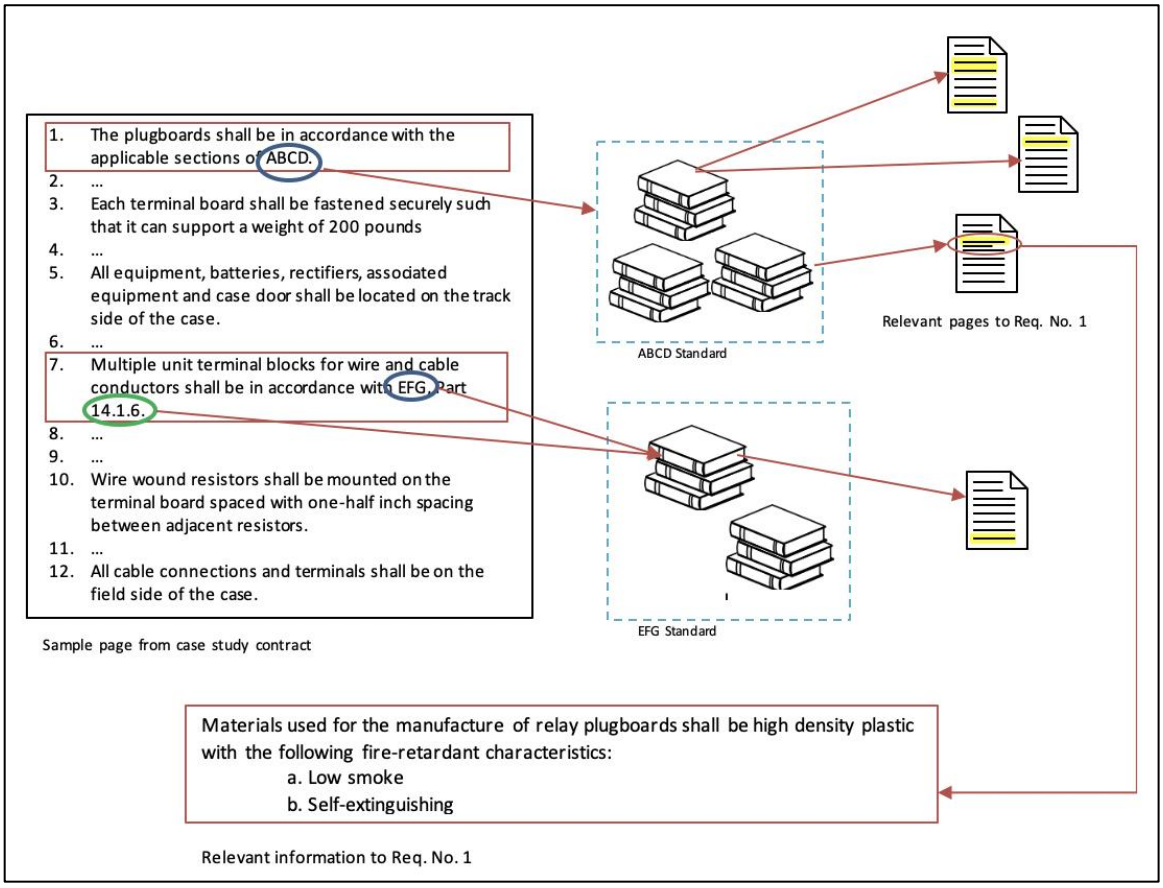

Figure 1.1 Example of requirement-related information cited in the project contract

1.3 Proposed Solution

Going through the previous studies concerned with external cross-references, we can see several papers that state the lack of an efficient tool to support external cross-reference analysis. However, there is only one paper that presents a solution for automatic external cross-reference identification. In (Rahmani et al., 2020), the requirements containing external cross-references, and the external cross-reference components in that requirement, were identified using Natural Language Processing (NLP) and Pattern Recognition techniques. However, in (Rahmani, 2020), only the title of regulatory documents or

standards was identified, and they did not extract any fragment of text from the identified document.

Therefore, there is the need for a supporting tool to build on the approach represented in (Rahmani, 2020) which will provide further information for each regulatory requirement in project contracts.

The proposed solution in this thesis consists of two main steps, each of which includes sub-steps. In step A., critical keywords and key phrases were extracted from each regulatory requirement in the project contract. To accomplish this task, first, the sentences in the project contract that contain external references were extracted. Then using NLP techniques, Target Phrase, which is a phrase elaborated by the content of the cited external document and Target Position, which is the place of referenced text within the external documents, were extracted.

After performing step A, the following keywords and key phrases were extracted for each regulatory requirement in the project contract:

- The name of the external document
- Target Phrase
- Target Position

After that, in step B, a search operation was conducted using Elasticsearch and query DSL to retrieve relevant information from the cited legal documents and standards based on keywords from Step A. First, an index was created to store the content of external documents. Then, utilizing document name, Target Phrase, and Target Position, a query was created. Finally, relevant text from external documents based on Target Phrase and Target Position and name of the external document was retrieved.

As a result, we were able to show the user every page containing relevant information to the regulatory requirement and highlight the specific part containing the keywords on each page ordered by a relevance score.

1.4 Key Contribution

This thesis describes a software solution (as described in Section 1.3), to our knowledge, for the first time to automatically extract requirement-related information from external documents cross-referenced in the contract. The gathered information is organized for the domain expert for analysis and uses, thereby saving a significant amount of time and effort.

1.5 Thesis Structure

The remainder of this thesis has the following structure:

Chapter 2 provides a general background and brief description of techniques and methods we used for the automatic extraction of requirement-related information for each regulatory requirement. Chapter 3 provides a summary of related work, and Chapter 4 describes the foundational work. Chapter 5 presents the methodology and detailed description of each step of the proposed solution. Chapter 6 provides details about the final output of the proposed solution, and chapter 7 describes the evaluation process and analysis of results. Chapter 8 compares the work accomplished in this study with related work and foundational work. Chapter 9 represents discussion. Chapter 10 provides the conclusion and describes future work.

Chapter 2

2 Background

This chapter provides a description of concepts and techniques pertinent to this study. First, contract-based projects and requirement engineering in this type of projects are described. After that, regulatory codes, standards and the concept of compliance and regulatory requirements are explained. Then one of the main challenges of developing a system to be compliant with standards and regulation, which is the cross-references, is described. Then this chapter would be continued with the description of the techniques and tools that we employed in this study. NLP techniques such as tokenization, dependency parsing, and POS tagging were used for the first step of our proposed framework. Then Information retrieval systems such as Elasticsearch and its features and services that we used for the purpose of extracting relevant information is reviewed, such as Python Elasticsearch Client, Bulk API, Mapping, Analyzers, Stemming, Stopwords, Searching, Query DSL, FullText Query, match Query, match_phrase Query, Boolean Query, Highlighting and Relevance score in addition to the Okapi BM25 scoring algorithm. Finally, the Confusion Matrix, Precision and recall that was used for evaluation is described.

2.1 Contract-Based Projects

In today's world, customer-centric organizations often use contracts to deliver large software systems from IT service providers and to ensure that the expected results are achieved at the end of the project (Daneva et al., 2014). In these projects, the supplier and the receiving customer must sign an agreement outlining the results or services (Chen et al., 2013).

In the contract, in order to provide a large software system, the customer and the seller undertake to do or not to do certain actions in the process of providing the system, and the customer-seller relationship is defined based on the definition of law, i.e., the liabilities and expectations of each party. Large contract-based projects are managed differently and are accomplished in a regulated and standardized manner compared to smaller projects (Berenbach et al., 2010; Daneva et al., 2014).

2.2 Contract-Based Requirements Engineering

Requirement engineering is critical to the successful implementation of the contract-based projects, and in addition, lack of knowledge and awareness can be extremely costly, as contract projects are large and often involve a large number of contractors in different areas, and there are also serious penalties for making mistakes or not meeting the deadlines. A key factor that increases the complexity and risk for requirement engineering due to the contract-oriented nature of the project is it requires the review and analysis of regulatory codes and standards in the requirement engineering process (Berenbach et al., 2010).

2.3 Regulatory Codes

They are the requirements that are provided by a legal entity and are required to be observed by law (Berenbach et al., 2010).

2.4 Standards

These are guidelines provided by professional organizations and are usually recommended by those organizations. In other words, they are the ‘best practices’ suggested by a particular organization. However, if a standard is mentioned in the contract, it must be executed like any other requirement in the contract. Therefore, it is necessary to consider all the regulatory requirements in order to finally have a system that is legally compliant (Berenbach et al., 2010; Nekvi & Madhavji, 2014).

2.5 Compliance

Compliance means to ensure that the business and operation processes, as well as the employed methods, are in accordance with a predetermined or agreed set of rules (Sadiq & Governatori, 2015). In the development of software systems, these systems are usually done in areas that must be subject to many rules and regulations, and these rules directly affect the development methods and requirements, and one of the priorities of software companies is to comply with the standards and regulations because if they do not succeed in compliance, in addition to high costs, there would be financial penalties and several irreparable consequences such as loss of credit and reputation of the company, getting

involved in legal issues as well as punishments for the violation of the law (Massey et al., 2014; Nekvi & Madhavji, 2014).

2.6 Regulatory Requirement

It is a requirement in which a reference is made to a regulatory document, and the items in the reference must be observed in that requirement. Regulatory requirements often refer to the standards and regulations, while non-regulatory requirements do not refer to standards or regulations (Sadiq & Governatori, 2015). Regulatory requirements are not explicitly labelled in the contract and are scattered among other requirements (Nekvi & Madhavji, 2014).

The medical services and finance section have a considerable number of regulatory requirements, and many researches have been conducted in this area in software engineering and regulatory compliance (Massey et al., 2014).

2.7 Cross-references

Cross-reference is defined as a citation that refers to a fragment of text in the same legal document or a fragment of text in an external document. (Maxwell et al., 2012). If a cross-reference points to a part of the text within the same document, it is called internal cross-reference. Conversely, it is called external cross-reference if it points to another document (Maxwell et al., 2013).

2.8 NLP

Natural language processing is a subfield of artificial intelligence and linguistic (Chopra et al., 2013). It is also a well-known technique for extracting desired elements from plain text and has been developed to help computers understand phrases or words written in natural languages.

Software requirements are usually compiled and expressed as a text in natural and human-readable language, which is time-consuming and difficult to analyze manually. On the other hand, NLP is a knowledge discovery approach for automated extraction and has many applications in various fields of software engineering, especially in requirement

engineering. Among the applications of requirement engineering, one can refer to ambiguity removal, requirement analysis, requirement assessment, Requirement Elicitation, classification and prioritization (Nazir et al., 2017).

2.9 Dependency Parsing

Dependency parsing specifies a binary head-dependent relationship between the words in a sentence based on the grammatical dependency structure of a sentence (Kübler et al., 2009); it is used for various purposes in NLP such as labelling semantic roles, extracting links between words and machine translation (Qi et al., 2019). examples of grammatical relations are root, auxiliary, an object of a preposition, direct object, nominal subject, etc.(de Marneffe & Manning, 2008)

2.10 Tokenization

Tokenization is the procedure of splitting a sentence or text into desired basic units such as words (Webster & Kit, 1992). Tokenization often is used as one of the first steps of text mining to provide input for further analysis. The output of tokenization is a list of meaningful elements of the input stream of text (Kannan et al., 2014).

2.11 Part of Speech (POS) tagging

Part of Speech tagging is the task of assigning parts of speech tags such as Noun, Verb, Adjective, Adverb, Pronoun, Preposition, Conjunction and Interjection to each word of an input text considering their individual meaning in addition to the context (Srinivasa-Desikan, 2018).

2.12 Spacy Library

Spacy is a free, open-source Python package that uses several language models to process text and perform advanced NLP tasks such as POS tagging, Dependency parsing and Named Entity Recognition (*SpaCy 101: Everything You Need to Know · SpaCy Usage Documentation*, n.d.; Srinivasa-Desikan, 2018). Some examples of dependency relations, their brief description and their label in Spacy are shown in Table 2.1.

Table 2.1 Examples of dependency relations (de Marneffe & Manning, 2008; *SpaCy 101: Everything You Need to Know · SpaCy Usage Documentation*, n.d.)

Dependency Relations	Dependency Labels in Spacy (token.dep_)	description
root	*ROOT*	the root of the sentence
direct object	*dobj*	(accusative) object of the verb
nominal subject	*nsubj*	syntactic subject of a clause
passive nominal subject	*nsubjpass*	syntactic subject of a passive clause
object of a preposition	*pobj*	head of a noun phrase following the preposition
adjectival modifier	*amod*	adjectival phrase that serves to modify the meaning of the NP
determiner	*det*	relation between the head of an NP and its determiner
conjunct	*conj*	relation between two elements connected by a coordinating conjunction, such as “and”, “or”, etc.
prepositional modifier	*prep*	is any prepositional phrase that serves to modify the meaning of the verb, adjective, noun, or even another preposition
auxiliary	*aux*	non-main verb of the clause

2.12.1 Displacy

Displacy is an advanced dependency visualizer which is part of the Spacy library. It depicts a sentence's syntactic structure using arrows and labels. Arrows from dependent word point

to its head and the labels under arrows show the type of head-dependent relationship (*SpaCy 101: Everything You Need to Know · SpaCy Usage Documentation*, n.d.). Examples of such relationships are described in Table 2.1.

2.13 NLTK Library

NLTK is the abbreviation for Natural Language Toolkit, which is a widely used platform in Python and supports several NLP tasks such as tokenization, POS tagging and etc. (*Natural Language Toolkit — NLTK 3.6.2 Documentation*, n.d.).

2.14 Unstructured Data

Unstructured data generally refers to the text written in natural language that is not compatible with a relational database or does not have any predefined semantic structure. Web pages are known as the primary source of unstructured data (Ceri et al., 2013; Kumar et al., 2020).

2.15 Information Retrieval

IR is described as a task of searching for relevant documents as a result of matching to the lexical patterns used in a query (Grossman & Frieder, 2012). Moreover, the IR's function is explained as to find the information which is provided through unstructured data from the large collections of documents. The main feature of this definition is the existence of large collections (Ceri et al., 2013; Schütze et al., 2008)

Information retrieval involves displaying, storing, organizing, searching, and accessing large sets of human language data. For the user, effectively achieving the relevant information is the fundamental objective of information. The most common utilization of IR is associated with the natural language text type data set. IR systems and services have become so widespread that millions of people are relying on them in order to take care of their daily routines such as business, education, and entertainment. Among the most widely used IR services, search engines are on top. An inverted index is the main data structure in most information retrieval systems (Büttcher et al., 2016). Since the textual IR query and the text sources, both of which are expressed in natural language, a number of text

operations are performed at the top of the retrieval process. In the application of IR techniques, it is anticipated that the process, leading to the production of a sorted list of documents that the most relevant documents appear on top of the list, this would save an enormous time for the user (Ceri et al., 2013).

2.16 Topic modelling

Topic modelling is one of the main statistical techniques that is used for understanding the concept of large-scale unstructured text-based data. It helps to identify various themes in the documents based on the groups of words that often appear together across the documents and thereby create a relationship between documents. Grouping social media users based on the similarity of the content of their posts and classifying news articles to different topics are examples of topic modelling applications. Latent Dirichlet Allocation (LDA) and Latent Semantic Analysis (LSA) are two of the most popular topic modelling methods (Jelodar et al., 2019; Vayansky & Kumar, 2020).

2.17 Elasticsearch

Elasticsearch is a distributed and open-source system that has the ability to efficiently search the text in a real-time manner. This system has been written in Java and is based on the Apache Lucene library and is designed for searching, analyzing and storing the data and is generally selected as the main technology in implementing document-oriented Big Data projects. Its most important features include high availability, its performance in real-time search, horizontal scalability and ease of use (Gormley & Tong, 2015; Zamfir et al., 2019)

In Elasticsearch, information is stored as JSON (*JSON*, n.d.) documents. Each field of the document is indexed to be searchable in real-time. Using the inverted index used in Lucene makes full-text queries faster. In addition, the users can search for the data through its exclusive and flexible search language called Query DSL (Zamfir et al., 2019).

The Document is the most basic Entity in Elasticsearch, which is equivalent to a row in a relational database. An index in Elasticsearch is a collection of documents with different properties that are organized through user-defined mapping that specifies the type of

document and fields for different data types. Elasticsearch is able to index numbers, text, dates and almost any type of data(Shah et al., 2018).

One of the most important benefits that have always been considered is the ability to manage large amounts of schema-less data, and JSON is the only compatible format with Elasticsearch (Taware & Shaikh, 2018).

Elasticsearch is a stable and independent platform that supports several languages, including Python (Shah et al., 2018). Elasticsearch Python plugin can be utilized in Python that provides the capability of interacting with Elasticsearch through Python (Badhya et al., 2019).

2.17.1 Python Elasticsearch

Python Elasticsearch is a low-level Python library that supports interaction between codes in Python and Elasticsearch (*Python Elasticsearch Client — Elasticsearch 8.0.0 Documentation*, n.d.).

2.17.2 Bulk API

Large amounts of data can be indexed using Bulk API. The advantage of using the Bulk API is that it indexes the data in chunks by only one API call, which is much faster than indexing data one by one (Badhya et al., 2019).

2.17.3 Mapping

Mapping is the process of defining how a document and the contents of its fields are stored and indexed. Mapping is similar to the schema definition in the SQL database and is an essential part of any index in Elasticsearch; mapping specifies the types of documents in the index and how to save and analyze the document fields (Kononenko et al., 2014; *Mapping | Elasticsearch Guide [7.13] | Elastic*, n.d.).

2.17.4 Analyzers

Analyzers are a set of rules that are used to change the input text in order to convert it to the desired format so that the text can be processed in the format defined further in the

project (Badhya et al., 2019). Only the text fields support the analyzer in Elasticsearch. The input sentence passes through these analyzers built into Elasticsearch in order to increase the efficiency and speed by making modifications. The analyzers can be made according to our needs and application. Character filters, tokenizers, and token filters are three main parts of an analyzer (*Analyzer | Elasticsearch Guide [7.13] | Elastic*, n.d.).

2.17.5 Stemming

Stemming combines various forms of a word into a single representation called a stem. For example, “present” is the stem of different terms such as: “presentation”, “presented”, “presenting”. This process could decrease the index size significantly (Kannan et al., 2014).

2.17.6 Stop word

Stop words repeatedly appear in various texts, although they do not help to convey crucial contextual meaning. Examples of stop words are “the”, “a”, “to”, “this”, “is”, etc., and they are employed to connect words in the text and often cause difficulties in information retrieval due to high occurrences. To enhance the system's performance and mitigate the text data, stop words are needed to be removed (Kannan et al., 2014).

2.17.7 Searching

Searching in Elasticsearch is accomplished in two ways: query or filter. The main difference between them is that the query calculates and assigns a relevance score for each returned document, while the filter does not do so. As a result, searching through a filter is faster than a query. The official document recommends using the query only in two situations: for full-text search or when the relevance score of each returned result is important (*Search Your Data | Elasticsearch Guide [7.13] | Elastic*, n.d.).

2.17.8 Query DSL (Query Domain Specific Language)

Elasticsearch provides an exclusive query language based on JSON called Query DSL.

Elasticsearch has a powerful Query DSL that supports advanced search features based on Lucene query syntax. There are actually two types of query clauses: 1) filter-context, which determines whether the query exactly matches the document. 2) query context, which

specifies to what degree the document matches the query according to the relevance score (*Query DSL | Elasticsearch Guide [7.13] | Elastic*, n.d.). Since we need a full-text search in this study to provide a list of ranked results, we will only use query-context clauses.

2.17.9 Full-Text Query

It allows you to search for the analyzed text fields, such as the content of an email. The query string generally is processed using the same analyzer that was applied to the field during indexing (*Query DSL | Elasticsearch Guide [7.13] | Elastic*, n.d.). There are several types of queries including, match Query, match_phrase Query and Boolean Query.

2.17.9.1 match Query

It is a standard query for full-text queries comprising fuzzy match and phrase matching.

2.17.9.2 match_phrase Query

It is similar to a match query, but its functionality is like exact matching.

2.17.9.3 Boolean Query

Boolean query (*Boolean Query | Elasticsearch Guide [7.13] | Elastic*, n.d.) uses different combinations of match queries with Boolean expressions. The Boolean query is written on the Lucene Boolean Query and is created using one or more Boolean clauses. The final score is calculated based on the must and should matches. The types of occurrences in the Boolean query are:

- Must: The clause (query) must be present in the matching documents and be calculated in the score. It is similar to “AND”.
- Filter: The clause (query) must be present in the matching documents; however, unlike the must type, the query score will be ignored.
- Should: The clause (query) should be present in matching documents. It is similar to “OR”.
- Must not: The clause (query) must not appear like matching documents.

2.17.10 Highlighting

The highlighters allow you to highlight one or more fields in the search results in order to show the user which part of the extracted text, the query matches and specifies the location. When using highlight in queries, the response contains an extra highlight element for each search that has highlighted fields and highlighted fragments (*Highlighting | Elasticsearch Guide [7.13] | Elastic*, n.d.).

2.17.10.1 Unified Highlighter

Unified highlighter uses Lucene Unified Highlighter. This highlighter converts the text into its constituent sentences and uses the BM25 algorithm to score each sentence, and supposedly they were individual documents in the corpus (*Highlighting | Elasticsearch Guide [7.13] | Elastic*, n.d.)

2.17.11 Relevance score

Relevance score is a floating-point number that indicates how similar the returned document is with the query using a similarity algorithm. Okapi BM25 is used as the default scoring algorithm in Elasticsearch. More relevant documents are represented by higher scores (Gormley & Tong, 2015).

2.17.11.1 Okapi BM25

Okapi BM25 is a scoring algorithm that search engines mostly utilize to determine how relevant the returned documents are to a specific search query. Eq. 1 calculates the Okapi BM25 for document D where q_i is the i^{th} keyword in the query, $IDF(q_i)$ is the inverse document frequency of the q_i, $f(q_i, D)$ is term frequency, $|D|$ is the word count of the document D, avgdl is the average document length (*Okapi BM25 - Wikipedia*, n.d.). Parameters b and k_1 are 0.75 and 1.2 respectively in Elasticsearch (*Practical BM25 - Part 3: Considerations for Picking b and K1 in Elasticsearch | Elastic Blog*, n.d.).

$$score(D,Q) = \sum_{i=1}^{n} IDF(q_i).\frac{f(q_i,D).(k_1+1)}{f(q_i,D)+k_1.(1-b+b.\frac{|D|}{avgdl})} \quad (1)$$

2.18 PyPDF2 toolkit

PyPDF2 is a helpful and practical tool for managing and manipulating PDF files in Python. This library is implemented in Python, so it is simply available and executable in any Python platform (*PyPDF2 · PyPI*, n.d.). one of the main capabilities is handling and working with each page of pdf files individually. It can also extract plain text from each page or whole files to be used later for analysis and implementing NLP techniques (Kekare et al., 2020).

2.19 Confusion Matrix

The confusion matrix is a 2×2 square matrix, as shown in Eq. 2, and it plays a fundamental role in the performance evaluation of machine learning models and information retrieval systems. In the confusion matrix, rows indicate the actual class, while columns represent the predicted class.

$$\begin{bmatrix} Number\ of\ True\ Positives & Number\ of\ False\ Negatives \\ Number\ of\ False\ Positives & Number\ of\ True\ Negatives \end{bmatrix} \quad (2)$$

- Number of True Positives (TP): number of correctly classified instances as positive
- Number of False Negatives (FN): number of instances incorrectly classified as negative
- Number of False Positives (FP): number of instances incorrectly classified as positive
- Number of True Negatives (TN): number of correctly classified instances as negative

There are several evaluation indicators derived from this matrix, such as Precision, Recall and F-measure (Caelen, 2017; Kuznetsova, 2021).

2.19.1 Precision

Precision is one of the evaluation indicators we can obtain from the confusion matrix, and it is the ratio of correctly classified positive instances to all positive classified instances (Kuznetsova, 2021).

$$precision = \frac{TP}{TP + FP} \tag{3}$$

2.19.2 Recall

Recall a performance indicator that shows the ratio of correctly classified positive instances to all positive instances (Kuznetsova, 2021).

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

2.19.3 F-measure

F-measure is the harmonic mean of precision and recall, and it can be obtained by Eq. 5 (Kuznetsova, 2021).

$$F - measure = \frac{2 \times precision \times recall}{precision + recall} \tag{5}$$

Chapter 3

3 Related work

Since crucial information about regulatory requirements is scattered within the same document and across different sources through cross-references, the requirement analysts must follow cross-references to relevant regulations and standards to ensure compliance at the requirement level. Various studies have been conducted to reflect the importance of analyzing internal and external cross-references, which we will review in this chapter. However, to our knowledge, no research has focused on extracting relevant information from external sources cited in regulatory requirements. In Section 3.1, the strands of work will be described, which proposed a solution to handle either internal cross-references automatically or external cross-references manually. In Section 3.2, we will analyze the related work based on the type of cross-reference, the accomplished task and whether it was done manually or automatically and finally, the type of text they examined.

Then in chapter 4 the only research that addressed identifying external cross-references with an automated approach which is the foundation of our work will be described and analyzed in detail.

3.1 Proposed solutions for handling cross-references

(Maxwell et al., 2012) aimed at helping requirement engineers to consider the importance of having several cross-references and analyzed their impacts on software requirements. They introduced different types of cross-references and manually examined the U.S. Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule, the Gramm– Leach–Bliley Act (GLBA), and the GLBA Financial Privacy Rule. They identified two-level external cross-references, and they depicted the graphs of external cross-references for each document. For the task of checking whether the cross-references point to a conflicting requirement or the referenced text is just for refining the pre-existing requirement, they manually locate the text that has been referenced. They also Specified five different types of conflicts and provided some descriptive strategies for each to solve contradictions in

such requirements to help requirement engineers prevent the costly consequences of non-compliance systems.

In addition, they proposed a taxonomy for cross-references classification and introduced seven variations of cross-references in compliance requirements which are (1) constraint, (2) exception, (3) definition, (4) unrelated, (5) incorrect, (6) general, and (7) prioritization. Their work was one of the first studies that investigated the impacts of cross-references on software requirements.

(Tran et al., 2014) examined the task of reference resolution in Japanese National Pension Law and proposed an automated solution through a four-step framework to identify references in the mentioned legal document and extract the referenced text within the same document. Their goal was to extract the smallest part of the text to which the reference referred and not only the whole grouping concept like a chapter or an article. In their study, a cross-reference is called “mention” and the text which cross-references refer to is called “antecedent”. They stated that two main components of mentions in the legal text are the position part and the content part. Examples of position part and content part in their work are shown in **Error! Reference source not found.**.

Table 3.1 Examples of position part and content part (Tran et al., 2014)

mention	The notification in the provision of Para 1	
Parts	Para 1	The notification in the provision of
	Position Part	Content Part
mention	The provision in article 27-3 and article 27-5	
Parts	article 27-3 and article 27-5	The provision in
	Position Part	Content Part

In the first step, they detected mentions by using the sequence labelling technique. They labelled each word with a tag to specify whether a word is part of a mention or not, and it is the first element, last element, or an element inside a mention. The identified mention was divided into the content part and the position part. Then mention was classified into two different classes using supervised machine learning. In the second step, they specified the context information based on the content and position part, references appear near the mentions, and also the obtained class of the mention. In the third step, they extracted several candidates for each mention based on the output of previous steps, the main noun of the mention or its synonym, and the punctuation marks in the sentence. In the fourth step, the best candidate chose as the exact antecedent. Finally, they achieved 67.02 % in the F1 score for identifying mentions and extracting the antecedents.

(Adedjouma et al., 2014) proposed an automatic solution for detection and resolution of internal cross-references in Luxembourg's Income Tax Law and obtained promising results. The main difference of their work compared to previous work was considering the legal text's schema as the first step. They manually illustrated how the content of their case study was organized into different grouping concepts such as sections, subsections, articles, etc. The relevant text schema was represented using a UML class diagram with each grouping concept as individual classes. In each class, the header attribute provides details such as the grouping concept marker, numbering formats and delimiters. Then the schema is used for transforming unstructured legal text to a markup text. However, this schema was designed for their case study, and it needs to be modified for any other legal document. For detecting cross-references, they extracted CRE patterns from their case study legal text and then classified them as simple or complex, implicit or explicit and internal or external. Then the part of the text that matched with predefined patterns, forwarded to an interpretation phase. In this phase, a detected cross-reference might be converted into several cross-references. For example, "Articles 99 to 102" converted to "Article 99, Article 100, Article 101, Article 102".

They also explained the structural format of legal documents which is such that the end of a specific part is recognized when confronted with a new grouping concept that is either at the current or higher levels. For example, if we are in section 3 of legal text, this part ends

if the next grouping concept is section 4, or if it does not exist, it is the starting point of the above level, for example, a new chapter. This helped them in interpreting complex cross-references. Based on the text schema they were able to interpret positions such as, current article; following paragraphs; alinea 2, sub a; articles 14, 61, 91 or 95; and paragraphs 1 to 3. Finally, after interpretation, each CRE was linked to their corresponding target provision to help easier navigation within the text.

Sannier et al. (Sannier et al., 2016), to support requirement analysts to efficiently handle and interpret the cross-references, conducted a qualitative study and classified the semantic intent of cross-references to eleven different types of intent such as Compliance, Constraint, Definition, Delegation, Exception, Refinement, and etc. In their taxonomy, first, cross-references in Luxembourg's Income Tax Law and Chamber of Deputies' Draft Law No. 6457 were identified, and the text before and after a cross-refer was examined manually to develop natural language patterns. Then the extracted patterns were used for the automatic classification of cross-references' semantic intent. Their work expands the intents addressed by (Breaux, 2009; Hamdaqa & Hamou-Lhadj, 2011; Maxwell et al., 2012). For each of the semantic intents, a definition, frequency in their case study legal text, and the patterns were presented. However, they did not investigate the content of the referenced text.

3.2 Analysis of Related Work

In this section, a detailed analysis of previous work is provided. As you can see in **Error! Reference source not found.**, we analyzed the studies that dealt with the challenge of cross-references, based on three main criteria, the examined type of cross-references, whether the work is done manually or automatically and what was the performed tasks, and also their case study type of text.

Table 3.2 summary of previous work attempted to deal with cross-references

Reference	Type of cross-references		Task					Case study legal text	
	internal	external	Manual	Automatic	identification	classification	Referenced text extraction	Law	Contract
(Maxwell et al., 2012)	✗	✓	✓	✗	✓	✓	✓	✓	✗
(Tran et al., 2014)	✓	✗	✗	✓	✓	✓	✓	✓	✗
(Adedjouma et al., 2014)	✓	✗	✗	✓	✓	✗	✓	✓	✗
(Sannier et al., 2016)	✓	✗	✗	✓	✗	✓	✗	✓	✗

3.2.1 Type of cross-references

The legal text contains numerous cross-references to different parts of the same document and also external resources. The former is called internal cross-reference the latter is called external cross-reference. To ensure a legally compliant system, both internal and external cross-references need to be considered and analyzed. However, as we can see in Table 3.2, previous work only focused on internal cross-references except for (Maxwell et al., 2012) that investigated external cross-references, although they performed the task manually.

3.2.2 Manual vs. Automatic

It is possible to manually handle the cross-references in smaller projects. Although in large-scale projects, including various technical domains and many voluminous standards and

regulations that need to be considered, doing the task manually is time-consuming and error-prone. Amongst previous work, all automated solutions are dedicated to internal cross-reference analysis.

3.2.3 Task

The most important actions taken to deal with the challenge of cross-references are their identification, classification and referenced text extraction. As we can see in Table 3.2, (Maxwell et al., 2012) performed all three tasks manually, (Tran et al., 2014) performed all three automatically but only for internal references, moreover, the classification was part of their solution to take the right strategy for each class of references. (Adedjouma et al., 2014) improved the text extraction solution represented by (Tran et al., 2014) by defining a text schema class diagram for their case study legal text. (Sannier et al., 2016) focused on the classification of internal cross-references based on their semantic intent and exclude the identification and text extraction from their work.

Amongst them, only (Adedjouma et al., 2014) and (Tran et al., 2014) extracted the referenced text. However, the referenced text extracted in previous studies was within the same document that contains cross-references and not from external sources.

3.2.4 Case study type of text

Previous work only focused on analyzing the text of the law for internal cross-reference handling.

In the next chapter, we will describe individual research by (Rahmani, 2020). For extracting requirement-related information, we adopted her approach for external cross-reference identification. The main difference between (Rahmani, 2020) and the studies described in this chapter is that she automatically identified external cross-references in a contract.

Chapter 4

4 Foundational work (by E. Rahmani)

This chapter provides a summary of a study about identifying external cross-references. Specifically, we refer to the work by E. Rahmani. We consider this work as our foundational work and continue their study by adding new information extraction steps for each regulatory requirement.

4.1 Identifying External Cross-references using NLP

(Rahmani et al., 2020) proposed an automated solution to identify two levels of external cross-references within the contract. To accomplish this task, she provided a list of semantic cues and a taxonomy of grammatical structures using POS tags to distinguish regulatory requirements in the contract. She used NLP techniques and Pattern Recognition to extract the reference component from each regulatory requirement, the extracted references were validated using APA properties. For second-level external cross-references identification, she used Web Scraping techniques to search the web if the external source was not available locally. Finally, she represented a table to the user with a list of identified external references for each requirement number. However, she only extracted the cross-reference component, which is the title of external sources, and extracting the relevant text cited by an external CR was not included in her study. Therefore, we put a step further and developed a requirement-related information extraction framework that builds on her work.

A detailed explanation of her first level external cross-reference identification is provided in the following.

4.1.1 Reporting Phrases

By investigating numerous contractual requirements, she found that specific phrases are often used for referring to a cross-reference. She collected a list of such phrases and employed it for the first step of their solution for external cross-reference identification. Parts of this list are available in Table 4.1. She checked each paragraph for the occurrences of reporting phrases to decide whether she wanted to keep the paragraph for analysis or the paragraph can be ignored. She kept paragraphs with at least one reporting phrase.

Table 4.1 Examples of reporting phrases

Examples of reporting phrases
in accordance with/to
conform to/with
by the warranty provisions of
permitted by
specified in
indicated in
required by
approved by
proposed by
recommended by
based on
indicated on
determined by
directed by
according to
in conformance with

4.1.2 Three formats of cross-reference expression

Utilizing the reporting phrases, she presented three formats of cross-reference expression. Among three formats of cross-reference expressions represented in Table 4.2, she concluded that the majority of requirements referred to a cross-reference using DC format. 83% in her case study. Accordingly, we will investigate regulatory requirements with DC format.

Table 4.2 Formats of cross-reference expression

Type of format	Abbreviation	Definition	Format
Direct Cue	DC	The reporting phrase will be followed by a reference.	< reporting phrase><reference>
InDirect Cue	IDC	reporting phrase is surrounded by parts of reference on each side	<reference-part2>< reporting phrase > <reference-part 1 >
No Cue	NC	There is not any reporting phrase to surrounds the reference.	<reference>

4.1.3 HasLeaf_Pattern taxonomy

(Rahmani, 2020) study of cross-reference expressions in contractual documents concluded that most of them are using particular patterns while referring to external documents. The patterns were gathered using POS tags based on their grammatical structure in a sentence Therefore, she tokenized identified requirements and assigned POS tags to each word and then searched for the matched patterns to find sentences that contain cross-reference expressions that were matched with the predefined patterns. Some examples of grammatical patterns are:

- {<IN><NN><IN><NN><NNP><NN><NNS><CC><DT><JJ><NN>+}
- {<IN><NN><IN><NNP>+<IN><NN>+<CC><NNP>+<IN><NN>+}
- {<IN><NN><IN><NNP>+<IN><NN>+<CC><NNP>+<IN><NN>+}
- {<NN><TO><NNP><CD><IN><DT><NNP>+<VBD>*<NNP>+<IN><NNP>+}
- {<IN><NN><IN><NNP>+<CD><,><NNP><IN><NNP>+<NNPS>}
- {<IN><NN><IN><NNP>+<CC><NNP>+<IN><NNP><NNPS>}
- {<IN><NN><IN><DT><IN><DT><NNS><IN><JJ><NNP>+}
- {<IN><NN><IN><JJ><NNP>+<,><NNP>+<CC><NNP>+}
- {<NN><IN><JJ><NNP>+<IN><NNP><\(><NNP>+<\)>}
- {<IN><NN><IN><DT><NN><POS><NNS><CC><NNP>+}
- {<IN><NN><IN><NNP>+<CD><CC><NNP><VBZ><CD>}
- {<IN><NN><IN><NNP>+<CD><CC><CD>}
- {<IN><NN><IN><NNP>+<CD><\,><JJ><CD>}

RegexpParser was used to find sentences that contain cross-reference expressions that were matched with the predefined patterns. RegexpParser utilizes a collection of regular expression patterns to define the procedure of the parsing, and its output is a tree, and the detected cross-reference expressions patterns are its leaves.

These leaves then were analyzed to check whether or not they contain any reporting phrases. If the leaves did not contain any reporting phrases, she ignored that paragraph. If they contain any reporting phrases, she needed to make sure reporting phrases refer to an external source and not internal fragments of a contract.

4.1.4 Reference Identifier

It was stated that the identified leaves from the previous step contain two parts:

1) reporting phrase
2) Perhaps the cross-reference component

At this step, for each identified leave, she removed reporting phrase and to make sure that the remained part is a reference, she developed a Has_APA_Properties step.

4.1.5 Has_APA_Properties

The reference title should be matched with one feature of the APA standard. APA standard has the following features:

1. Abbreviations in all uppercase alphabets (example: CSA (**Canadian** Standards Association)
2. One or more words that start with an uppercase (example: Software Quality Assurance Plan)
3. numbers (example: CSA C22.2 No. 124)
4. special characters such as: “/, -, ., _”, etc. (example: CAN/CSA C22.2).

After passing this step, the reference component was identified for each regulatory requirement. Table 4.3 depicts the output of their solution for first level reference identification.

The first column is the contractual requirement number in the contract and the second column is the identified external references and the total number of identified external references.

Table 4.3 Output of (Rahmani, 2020)**'s system**

Contractual Requirement Num	**Reference Level 1**
CR 2.4.1	1- CBC/ASC-G30.18 2- CRL/AMQ G164 3- FBC 41-GP-34M Type G 4- McFfoy Foundry Co. Ltd. MH332 **Total: 4**
CR 4.1.1	1- Document 00500 SPECIAL CONDITONS 2- CAN/ACA-A6/A362 Portland Type **Total:6**
CR 2.6.1.4	1- CBC 41-GP-34M Type III **Total: 7**
CR 2.7.1	1- National Building Code 2- XAXM Manual Standard Practice 3- IXAZ Std 1026 **Total: 10**

Chapter 5

5 Research Methodology

This chapter describes our research methodology. First, in Section 5.1 we go through an overview of our methodology for automatically extracting requirement-related information from external cross-references in the contract and review the steps of our proposed solution. Then in Section 5.2, a detailed explanation of each step is provided. First, we describe a preprocessing step using the foundational work for parsing the contract, identifying regulatory requirements and identifying the reference component, then we continue with Step A, the fundamental part of our framework, which is extracting the keywords from each regulatory requirement. These keywords will be then used to retrieve relevant information from external sources in step B.

Step A involves three following sub-steps: extract the sentence that contains external references, extract the Target Phrase and extract the Target Position from each sentence. After that, this chapter would be continued with explaining the step B which is designed to retrieve relevant information from the cited legal documents and standards. Using the extracted keywords from step A, we describe how relevant information from external sources was retrieved. Step B involves three following sub-steps: create an index to store the content of external documents, create queries using document name, Target Phrase and Target Position and finally retrieve relevant text from external documents based on Target Phrase and Target Position.

5.1 Overview of methodology

This research aims to extract requirements-related information from external regulatory documents cited in the project contract. Since the project contract and external regulatory documents are written in natural language, we need to develop methods that rely on natural language processing to analyze such data types.

To achieve the goal of this research, a framework was proposed that has two main steps, and each of them contains sub-steps. There is also a preprocessing step based on the foundational work by (Rahmani, 2020) that was described in Chapter 4. Steps and sub-steps are listed as the following in this Section:

- Preprocessing step:
    - Parse the contract
    - Identify regulatory requirements
    - Identify the reference component
- Step A. Extract the keywords and key phrases from each regulatory requirement
    - Sub-step A.1- Extract the sentence that contains external references
    - Sub-step A.2- Extract the Target Phrase
    - Sub-step A.3- Extract the Target Position
- Step B. Retrieve relevant information from the cited legal documents and standards
    - Sub-step B.1- Create an index to store the content of external documents
    - Sub-step B.2- Create queries using document name, Target Phrase and Target Position
    - Sub-step B.3- Retrieve relevant text from external documents based on Target Phrase and Target Position

5.2 Detailed steps of the methodology

This chapter describes the details of the techniques and tools which were used in each step and sub-step. Firstly, a brief explanation of preprocessing step is provided. Preprocessing step performed based on the techniques suggested in foundational work (chapter 4).

Preprocessing step

The preprocessing step identifies regulatory requirements and the reference component in each regulatory requirement in the contract.

A regulatory requirement is a requirement that refers to regulation or standard. To automatically extract information related to each regulatory requirement from external documents cited in the project contract, the first step is to identify them in the contract's content. It is an essential step because, often, regulatory requirements are mixed with all other types of requirements. After identifying regulatory requirements, we need the name of the reference itself, which is the title of the referenced standard or any external cross-reference. The output of preprocessing step is 584 paragraphs in the contract with DC format and their corresponding external cross-references.

Step A. Extract the keywords and key phrases from each regulatory requirement

Step A, which contains three sub-steps developed to analyze each regulatory requirement to extract the keywords and key phrases. Output of Step A are Target Phrase and Target Position that would be helpful to retrieve relevant information to each regulatory requirement from the external document in Step B. Step A contains following sub-steps:

Sub-step A.1 Extract the sentence that contains external references

584 paragraphs of requirements were examined that refer to an external document with DC format; 398 (68%) contain one sentence, 112 (19%) include two sentences, 40 (6%) have three sentences, 14 (2%) include four sentences, and 20 (3%) contain more than five sentences. Since the format of requirements, we are analyzing are DC, by checking requirements with more than one sentence, it was observed that in 99% the sentence that contains the cross-reference, contains the Target Phrase which the external reference elaborates with more detail. So, we extracted sentences containing external references in addition to the reference title for further analysis. If there was more than one reference in a sentence, we considered that sentence with each reference. For example, if there are two references in a sentence, we considered that sentence with each individual reference. This sub-step led to 657 sentences with corresponding external cross-reference. Hereon, we call these sentences regulatory requirements.

Example:

- Ensure that air entraining admixture conforms to EFG 23 and AFS C260 :
  - Ensure that air entraining admixture conforms to EFG 23 and AFS C260 → CAN/XX 23
  - Ensure that air entraining admixture conforms to EFG 23 and AFS C260→ AFS C260

Sub-step A.2 extract the Target Phrase

By analyzing regulatory requirements in contractual documents, it was found that the phrases before and after the external reference help retrieve information from external sources. We named these phrases "Target Phrase" and "Target Position".

In this study, a Target Phrase, is a phrase in regulatory requirements which is elaborated by the content of the cited external document which provides additional information and more details about the Target Phrase.

By examining several regulatory documents with DC format (chapter 4), we found that some particular grammar patterns were frequently used in the contractual document for referring to an external rule or standard. Examples are provided in following tables.

The following regulatory requirements in the Figure 5.1 are gathered from the case study contract. The Target Phrases are highlighted with yellow and the Target Positions are highlighted with green. Reporting phrases are shown in red frames.

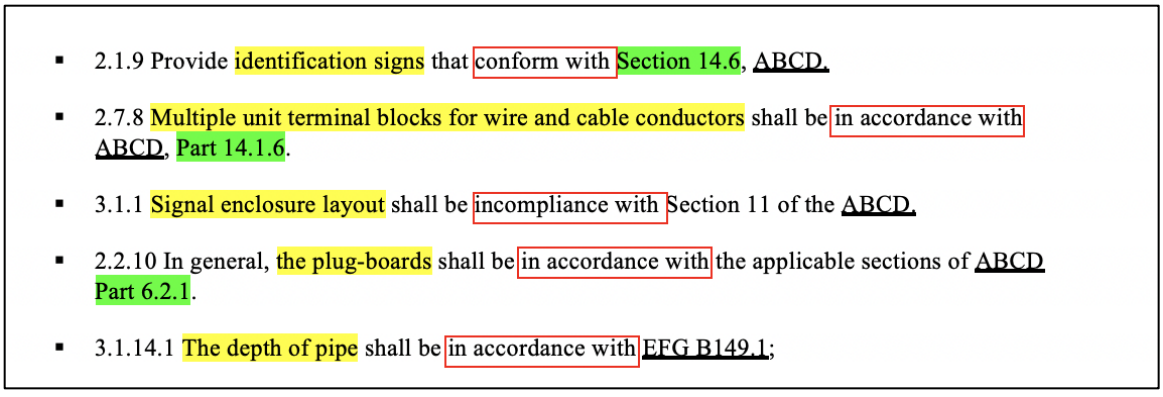

Figure 5.1 Sample of regulatory requirements from the case study contact

As mentioned in (Rahmani et al., 2020) reporting phrases are commonly used for external referencing. In our case study, some reporting phrases were used more than others. Examples include 'in accordance with'; different tenses and forms of the verb "conform" for example ‘conforms to’, ‘conform with’, and ‘conforming to’; and different tenses and forms of the verb "comply" for example ‘comply with’. The meanings of these reporting phrases are defined and their use in regulatory requirements are described in the following subsections. To provide better understanding of grammatical structures in regulatory requirements.

The reporting phrase *in accordance with*

in accordance with is the most widely used reporting phrase in our contractual document case study. The Merriam-Webster dictionary defines this phrase as "in a way that agrees with or follows (something, such as a rule or request),". Formats and examples of its use from the actual contract are shown in Table 5.1, Table 5.2 and Table 5.3.

Table 5.1 Regulatory requirement decomposition format A

Format A:			
Something shall be done *in accordance with* the External Document.			
Target Phrase	**Reporting phrase**	**External source**	**Target Position**
something	*in accordance with*	External Document	-
Example:			
feeder breakers for Fire Alarm Panels shall be designed in accordance with CAN123			
Target Phrase	**Reporting phrase**	**External source**	**Target Position**
feeder breakers for Fire Alarm Panels	in accordance with	CAN123	-

Table 5.2 Regulatory requirement decomposition format B

Format B:			
Do *Something in accordance with the* External Document			
Target Phrase	**Reporting phrase**	**External source**	**Target Position**
something	*in accordance with*	External Document	-
Example:			
Do precast concrete work in accordance with CAN123 and CAN456.			
Target Phrase	**Reporting phrase**	**External source**	**Target Position**
precast concrete work	in accordance with	CAN123	-
precast concrete work	in accordance with	CAN456	-

Table 5.3 Regulatory requirement decomposition format C

Format C:			
Something shall be in accordance with the External Document.			
Target Phrase	**Reporting phrase**	**External source**	**Target Position**
something	*in accordance with*	External Document	-
Example:			
Multiple unit terminal blocks for wire and cable conductors shall be in accordance with ABCD, Part 14.1.6			
Target Phrase	**Reporting phrase**	**External source**	**Target Position**
Multiple unit terminal blocks for wire and cable conductors	in accordance with	ABCD	Part 14.1.6

The reporting phrase *Conform to/with*

Another commonly used reporting phrase is "conform", which is usually used with the preposition "to" and "with", and according to the Cambridge dictionary, *conform to/with*, means "to obey a rule or reach the necessary stated standard". Some examples of using this verb in different regulatory requirements are shown in Table 5.4 and Table 5.5.

Table 5.4 Regulatory requirement decomposition format D

Format D:			
Something shall *conform to* the External Document.			
Target Phrase	**Reporting phrase**	**External source**	**Target Position**
something	*conforms to*	External Document	-
Example D:			
The direct current resistance of the conductors shall conform to CAN123			
Target Phrase	**Reporting phrase**	**External source**	**Target Position**
The direct current resistance of the conductors	*conform to*	CAN123	-

Table 5.5 Regulatory requirement decomposition format E

Format E:			
Do *Something* conforming to the External Document			
Target Phrase	**Reporting phrase**	**External source**	**Target Position**
something	*conforming to*	External Document	-
Example E:			
Provide ladder rungs conforming to CAN123			
Target Phrase	**Reporting phrase**	**External source**	**Target Position**

ladder rungs	conforming to	CAN123	-

The reporting phrase *Comply*

Comply is another formal verb which is used widely for referencing a rule or regulation, and its definition according to Cambridge dictionary is "to act according to an order, set of rules, or request". The way it is used in regulatory requirements with an example is shown in Table 5.6.

Table 5.6 Regulatory requirement decomposition format F

Format F:			
Something shall comply with the External Document			
Target Phrase	**Reporting phrase**	**External source**	**Target Position**
something	*comply*	External Document	-
Example F:			
The Control System installation shall comply with CAN123, Part 1.			
Target Phrase	**Reporting phrase**	**External source**	**Target Position**
The Control System installation	comply with	CAN123	Part 1

These are just a few examples of reporting phrases. These formats can be generalized to other reporting phrases.

In the examples shown in above tables, for each reporting phrase, we are looking for answers to the question of which action or entity must comply with the law or the relevant standard.

The answer to this question can be obtained by knowing the grammatical structure of each sentence. In the above formats, we are looking for additional information about the Target Phrase *something* in the external documents. In order to extract this information from sentences each of which has a different grammatical structure, we need to find the connection between different words in the sentence. Tokenization, Dependency parsing, and POS tagging was used for this purpose.

Sub-step A.2.1 Tokenization

In this sub-step, requirements were tokenized and split into separate words to be prepared for assigning POS tags and dependency tags in following sub-steps. For word tokenization, the Spacy library (Section 2.12) was used.

Sub-step A.2.2 Dependency parsing

Dependency parsing is the process of extracting a dependency parsing tree of a sentence that represents the binary relation between words of a sentence based on syntactic structure. dependency grammar is used to show the relation between each dependent word to its head and head of the entire structure (root), which is also root of the generated tree. key dependency relations (Section 2.12, Table 2.1) that are used in this study are:

nsubj: nominal subject

nsubjpass: passive nominal subject

dobj: direct object

For extracting this relationship between words in regulatory requirements, the Spacy library was used. In Spacy, *dep_* attribute is used for assigning syntactic dependency relation to words in a sentence. Therefore, for extracting a word with particular dependency

relation, for example, *nsubj*, we can iterate over the words and extract the *dep_* that is *nsubj*.

Sub-step A.2.3 POS tagging

POS tagging technique (Section 2.11) assigns a grammatical tag to each word in a sentence based on the context and meaning of a word. For POS tagging, the Spacy library was used. In Spacy, *pos_* attribute assigns string type of universal POS tags to each token.

Below you can see a graphical representation of dependency parsing for different regulatory requirements formats that refer to an external source. For demonstrating the links between words, Displacy visualizer from Spacy library was used (Section 2.12.1). Arrows depict the dependency relation from dependent to head and the labels on arcs indicate the type of relationship. Under each word simple POS tags are depicted.

Figure 5.2 illustrates the dependency structure of the format A of regulatory requirements from Table 5.1. As we can see, this is a passive sentence, and *something* is a Target Phrase that we want to extract from the sentence. *done* is the main verb and also root of the sentence, and it is the head of *something*, and the dependency relation is *nsubjpass*. So, by identifying a noun phrase which is the *nsubjpass* of the sentence, we can extract the Target Phrase.

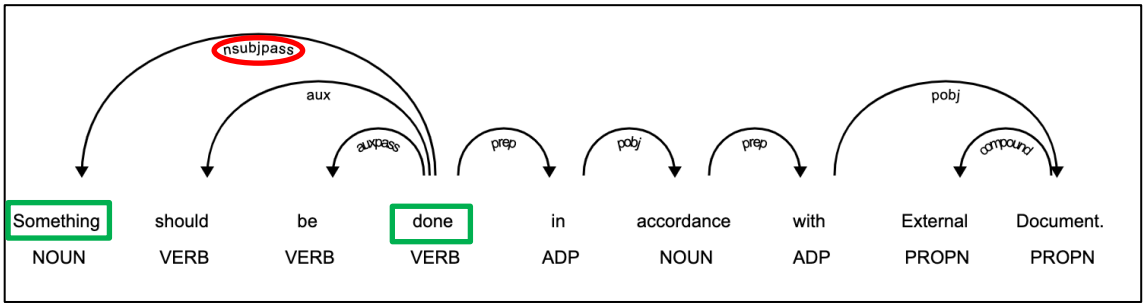

Figure 5.2 Dependency structure of the regulatory requirement Format A [see Table 5.1]

Sub-step A.2.4 Navigating the dependency parsing Tree and Subtree

The output of dependency parsing is a tree, so it has all the features of a tree. To extract information about the grammatical structure of a sentence, we can navigate in several ways to extract links between words. In real world data, a Target Phrase is not just a word, and extracting single components like *nsubj*, *nsubjpass*, and *dobj* of the sentence is not enough for our purpose. So, we should extract the subtree of relevant components and to accomplish this task, we have to look for a sequence of words from the leftmost token to the rightmost token which have a syntactic relation with each of the aforementioned components that form a subtree.

As you can see in Figure 5.3, which is a dependency structure of a sample requirement from our case study contract with format A, the Target Phrase is not a single word. To extract the Target Phrase, we determine its subtree. In Figure 5.3, the subtree of the *nsubjpass* is illustrated by a blue dotted rectangle.

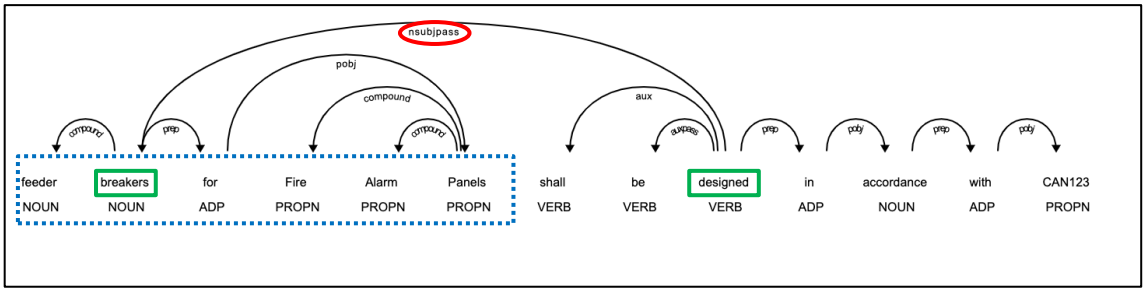

Figure 5.3 Dependency structure of regulatory requirement Example A [see Table 5.1]

Figure 5.4 illustrates the dependency structure of the format B of regulatory requirements in Table 5.2. In this type of sentences *something* is the Target Phrase that we want to extract from the sentence. *do* is the main verb and also root of the sentence, and it is the head of

something, and the dependency relation is *dobj*. So, by identifying the *dobj* of the sentence, we can extract the Target Phrase.

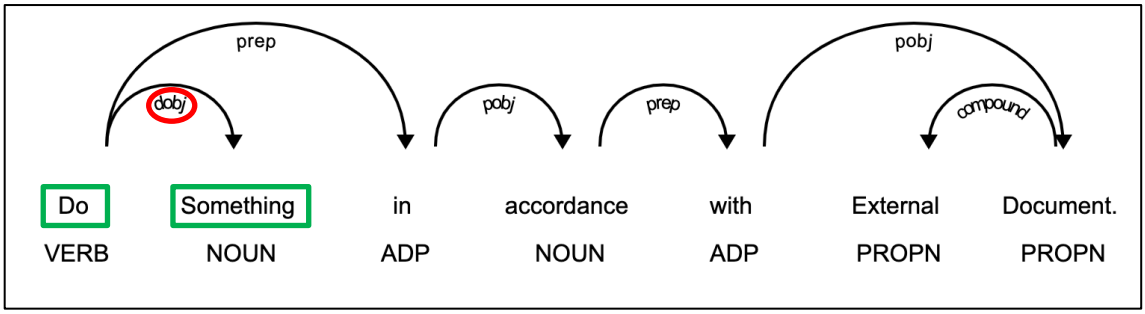

Figure 5.4 Dependency structure of regulatory requirement format B [see Table 5.2]

As you can see in Figure 4.4, which is dependency structure of a sample requirement from our case study contract, same as previous examples the Target Phrase is not a single word, so we should extract its subtree to extract the Target Phrase. In Figure 5.5, the subtree of the *dobj* is illustrated by a blue dotted rectangle.

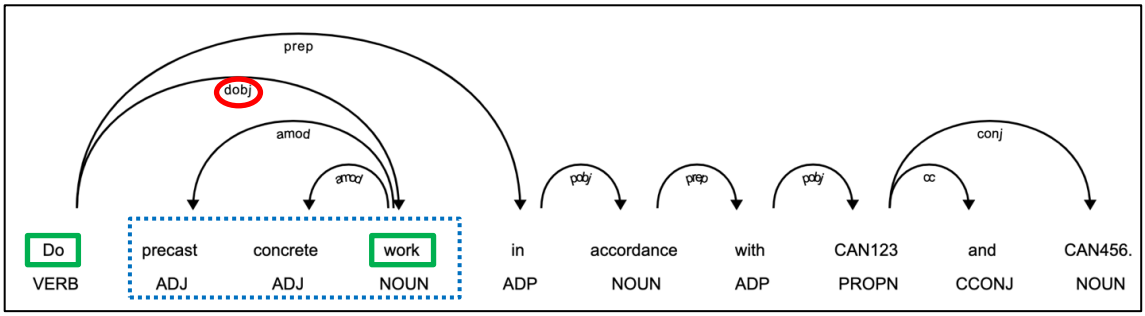

Figure 5.5 Dependency structure of regulatory requirement example B [see Table 5.2]

Figure 5.6 depicts the dependency structure of the format C of regulatory requirements. *be* here is our main verb and also root of the sentence and there is *nsubj* dependency relation between *be* as head with its dependent which is *something.* In other words, *something* is the Nominal Subject of *be*, so by identifying *nsubj* of the sentence we can extract the Target Phrase.

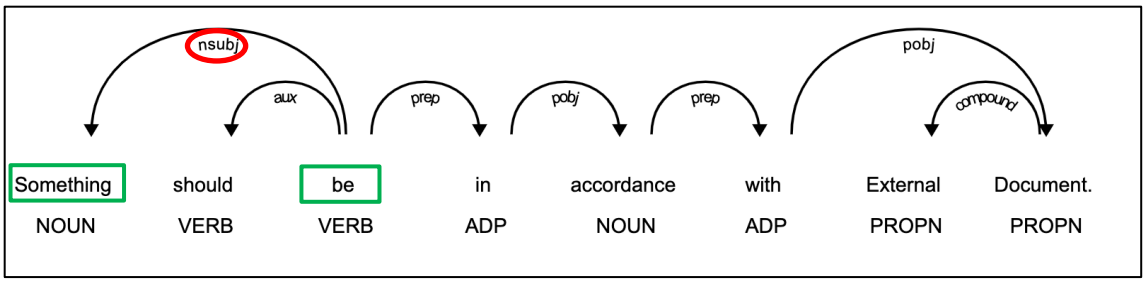

Figure 5.6 Dependency structure of regulatory requirement format C [see Table 5.3]

Figure 5.7 is illustrating the dependency relation between words in example C and as well as previous examples, we need to extract the subtree of *nsubj* to extract the Target Phrase.

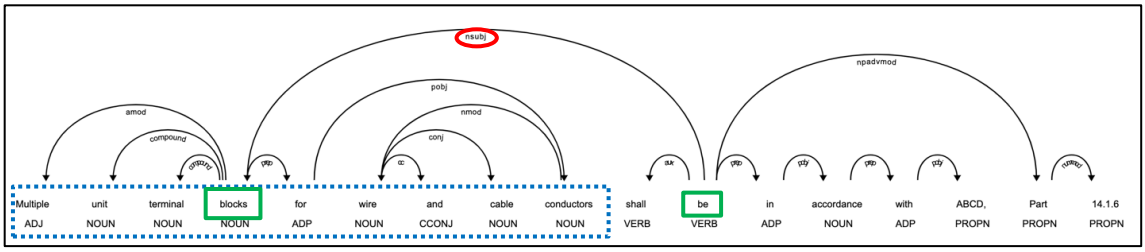

Figure 5.7 Dependency structure of regulatory requirement example C [see Table 5.3]

Figure 5.8 shows format D regulatory requirement which is similar to format B except that the reporting phrase is different. So, similar to format B, by extracting *dobj*, we can extract the Target Phrase.

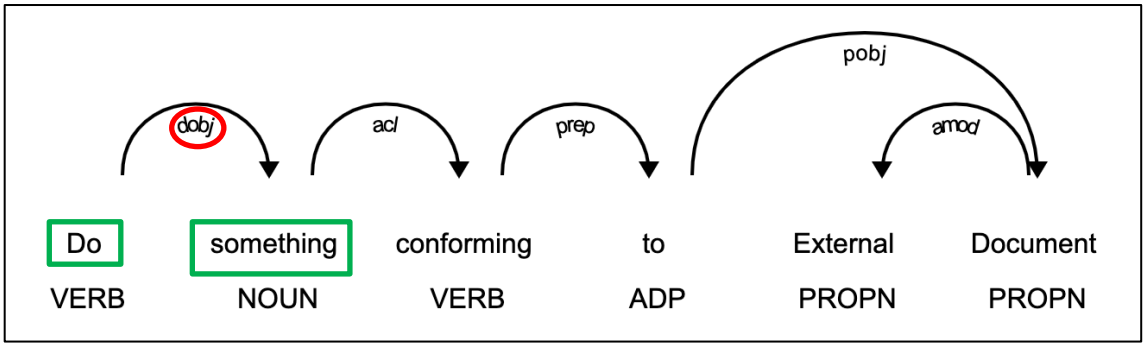

Figure 5.8 Dependency structure of regulatory requirement format D [see Table 5.4]

Figure 5.9 which is the visualization of dependency structure of regulatory requirement in Table 5.4, shows that we need to extract *dobj* subtree to retrieve the Target Phrase.

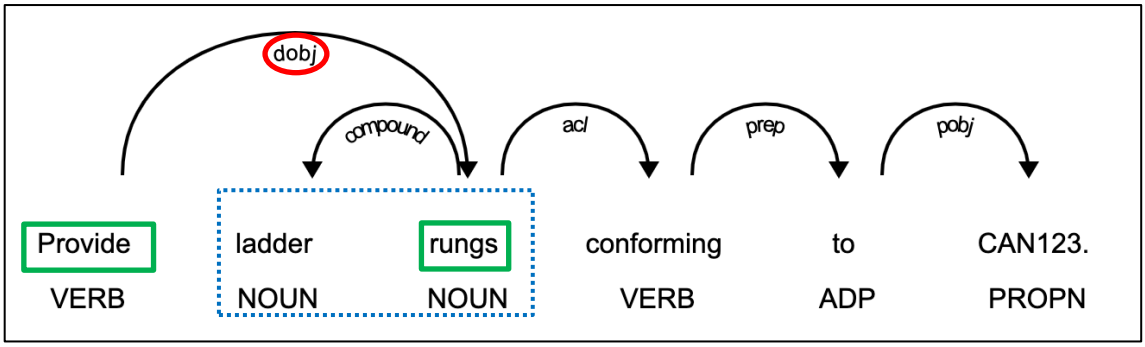

Figure 5.9 Dependency structure of regulatory requirement example D [see Table 5.4]

Figure 5.10 shows the dependency analysis of another format using the reporting phrase, *conform*. In format E, *something* has *nsubj* relation with its head *conform*, so by extracting *nsubj* of the sentence we could extract the Target Phrase.

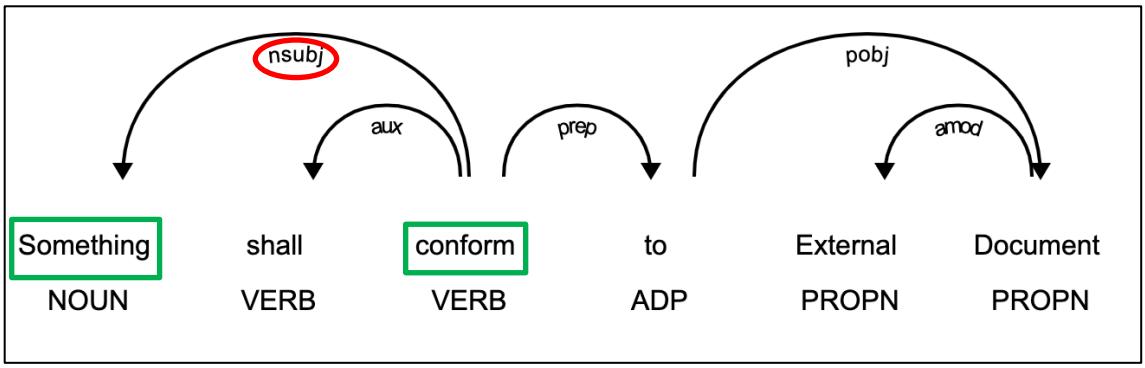

Figure 5.10 Dependency structure of regulatory requirement format E [see Table 5.5]

Figure 5.11 is showing the dependency relation between words in example E and as well as previous examples, we need to extract the subtree of *nsubj* to extract the Target Phrase. In this example *resistance* is *nsubj* of the verb *conform*. Our Target Phrase in this sentence is subtree of *resistance* which is specified in dotted blue rectangle.

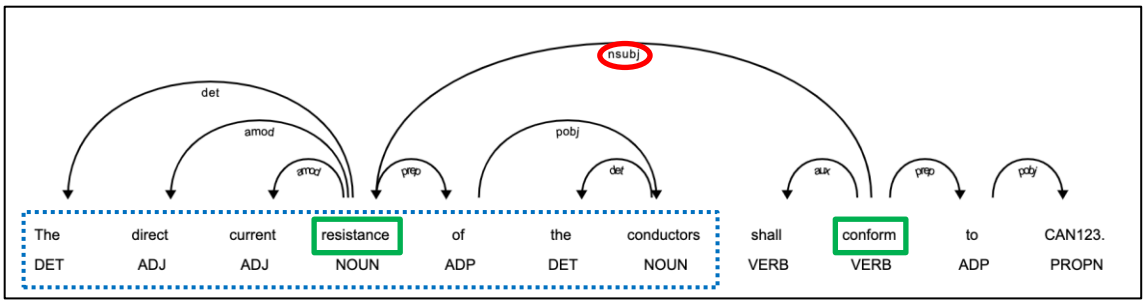

Figure 5.11 Dependency structure of regulatory requirement example E [see Table 5.5]

As shown in the figures, in sentences with the following format, the main component of the Target Phrase is a noun phrase which is the *nsubj* of the root (main verb).

<Tareget phrase> shall be <Reporting phrase><the External Document>

If the sentence is passive and similar to the following formats, the main component of the Target Phrase is a noun phrase which is the *nsubjpass* of the sentence.

<Tareget phrase> shall be done <Reporting phrase><the External Document>

<Tareget phrase> shall be <verb> <Reporting phrase><the External Document>

In sentences with the following structure, Target Phrase is a noun phrase which is the *dobj* of the sentence.

Do <Tareget phrase> <Reporting phrase><the External Document>

<verb><Tareget phrase> <Reporting phrase><the External Document>

If after passing the previous steps, the Target Phrase does not have any values, there might be because of the format and structure of the sentence. In our case study contract, we faced with several regulatory requirements such as:

Marking of terminals and leads : In accordance with NEMA MG1 .

Ground transformers as required by EFG C9 , EFG C22.2.

In this situation, a sentence was divided into two parts: before reporting phrase and the remaining of the sentence. We cannot employ this method for all the requirements because it might lead to extracting extra tokens as part of the Target Phrase.

Verbs in regulatory requirements

We found that not all verbs are useful for retrieving information from external sources because of their meaning. For example, many regulatory requirements contain verbs which are not technically specific such as ensure, provide, do, prepare, conduct and perform, therefore, there is the possibility they will have a negative effect on returned results in the information retrieval step. verbs will be examined closely in our future studies.

Sub-step A.3 extract the Target Position

Legal texts are written in a hierarchical structure and contain grouping concepts that are ranked in order. Each grouping concept is represented by a marker such as a section or part and its corresponding number, divided by delimiters such as a dot or a dash to show the level of that grouping concept within the legal text.

The Target Position helps to understand the place of referenced text within the external documents, which provide additional information to existing regulatory requirements. The Target Position makes the process of finding relevant text easier and the results more accurate. However, in our case study contract, only 12% of regulatory requirements contain the Target Position. Among all Target Positions in our case study, 81% was a single well-formed position that shows the referenced text's exact position. 2% of regulatory requirements contains multiple positions, or there might be several sub-sections between them. Since we do not have the text schema for external documents, we cannot interpret positions such as parts 7.2.36 to 7.3.50.

In our study, we identified Target Positions by using rule-based matcher and regular expressions in the Spacy library. it finds the array of tokens utilizing rules that specified the attributes of the tokens.

In this study, Although, we can extract other forms of Target Positions from regulatory requirements, only single well-formed positions which contain marker in external sources could be used for information extraction.

Step B. Retrieve relevant information from the cited legal documents or standards

The purpose of this step is to extract the text from the cited external sources related to each regulatory requirement, based on Target Phrase and Target Position. This requires a search engine and we chose Elasticsearch due to its real-time search capability and ease of use. For the purpose of this study we used Elasticsearch's default setup configuration.

First, the unstructured content of each page of available external documents was converted to JSON semi-structured format.

Then, using bulk API, the content of external documents which the contract refers to for additional information stored in an Elasticsearch index.

For this investigation, topic modelling is not helpful because we know what we are looking for, which is the target phrases in external documents. In addition, each page of the external document belongs to a section of a regulation or standard, and each section has a specific topic.

Topic modelling is practical when someone needs to understand the high-level themes in several documents and find the documents of interest (Hu et al., 2014).

In the following, we will go into the details of each sub-steps.

Sub-step B.1 Create an index to store the content of external documents

In this sub-step, we created an index to store the content of external documents that the contract refers to for additional information. An index is equal to a database in RDBMS. The documents can be stored as JSON objects in the Elasticsearch index, so we converted

the unstructured content of available external documents to JSON format to become ready for conducting a search operation in Elasticsearch.

The name of each external document, content of each page and also the page number of each file were stored in an Elasticsearch index. Since the structure of each documents is different and there are a large number of them all in pdf format, so the fastest and easiest way is to store the content of each page with its page number rather than any other structure such as sections.

This step provides our tool to be prepared to retrieve relevant information by querying.

Sub-step B.1.1 Index Creation

First, an index in Elasticsearch was created. Since the main type of data, we work with is text, for the text analysis configuration, we created an analyzer (Section 2.17.4) with the following building blocks in the index:

- Tokenizer:
  - *standard* tokenizer to split the text into word boundaries
- Token filter:
  - *Lowercase* for converting all tokens to lowercase
  - *porter_stem* to stem the text
  - *stop* to remove English stop words

The analyzer was created for both searching and indexing the content of each page to ensure the format of inverted index matches with the text in our query. For the mapping properties(Section 2.17.3), we defined text for files name and content of each page and a numeric type for page numbers.

Sub-step B.1.2 prepare data for bulk indexing

For preparing data to be indexed in Elasticsearch, we gathered all available external documents in a directory. Since all the external documents were available in pdf format, we defined a function to extract the plain text from each page of individual files and create a dataframe with three main columns for the convenience of working with data. Columns were file_name, content, and page_no. The names of the external documents were stored in the file_name column, the text of each page and the corresponding page number were stored in the content column and page_no column, respectively. To capture the text from each pages of our pdf files, we used the PYPDF2 toolkit which is capable of extracting text from pdf files page by page (Section 2.18). After that to prepare the data for bulk indexing (Section 2.17.2), we needed to iterate over the rows of the dataframe and create data with two elements including the primary data, combined with the operational meta-data like "_index," and "_id,".

Sub-step B.1.3 Insert the data into Elasticsearch index

We used bulk API (Section 2.17.2), in Python Elasticsearch client (Section 2.17.1) to insert all the data into the index with a single request. Using bulk API, several operations such as indexing, deleting, etc. could be performed in only one command, leading to significant improvement in indexing speed. By performing this sub-step, the external documents are stored in an Elasticsearch index, and it is ready for querying step for each regulatory requirement.

Sub-step B.2 Create queries by document name, Target Phrase, Target Position

Using Elasticsearch client in Python, we developed a querying step based on the outputs of step A. This can search from the content of external documents and find the relevant text in each page of the documents. The query, search for the exact matches (Section 2.17.9.2) with the name of external source and Target Position (if there is any) and match (Section 2.17.9.2) with the Target Phrase. In order to make sure the name of reference in our directory matches with the extracted external cross-reference we manually checked and fixed any contradiction.

Sub-step B.2.1 Full-text Query builder

We obtained the following keywords from preprocessing step and output of step A.

- Reference Title
- Target Phrase
- Target Position (if present)

Moreover, we have an index with the content of each page of external documents, the name of reference and its page number.

Since a Target Position might not exist, we first need to check the presence of the Target Position both in the requirement and in the content of external references to provide the requirement analyst with an appropriate message. If there is no Target Position in our requirement, we only search the Target Phrase in the content and reference name in the file name.

```
{
"query":{
  "bool": {
    "must": [
      {"match_phrase":{"name": reference_title}},
      {"match": {"content": target_phrase}}
      ]
    }
  }
}
```

Figure 5.12 Query without Target Position

If the Target Position is well-formed and refers to an exact position, we will consider all three keywords in the query.

```
{
  "query": {
    "bool": {
      "must": [
        {"match_phrase": {"name": reference_title }},
        {"match": {"content": target_phrase}},
        {"match_phrase": {"content": target_position}}
        ]
    }
  }
}
```

Figure 5.13 Query with Target Position

Another situation is when the Target Position is available in the requirement, but it is not a single well-formed position, so it would not be helpful in information extraction in our solution. In this case, we proceed as if there is no Target Position and use the query in Figure 5.12.

Additionally, we used highlighting (Section 2.17.10) as part of our query. It is a feature that allows us to show the user the exact words and sentences in the content of each page that led that page displayed as a result by executing the query.

Sub-step B.3 Retrieve relevant text from external documents based on Target Phrase and Target Position

By running the query, the outcome would be the name of the relevant external documents, the content of relevant pages and the file's page number, all ordered by a relevancy score. In addition, it shows the text in each page that makes the program return that page.

Elasticsearch is capable of sorting results based on relevance score. Relevance score (Section 2.17.11)calculates how similar is the retrieved document with the query, and greater scores represent more similarity. In this study, BM25 (Section 2.17.11.1) was used to rank the relevancy of extracted pages from external documents.

Chapter 6

6 Output of the proposed solution

This Section provides details about the final output of proposed solution and two different ways of displaying the output to the requirement analyst. After executing the query as the last step in methodology Section, the relevant information for each requirement was extracted from the cited reference in the contract to support the analyst with this challenging task. The first view is the extracted text from relevant external document and the second one is a list view of relevant pages of each external document, both in order with higher relevance scores on top.

The first view is the content of each page of external documents related to the regulatory requirement in which the sentences and words that lead to extraction of that page are also highlighted.

After running the query, we show a message to requirement analyst about the status of Target Position and if it is considered in the output or not.

Consider the regulatory requirement in Figure 6.1

Provide identification and information signs in accordance with ABCD.

Figure 6.1 Example of regulatory requirement

Figure 6.2 depicts a relevant page to the regulatory requirement in Figure 6.1 which is extracted based on the name of external document cited in the requirement and the Target Phrase. However, both the name of external document and the Target Phrase were extracted automatically using preprocessing step and step A of the proposed solution respectively.

```
score: 10.941795

page: 73

Standard: ABCD.pdf

C&S Manual Part 8.3.1
2002 _____
- 2 Œ E. Dielectric Requirements  1. See Manual Part 1.4.1, Section E.  2.
See Manual Part 11.5.1 (Recommended Environmental Requirements for Electri
cal & Electronic Railroad Signal System Equipment), Section C.1., Class C.
F. Identification  1. Required identification and information should be ei
ther engraved or stamped on the unit, or the name plate should be securely
fastened or adhered to each unit to give the following information:  a. Na
me of manufacturer.  b. Manufacturer's drawing or reference number.  c. Se
rial number.  d. Normal operating voltage.  e. Other information as requir
ed by application of Section B.3. through B.5.  2. A plate or tag should b
e securely fastened or adhered to each unit showing a connection diagram,
as applicable.
```

Figure 6.2 An extracted page for regulatory requirement in Figure 6.1

The output text contains:

- name of standard: ABCD
- page number: page 73
- score: 10.941795
- content of the page
- relevant fragment of text highlighted in each page

The name of the standard is shown because sometimes, the name of the standard cited in the requirement is only the name of an association or an organization that issued the codes or standards and not the exact name of a document. Since they have developed many codes and standards, it is helpful to show the requirement analyst where exactly the extracted content arises. Therefore, the exact name of the reference based on the extracted content is helpful.

For example, in the requirement in Figure 6.3, EFG is the acronym for a standard development organization which in our case study we have more than fifty different files issued by that organization for different parts of the system.

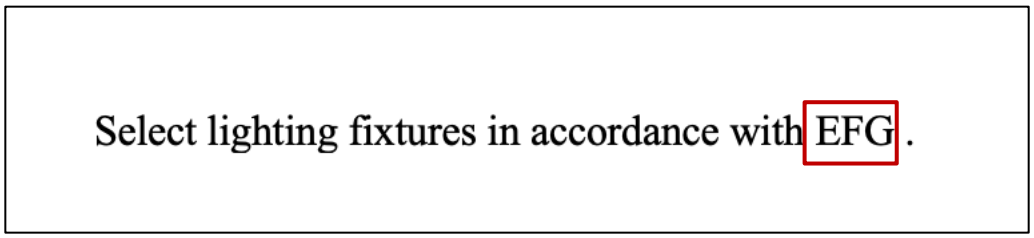

Figure 6.3 Example of regulatory requirement

And as you can see in Figure 6.4 the relevant text returned for this requirement scattered a cross several different standards, which are highlighted in below figure, such as EFG-W47.2-M1987R2008, EFG-Z460-05 , EFG-C22.2No.56-04, EFG-W117.2-06, EFG-B137Series-05, EFG-Z412-00, etc.

```
page697 EFG-S6-06.pdf score: 11.363991
page140 EFG-Z412-00.pdf score: 10.755578
page124 EFG-Z412-00.pdf score: 10.360926
page614 EFG-S6-06.pdf score: 9.413914
page429 EFG-B137Series-05.pdf score: 9.217261
page210 EFG-B137Series-05.pdf score: 9.188959
page209 EFG-B137Series-05.pdf score: 9.058483
page20 EFG-C22.2No.56-04.pdf score: 7.9575996
page94 EFG-Z460-05.pdf score: 7.761076
page32 EFG-W59.2-M1991.pdf score: 7.7472954
page223 EFG-B137Series-05.pdf score: 7.7177124
page67 EFG-B137Series-05.pdf score: 7.5533037
page68 EFG-W47.2-M1987R2008.pdf score: 7.028804
page93 EFG-Z460-05.pdf score: 6.6964273
page30 EFG-C22.2No.56-04.pdf score: 6.6164656
page57 EFG-W117.2-06.pdf score: 6.609292
page172 EFG-B137Series-05.pdf score: 6.5184145
page128 EFG-Z412-00.pdf score: 6.2784395
page126 EFG-Z412-00.pdf score: 6.168564
```

Figure 6.4 Relevant pages extracted for regulatory requirement in Figure 6.3

The page number is provided to help analysts directly access the extracted information in the original file.

The relevance score which is calculated using BM25 algorithm is provided to sort the extracted content based on the matches with the executed query.

The content of each page is provided because sometimes an important information about the Target Phrase come as a list of features or constraints after the extracted text.

The fragments of each page that made the page to be extracted also displayed in highlighted format to show the analyst where the matches are in the content of each page.

Chapter 7

7 Empirical Evaluation

This chapter provides the evaluation of our two-step framework through our case study contract. We start this chapter by evaluating the Target Phrase extraction, then how the ground-truth was built and explanation about the notions that we used for assessment. Then, the evaluation procedure for extraction of Target Position is explained, and finally, the methods which was used to show the efficiency and effectiveness of Step B of our proposed solution (information extraction from external sources) is described.

7.1 Target Phrase extraction evaluation

To measure the accuracy of our proposed method for Target Phrase extraction we used the elements of confusion matrix (Section 2.19) to calculate Precision (Section 2.19.1), Recall (Section 2.19.2) and F-measure (Section 2.19.3).

7.2 Ground truth construction for Target Phrase extraction

We created the ground truth to evaluate the Target Phrase extraction because our case study data set is a raw contract text, and we do not have any train data set and test data set. Therefore, for this task, 200 of 657 regulatory requirements were selected randomly which is 30% of data set and it is recommended for test data (Singh et al., 2021). Then all 200 regulatory requirements were analyzed manually to extract the Target Phrase in each of them to assess the output of our automated solution against 200 ground truth. In this procedure, to prevent biased conclusions first, 200 regulatory requirements were analyzed by the first author. Then, for examining reliability, the second author independently analyzed 20 (10%) of regulatory requirements. The agreement was calculated employing Cohen’s kappa coefficient, and “perfect agreement” was obtained (Cohen, 1960).

7.3 Evaluation metrics for Target Phrase

We assessed the output of automatic extraction of Target Phrases against ground truth utilizing below notions:

- TP: if the extracted Target Phrases exactly matched with ground truth it was considered as True Positive
- FP: if the extracted Target Phrases did not match with ground truth it was considered as False Positive
- FN: if our method was not able to extract any Target Phrase it was considered as False Negative
- TN: if the sentence did not contain any Target Phrase and our method did not extract any Target Phrase it was considered as True Negative

7.4 Target Position extraction evaluation

We manually extracted all the Target Positions from all the 657 regulatory requirements, then the output of automated solution was compared with the actual Target Positions. In our case study contract among 657 regulatory requirements, 81 contained Target Position which is only 12% and we were able to extract all of them.

7.5 Evaluation metrics for Target Position

We used elements of confusion matrix to measure the performance of Target Position extraction.

- TP: if the extracted Target Position exactly matched with manual extraction of Target Position it was considered as True Positive
- FP: if the extracted Target Phrases did not match with manual extraction of Target Position it was considered as False Positive
- FN: if our method was not able to extract any Target Position it was considered as False Negative

- TN: if the sentence did not contain any Target Position and our method did not extract any Target Phrase it was considered as True Negative

7.6 Statistics

Our evaluation results for Target Phrase and Target Position extraction is presented in Table 7.1

Table 7.1 statistics for Target Phrase and Target Position extraction

	Result of automatic extraction				Accuracy		
	TP	FP	FN	TN	Precision	Recall	F-measure
Target phrase	159	38	2	1	0.81	0.98	0.89
Target Position	81	0	0	576	1.00	1.00	1.00

7.6.1 Analysis of Target Phrase extraction

In the Target Phrase extraction sub step, our solution obtained Precision = 0.81, which show we were able to identify 81% of the Target Phrases correctly in regulatory requirements.

In the following subsection we will analyze samples of False positives, False Negatives and True Negatives for automatic Target Phrase extraction by using the 200 random regulatory requirements.

7.6.1.1 Analysis of False Positives in Target Phrase extraction

By analyzing the FPs we found that the main reason is due to complexity of the sentences and incorrectly identifying dependency relations.

see the examples below:

Example 1: Fouling, switch heel, and frog bonds shall be installed at all turnouts , crossovers , double slip switches , as indicated in the CDS.

- Incorrectly identified Target Phrase: heel (as *dobj* of the verb switch)
- “switch” in this sentence is not a verb and “switch heel” is a physical component that must be installed in accordance with a standard.

Example 2: Ensure that reinforcing steel conforms to EFG G30.18-M.

- Incorrectly identified Target Phrase: steel (as *dobj* of the verb reinforcing)
- “Reinforcing” in this sentence is not a verb and “reinforcing steel” is a physical component that must be in accordance with a standard.

7.6.1.2 Analysis of False Negatives in Target Phrase extraction

By analyzing the FNs we found that the main reason that our solution could not extract the example sentence is that in such rare cases Target Phrase are object of preposition and not direct object.

Example: Refer to the CDS for conceptual details of signal bridges.

- Incorrectly no Target Phrase was extracted
- Target Phrase: conceptual details of signal bridges

7.6.1.3 Analysis of True Negative in Target Phrase extraction

True Negatives are those sentences we mentioned in Step A.1 that are among 1% that does not have any Target Phrase and our solution also did returned any values for such sentences.

Example: Comply with the Occupational Health and Safety Act , and Regulations for Construction Projects Ontario Regulation 659/79.

- correctly no Target Phrase was extracted

7.6.2 Analysis of Target Position extraction

In our case study contract, for Target Position only part, section, and table were used as markers following by numbers. The format of the numbers was all in numeric with dots as delimiters.

7.7 Evaluation metrics for requirement-related information extraction

Two main aspects that often is used for evaluating IR systems are efficiency and effectiveness. Efficiency is measured in terms of time of response, and measuring effectiveness is based on the concept of relevance. The time of response which is the time between running a query and receiving the output is the most important feature which the user experiences. (Büttcher et al., 2016).

Table 7.2 shows the query processing time in milliseconds and the top 10 relevance score (Section 2.17.11) for 20 individual queries, calculated using BM25 (Section 2.17.11.1). As we can see, the average query processing time is 52.55 milliseconds.

Table 7.2 processing time and relevance score for samples of queries

Query	Query Processing Time (ms)	score 1	score 2	score 3	score 4	score 5	score 6	score 7	score 8	score 9	score 10
Q1	143	15.77	15.56	15.51	14.39	13.80	13.59	13.35	12.80	12.20	12.14
Q2	136	24.12	21.83	21.53	21.08	21.05	21.05	20.74	20.41	20.06	19.79
Q3	41	13.76	13.71	13.67	13.59	13.50	13.17	12.97	12.93	12.81	12.59
Q4	58	15.08	14.04	13.59	13.59	13.31	13.29	12.89	12.84	12.79	12.46
Q5	49	19.88	19.87	19.30	19.22	18.77	18.63	18.62	18.62	18.59	18.31
Q6	27	19.35	-	-	-	-	-	-	-	-	-
Q7	49	19.67	19.43	19.39	19.30	18.62	18.54	18.49	18.42	18.41	18.31
Q8	41	22.89	22.73	22.71	21.49	21.41	21.16	21.06	20.62	20.41	20.39
Q9	78	28.99	26.44	25.59	25.47	24.21	23.96	23.62	23.31	23.24	21.38
Q10	39	18.00	17.85	17.63	17.63	17.61	16.20	14.29	14.21	14.19	14.05
Q11	45	16.13	15.54	15.30	15.23	14.95	14.86	14.56	14.37	14.00	13.99
Q12	20	32.53	29.68	25.39	24.21	24.21	21.42	-	-	-	-
Q13	16	27.44	24.79	23.93	23.04	22.46	21.18	-	-	-	-
Q14	26	16.66	15.23	15.11	14.99	14.28	14.27	13.58	-	-	-
Q15	19	38.92	36.24	34.92	19.53	-	-	-	-	-	-
Q16	22	31.73	29.74	29.31	27.99	27.54	20.30	19.97	19.66	-	-
Q17	15	38.79	14.41	12.81	12.51	12.07	11.55	9.65	-	-	-
Q18	82	12.20	11.95	11.88	11.82	11.71	11.30	11.27	10.74	10.55	9.65
Q19	109	11.97	11.48	10.94	10.45	10.15	10.08	9.66	9.59	9.55	9.55
Q20	36	14.11	13.59	10.90	10.79	10.67	10.37	9.74	9.08	8.10	7.97
Average query processing time (ms)	**52.55**										

The output of our proposed solution supports requirement analysts with automatic extraction of requirement-related information from external cross-references in the project contract, which was previously possible to be done manually. Therefore, considering the importance of system compliance and consequences of non-compliance, the presented

solution would save an enormous amount of time and effort for the requirement analyst while eliciting requirements from external documents such as standards and regulations to ensure compliance at requirement level.

Manual handling of external cross-references in the contract can take several days. However, our solution is capable of extracting following information for each regulatory requirement in a matter of milliseconds.

- name of external document
- relevant pages
- relevance score
- content of the page
- relevant fragment of text highlighted in each page

Chapter 8

8 Comparison with related and foundational work

This chapter compares our solution with related work (Chapter 3) and foundational work (Chapter 4). First, in Section 8.1, a high level comparison with previous work and expansion of Table 3.2 is provided. Then in Section 8.2, we explained how our solution compares with related work before (Rahmani, 2020), and finally, in Section 8.3, we described what we build on (Rahmani, 2020)’s achievements.

8.1 High level comparison with previous work

Table 8.1 shows that automatic solutions which was proposed before (Rahmani, 2020) focused on internal cross-references. Then (Rahmani, 2020) represented a novel solution which was described in Chapter 4 for identifying external cross-references in a project contract. Our work focused on automatically extracting information from external cross-references in the project contract.

Table 8.1 summary of works attempted to deal with cross-references

Reference	Type of cross-references		Task					Case study legal text	
	internal	external	Manual	Automatic	identification	classification	Referenced text extraction	Law	Contract
(Maxwell et al., 2012)	✗	✓	✓	✗	✓	✓	✓	✓	✗
(Tran et al., 2014)	✓	✗	✗	✓	✓	✓	✓	✓	✗
(Adedjouma et al., 2014)	✓	✗	✗	✓	✓	✗	✓	✓	✗
(Sannier et al., 2016)	✓	✗	✗	✓	✗	✓	✗	✓	✗
(Rahmani, 2020)	✗	✓	✗	✓	✓	✗	✗	✗	✓
Our Work	✗	✓	✗	✓	✗	✗	✓	✗	✓

8.2 Comparison with related work (Chapter 3)

Table 8.2 compares the related work which was described in Chapter 3 with our work in terms of type of cross-references, case study legal text, keywords for extracting referenced text, extractable position and referenced document structure.

Table 8.2 Comparison with related work

No.	Comparable subject	Related work	Our solution

1	**Type of cross-references**	• (Maxwell et al., 2012)
		  ○ External cross-references
		• (Tran et al., 2014)
		  ○ Internal cross-references
		• (Adedjouma et al., 2014)
		  ○ Internal cross-references
		• (Sannier et al., 2016)
		  ○ Internal cross-references
			External cross-references

2	**Case study legal text**	• (Maxwell et al., 2012)
		  ○ HIPAA
		  ○ GLBA
		• (Tran et al., 2014)
		  ○ Japanese National Pension Law
		• (Adedjouma et al., 2014)
		  ○ Luxembourg's Income Tax Law
		• (Sannier et al., 2016)
		  ○ Luxembourgish legislative texts
		  ○ French editions of the Personal Health Information Protection Act (PHIPA)
			An engineering project contract which contains requirements

3	**Keywords for extracting referenced text**	• (Tran et al., 2014)
		  ○ Position recognition
		  ○ Main noun of the mention
		  ○ Punctuation mark
		• (Adedjouma et al., 2014)
		  ○ Positions based on text schema
			• Target Phrase
			• Target Position
			• External cross-references

4	**Extractable position**	• (Tran et al., 2014)
		  ○ single (well-formed)
		  ○ Anaphora Indirect position
		  ○ Coordinated positions
		• (Adedjouma et al., 2014)
		  ○ Explicit/implicit
		  ○ simple/complex
			• single (well-formed)

8.2.1 Type of cross-references

As explained in Section 3.2.1, previous work only represented automatic solutions for handling internal cross-references. Our solution on the other hand, supports analysts in handling external cross-references. Cross-references to external documents are more challenging in comparison with internal cross-references (Maxwell et al., 2012).

8.2.2 Case study legal text

Previous work investigates internal cross-references in law texts, but our work examined external cross-references in an engineering project contract. To ensure compliance, requirement engineers must elicit the requirements from the standards and regulations cited in the contract in addition to all other requirements in the contract which is a complex task (Nekvi & Madhavji, 2014) .

8.2.3 Keywords for extracting referenced text

Positions are main element for extracting referenced text from internal cross-references, therefore previous work by knowing the position of the cross-reference and also identifying the position of referenced text, were able to extract referenced text. (Tran et al., 2014) first, identified the position of referenced text and then extract several candidates by checking whether the main noun of the mention (n_{head}) or its synonym exist in the antecedent. Another strategy was to extract the text from right to left until a punctuation mark. (Adedjouma et al., 2014) link the internal cross-references to the referenced text based on the text schema and position extraction. In our work, although we used Target Position,

88% of the requirements did not contain Target Position, so we extract most of the requirement-related information based on Target Phrase from the external cross-references.

8.2.4 Extractable position

Since previous work only examined internal cross-references in a single document, they were able to understand the hierarchical structure of their legal text case study. (Tran et al., 2014) manually extracted architecture of their case study legal text and (Adedjouma et al., 2014) provided a text schema class diagram for transforming the text to markup. Therefore, handling more complex positions was possible. In our work, due to large number of external documents with different structure we are able to extract single well-formed positions which point to an exact location within the external document.

8.3 Output comparison with foundational work (Rahmani's work, Chapter 4)

Consider the requirement in Figure 8.1:

2.4.1 Motor nameplates shall conform to EFG C22.2 No. 100. Stainless steel or non-corrodible alloy fixed to a non-removable part of motor enclosure with stainless steel screws.

Figure 8.1 Example of a regulatory requirement

The output of proposed solution in (Rahmani, 2020) for this requirement is shown in Table 8.3. She identified the external cross-reference in the regulatory requirement which is EFG C22.2 No. 100.

Table 8.3 output of foundational work for the requirement in Figure 8.1

Contractual Requirement Num	**Reference Level 1**
CR 2.4.1	1- EFG C22.2 No. 100 **Total: 1**

After applying Step A of our proposed solution on the requirement in Figure 8.1 we can extract *Motor nameplates* as Target Phrase which we need to find additional information about it in EFG C22.2 No. 100.

Then in Step B for extraction of related information, a simplified output is shown in Figure 8.2 for the top returned values.

```
Target Position does not exist !

Number of returned Items 60

took (milliseconds) 167

score: 17.99804

page: 58

Standard: EFG C22.2No.100-04.pdf

<other lines in the page>

shall meet the requirements of Table4 and Clauses6.2.4, 6.2.7, and 6.2
.8 when tested as follows:(a)for motors without a marked rated output,
either by loading the motor or by raising the input voltage to obtain
the rated input amperes (with the motor running); or(b)for motors with
a marked rated output, at the rated frequency or speed and delivering
the rated output for the period of time specified on the nameplate or
until constant temperatures are reached for machines having a continuo
us rating.

 <other lines in the page>

**********

score: 17.853388

page: 55

Standard: EFG C22.2No.100-04.pdf

<other lines in the page>

The following information shall be marked on machines, as applicable,
and shall appear on a nameplate, be die-stamped in a readily visible l
ocation on the frame or enclosure, or be marked in some equivalent, pe
rmanent manner:(a)the manufacturers name, trademark, trade name, or ot
her symbol of identification;(b)the model, catalogue, style, or other
type of designation;(c)the rated voltage(s);(d)the rated input for mot
ors in amperes, or for motors of less than 100 W output, the rated inp
ut in amperes or watts, and as applicable, one of the following:(i)the
service factor amperes for motors having a service factor greater than
```

1.15. (ii)the rated amperes for multiconnection motors for each connection (permanent split-capacitor and shaded-pole motors shall be marked with the highest rated amperes only);(iii)the rated amperes for part-winding-start motors having a rated load exceeding 10 A and an additional marking of the full-load current in each supply circuit conductor if there are unequal values of current in each winding supply circuit;

<other lines in the page>

****

score: 17.626814

page: 56

Standard: EFG C22.2No.100-04.pdf

<other lines in the page>

The marking shall appear on the nameplate, in the terminal box, or near the point where the supply connections are made. Motors intended for use only as components of specific equipment need not be so marked if this information is provided separately. The information need not be supplied with each motor;(v)DRIP-PROOF or DP if a drip-proof motor, WEATHER-PROOF or WP if a weather-proof motor, or TOTALLY ENCLOSED or TE if a totally enclosed motor. Additional characters may be used, e.g., DPGD (drip-proof guarded) or TENV (totally enclosed non-ventilated). This marking may be combined with that of Item (w);Note: This information may be omitted from the marking of the motor and provided separately, but not necessarily with each motor, for motors that are intended to be used only as components of specific equipment.(w)AIR OVER or AO, or AIR-OVER MOTOR or AOM, for air-over motors. This marking may be combined with that of Item (v);

<other lines in the page>

****

score: 16.196255

page: 91

Standard: EFG C22.2No.100-04.pdf

<other lines in the page>

The difference between the rated ambient, as marked on the motor nameplate, and the test ambient, where the test ambient is below (or above) the rated ambient, shall be added to (or subtracted from) the observed motor temperature before comparison with the values in this Table.

<other lines in the page>

Figure 8.2 Information extraction for requirement in figure 8.1

As shown in Figure 8.2, our solution searched inside the EFG C22.2 No. 100 as an external document and provided relevant pages and text for the regulatory requirement that the domain expert or analyst needs to consider while eliciting requirements. The output is displayed in order based on the relevance score. Since the exact name of the external document is mentioned as a cross-reference in Figure 8.1, the standard name is the same for all returned pages. However, some requirements only mention, for example, EFG as an external cross-reference which has more than 50 different standards for various areas; in that case, our solution provides the name of the standard from which the information is extracted.

Chapter 9

9 Discussion

This study has shown that automatic extraction of relevant text from external documents is feasible, especially for explicit regulatory requirements. The Target Phrase is a specific part of a system or a particular action in explicit regulatory requirements. Examples of explicit regulatory requirements is provided in Figure 9.1.

Motor nameplates shall conform to EFG C22.2.

Figure 9.1 Example of explicit regulatory requirements

In contrast, if the Target Phrase is a major part of a system or a main task applicable to various parts of the system, our solution's information extraction is inadequate, and a domain expert needs to determine in which parts of the system it can be employed. Examples is shown in Figure 9.2 and Figure 9.3.

All products shall comply with EFG B149.1.

Figure 9.2 Example of a regulatory requirement in which Target Phrase refer to a major part a system

Electrical Work shall comply with all ESA , CSA , applicable authorities having jurisdiction , codes and standards .

Figure 9.3 Example of a regulatory requirement in which the Target Phrase refer to a main task

Chapter 10

10 Conclusion and Future Work.

10.1 Conclusion

In this thesis we proposed a two-step framework for requirement-related information extraction from external documents cited in the project contract. We introduced Target Phrase in regulatory requirements (Sub-step A.2), and we developed flexible and reusable rules for Target Phrase extraction (Sub-step A.2) from regulatory requirements with DC references (Section 4.1.2). Using dependency parsing (Section 2.9) and POS tagging (2.11) for Target Phrase extraction we obtained Precision = 0.81, Recall = 0.98 and F-measure = 0.89 (Section 7).

In the case study project contract, the Target Positions were contained in 12% of the regulatory requirements. We obtained Precision = 1 and Recall = 1 in Target Position extraction. However, we only used single well-formed positions for information extraction (Sub-step A.3).

Using external cross-reference, Target Phrase, and Target Position in each regulatory requirement, we extracted relevant information for each regulatory requirement from external documents using Elasticsearch (Section 2.16, and Step B in Chapter 5).

The final output Figure 8.2 shows the content of relevant pages and the page number for a corresponding regulatory requirement ordered by relevance score and also highlighted fragment of text relevant to each regulatory document in each page is depicted in the output.

Such extraction and assembly of requirements information enables the analyst and domain expert to focus on the content and frees them from the drudgery of the analysis and extraction process saving a significant amount of time in a large project.

10.2 Future Work

Future research activities can be dedicated to the following:

10.2.1 Regulatory requirements with more than one Target Phrase

In this study the Target Phrase extraction approach is capable of extracting a Target Phrase from the below formats of regulatory requirements:

- one Target Phrase and one external cross-reference (Figure 10.1)
- one Target Phrase and more than one external cross-refences (Figure 10.2)

Ensure that steel forms conform to EFG3-A23.4.

Figure 10.1 Example of regulatory requirement with one Target Phrase and one external cross-reference

Make the grounding connections in accordance with OESC and ESA requirements.

Figure 10.2 Example of regulatory requirement with one Target Phrase and more than one external cross-reference

For future work, it would be beneficial to consider regulatory requirements with several Target Phrases and several external cross-refences and specify which Target Phrase belongs to which reference. Example is provided in Figure 10.3.

Tie ducts shall be constructed of steel conforming to EFG-G40.20/G40.21M and galvanized in accordance with EFG-G164-M .

Figure 10.3 Example of regulatory requirement with more than one Target Phrase and more than one external cross-reference

10.2.2 Examining verbs in regulatory requirements

Regulatory requirements involve verbs such as install, test, design, etc. that have technical meaning and would be helpful for information extraction. There are also verbs such as provide, ensure, etc. and there is the possibility that by considering them as part of Target Phrase, they will have a negative effect on returned results in the information retrieval step. Determining whether to consider a verb as part of the Target Phrase will improve the result of information retrieval step.

References

Adedjouma, M., Sabetzadeh, M., & Briand, L. C. (2014). Automated detection and resolution of legal cross references: Approach and a study of luxembourg's legislation. *2014 IEEE 22nd International Requirements Engineering Conference (RE)*, 63–72.

analyzer | Elasticsearch Guide [7.13] | Elastic. (n.d.). Retrieved July 22, 2021, from https://www.elastic.co/guide/en/elasticsearch/reference/current/analyzer.html

Badhya, S. S., Prasad, A., Rohan, S., Yashwanth, Y. S., Deepamala, N., & Shobha, G. (2019). Natural Language to Structured Query Language using Elasticsearch for descriptive columns. *2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, *4*, 1–5.

Berenbach, B., Lo, R.-Y., & Sherman, B. (2010). Contract-based requirements engineering. *2010 Third International Workshop on Requirements Engineering and Law*, 27–33.

Boolean query | Elasticsearch Guide [7.13] | Elastic. (n.d.). Retrieved July 20, 2021, from https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-bool-query.html

Breaux, T. D. (2009). *Legal requirements acquisition for the specification of legally compliant information systems*. North Carolina State University.

Büttcher, S., Clarke, C. L. A., & Cormack, G. v. (2016). *Information retrieval: Implementing and evaluating search engines*. Mit Press.

Caelen, O. (2017). A Bayesian interpretation of the confusion matrix. *Annals of Mathematics and Artificial Intelligence*, *81*(3), 429–450.

Ceri, S., Bozzon, A., Brambilla, M., della Valle, E., Fraternali, P., & Quarteroni, S. (2013). *Web information retrieval*. Springer Science & Business Media.

Chen, C. Y., Chen, P. C., & Lu, Y. E. (2013). The coordination processes and dynamics within the inter-organizational context of contract-based outsourced engineering projects. *Journal of Engineering and Technology Management - JET-M*, *30*(2), 113–135. https://doi.org/10.1016/j.jengtecman.2013.01.001

Chopra, A., Prashar, A., & Sain, C. (2013). Natural language processing. *International Journal of Technology Enhancements and Emerging Engineering Research*, *1*(4), 131–134.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, *20*(1), 37–46.

Daneva, M., Herrmann, A., & Buglione, L. (2014). Coping with quality requirements in large, contract-based projects. *IEEE Software*, *32*(6), 84–91.

de Marneffe, M.-C., & Manning, C. D. (2008). *Stanford typed dependencies manual*. Technical report, Stanford University.

Gormley, C., & Tong, Z. (2015). *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine*. " O'Reilly Media, Inc."

Grossman, D. A., & Frieder, O. (2012). *Information retrieval: Algorithms and heuristics* (Vol. 15). Springer Science & Business Media.

Hamdaqa, M., & Hamou-Lhadj, A. (2011). An approach based on citation analysis to support effective handling of regulatory compliance. *Future Generation Computer Systems*, *27*(4), 395–410.

Highlighting | Elasticsearch Guide [7.13] | Elastic. (n.d.). Retrieved July 20, 2021, from https://www.elastic.co/guide/en/elasticsearch/reference/current/highlighting.html

Hu, Y., Boyd-Graber, J., Satinoff, B., & Smith, A. (2014). Interactive topic modeling. *Machine Learning*, *95*(3), 423–469.

Ingolfo, S., Siena, A., Mylopoulos, J., Susi, A., & Perini, A. (2013). Arguing regulatory compliance of software requirements. *Data & Knowledge Engineering*, *87*, 279–296.

Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y., & Zhao, L. (2019). Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications*, *78*(11), 15169–15211.

JSON. (n.d.). Retrieved June 21, 2021, from https://www.json.org/json-en.html

Kannan, S., Gurusamy, V., Vijayarani, S., Ilamathi, J., Nithya, M., Kannan, S., & Gurusamy, V. (2014). Preprocessing techniques for text mining. *International Journal of Computer Science & Communication Networks*, *5*(1), 7–16.

Kekare, A., Jachak, A., Gosavi, A., & Hanwate, P. S. (2020). Techniques for Detecting and Extracting Tabular Data from PDFs and Scanned Documents: A Survey. *Tabula*, *7*(01).

Kiyavitskaya, N., Zeni, N., Breaux, T. D., Antón, A. I., Cordy, J. R., Mich, L., & Mylopoulos, J. (2008). Automating the extraction of rights and obligations for regulatory compliance. *International Conference on Conceptual Modeling*, 154–168.

Kokaly, S., Salay, R., Sabetzadeh, M., Chechik, M., & Maibaum, T. (2016). Model management for regulatory compliance: a position paper. *2016 IEEE/ACM 8th International Workshop on Modeling in Software Engineering (MiSE)*, 74–80.

Kononenko, O., Baysal, O., Holmes, R., & Godfrey, M. W. (2014). Mining modern repositories with elasticsearch. *Proceedings of the 11th Working Conference on Mining Software Repositories*, 328–331.

Kübler, S., McDonald, R., & Nivre, J. (2009). Dependency parsing. *Synthesis Lectures on Human Language Technologies*, *1*(1), 1–127.

Kumar, A., Dabas, V., & Hooda, P. (2020). Text classification algorithms for mining unstructured data: a SWOT analysis. *International Journal of Information Technology*, *12*(4), 1159–1169.

Kuznetsova, A. A. (2021). Statistical Precision-Recall Curves for Object Detection Algorithms Performance Measurement. *Cyber-Physical Systems Modelling and Intelligent Control; Springer: Berlin/Heidelberg, Germany*, 335–348.

Mapping | Elasticsearch Guide [7.13] | Elastic. (n.d.). Retrieved July 22, 2021, from https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping.html

Massey, A. K., Rutledge, R. L., Anton, A. I., & Swire, P. P. (2014). Identifying and classifying ambiguity for regulatory requirements. In *RE* (pp. 83–92). IEEE. https://doi.org/10.1109/RE.2014.6912250

Maxwell, J. C., Anton, A. I., & Earp, J. B. (2013). An empirical investigation of software engineers' ability to classify legal cross-references. *2013 21st IEEE International Requirements Engineering Conference (RE)*, 24–31.

Maxwell, J. C., Antón, A. I., Swire, P., Riaz, M., & McCraw, C. M. (2012). A legal cross-references taxonomy for reasoning about compliance requirements. *Requirements Engineering*, *17*(2), 99–115. https://doi.org/10.1007/s00766-012-0152-5

Natural Language Toolkit — NLTK 3.6.2 documentation. (n.d.). Retrieved July 22, 2021, from https://www.nltk.org/

Nazir, F., Butt, W. H., Anwar, M. W., & Khan Khattak, M. A. (2017). The Applications of Natural Language Processing (NLP) for Software Requirement Engineering - A Systematic Literature Review. In K. Kim & N. Joukov (Eds.), *Information Science and Applications 2017* (pp. 485–493). Springer Singapore.

Nekvi, M. R. I., & Madhavji, N. H. (2014). Impediments to regulatory compliance of requirements in contractual systems engineering projects: a case study. *ACM Transactions on Management Information Systems (TMIS)*, *5*(3), 1–35.

Okapi BM25 - Wikipedia. (n.d.). Retrieved July 22, 2021, from https://en.wikipedia.org/wiki/Okapi_BM25

Practical BM25 - Part 3: Considerations for Picking b and k1 in Elasticsearch | Elastic Blog. (n.d.). Retrieved July 22, 2021, from https://www.elastic.co/blog/practical-bm25-part-3-considerations-for-picking-b-and-k1-in-elasticsearch

PyPDF2 · PyPI. (n.d.). Retrieved July 20, 2021, from https://pypi.org/project/PyPDF2/

Python Elasticsearch Client — Elasticsearch 8.0.0 documentation. (n.d.). Retrieved July 18, 2021, from https://elasticsearch-py.readthedocs.io/en/master/index.html

Qi, P., Dozat, T., Zhang, Y., & Manning, C. D. (2019). Universal dependency parsing from scratch. *ArXiv Preprint ArXiv:1901.10457*.

Query DSL | Elasticsearch Guide [7.13] | Elastic. (n.d.). Retrieved July 20, 2021, from https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl.html

Rahmani, E. (2020). *Identifying External Cross-references using Natural Language Processing (NLP)*.

Rahmani, E., Madhavji, N. H., & Noorwali, I. (2020). Identifying external cross-references using natural language processing. *Proceedings of the 30th Annual International Conference on Computer Science and Software Engineering*, 143–152.

Sadiq, S., & Governatori, G. (2015). Managing regulatory compliance in business processes. In *Handbook on business process management 2* (pp. 265–288). Springer.

Sannier, N., Adedjouma, M., Sabetzadeh, M., & Briand, L. (2017). An automated framework for detection and resolution of cross references in legal texts. *Requirements Engineering*, *22*(2), 215–237.

Sannier, N., Adedjouma, M., Sabetzadeh, M., & Briand, L. (2016). Automated classification of legal cross references based on semantic intent. *International*

Working Conference on Requirements Engineering: Foundation for Software Quality, 119–134.

Schütze, H., Manning, C. D., & Raghavan, P. (2008). *Introduction to information retrieval* (Vol. 39). Cambridge University Press Cambridge.

Search your data | Elasticsearch Guide [7.13] | Elastic. (n.d.). Retrieved July 22, 2021, from https://www.elastic.co/guide/en/elasticsearch/reference/current/search-your-data.html

Shah, N., Willick, D., & Mago, V. (2018). A framework for social media data analytics using Elasticsearch and Kibana. *Wireless Networks*, 1–9.

Singh, V., Pencina, M., Einstein, A. J., Liang, J. X., Berman, D. S., & Slomka, P. (2021). Impact of train/test sample regimen on performance estimate stability of machine learning in cardiovascular imaging. *Scientific Reports*, *11*(1), 1–8.

spaCy 101: Everything you need to know · spaCy Usage Documentation. (n.d.). Retrieved July 17, 2021, from https://spacy.io/usage/spacy-101

Srinivasa-Desikan, B. (2018). *Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras*. Packt Publishing Ltd.

Taware, U., & Shaikh, N. (2018). Heterogeneous database system for faster data querying using elasticsearch. *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 1–4.

Tran, O. T., Ngo, B. X., le Nguyen, M., & Shimazu, A. (2014). Automated reference resolution in legal texts. *Artificial Intelligence and Law*, *22*(1), 29–60.

Vayansky, I., & Kumar, S. A. P. (2020). A review of topic modeling methods. *Information Systems*, *94*, 101582.

Webster, J. J., & Kit, C. (1992). Tokenization as the initial phase in NLP. *COLING 1992 Volume 4: The 14th International Conference on Computational Linguistics*.

Zamfir, V.-A., Carabas, M., Carabas, C., & Tapus, N. (2019). Systems monitoring and big data analysis using the elasticsearch system. *2019 22nd International Conference on Control Systems and Computer Science (CSCS)*, 188–193.

Curriculum Vitae

Name: Sara Fotouhi

Post-secondary Education and Degrees:

Azad University
Qazvin, Iran
2008-2012 B.Sc.

Azad University
Tehran, Iran
2015-2018 M.Sc.

The University of Western Ontario
London, Ontario, Canada
2019-2021 M.Sc.

Related Work Experience

Teaching Assistant and Research Assistant
University of Western Ontario
2019-2021

Publications:

Sara Fotouhi, Shahrokh Asadi, and Michael W. Kattan.
"A comprehensive data level analysis for cancer diagnosis on imbalanced data."
Journal of biomedical informatics 90 (2019): 103089.