4-23-2020 2:45 PM

Identifying External Cross-references using Natural Language Processing (NLP)

Elham Rahmani, *The University of Western Ontario*

Supervisor: Nazim H. Madhavji, *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] Software engineers build systems that need to be compliant with relevant regulations. These regulations are stated in authoritative documents from which regulatory requirements need to be elicited. Project contract contains cross-references to these regulatory requirements in external documents. [**Problem]** Exploring and identifying the regulatory requirements in voluminous textual data is enormously time consuming, and hence costly, and error-prone in sizable software projects. [**Principal idea and novelty]** We use Natural Language Processing (NLP), Pattern Recognition and Web Scraping techniques for automatically extracting external cross-references from contractual requirements and prepare a map for representing related external cross-references to each contractual requirement. This map is also automatically extended to the world-wide web using previously identified references that are not located in local resources. The novel aspects in our approach involve: (i) a list of semantic cues for identifying cross-references, (ii) a taxonomy of grammatical structures for supporting various combinations of word roles in a sentence, (iii) APA standards for validating cross-references, and (iv) third party access for unavailable resources. **[Research Contribution**] The key research contribution is a tool implementing the mentioned techniques for identifying cross-references in contractual documents and related regulatory documents and the web. The tool produces high-level and detailed views of cross-references amongst documents that can be used by various stakeholders for project management, requirements elicitation, testing, and other purposes. We anticipate that this would save an enormous amount of time and effort needed to do this task manually in contractual projects. **[Conclusion]** The output cross-references produced by the tool suggests a precision of 99%, and recall of 87% from contractual requirements. Further work is identified.

Keywords

Regulatory Compliance, Regulatory Requirements, Cross-references, Natural Language Processing, Pattern Recognition, Web Scraper

Summary for Lay Audience

In this thesis, we implemented an approach for automatically identifying external cross-references (references that refer to the existing external documents) from a contract document which is an official agreement between supplier and customer organizations. We categorized external references into three groups based on their differing formats: Direct Cue (DC), Indirect Cue (DC) and No Cue (NC) references. In the case study contract with 683 pages and 10345 paragraphs, we identified 667 DC references (83% of the total external references). Therefore, we focused on identifying DC references in this thesis.

As data preparation, we created two taxonomies: (i) “whitelist” list consists of a number of “reporting phrases” that precede cross-references in the contract, (ii) and ‘Hasleaf_Pattern” taxonomy consists of patterns that aid in finding the boundaries of references.

By utilizing Natural Language Processing (one of the artificial intelligence disciplines contains a set of functionalities designed for interacting between computers and human natural languages and then making them understandable for machines), RegexParser (a mini programming language enabling you for describing and parsing the texts) and the mentioned taxonomies, we have created a tool that can identify DC references from contracts with 99% average accuracy. For cross-references with target documents not available locally, the tool searches the world wide web using Web Scraping techniques (an automated approach enabling to extract data from HTML web pages). With the target resource determined, the tool attempts to find second level references. Currently, the tool is limited to two levels of reference identification. This tabulated reference shows the relations between the references in the contract and the target resources with domain information.

This tabulated information can be used by different stakeholders including: project managers for scoping the effort and time for compliance analysis; analysts for eliciting project requirements; testers for creating test cases, and others. The case study contract was processed for cross-references by the tool in approx. 17 seconds; manually identifying these references would take a number of days, thus saving an enormous amount of time and effort, not to mention the quality of the work.

Acknowledgement

First and foremost, I would like to express my very profound gratitude to my supervisor, Professor Nazim Madhavji for his support, continuous guidance, supervision, and encouragement during the course of my research. He is not only an outstanding mentor, but a compassionate friend. I highly appreciate his dedication, time, consideration and ideas to make me have a pleasant research experience in Western University.

I would like to thank the Department of Computer Science at the University of Western Ontario for the computing infrastructure and facilities provided during my graduate studies.

My deepest appreciation goes to my lovely parents and siblings for their never-ending loves, sacrifices and encouragements. They constantly support me with endless inspirations and I would not be able to pass this experience without their positive energy and support even from far away.

My special thank goes to my best friend A.Farzaneh who has always given me never-ending support, encouragement and confidence needed to deal with the challenges I have had in my life.

Glossary of Terms

CR	Cross-reference: A reference made from one part of a book, register, dictionary, etc. to another part where the same word or subject is treated of (Oxford University, 2019)". It is like a purposeful object in a textual document which refers to related information somewhere in the same document or somewhere out of that document (see Section 2.4).
CRE	Cross Reference Expression: A Cross-reference Expression is a Natural Language phrase that can represent one or more cross-references (see Section 2.4.2).
POS Tagging / POST	Part of Speech Tagging: is the process of marking up words in a textual input with their appropriate part of speech. It means it determines the role of each word (verb, noun, adjective, etc.) in a sentence (see Section 2.7.1).
DC Reference	Direct Cue Reference: Direct Cue References include a reporting phrase in the middle of the CRE and the reference is coming after those phrases (see Section 4.2.2.1).
IDC Reference	InDirect Cue Reference: InDirect Cue References include a reporting phrase in the middle of a CRE and just a part of the reference is coming after the reporting phrases and the remaining part of the reference might coming at any part of the CRE; it might be in the first part of the sentence or before the reporting phrase (see Section 4.2.2.2).
NC Reference	No Cue Reference: No Cue Reference don't have any reporting phrases or any extra explanation. In this model most of the time all the CRE is a cross-reference (see Section 4.2.2.3).
APA	American Psychological Association (APA): It is standards which are providing academic writing foundation that allows authors to write their ideas clearly and accurately (see Section 4.2.3).
APA Properties	If a reference has the following properties it means it has APA Properties: 1) References might consist of acronym words with all Capital letters (Example: In accordance with IXXX No. 85.)

	2) References might consist of some sequential words that each word start with capital letters (Example: in accordance with National Building Code) 3) References can consist Digits (Example: in accordance with IXXX Std IXXX No. 85.) 4) References can consist of special characters like “/, -, ., _”, etc. (Example: given in NAC/ASC A23.1/A23.2-M.) 5) CREs may have more than one Reference. In this case, references could be separated by different ways: ▪ Separated by and (Example: in accordance with the Manufacturer's Instructions and CAN/CSA A23.1/A23.2-M) ▪ Separated by or (Example: shown in the Contract Documents or determined by the GO Wheel) ▪ Separated by and or (Example: in accordance with the CAN/CSA S16.1-M and or National Building Code) ▪ Separated by comma (in accordance with IXXX Std 1058, Standard for Software Project Management Plans) (see Section 4.2.3).
Reporting Phrases	Reporting phrases are also known as “referring phrases” or “referencing phrases” (Sydney, 2019) (EAP Foundaton, 2019). They are lingual structures which are used for referring to the key details that uniquely identify a source of information. This source of information can be references, important ideas, discoveries or writings of experts in a special field of study (Sydney, 2019) (see Section 4.2.2.4).
Whitelist items	It is a text file contains all the reporting phrases collected from different established English literature (see Tables 4.7).
HasLeaf_Pattern Taxonomy	This is a text file containing various grammatical structures. Each pattern in this taxonomy is a representative of a part of a standard English

	sentence. Such grammatical structures are the most common patterns which are used in different CREs and are collected from contractual requirement documents (see Figure 5.4).
TP (True Positive)	True Positive: Observations which are references, and an algorithm recognizes them correctly (see Section 7.1.3).
TN (True Negative)	True Negative: Observations which are not references, and an algorithm correctly doesn’t recognize them as references (see Section 7.1.4).
FP (False Positive)	False Positive: Observations which are not references, but an algorithm falsely recognizes them as references (see Section 7.1.5).
FN (False Negative)	False Negative: Observations which are references, but an algorithm doesn’t recognize them as references (see Section 7.1.6).

Table of Contents

List of Tables

List of Figures

Chapter 1

1 Introduction

This thesis focuses on the problem of identifying cross-references to external documents from a project contract. This introductory chapter starts by describing the context that motivates us to choose this topic. We then describe the problem, solution approach, the solution, its novelty, and its anticipated impact.

1.1 Context and Motivation

Software systems are required to comply with the relevant regulations and standards *(Maxwell, et al.*, 2011) (Brian Berenbach, Ren-Yi Lo, 2010).With increasing automation and digitalization in different application domains, there is corresponding increase in government regulations and standards that act as constraints on the software systems that are developed and deployed. However, a complication arises when regulations and standards change while outdated versions are still being used in projects (Nekvi and Madhavji, 2014).

Regardless, with increasing awareness and needs for system compliance in modern systems, the applicable regulations and standards are permeating the functional and nonfunctional (or quality) requirements of both legacy and new systems. For example, in the US alone, billions of dollars are spent annually in compliance (Ingolfo *et al.*, 2013). In the Healthcare sector alone, approx. $20 billion were invested over a number of years on system compliance with (Ingolfo *et al.*, 2013) HIPPA (Health Insurance Portability and Accountability Act) (Office for Civil Rights, 2003). In 2005, in the Business domain, organizations spent approx. $6 billion on compliance of their reporting and risk management procedures with the (Ingolfo *et al.*, 2013) Sarbanes–Oxley Act (SOX) (Of, 2015).

It is important to note that implementation of system compliance can be challenging because regulatory requirements are known to be ambiguous, and often contain semantic errors, undefined acronyms, numerous cross-references, and domain-specific terms (Antón, 2007) (Ingolfo *et al.*, 2013) (Nekvi and Madhavji, 2014). While defining and elaborating compliance requirements to make them fit in a project, analysts often need to identify and follow cross-references (both internal to the document at hand or externally to third-party documents such as regulations and standards (Nekvi and Madhavji, 2014). Generally, it can be stated that cross-references forming a complex relational network, capture relationships between different pieces of texts (Sannier *et al.*, 2017) making the task of requirements elicitation quite complex (Nekvi and Madhavji, 2014). Therefore, an automated cross-reference identification tool can reduce the effort significantly while also improving the accuracy of the task.

1.2 Problem Description

From the business point of view, the cost of noncompliance is significant. Organizations that do not comply with regulations are liable to penalties and may suffer tarnished reputation (Nekvi and Madhavji, 2014). While system compliance is clearly important, its implementation can be challenging because the regulatory requirements are not trivial to determine in many projects due to numerous cross-references, known to be ambiguous, and often contain semantic errors, undefined acronyms, and domain-specific terms (Antón, 2007). Among these issues, cross-references to external documents are arduous and error-prone to deal with (Nekvi and Madhavji, 2014). It forces developers to analyze a great deal of data (Antón, 2007) among a network of documents in a non-sequential manner (Breaux and Antón, 2008).

Considering the importance of cross-references for compliance, this thesis focuses on automatic identification of external cross-references in a contractual document in software projects.

1.3 Solution Approach

To our knowledge, previous work has focused on extracting cross-references *internal* to a given document and not on cross-references *external* to the primary document. For dealing with internal cross-references, all the elements of the document (such as: headers, footers, titles, chapters, sections, sub-sections, and paragraphs) may refer to one another as necessary (Adedjouma *et al.*, 2014). While this can be complex due to the size of the document, it is at least contained in that one document. If the document is online, as most are these days, then hyperlinks help in traversing the document.

In contrast, for dealing with cross-references to external documents, one needs to, first, access the target document, which can be one of many according to the complexity of the primary document or project. Once the relevant external document is accessed, finding the target text (or information) can be quite time consuming given that the structure of all relevant documents is not uniform. Once the target piece of information is identified, it may contain yet more external cross-references.

The overall solution entails the following five steps:

1. Examine external cross-references (in the primary document) for different formats, and categorize them into the following three types:
   - ✓ Direct Cue References (Identifying this type of references is the focus of this thesis)
   - ✓ InDirect Cue References (future work)
   - ✓ No Cue References (future work)
2. Create a list of reporting phrases for supporting the properties of DC references
3. Create a taxonomy of grammatical structures for supporting word-ambiguities that exist in different sentences.
4. Implement a tool to identify cross-references in the primary document and indirectly from cross-referenced documents that may even be on the world wide web (WWW). The automated identification process consists of the following two steps:
   - (i) External cross-references with the properties of DC references are captured from contractual documents.
   - (ii) The identified references captured in step one are now considered as the keywords for finding their resources. The following two conditions prevail:
     - If there is a local file/resource for the intended keyword (the reference which is identified in step 1), the related resource would be chosen from the available local resources. At this time, investigation process for identifying the next level of cross-references is started over the local resource.

    - If there is not any local file/resource for the intended keyword, the keyword would be posted to google search engine API. A textual resource for each keyword would be identified and recommended by google. At this time, investigation process for identifying the next level of cross-references is started over the content of web pages recommended by google.

5. A table would be generated for representing all the identified cross-references related to each contractual requirement. In this thesis, the level of reference identification is done up to two levels; however, in theory, there can be more indirect levels.

Our approach is quite different from those in the literature in two principal ways:

(i) internal vs. external references, as described above.

(ii) in the technical details in how each cross-reference is identified (described in the thesis).

1.4 Impact of Result

The output of our tool is a map in the form of a table. Each row of this table represents a contractual requirement and shows associated cross-references. Generally, this table shows multiple views of cross-references amongst the documents. In the first view, the identified references extracted from contract document are shown in one column of a table. In the second view, the identified references extracted from standard or legal documents are represented in the next column of the table. From the software engineering perspective, this result can be helpful in the process of designing software systems involving cross-references. For example, health systems that need to comply with HIPAA requirements, or financial systems with Sarbanes-Oxley Act (Of, 2015) (Hamou-lhadj, 2010), would need to deal with numerous documents with inter-twining cross-references. In such systems, our result can significantly reduce the manual labor of eliciting requirements from cross-referenced documents. The tabulated format of identified cross-references can be used by various stakeholders, e.g.: project management (for cost estimation of tasks); requirements engineers (for elicitation of requirements), testers (for creating test cases), and domain experts (for verifying development against domain knowledge), etc.

1.5 Thesis Structure

Chapter 2 describes general background literature. Chapter 3 describes core literature against which our approach can be compared. Chapter 4 provides high-level analyses of the problem and solution-approach. Chapter 5 describes the proposed solution for the implementation phase of the work. Chapter 6 gives the output produced by the proposed solution. Chapter 7 describes the quality of algorithm, including accuracy, precision, recall and f-measure. Chapter 8 provides compares our work with related work. Chapter 9 discusses the anticipated impact of our results. Chapter 10 summarizes the thesis and future work.

Chapter 2

2 Background

This chapter describes the software engineering background that is relevant to the thesis topic. It is Regulatory Compliance and introducing three types of documents which are important in compliance area (Contractual Requirement Documents, Technical Standard Documents, Legal Requirement Documents), System Compliance and its Challenges, Costs of Noncompliance and Cross-References. This chapter is then continued by explaining about the background required for supporting the practical approach of implementing the idea of this thesis. They are Natural Language Programing, Pattern Recognition, Sequence Labeling with Part Of Speech Tagging and Named Entity Recognition, Word Lexical Disambiguation, Tokenization, Regular Expression with its two types of characters Metacharacters and Quantifiers. It would be then continued by explaining about Web Scraping technique and its need for two of important libraries named Requests and BeautifulSoup. Finally, the chapter would be terminated by explaining about Classification problems and Confusion Matrix with all its elements including: Observation, Positive and Negative Observations, True Positive, True Negative, False Positive, False Negative and Recall, Precision, F-Measure.

2.1 Regulatory Compliance

There are numerous regulations and standards which are released by the governments in different times. These regulations are proposed in the form of authoritative documents and they are created at the local, state, federal, or international levels. These documents are called “Regulatory Requirements” (Hamou-lhadj, 2010). Software systems are widely affected by these regulations because they can impact on both functional and nonfunctional Software requirements. Therefore, there is a serious and vital need that a software system complies with all the relevant constraints which are proposed in Regulatory Requirements. Applying compliance in software systems is a key activity which is called “Regulatory Compliance”, and it should be done in all the Software Development Life Cycle (SDLF) (Nekvi and Madhavji, 2014).

In the following three types of documents which are important in compliance area are introduced:

2.1.1 Contractual Requirement Documents

In any requirement engineering project, contract is an official agreement between supplier and customer organizations. Contract is a textual document containing large number of requirements. This is called contractual requirements (Nekvi and Madhavji, 2014). Such requirements are expressed in form of short abstracts or just a general idea which are categorized into two types:

- Regulatory requirements: For instance, “The transfer switch shall comply with Electrical Code” (Nekvi and Madhavji, 2014). It refers to a regulatory document with which the transfer switch has to comply.
- Nonregulatory requirements: These are kinds of requirements which do not refer to a regulation or a standard. This is the reason they are called nonregulatory requirements. (Nekvi and Madhavji, 2014).

2.1.2 Technical Standard Documents

Standards are developed by professional organizations; for example, CBC Std. 1003.1 (Committee, 2001) is a standard which is proposed by CBC organization (Brian Berenbach, Ren-Yi Lo, 2010). Such standards are considered as the best proposed practice approach and instructions which are in the form of textual documents. Regarding the importance of standards, it is quite common that some of the standards are referenced in contractual requirements. In such circumstances, it then becomes contractually mandated for execution. In addition, in such situations its degree of importance and impact is the same as any other contractual requirement (Brian Berenbach, Ren-Yi Lo, 2010).

2.1.3 Legal Requirement Documents

There are requirements that are enforced by law and are created by a legal entity (Brian Berenbach, Ren-Yi Lo, 2010). Documents which keep these kinds of requirements are called legal Requirements. For example, “Under Part 2 of these Regulations, which is applicable to fixed workplace premises, employers are required to ensure that pedestrians and vehicles can move in a safe manner and that traffic routes are clearly identified and appropriately dimensioned. Traffic rules for mobile work equipment are also required.(Authority, 2020)”, is a requirement enforced by law and it refers to safety, health and welfare at work.

2.2 System Compliance and Challenges

Based on the Regulatory Compliance definition, system compliance Software Development Life Cycle (SDLF), is a term that can be assigned to any operational system which is satisfying regulatory constraints. It should be noted that while system compliance is important, its implementation can be challenging because the documented texts in regulatory requirements are known to be ambiguous and often contain semantic errors, undefined acronyms, numerous cross-references, and domain-specific definitions (Antón, 2007) (Ingolfo *et al.*, 2013) (Nekvi and Madhavji, 2014). Also, while these documents are revised due to changes in laws, regulations, and standards, the outdated versions are still accessible and hence often inadvertently utilized in projects (Nekvi and Madhavji, 2014).

2.3 Costs of Noncompliance

From the business point of view, organizations that do not comply their software with regulations are liable to penalties and may suffer tarnished reputation (Nekvi and Madhavji, 2014). The reason is that noncompliance can leads to nonstandard systems, customer satisfaction issues, lost reputation, expensive penalties, damage to brand name, and criminal charges (Antón, 2007). For example, noncompliance in the Canadian Environment Act, 1999 (Government, 2019) can result in a fine in the range of $75,000 to $4 million (Nekvi and Madhavji, 2014). Therefore, companies must have a serious plan for complying their software with related regulations in order to protect their business from the consequences of noncompliance.

2.4 Cross-references

A cross reference means: “A reference made from one part of a book, register, dictionary, etc. to another part where the same word or subject is treated of (Oxford University, 2019)”. It is like an instance or a kind of purposeful object in a textual document which refers to related information elsewhere; somewhere in the same document or somewhere outside that document. We can also refer to cross-references simply as “references”. The following two items are two common definitions in cross-reference area which are important to be noted here.

2.4.1 Legal Cross-references

Regulatory requirements consist of numerous domain-specific terms and definitions. Understanding such terms and definitions are complex for anyone who is not a domain expert. In order to achieve compliance, requirements engineers need to figure out such domain-specific definitions and vocabularies before deriving compliance requirements from regulatory texts. A legal Cross Reference (CR) is a kind of such domain-specific terms which is expressed in form of a citation that links one legal provision to another (Maxwell *et al*., 2011).

2.4.2 Cross-reference Expressions

A Cross Reference Expression (CRE) is a natural language phrase that can represent one or more cross references. For example: "Submit shop drawings, diagrams, plans and details associated with demolition and removals work in accordance with Section 01340 SHOP DRAWINGS." is a cross reference expression. This expression embodies one cross reference: "Section 01340 SHOP DRAWINGS ". This type of cross reference is called internal cross reference by virtue of the fact that no external document name is cited, and the reference mentions a section within the same document (Maxwell *et al*., 2011). On the other hand, "Software Quality Assurance Plans in accordance with IXXX Std 730, Standard for Software Quality Assurance Plans;" is another type of cross reference expression which contains an external cross reference because here the reference is pointing to provisions in a different legal text (Maxwell *et al*., 2011); not in the current legal text (Adedjouma *et al*., 2014).

2.5 Natural Language Processing (NLP)

Natural Language Processing is one of the artificial intelligence disciplines and linguistics subfields. It contains a set of functionalities designed for interacting between computers and human natural languages and then making them understandable for machines (Eisenstein, 2019). Our daily life during the past decades have been indeed undergone many positive changes as the results of this invaluable technology. Extracting information from texts, translating between languages, answering questions, holding a conversation, taking instructions and so forth, are some examples of the NLP functionalities (Eisenstein, 2019).

2.6 Pattern Recognition

Pattern recognition is the automated recognition of patterns in data. Pattern Recognition underpins developments in cognate fields such as computer vision, image processing, text and document analysis and neural networks (Hancock, 1968). In the area of text and document analysis, phrase pattern recognition (phrase chunking) is a kind of pattern recognition which is used to identify predefined phrase structures (such as noun phrase (NP), verb phrase (VP), prepositional phrase (PP), etc.) in a stream of text (Wu, Lee and Yang, 2008).

2.7 Sequence Labelling

Sequence labelling is a typical NLP task which lies in the group of pattern recognition tasks. It is the process of assigning a label to each token in a given input sequence (Sun *et al.*, 2019). In this context, a single word referred to as a "token". For example, Figure 2.1 shows the tokens (and, now, for, something, completely, different) of the provided sample sentence "and now for something completely different", which are labeled by different tags (CC, RB, IN, NN, RB, JJ) (Bird, Klein, 2009). The process of assigning such tags to each token is called sequence labelling.

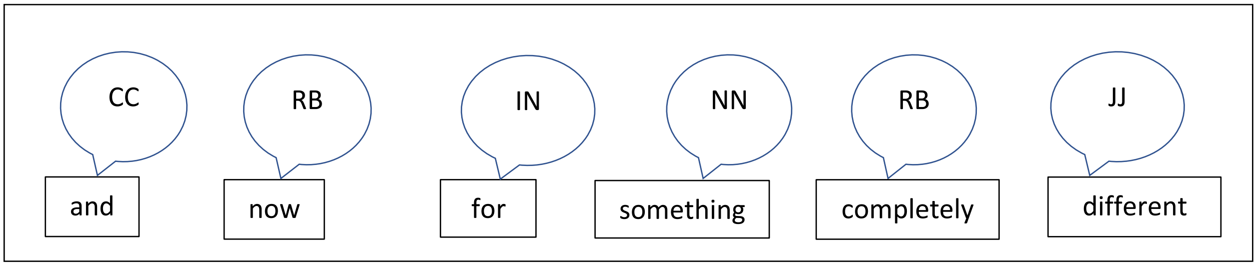


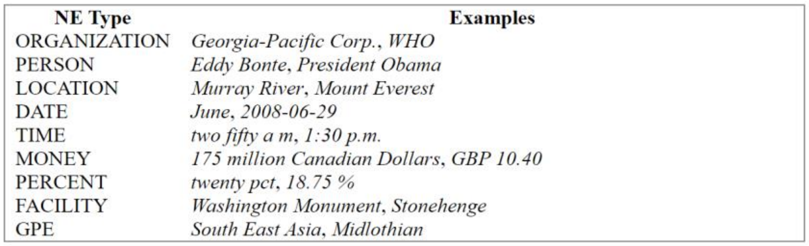
Figure 2. 1: Example of sequence labeling or POST

2.7.1 Part of Speech Tagging (POST)

Part of Speech Tagging (POS tagging or PoS tagging or POST) is a common example of a sequence labeling task (Bird, Klein, 2009). POS is the process of assigning a unique part-of-speech to each word in a sentence, e.g. noun, verb, pronoun and preposition (Santos and Zadrozny, 2014). Figure 2.1 shows an example of a sentence tagged with POS. In this example, the grammatical role of "She" in this sample sentence is pronoun which is shown with "PRP", or the grammatical role of "seashore" in this sentence is singular noun which is tagged with "NN". Therefore, it can be state that, POS is refer as to “Token Labeling”. It means each token is tagged with a specific label which is related to the grammatical role of the token in the sentence. These tags are assigned by POS. A complete list of these tags with their meaning is provided in Table 5.1.

2.7.2 Named Entity Recognition (NER)

Named entities are certain noun phrases that refer to specific types of individuals, such as of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, in a raw text (Bird, Klein, 2009). In other words, it tries to find out whether or not a word is a named entity (persons, locations, organizations, time expressions etc.). This problem can be broken down into detection of names followed by classification of name into the corresponding categories. NER is also refer as to “Span Labeling”, which means labeling segments or groups of words that contain one tag. Figure 2.2 lists some of the more commonly used types of NEs.



NE Type	Examples
ORGANIZATION	*Georgia-Pacific Corp., WHO*
PERSON	*Eddy Bonte, President Obama*
LOCATION	*Murray River, Mount Everest*
DATE	*June, 2008-06-29*
TIME	*two fifty a m, 1:30 p.m.*
MONEY	*175 million Canadian Dollars, GBP 10.40*
PERCENT	*twenty pct, 18.75 %*
FACILITY	*Washington Monument, Stonehenge*
GPE	*South East Asia, Midlothian*

Figure 2. 2: Commonly used types of named entity (Bird, Klein, 2009)

2.8 Word Lexical Disambiguation

Ambiguity is a type of meaning in which a phrase, statement or resolution is not explicitly defined, and it can making several plausible interpretations . Once we talk about human languages, then we are also face with ambiguous (semantically and syntactically), because many words can be interpreted in multiple ways depending upon the context of their occurrence.

Semantic ambiguity occurs when a word, phrase or sentence of a context, has more than one interpretation (Bird, Klein, 2009). For example, the word “duck” in a sentence can refer either *a bird* or *the motion of alive creature*. Therefore, it can be stated that in semantic ambiguity the structure of the sentence is the same, but the individual words are interpreted differently. Word Sense Disambiguation (WSD), in natural language processing is defined as the ability to determine which meaning of word is activated by the use of word in a particular context (Navigli, 2009).

On the other hand, syntactic ambiguity is a situation where a sentence may be interpreted in more than one way because of the ambiguous sentence structure (Bird, Klein, 2009). For example, a sentence is syntactically ambiguous when a reader or listener can reasonably interpret one sentence as having more than one possible structure.

2.9 Tokenization

Tokenization is a process which is commonly applied in the first step of any kind of natural language processing tasks and it underlies the preprocessing phase (Kit, 1992). The major goal of this early task is cutting a string of words and converting it into identifiable linguistic units which forms a piece of language data (Bird, Klein, 2009). Therefore, the result of this process is generating units which are called tokens (Bird, Klein, 2009).

2.10 RegexpParser

Regular Expression (Regexp) parser is a grammar-based chunk parser playing a vital role in having a powerful, flexible, and efficient text processing (Bird, Klein, 2009). It uses a set of regular expression patterns and then specify patterns which are matched with a particular pattern and doesn't match with others (Bird, Klein, 2009). By using a well-structured regular expression, both complex and obvious text processing can be performed in a way that it reduces hours of tedious labors to an automated solution of a few seconds (Friedl, 2006).

2.10.1 Metacharacters

The building blocks of regular expressions are composed of two types of characters. Characters with special meaning which are called Metacharacters and the rest which are regular characters are called literal characters. Table 2.1 and 2.2 (Friedl, 2009) show some of the most common RegEx metacharacters and the examples of what they would match in RegEx.

Table 2. 1: Summary of Metacharacters seen so far (Friedl, 2006)

Metacharacter	Name	Matches
.	Dot	any one character
[…]	character class	any character listed
[^…]	negated character class	any character not listed
^	Caret	The position at the start of the line
$	Dollar	The position at the end of the line
\<	Backslash less-than	The position at the start of a word
\>	Backslash greater-than	The position at the end of a word
|	Or, bar	Matches either expression it separates
(…)	parentheses	Indicates grouping; use to limit the scope

2.10.2 Quantifiers

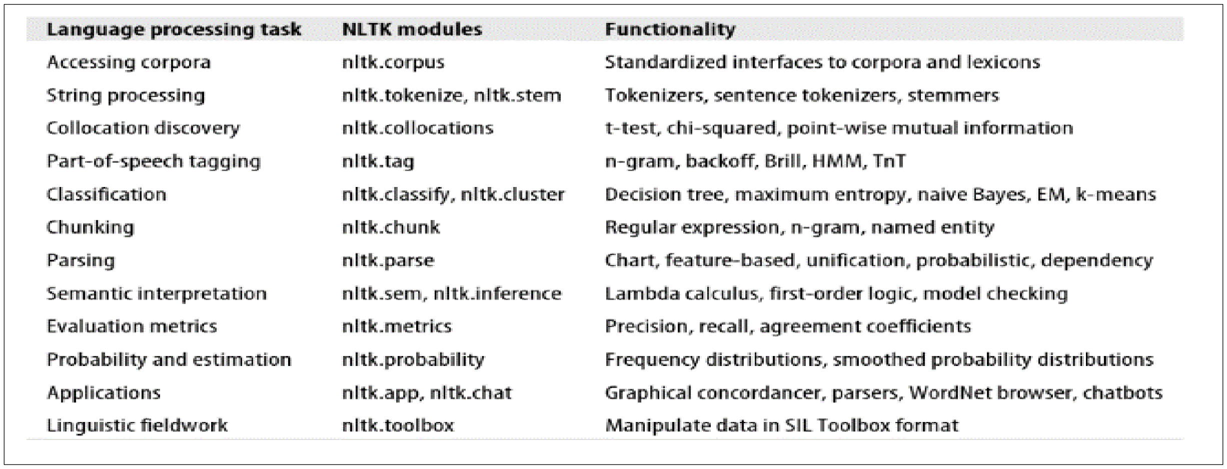
One or multiple quantifiers could be used by RegEx in order to determine the scope of a search string. In table 2.2 the most common examples of quantifiers are listed.

Table 2. 2: Summary of Quantifier "Repetition Metacharacters" (Friedl, 2006)

	Minimum Required	Maximum to Try	Meaning
?	None	1	one allowed; none required ("one optional")
*	None	no limit	unlimited allowed; none required ("any amount OK")
+	1	no limit	unlimited allowed; one required ("at least one")

2.11 NLTK Library

NLTK or Natural Language Toolkit, is a leading library in Python programming language which is widely used in natural language processing for English written human language data (Hardeniya, 2015). It has been also used as a teaching study tool and as a platform suitable for building research systems (Bird, Klein, 2009). The power of NLTK is providing simplicity, consistency, extensibility and modularity (Hardeniya, 2015). Figure 2.3 shows some of the most important NLTK modules:



Language processing task	NLTK modules	Functionality
Accessing corpora	nltk.corpus	Standardized interfaces to corpora and lexicons
String processing	nltk.tokenize, nltk.stem	Tokenizers, sentence tokenizers, stemmers
Collocation discovery	nltk.collocations	t-test, chi-squared, point-wise mutual information
Part-of-speech tagging	nltk.tag	n-gram, backoff, Brill, HMM, TnT
Classification	nltk.classify, nltk.cluster	Decision tree, maximum entropy, naive Bayes, EM, k-means
Chunking	nltk.chunk	Regular expression, n-gram, named entity
Parsing	nltk.parse	Chart, feature-based, unification, probabilistic, dependency
Semantic interpretation	nltk.sem, nltk.inference	Lambda calculus, first-order logic, model checking
Evaluation metrics	nltk.metrics	Precision, recall, agreement coefficients
Probability and estimation	nltk.probability	Frequency distributions, smoothed probability distributions
Applications	nltk.app, nltk.chat	Graphical concordancer, parsers, WordNet browser, chatbots
Linguistic fieldwork	nltk.toolbox	Manipulate data in SIL Toolbox format

Figure 2. 3: Language processing tasks and corresponding NLTK modules with examples of functionalities (Hardeniya, 2015)

2.12 Web Scraping

Web scraping is an automated approach enabling to extract data from HTML web pages. Its main focus is on transforming the unstructured web content into the proper structured and accessible form of data for analyzing (Mitchell, 2018). For example, converting the context of <p> tag (in HTML) into simple string format. In cases that your only access to internet is through browsers, web scraping can be an excellent choice for gathering and processing large amount of data quickly on web pages. The reason is because, the browsers have proper human readable format which can provide you with viewing thousands or even millions of pages at once (Mitchell, 2018). Online price comparison, weather data monitoring, website change detection, web research, web content mashup and web data integration are some of the potential instances of web scraping (Mitchell, 2018). Following libraries are two of the most common open source Python frameworks used for web scraping:

2.12.1 Requests

Sending an HTTP request to the servers of websites is the first step in any web scraping task. This step is done for fetching the displayed data on the target web pages (Mitchell, 2018). Traditionally, it is done manually by adding query strings to URLS. However, by using Requests library which is an elegant Python framework, the process of making HTTP requests is significantly simplified and it allows you to easily retrieve the web page data (Mitchell, 2018).

2.12.2 BeautifulSoup

BeautifulSoup is the other popular Python library which has a leading role in web scraping tasks. It is often used with Requests library because the main functionality of BeautifulSoup is parsing the data which has been already extracted from HTML or XML documents through Requests library. The power of BeautifulSoup is its simplicity in automating some of the parsing steps which are recurring during web scraping time. It is also capable of finding any kind of data and even detecting special characters (Mitchell, 2018).

2.13 Classification Problems

Classification is a central topic in today's world, where big data is used. It is simply grouping data together according to similar features and attributes (Krzystof Jajuga, Andrzej Sokolowski, 2002). In real world classification is a basic cultural activity of humanity for naming of appearances (Krzystof Jajuga, Andrzej Sokolowski, 2002). An example of classification in real world is grouping and naming the creatures into humans, animals, plants, etc. (Krzystof Jajuga, Andrzej Sokolowski, 2002). In AI techniques, classification is all about teaching computers to do the same thing.

2.14 Confusion matrix

The decision made by the classifier can be shown in a structure known as a confusion matrix (Davis and Goadrich, 2006). In other words, Confusion matrix is a table which is designed to be applied over a set of test data for representing the performance of a classification model. Following items are the main elements of confusion matrix:

2.14.1 Observation

"In statistics, a unit of observation is the unit described by the data that one analyzes. For example, in a study of the demand for money, the unit of observation might be chosen as the individual, with different observations (data points) for a given point in time differing as to which individual they refer to; or the unit of observation might be the country, with different observations differing only in regard to the country they refer to" (Wikipedia, 2019). Therefore, sometimes observations are determined based on the absence or presence of a property. For example, in a text document containing capital and small words, observations can only refer to the words with capital letters.

2.14.2 Positive and Negative Observations

In a simple classification problem, all the observations should be evaluated based on two positive and negative classes (Davis and Goadrich, 2006). Observation is positive when for example in a textual document, a word is written with all capital letters. Observation is negative when for example a word is not written with all capital letters.

2.14.3 True Positive (TP)

This element is for observations which are positive, and are predicted to be positive (Davis and Goadrich, 2006).

2.14.4 True Negative (TN)

This element is for observations which are negative, and are predicted to be negative (Davis and Goadrich, 2006).

2.14.5 False Positive (FP)

This element is for observations which are negative, and are predicted to be positive (Davis and Goadrich, 2006).

2.14.6 False Negative (FN)

This element is for observations which are positive, and are predicted to be negative (Davis and Goadrich, 2006).

2.14.7 Recall

Recall is another element which can be calculated from the confusion matrix elements. It is the ratio of the total number of correctly classified positive examples (TP) divide into the total number of positive examples (TP and FN) (Davis and Goadrich, 2006).

2.14.8 Precision

Precision is the total number of correctly classified positive observations (TP) divided by the total number of predicted positive examples (TP and FP) (Davis and Goadrich, 2006).

2.14.9 F-Measure

F-measure is a standard that evaluate the average probability of success in recognizing the right class of an instance (Maratea *et al.*, 2014). It conveys the balance between the precision and the recall.

Chapter 3

3 Related Work

To our knowledge, previous work has focused on extracting cross-references internal to a given document and not on cross-references external to the primary document. In this chapter three kind of related work is discussed. Firstly, we review an automated reference extraction approach which is applied on Japanese law corpus. Secondly, we review a text schema automatic model for enhancing non-mark legal texts and provide a study over interpreting cross-reference expression. Finally, we review a study over an automated classification of legal cross-references based on semantic intent.

In (Tran *et al.*, 2014), the authors proposed a four step automatic framework for recognizing internal mentions (in this work they use the word “mention” instead of cross-reference) and then extracting them from a Japanese National Pension Law corpus. In the first step, they focused on identifying the references and then splitting them. To do this, they used sequence labeling and POST techniques in order to see where each word is located in an input string. They defined the following notations for tagging different parts of each mention:

- B_M (Begin of Mention): The first element of a mention is tagged with B_M
- I_M (Inner of Mention): The remaining elements of the mention are tagged with I_M
- E_M (End of Mention): The last element of the mention is tagged with E_M
- O (Others): All elements outside the mention are tagged with O

In the next step, the extracted mentions are categorized into two classes. To apply this categorization, they used supervised machine learning methods to determine the status of each mention in each class. In the third step, they focused on mention position recognition. Here their main objective was finding articles, paragraphs, items, etc. In the last step, they prepared a list of mention candidates which was extracted based on two approaches: 1) Dependency tree 2) punctuation marks (usually a comma). In the first one they supposed that each node has a mention head. In the later one they used a right-to left scan from the position of the mention head or from its synonyms. Once the scanning process is meeting a comma or meeting the beginning of a sentence, the scanned text is extracted and is considered as a reference candidate. Finally, among all the candidates the best reference is chose. This system succeeds to achieve 80.06 % for detecting references and 85.61 % accuracy for resolving them.

In (Adedjouma *et al.*, 2014) the authors focused on automatically identifying internal references. They built a text schema for enhancing non-mark legal texts and provided an automatic approach for interpreting cross-reference expressions. They used a four steps approach for implementing their idea. Firstly, they built manually a UML class diagram for showing relationships between different parts of a legal text including book, title, chapter, section, etc. Each part is a class and each class are distinguished by a relevant "Header" and "id" labels. These labels were assigned to recognize the beginning of each part. Then, by using an algorithm, they generated and executed some kinds of regular expressions; for example, "HeaderRegEx" for headers or "SegmentRegEX" for segmentations. Therefore, it can be stated that in this work such expressions are working as indicators to show that a section is terminated and a new section is beginning. Furthermore, they automatically generated a hierarchical structure from the relation between different parts/classes. For example, the sign "<" was used for showing relation between different parts. For instance, $c_i <$ c_j means that class j directly or indirectly contains class i. This is how they converted a non-markup format into a markup format (like XML). For using such regular expressions, they also needed another phase to automatically detect CREs in a given legal text. For this, they had a study over CREs and categorized them into explicit and implicit groups. Based on this categorization they defined some terms and patterns and then interpret the detected CREs into a set of individual CRs. For example: "article 102" is a reference consist of a term-pattern ("article") and a number-pattern ("102"). So, the CRE which contains this reference is representing the reference explicitly. "article", "articles", "art" and "paragraph" are the predefined terms in explicit CREs (see Table 3.1). On the other hand, implicit CREs consist of the other predefined terms, including:" above", "below"," preceding"," following"," that follows", "next", "previous", "this", "in question", "same" (see Table 3.1). In implicit CREs these predefined terms are actually referring to the references. It means the CRE is referring to the reference implicitly. For example, if a CRE consist of "*following paragraph*", it means the current CRE is implicitly referring to a reference which is cited/located in the *following paragraph* of the CRE.

Table 3.1 shows a view from explicit and implicit patterns.

Table 3. 1: A view from explicit and implicit patterns

Elements	Description	Patterns	Examples
Explicit CRE	Explicit CREs have references that for example start with certain marker terms and followed by a number	<marker-term><numbers>	article 102
Implicit CRE	Implicit CREs have references that for example start with an implicit term and followed by a marker term	<implicit-term><marker-term>	following paragraph
<marker-term>	Are some certain terms	< “article” | “articles” | “art” | “paragraph”>	N/A
<implicit-term>	Are some certain terms	<” above” |” below” |” preceding” |” following” |” that follows” | “next” | “previous” | “this” | “in question” | “same” |…>	N/A

In (Sannier *et al.*, 2016) an automated approach for classifying cross reference in legal texts is created. This classification is done based on two Luxembourgish legislative Texts. Generally, in this work the important thing was building a taxonomy (Table 3.2) of semantic intent types that for the first time was proposed by (Maxwell *et al.*, 2011). In this taxonomy, they compare their classifications with three other taxonomies which were previously proposed by (Breaux, 2009), (Hamdaqa and Hamou-Lhadj, 2011) and (Maxwell *et al.*, 2011). It is important to note that, in (Sannier *et al.*, 2016) the main focus is on classifying the CRs; not on automatically identifying CRs. In this work, their main idea for classifying CREs is appearing CRs before or after specific phrases, because they believe these phrases are helpful in identifying the CRs. Their phrase-classifications is listed in Table 3.2. This taxonomy distinguishes eleven intent types. These intent types are listed in the first column of the table and are collected from two legal texts. The second column which is named “Most Frequent Patterns”, is for showing the most frequent phrases that they have found for each content type. The values of the third column named “Mapping” show that, except ‘General Amendment type”, all types in their taxonomy have a corresponding type in the taxonomies of Breaux's, Hamdaqa *et al.*'s, and Maxwell *et al.*'s. The frequency of values in Hamdaqa *et al*. and Maxwell et al, in the "Mapping" column, shows that this taxonomy (Table 3.2) is primarily an amalgamation of those by Hamdaqa *et al*. and Maxwell *et al.* The fourth column named "Frequency" is for showing the frequency of each intent type that they found in the legal documents. The fifth column named "# of phrases" shows the number of related phrases that they identify for to each intent type. The last column named "# of distinct patterns" shows the number of patterns which are different from the taxonomies of Breaux**,** Hamdaqa et al. and Maxwell.

Table 3. 2: Cross-reference classification (Sannier *et al*., 2016)

Intent Type	**Most Frequent Patterns (% of all patterns for intent type)**	**Mapping**			**Frequency**	**# of phrases**	**# of distinct patterns**
		(Breaux, 2009)	**(Hamdaqa and Hamou-Lhadj, 2011)**	**(Maxwell *et al.*, 2011)**			
(rare patterns: 36.68%)	applicable (22.10%) by virtue of (18.23%) conforming to / in accordance with (13.81%)	—	compliance	—	16.03%	173	24
Constraint *(rare patterns: 10.53%)*	within the conditions of (68.42%) within the limits of (21.05%)	constraint	—	constraint	1.76%	19	4
Definition *(rare patterns: 4.18%)*	under (67.67%) within the meaning of (22.16%) specified / defined (5.99%)	definition	definition	definition	30.95%	334	7

Delegation *(rare patterns: 5.31%)*	future tense (in French) (55.75%)	—	—	general	10.47%	113	4
	infinitive form (in French) (26.55%)						
	modals (may / can / will) (12.39%)						
Exception *(rare patterns: 8.63%)*	negative form (53.44%)	exception	—	exception	6.12%	66	11
	derogation (29.31%)						
Refinement *(rare patterns: 7.40%)*	applies to (66.67%)	refinement	specification	—	2.50%	27	8
	for the application of (18.52%)						
	also concerns (7.41%)						
General Amendment *(rare patterns:0%)*	modified (62.35%)	—	—	—	15.01%	162	3
	Following [+addition] (37.65%)						
Amendment By Addition *(rare patterns:0%)*	is added (40.91%)	—	Amend. by Addition	—	4.08%	44	6
	is completed (36.36%)						
	is inserted (22.71%)						
Amendment By Deletion *(rare patterns:0%)*	is deleted	—	Amend. by Deletion	—	3.52%	38	3
Amendment By Redesignation *(rare patterns:0%)*	becomes the new	—	Amend. by Redesignation	—	1.48%	16	1

Amendment By Replacement *(rare patterns:0%)*	is replaced by	__	Amend. by Striking	__	7.41%	80	1

Their idea for classifying CRs based on such phrases was very helpful for us because it lights in our mind that we can use the same approach for identifying CRs. Therefore, the similarity of our work with their work is the idea of creating a list of specific phrases which comes before or after the references. However, the important difference is in the essence of phrases that we use. With respect to their invaluable work, we did not use their classifications. Firstly because, they needed to collect their phrases from legal documents (Section 2.1.3), not contractual requirement documents (Section 2.1.1). Furthermore, their categorization is limited to two legal texts. In addition, they only get the result from the CREs that their phrases are semantically subset of those 11 categories (Table 3.2). In contrast, in our thesis for creating a phrase-list, we have created a list consist of “Reporting Phrases” which are also known as “Referring phrases” or “Referencing phrases” (Sydney, 2019) (EAP Foundaton, 2019). Such phrases are lingual structures which are used for referring to the relevant target information. This source of information can be references, important ideas, discoveries or writings of experts in a special field of study (Sydney, 2019). Therefore, such phrases helped us to identify references. By using such references we don’t limit our algorithm to the phrase-classification used in (Sannier *et al*., 2016). Although, reporting phrases cannot cover all the phrases comes before the references, by using our taxonym not only we succeeded to cover majority of CRs in contractual requirement documents, but also, we could identify numerous CRs which are used in other textual documents.

Chapter 4

4 Problem and Solution-approach Analyses

In this chapter we provide high-level analyses of the problem and solution that we are working on. The Problem Analysis section focuses on analyzing a contractual document, requirements and cross-references. The analysis of the solution approach focuses on the analysis of cross-reference components, phrases leading to cross-references, and related material. We also discuss the applicable NLP techniques for processing the phrases.

4.1 Problem Analysis

As mentioned in Section 2.1.1, the contract is an official agreement between supplier and customer organizations. It is a textual document containing large number of regulatory requirements and nonregulatory requirements. Each of these requirements is named "contractual requirement". These requirements are scattered in different parts of the contract and their types ("regulatory" or "nonregulatory") are not tagged explicitly in the contract. In other words, rather regulatory requirements are mixed up with other general requirements in each page of the contract. Furthermore, any contractual document can be organized into different number of domain-specific "divisions" (such as Electrical, Mechanical, Doors and windows, Metals, etc.) (Nekvi and Madhavji, 2014). Therefore, identifying regulatory requirements from the contract for a particular subsystem (e.g., power supply), needs to explore all the divisions of the contract (a thousand pages in this case study) carefully in order to identify the corresponding regulatory requirements from the mixed set of requirements (Nekvi and Madhavji, 2014).

The described complexity is compounded when one contractual requirement refers to several references, each pointing to a different standard or legal document. For instance, the 6-page standard "CGSB 1-GP-81" contains 48 cross-references, which are interreferences to 12 unique, external regulatory documents (e.g., CGSB 1- GP-70M, ASTM D1210, ASTM D2621, and nine others) (Nekvi and Madhavji, 2014). Furthermore, the 12 externally referenced documents refer to yet other documents, which, in turn, may refer to yet others, and so on (Nekvi and Madhavji, 2014).

Thus, without automated support, identifying these myriads of references is arduous and can easily lead to non-compliant systems.

This thesis uses a contract document for the purpose of cross-reference analysis. Reason is that our case study contract contains external references with indirections, as described above. Thus, it is suitable for our analysis. In addition, a contract document is quite typical in large systems engineering projects. Thus, our results would be generalizable in such situations which abound in industry.

4.2 Solution-approach Analysis

4.2.1 Cross-reference components

We start this section with providing five samples of cross-references listed in Table 4.1:

Table 4. 1: Five samples of cross-references

Software Quality Assurance Plan
IXXX Std 829
CGSB, 1.40
IXXX 587 (A)
Electrical Code

Based on our investigation over the cross-references of contractual documents, a cross-reference can entail different components, each in different combinations of alphabets, numbers and special characters (see Table 4.1).

- ✓ sequential of words
- ✓ acronyms
- ✓ combination of words and numbers
- ✓ combination of words and special characters
- ✓ combination of acronyms and numbers
- ✓ combination of acronyms and special characters
- ✓ combination of words, acronyms, numbers and special characters

Therefore, considering what we described, we are facing a problem that the components of cross-references are often unmeaningful tokens. Also, combination of these components is not presented in static and definite formats. So, based on the mentioned circumstances, we decided to think about the identity of each component in a CRE (Section 2.4.2), because each component of a cross-reference has a unique identity in the structure of an English sentence. The five following tables show example decompositions of cross-references.

Table 4. 2: Cross-reference-Decomposition example 1

CRE	GO Wheel reserves the right to participate in all SQA activities. Provide a detailed schedule of such activities in accordance with the **Software Quality Assurance Plan** and update such schedules as required.			
Cross-reference components	Software	Quality	Assurance	Plan
Component-Identity	proper singular noun	proper singular noun	proper singular noun	proper singular noun

Table 4. 3: Cross-reference-Decomposition example 2

CRE	Software Test Documentation in accordance with **IXXX Std 829**		
Cross-reference components	IXXX	Std	829
Component-Identity	proper singular noun	proper singular noun	cardinal number

Table 4. 4: Cross-reference-Decomposition example 3

CRE	Finish coats: Alkyd base enamel conforming to **CGSB, 1.40**		
Cross-reference components	CGSB	,	1.40
Component-Identity	proper singular noun	punctuation mark, comma	cardinal number

Table 4. 5: Cross-reference-Decomposition example 4

CRE	Input Surge Withstand Capability: The UPS shall comply with **IXXX 587 (A)**				
CR-Components	IXXX	587	(	A	)
Component-Identity	proper singular noun	cardinal number	contextual separator, left parentheses	proper singular noun	contextual separator, right parentheses

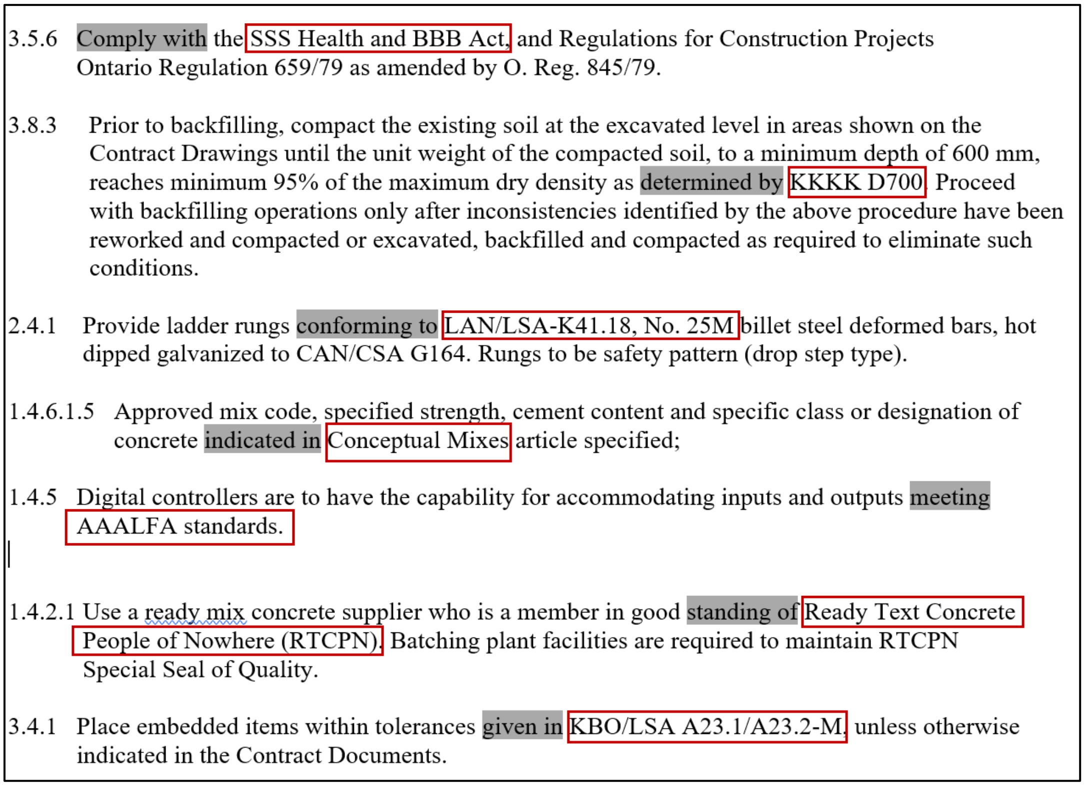
Table 4. 6: Cross-reference-Decomposition example 5

CRE	Design circuit breakers in accordance with **Electrical Code**	
CR-Components	Electrical	Code
Component-Identity	Adjective	proper singular noun

Our technical approach for dealing with the provided analysis, is discussed in Section 5.3.1.2.

4.2.2 Key-phrases

Based on our analysis over the CREs in a contractual document, we realized that in majority of CREs, cross-references are referenced by specific key-phrases. In other words, it can be stated that presenting key-phrases and cross-references with each other is a combination that can be observed in majority of CREs. Therefore, these key-phrases play a vital role in our proposed solution approach for accessing the references. Figure 4.1 depicts such key-phrases which are used surrounding the external references gathered from different parts of a contractual document. The gray highlighted text represents key-phrases that refer to the cross-references that are specified in red rectangles.



3.5.6 Comply with the SSS Health and BBB Act, and Regulations for Construction Projects Ontario Regulation 659/79 as amended by O. Reg. 845/79.

3.8.3 Prior to backfilling, compact the existing soil at the excavated level in areas shown on the Contract Drawings until the unit weight of the compacted soil, to a minimum depth of 600 mm, reaches minimum 95% of the maximum dry density as determined by KKKK D700. Proceed with backfilling operations only after inconsistencies identified by the above procedure have been reworked and compacted or excavated, backfilled and compacted as required to eliminate such conditions.

2.4.1 Provide ladder rungs conforming to LAN/LSA-K41.18, No. 25M billet steel deformed bars, hot dipped galvanized to CAN/CSA G164. Rungs to be safety pattern (drop step type).

1.4.6.1.5 Approved mix code, specified strength, cement content and specific class or designation of concrete indicated in Conceptual Mixes article specified;

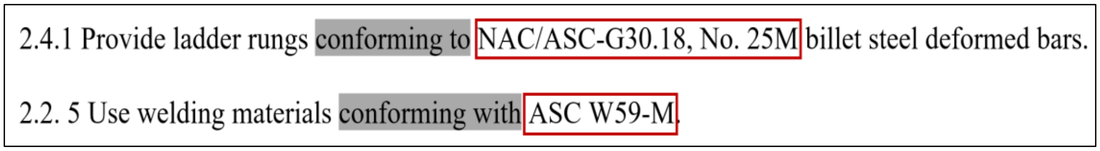
1.4.5 Digital controllers are to have the capability for accommodating inputs and outputs meeting AAALFA standards.

1.4.2.1 Use a ready mix concrete supplier who is a member in good standing of Ready Text Concrete People of Nowhere (RTCPN). Batching plant facilities are required to maintain RTCPN Special Seal of Quality.

3.4.1 Place embedded items within tolerances given in KBO/LSA A23.1/A23.2-M, unless otherwise indicated in the Contract Documents.

Figure 4. 1: A view of key-phrases surrounding external references

An important aspect to note is the tenses/forms of key-phrases in CREs. For example, as shown in Figure 4.1, “comply with” (clause 3.5.6) is in simple present tense, “Determined by” (clause 3.8.3) is in simple present passive voice, “conforming to” (clause 2.4.1) is a present participle statement, etc. Also, sometimes, a key-phrase can be presented with different propositions. For example, Figure 4.2 shows that “conforming” key-phrase has “to” in one CRE, and “with” in another.



2.4.1 Provide ladder rungs conforming to NAC/ASC-G30.18, No. 25M billet steel deformed bars.

2.2. 5 Use welding materials conforming with ASC W59-M.

Figure 4. 2: A view of a key-phrase with two different propositions

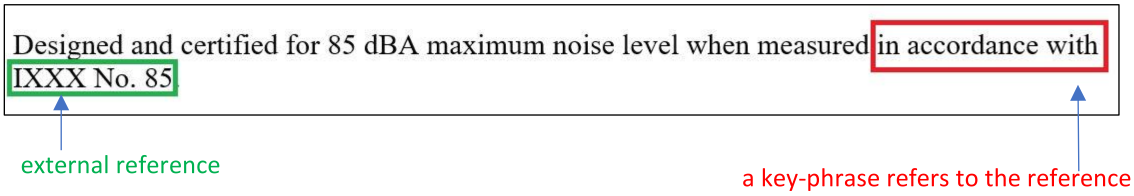
Furthermore, our analysis suggests that key-phrases are juxtaposed in three forms with a CRE, what we call: (i) Direct Cue (DC), (ii) InDirect Cue (IDC) and (iii) No Cue (NC) and explained below.

4.2.2.1 Direct Cue (DC):

Direct Cue CRE include a key-phrase presents somewhere in the CRE and the reference is coming after the key-phrase. The main format of a Direct Cue CRE is:

<DC_CRE>::= <key-phrase><reference>

An example is provided in Figure 4.3:



Designed and certified for 85 dBA maximum noise level when measured in accordance with IXXX No. 85.

external reference
a key-phrase refers to the reference

Figure 4. 3: Cross-reference expression example with DC type (example is extracted from contractual document)

As shown in Figure 4.3, there are some words in this CRE and then in the middle of the sentence there is a key-phrase “in accordance with” which refers to the reference. Immediately after that key-phrase, the reference “IXXX No. 85” is coming. Since the reference is directly coming after the reporting phrase, we named it CRE with direct cue. Based on our investigation in one contractual document with 683 pages and 10345 paragraphs, approximately 667 references (83%) out 802 total number of external references are DC references.

Since DC references are the most common type of references in contractual documents, we decide to focus on identifying these types of references in this thesis.

4.2.2.2 InDirect Cue (IDC)

InDirect Cue CREs include a key-phrase in the middle of a CRE and just a part of the reference is coming after the key-phrase and the rest of the reference is coming at any part of the CRE; it might be in the first part of the sentence or before the key-phrase. The main format of an InDirect Cue CRE is:

<IDC_CRE>::=<reference-part2><key-phrase><reference-part 1 >

An example is provided in Figure 4.4.

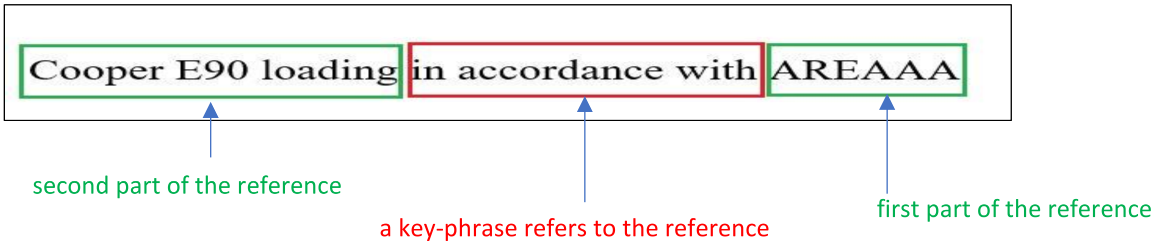

Cooper E90 loading
in accordance with
AREAAA
second part of the reference
a key-phrase refers to the reference
first part of the reference

Figure 4. 4: Cross-reference expresson example with IDC type (example is extracted from contractual document)

As shown in Figure 4.4, “Cooper E90” is a part of the reference coming before the key-phrase. The rest of this reference is “AREAAA” which is coming after the key-phrase. Since the reference can be presented at any part of the sentence, we named it InDirect Cue CREs. Based on our investigation in one contractual document with 683 pages and 10345 paragraphs, approximately 15 references out 802 total number of external references are IDC references (1.8%). Recognizing this type of references is not done in this thesis and it would be supported in future work.

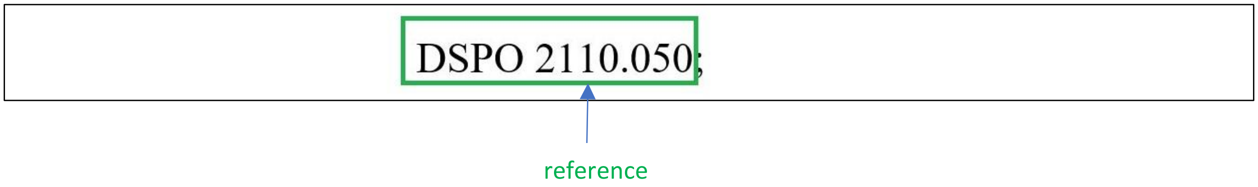
4.2.2.3 No Cue (NC)

No Cue CREs don’t have auxiliary key-phrases in the paragraph that the cross-reference is presented. This is the reason we named it No Cue CREs.

The main format of a No Cue CRE is:

<NC_CRE>::=<reference>

An example is provided in Figure 4.5:



DSPO 2110.050;

Figure 4. 5: Cross-reference expression example with NC type (example is extracted from contractual document)

As shown in Figure 4.5, there is no key-phrase around the reference. Based on our study in one contractual document with 683 pages and 10345 paragraphs, approximately 112 references out 802 total number of external references are NC References (13.96%). Investigating and recognizing this type of references is not done in the current work and it would be supported in future work.

4.2.2.4 Reporting Phrases

As described earlier (Sections 4.2.2.1 and 4.2.2.2), in majority of CREs, cross-references are referenced by specific kinds of key-phrases. Therefore, we realized that the first step for identifying external references, can be using such key-phrases. We had a study in English language literature to acquire more information about the essence of these key-phrases. In this study, we found different classifications for English phrases. For instance, helping phrases, linking phrases, reporting phrases, action phrases, etc. In this study, we observed that, the majority of phrases which are used in one contractual document with 683 pages, are categorized in the group of reporting phrases. Table 4.7 shows these phrases which are collected from different resources ((AUSB, 2017), (Toronto, 2005), (Centre and Guide, 2014), (EAP Foundaton, 2019), (University, 2007)). Based on the provided established definitions in English Language literature, reporting phrases are also known as “referring phrases” or “referencing phrases” (Sydney, 2019) (EAP Foundaton, 2019). They are lingual structures which are used for referring to the key details that uniquely identify a source of information. This source of information can be references, important ideas, discoveries or writings of experts in a special field of study (Sydney, 2019). Technically, these phrases connect the in-text citation to the information which you are citing (PennState Library University, 2020). Therefore, we realized that these phrases are exactly types of phrases that can help us to find the references. This is because, as we explained previously in Sections 4.2.2.1 and 4.2.2.2, in each DC or IDC CRE, such key-phrases are referring to a piece of information. In our work, this piece of information is the cross-references that we are supposed to identify. Therefore, we decided to use reporting phrases, since they are specifically used for mentioning the references. In this way, not only we can cover majority of CRs in contractual requirement documents, but also, we can identify numerous CRs which are used in other textual documents. The result and quality of our work which are respectively provided in Chapters 6 and 7, can attest this statement.

List of reporting phrases is provided in Table 4.7. Provided phrases in this table are gathered from different resources. It is important to note that, in this table we intentionally repeat some of the phrases by different tenses and different propositions. This is because we intend to show that in CREs these key-phrases can be presented by different tenses and different propositions and our proposed algorithm should tack care all of these tenses and propositions for each key-phrase.

Table 4. 7: Reporting phrases collected experimentally from different resources (AUSB, 2017)(Toronto, 2005) (Centre and Guide, 2014)(EAP Foundaton, 2019) (University, 2007)

Reporting/Referring phrases		
along with	based on	by the warranty provisions of
conform to	conform with	confirmation from
in conformance with	comply with	in compliance with
covered in	covered by	contained in
determined by	define by	define as
define in	definition from	definition of
directed by	described in	fitting to
found in	find in	found by
according to	in accordance with	in accordance to
given in	in conjunction with	itemized in
included in	indicated in	indicated on
meeting	identified in	identify
mention	mentioned by	mentioned for
map to	noted in	provided in
provided by	permitted within	permitted by
required by	refer to	refers to
standing of	stated in	specified in
see	satisfy	maintain through
supplemented by	set by	with the requirements of
proclaim	state	pointed out
point to	comment	observe
reported by	express	consider
explored	illustrated	emphasized
proposed by	acknowledge	admit
concede	remark	conclude
confirmed by	discovers	established by
admits	recognize	declares
reflect	realize	requests
imply	estimate	investigate
inform	list in	outlined in
reveal	restate	presented in
uses	studied by	hypothesize
theorizes	contradict	approved by
assert	recommended by	suggested by
view	demonstrate with	calculate
suppose	stipulate	take into consideration
investigate	contrast	compare
develop	infer	recognize
elaborate	contribute	address

We have categorized all the above reporting phrases into "**Compliance"** and "**None-Compliance**" phrases. Table 4.8 shows this categorization. In "Frequency" column you can see number of representations that each phrase appears before external cross-references.

Table 4. 8: Classifying reporting phrases into "Compliance" and "None-Compliance" phrases

Compliance		**None-Compliance**	
Phrase	**Frequency**	**Phrase**	**Frequency**
along with	1	by the warranty provisions of	41
conform to	65	determined by	10
conform with	3	directed by	10
confirmation from	0	indicated in	26
in conformance with	7	indicated on	12
fitting to	0	permitted by	41
according to	9	required by	23
in accordance with	241	specified in	32
in accordance to	2	proposed by	16
standing of	1	approved by	16
based on	12	recommended by	13

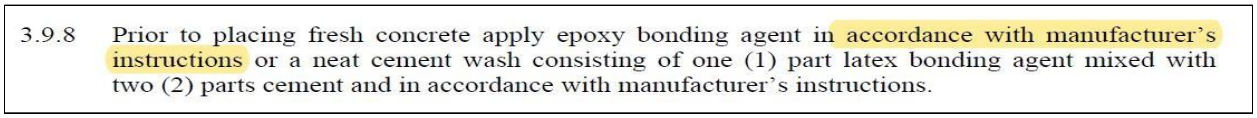
All of the rest phrases which are in Table 4.7 and aren't in Table 4.8 are shown under 10 times in our contract case study (683 pages). So, we don't list them here anymore.

4.2.3 Cross-Reference Validation Conditions

As elaborated in Section 4.2.1, cross-references in contractual requirements are represented in different forms (see Tables 4.2, 4.2, 4.3, 4.4 and 4.6). Therefore, in our proposed algorithm (Chapter 5) we decide to define standard properties for approving the identified references as valid references. We defined our standards based on the properties of APA in-text citations from American Psychological Association (APA) standards. APA in-text citations provide academic writing foundation and standards that allows authors to write clearly and accurately (Association, 2019a). In scholarly writing, APA in-text citations standards are used for referring to, summarizing, paraphrasing or quoting from another source (Association, 2019b). Therefore, in our proposed solution, once a reference is identified, it should meet some properties to be accepted as a valid cross-reference. To do this, we defined the following properties which are gathered experimentally form APA in-text citing properties (University, 2018) and contractual document:

1. References can consist of acronyms with all capital letters (Example: In accordance with IXXX No. 85.)
2. References can consist of sequential words (camel case format) that each word starts with capital letters (Example: in accordance with National Building Code)
3. References can consist of digits (Example: in accordance with IXXX Std No. 85.)
4. References can consist of special characters like “/, -, _, &, (,)”, etc. (Example: given in NAC/ASC A23.1/A23.2-M.)
5. CREs may have more than one reference. In this case, references can be split by different ways:
    - i. Split by and (Example: in accordance with the Manufacturer's Instructions and NAC/ASC A23.1/A23.2-M)
    - ii. Split by or (Example: shown in the Contract Documents or as determined by the GO Wheel)
    - iii. Split by and or (Example: in accordance with the NAC / ASC S16.1-M and or National Building Code)
    - iv. Split by comma (in accordance with IXXX Std 1058, Standard for Software Project Management Plans)

It is important to note that, based on our investigation, in one contractual document with 683 pages and 10345 paragraphs, 8 references out 802 total number of external references are references which doesn’t meet APA properties (0.99%). Figure 4.6 shows an example from external reference which does not meet APA properties.



3.9.8 Prior to placing fresh concrete apply epoxy bonding agent in accordance with manufacturer’s instructions or a neat cement wash consisting of one (1) part latex bonding agent mixed with two (2) parts cement and in accordance with manufacturer’s instructions.

Figure 4. 6: An example from external reference that doesn’t meet APA properties

In this figure, “manufacturer’s instructions” is an external reference which is not written in capital words or camel case format. Therefore, it doesn’t meet APA properties #1 or #2. We don’t support these types of references in our thesis, since we believe that, references must be written in standard format from scratch; the standard format which is proposed by APA. Supporting these kinds of references are not align with the standards that we define in our algorithm.

4.2.4 Analysis of NLP techniques applicable for processing the phrases

From the analyses provided in the previous sections, it’s obvious that for identifying cross-references, we need to deal with various types of phrases, words, alphabets, numbers and special characters. NLP is a useful tool for dealing with the complexity and ambiguities of human languages.

Sequence labelling (Section 2.7) is a typical NLP task which would be raised once we are facing with sequences of tokens (Section 2.9). It is the process of assigning a label to each token in a given input sequence. Token Labeling (Section 2.7.1) and Span Labeling (Section 2.7.2) are two different types of sequence labeling tasks that can apply on different components of a sentence. The first one is applicable by Part of Speech Tagging (POST) (Section 2.7.1) and the later one can be achieved by Named Entity Recognition (NER) (Section 2.7.2). Part of speech tagging aims on identifying which grammatical group a word belongs to, so whether it is a NOUN, ADJECTIVE, VERB, ADVERBS and so forth, based on the context. This means it looks for relationships within the sentence and gives a corresponding tag to each word in a sentence. Named Entity Recognition on the other hand, tries to find out whether or not a word is a named entity (persons, locations, organizations, time expressions etc.). This problem can be broken down into detection of names followed by classification of name into the corresponding categories (see Figure 2.2). Most often once a word is recognized by NER, it may be recognized as a noun by POST. So, POST is more global, since it can determine the relationships between the first and the last word of a sentence (Quora, 2017).

For dealing with words and phrases in natural language, Word Lexical Disambiguation (Semantic or Syntactic) (Section 2.8) is one of the very first approaches that comes to mind. Semantic ambiguity occurs when a word, phrase or sentence of a context, has more than one interpretation. Word Sense Disambiguation (WSD) (Section 2.8), is defined as the ability to determine which meaning of word is activated by the use of word in a particular context. On the other hand, syntactic ambiguity is a situation where a sentence may be interpreted in more than one way due to ambiguous sentence structure. One of the abilities of Part of speech taggers with high level of accuracy is solving word’s syntactic ambiguity.

Therefore, considering the above, in our problem, POST can be the choice NLP technique for figuring out the essence of each token in a CRE. This is because, we should recall that in our problem we are not only faced with recognizing the names in a sentence; we are also facing various formats and identities of words (see Tables 4.2, 4.2, 4.3, 4.4 and 4.6). For dealing with these complexities, NER cannot provide us sufficient power to identify the identity of all the tokens. On the other hand, we are not facing a problem where individual words are interpreted differently and causes ambiguity. Thus, the techniques for word lexical disambiguation (Section 2.8) cannot help us to figure out our problem.

Chapter 5

5 Proposed Solution

In this chapter, we go through different methods used for automatically identifying cross-references. Figure 5.1, shows the algorithm of our work. This algorithm is our own creation. As shown in the flowchart (Figure 5.1), this algorithm consists of six main steps, which are further categorized into the sub-steps. This chapter starts with the flowchart of our algorithm which depicts all the major steps and important sub steps of our work. Then it would be continued with describing the first step of the algorithm which is referred to as Structuring Data with the following sub-steps: Reading Contractual Requirements, Noise Removal, and Paragraph Splitting. After that we continue with Pattern Setting. The core and concrete part of our work is Fundamental Data Preparation, which is the first step that will be explained in Pattern Setting section. In this part, we will introduce two important taxonomies and then we will describe the functionality of the module referred to as Has_Whitelist. Then, we explain Pattern Recognition sub-steps including Word Tokenization, Part of Speech Tagging, Pattern Recognition Parser, Generating Tree, and again Has_ Whitelist, which all perform different functionalities at this phase. We then explain the next step of the algorithm Reference Identifier, with its module Has_APA_Properties, which is designed to determine the validity of identified references based on the standards of American Psychological Association. After describing all of these steps, you will understand how the references are extracted from contractual requirements. The chapter would be then continued with explaining the techniques applied in the second step of reference identification which is done over the web pages. This step starts with Web Scraping and then it’s all subsets including Google Search API, Choosing URL and Extracting HTML Context. Finally, we will terminate this section with describing the last step of the algorithm which is call Visualization designing to represent all of the identified references.

5.1 Flowchart of algorithm for extracting external cross-references (DC references)

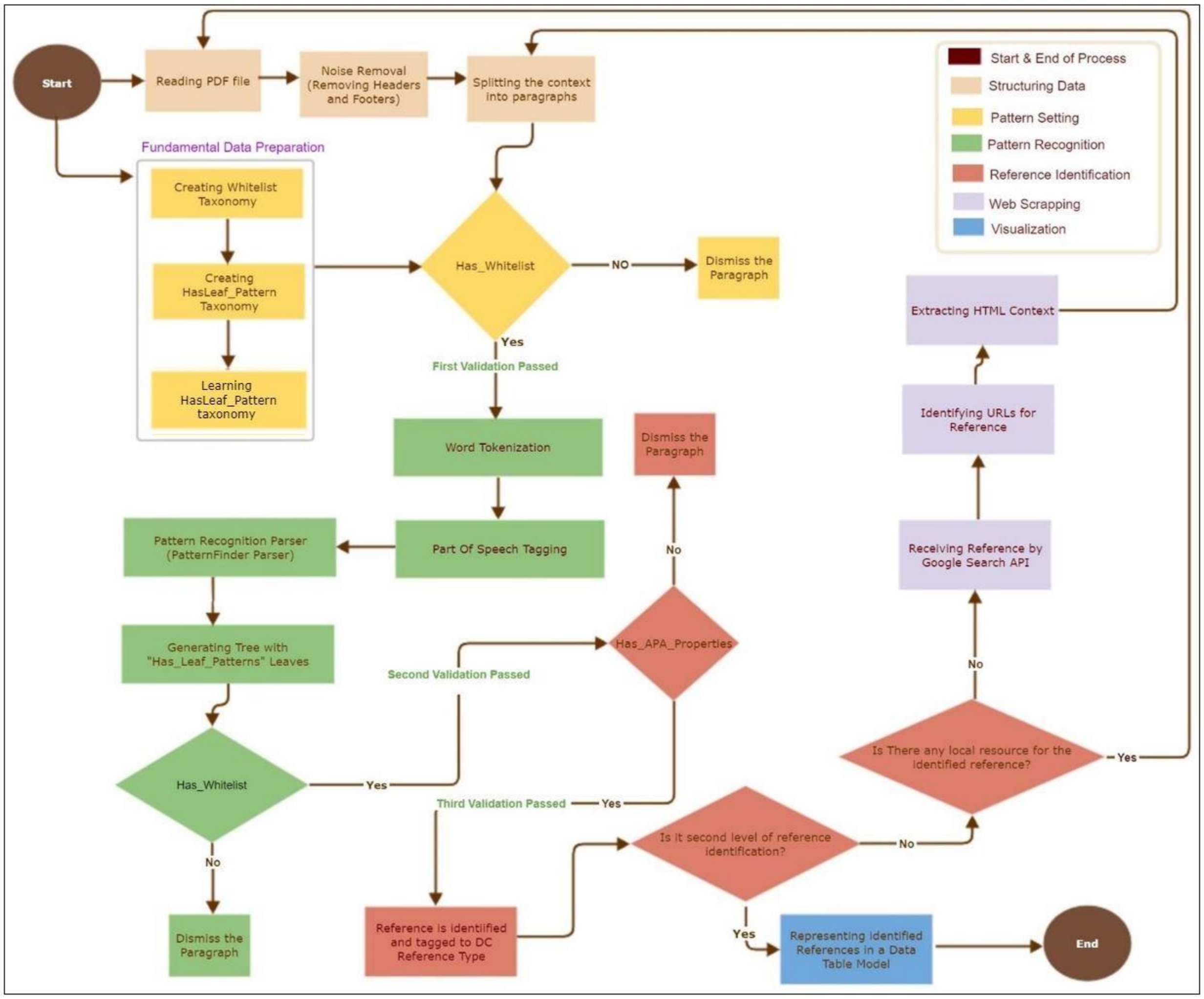

Start
Reading PDF file
Noise Removal (Removing Headers and Footers)
Splitting the context into paragraphs
Start & End of Process
Structuring Data
Pattern Setting
Pattern Recognition
Reference Identification
Web Scrapping
Visualization
Fundamental Data Preparation
Creating Whitelist Taxonomy
Creating HasLeaf_Pattern Taxonomy
Learning HasLeaf_Pattern taxonomy
Has_Whitelist
NO
Dismiss the Paragraph
Yes
First Validation Passed
Extracting HTML Context
Identifying URLs for Reference
Word Tokenization
Dismiss the Paragraph
Receiving Reference by Google Search API
Pattern Recognition Parser (PatternFinder Parser)
Part Of Speech Tagging
No
Generating Tree with "Has_Leaf_Patterns" Leaves
Has_APA_Properties
Second Validation Passed
No
Is There any local resource for the identified reference?
Yes
Has_Whitelist
Yes
Third Validation Passed
Yes
Is it second level of reference identification?
No
No
Dismiss the Paragraph
Reference is identified and tagged to DC Reference Type
Yes
Representing identified References in a Data Table Model
End

Figure 5. 1: Flowchart of algorithm for Extracting External Cross-references (DC References)

The flowchart in Figure 5.1 is explained through the sub-sections 5.2 to 5.7.

5.2 Structuring data

5.2.1 Reading Contractual Requirement document:

The first step in implementing of our problem is having the proper ability for reading significant number of PDF documents. In order to perfectly accomplish this, we used the parser module from

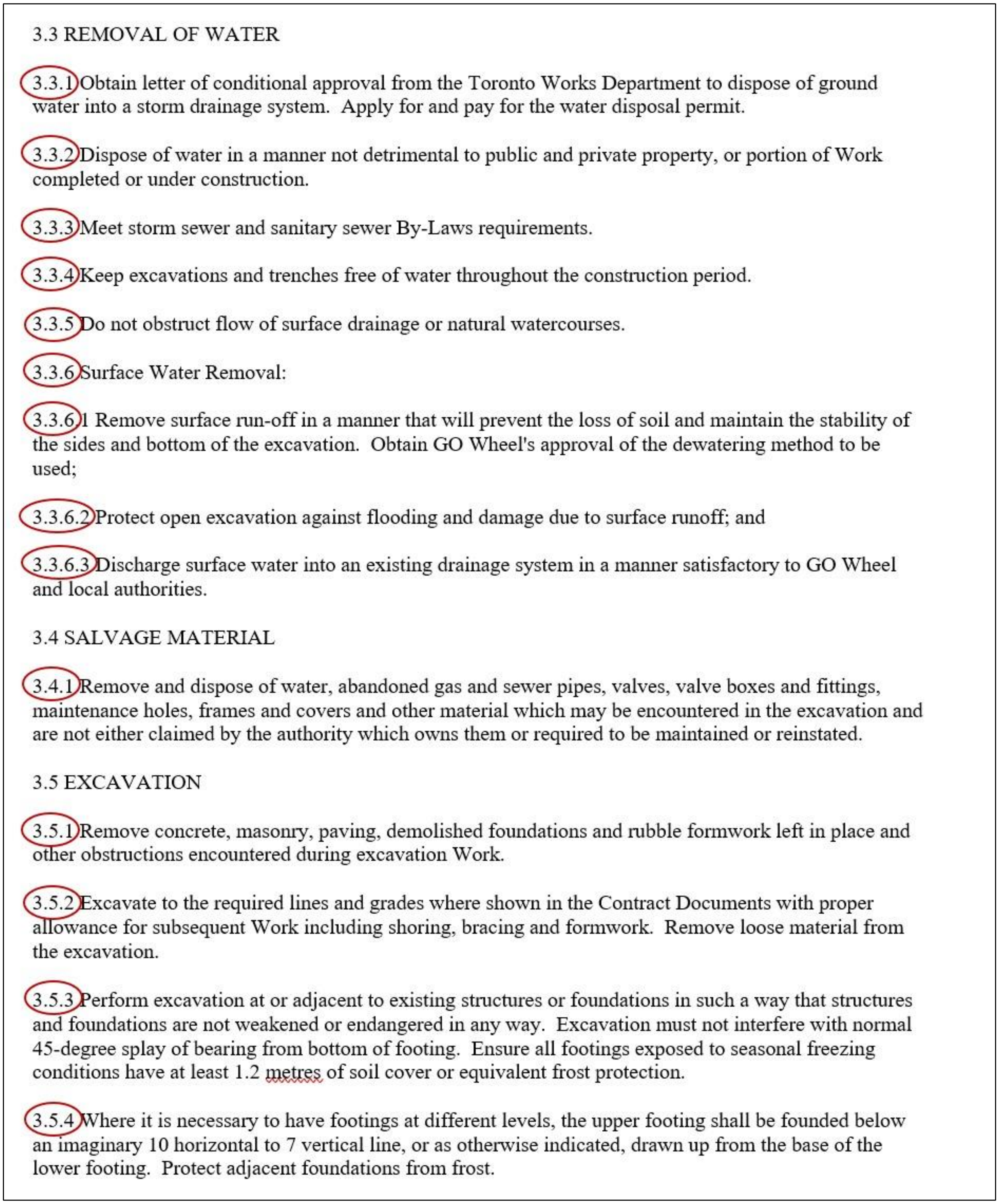
Tika library. Tika is capable of detecting and deriving metadata and text from various types of files such as PPT, XLS, and PDF (David Ascer, 2004). Therefore, it can be stated that using Tika parser causes to parse the PDFs accurately and readily and finally the parser returns a dictionary containing the main context of the PDF file.

5.2.2 Noise Removal

Since the subject of our work is extracting external references, we don’t need the headers and footers of the pages of contractual requirements. Therefore, in this work these parts are extra information that are considered as the first noises of the work. For removing such noises, it is just needed to extract the main content value from the dictionary that was generated in the previous step (Section 5.2.1). Therefore, the result of this step has a string type containing the main context of the PDF file which are presented in the form of sequential paragraphs.

5.2.3 Paragraph Splitting

Based on the outputs of the previous steps, at this step we face a string format containing a large number of paragraphs. Contractual requirement documents consist of a number of contractual requirements with each of the paragraphs starting with a number for representing the number of that contractual requirement. Since the main purpose of our work at the first level was identifying external references from each contractual requirement, we needed to split the whole context into the existing number of paragraphs in the document. For this, we split all the content parts into “\n\t”. In Python programing language, “\n” means a new line and “\t” means a tab (David Ascer, 2004). Figure 5.2 represents a view from different contractual requirements in a contractual requirement document and red circles around the numbers highlight the number of each contractual requirement.



3.3 REMOVAL OF WATER

3.3.1 Obtain letter of conditional approval from the Toronto Works Department to dispose of ground water into a storm drainage system. Apply for and pay for the water disposal permit.

3.3.2 Dispose of water in a manner not detrimental to public and private property, or portion of Work completed or under construction.

3.3.3 Meet storm sewer and sanitary sewer By-Laws requirements.

3.3.4 Keep excavations and trenches free of water throughout the construction period.

3.3.5 Do not obstruct flow of surface drainage or natural watercourses.

3.3.6 Surface Water Removal:

3.3.6.1 Remove surface run-off in a manner that will prevent the loss of soil and maintain the stability of the sides and bottom of the excavation. Obtain GO Wheel's approval of the dewatering method to be used;

3.3.6.2 Protect open excavation against flooding and damage due to surface runoff; and

3.3.6.3 Discharge surface water into an existing drainage system in a manner satisfactory to GO Wheel and local authorities.

3.4 SALVAGE MATERIAL

3.4.1 Remove and dispose of water, abandoned gas and sewer pipes, valves, valve boxes and fittings, maintenance holes, frames and covers and other material which may be encountered in the excavation and are not either claimed by the authority which owns them or required to be maintained or reinstated.

3.5 EXCAVATION

3.5.1 Remove concrete, masonry, paving, demolished foundations and rubble formwork left in place and other obstructions encountered during excavation Work.

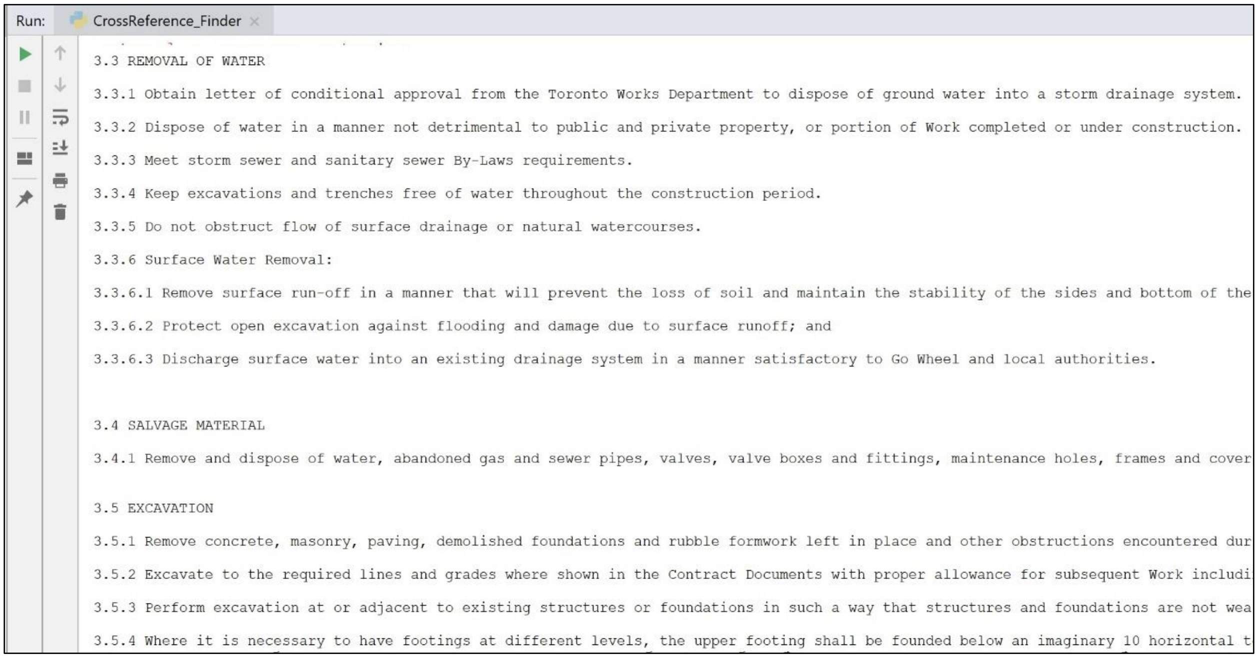
3.5.2 Excavate to the required lines and grades where shown in the Contract Documents with proper allowance for subsequent Work including shoring, bracing and formwork. Remove loose material from the excavation.

3.5.3 Perform excavation at or adjacent to existing structures or foundations in such a way that structures and foundations are not weakened or endangered in any way. Excavation must not interfere with normal 45-degree splay of bearing from bottom of footing. Ensure all footings exposed to seasonal freezing conditions have at least 1.2 metres of soil cover or equivalent frost protection.

3.5.4 Where it is necessary to have footings at different levels, the upper footing shall be founded below an imaginary 10 horizontal to 7 vertical line, or as otherwise indicated, drawn up from the base of the lower footing. Protect adjacent foundations from frost.

Figure 5. 2: Overview of contractual requirement numbers and contexts

Figure 5.3 shows the result of the structuring data step.



```
3.3 REMOVAL OF WATER

3.3.1 Obtain letter of conditional approval from the Toronto Works Department to dispose of ground water into a storm drainage system.

3.3.2 Dispose of water in a manner not detrimental to public and private property, or portion of Work completed or under construction.

3.3.3 Meet storm sewer and sanitary sewer By-Laws requirements.

3.3.4 Keep excavations and trenches free of water throughout the construction period.

3.3.5 Do not obstruct flow of surface drainage or natural watercourses.

3.3.6 Surface Water Removal:

3.3.6.1 Remove surface run-off in a manner that will prevent the loss of soil and maintain the stability of the sides and bottom of the

3.3.6.2 Protect open excavation against flooding and damage due to surface runoff; and

3.3.6.3 Discharge surface water into an existing drainage system in a manner satisfactory to Go Wheel and local authorities.


3.4 SALVAGE MATERIAL

3.4.1 Remove and dispose of water, abandoned gas and sewer pipes, valves, valve boxes and fittings, maintenance holes, frames and cover

3.5 EXCAVATION

3.5.1 Remove concrete, masonry, paving, demolished foundations and rubble formwork left in place and other obstructions encountered dur

3.5.2 Excavate to the required lines and grades where shown in the Contract Documents with proper allowance for subsequent Work includi

3.5.3 Perform excavation at or adjacent to existing structures or foundations in such a way that structures and foundations are not wea

3.5.4 Where it is necessary to have footings at different levels, the upper footing shall be founded below an imaginary 10 horizontal t
```

Figure 5. 3: A view from final result of structuring data

5.3 Pattern setting

5.3.1 Fundamental Data Preparation

5.3.1.1 Creating Whitelist

As elaborated on in Section 4.2.2, paragraphs which contains external references use specific kinds of key-phrases (see Figure 4.1). For supporting these key-phrases we decided to use a more general type of phrase called “Reporting Phrases” (Section 4.2.2.4) which are specifically used for mentioning the references. Therefore, to have the access to all of these phrases, we created a text file called “whitelist”. This whitelist file contains all the phrases that have been collected in Table 4.7.

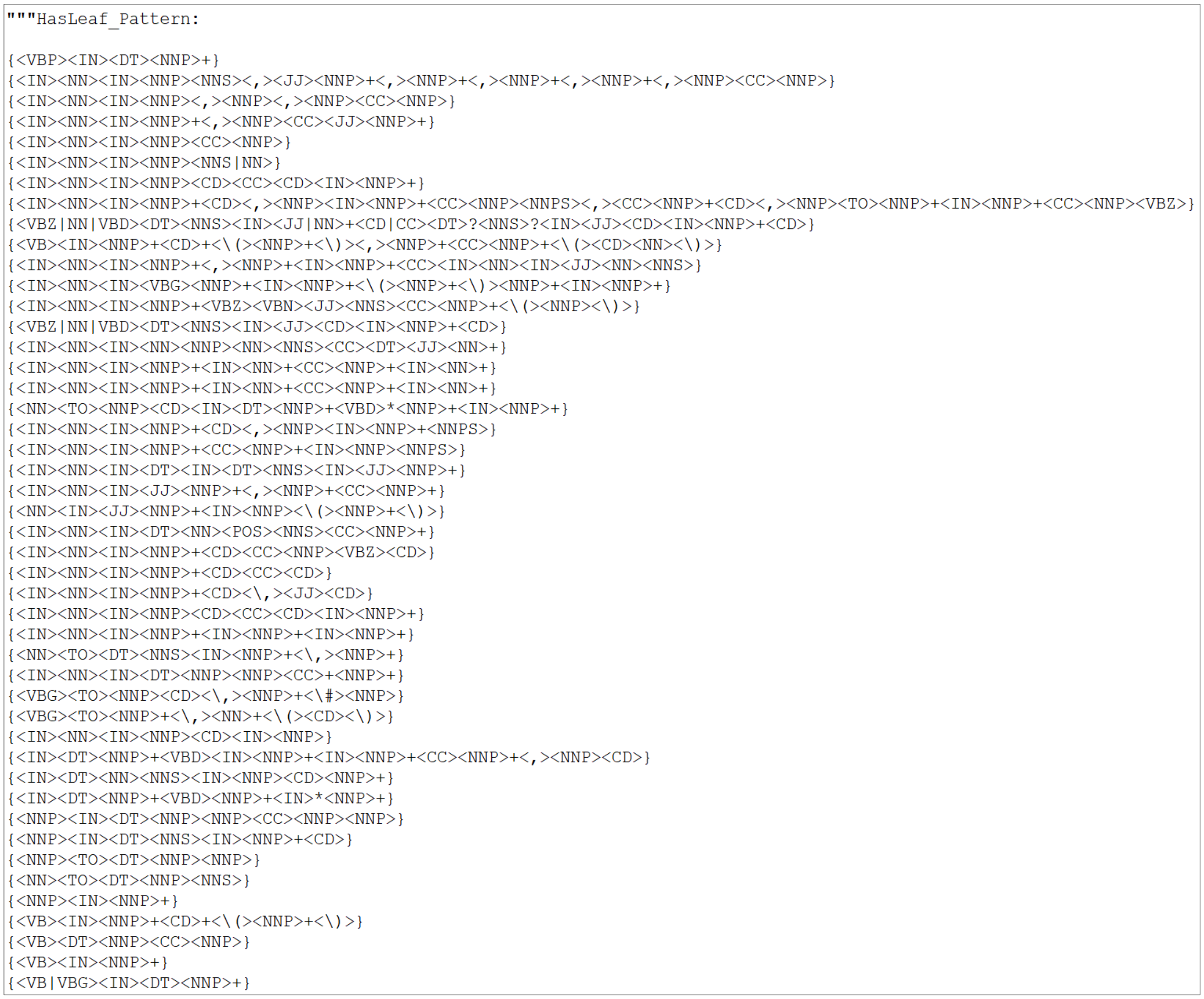
5.3.1.2 Creating HasLeaf_Pattern Taxonomy

We examined a number of the cross-reference expressions and concluded that many of them are using some specific patterns in stating the references. These patterns are repeated throughout the contractual requirement documents. Therefore, we decided to first collect these patterns and create them as a taxonomy, and then find all the matched patterns. To define these patterns, we needed to know how English language words are sitting beside each other in English grammar. We also needed to know and specify the role of each word of the sentences. Part of Speech Tagging (POST) (Section 2.7.1) provides us the capability of labeling or tagging the words. It is assigned to a single word according to its role in the sentence. In other words, POS uses specific kinds of labels for any single word. For example, it uses **VB** for verbs, **NN** for the nouns, **PR**+**DT** for pronouns, **JJ** for adjectives, **RB** for adverbs, **IN** for prepositions, **CC** for conjunctions, **UH** for interjections. Table 5.1 shows many other tags for the rest categories of words.

Table 5. 1: Different types of tags for the categories of words (Bird, Klein, 2009)

Tag	Meaning	Examples
ADJ	adjective	new, good, high, special, big, local
ADV	adverb	really, already, still, early, now
CNJ	conjunction	and, or, but, if, while, although
DET	determiner	the, a, some, most, every, no
EX	existential	there, there's
FW	foreign word	dolce, ersatz, esprit, quo, maitre
MOD	modal verb	will, can, would, may, must, should
N	noun	year, home, costs, time, education
NP	proper noun	Alison, Africa, April, Washington
NUM	number	twenty-four, fourth, 1991, 14:24
PRO	pronoun	he, their, her, its, my, I, us
P	preposition	on, of, at, with, by, into, under
TO	the word to	to
UH	interjection	ah, bang, ha, whee, hmpf, oops
V	verb	is, has, get, do, make, see, run
VD	past tense	said, took, told, made, asked
VG	present participle	making, going, playing, working
VN	past participle	given, taken, begun, sung
WH	wh determiner	who, which, when, what, where, how

Based on the provided information of Table 5.1, we created a taxonomy of grammatical patterns. This taxonomy is a text file and we named it “HasLeaf_Pattern”. Figure 5.4, shows a view from these patterns. Each pattern starts with an opening bracket ({) and ends with a closing bracket (}). Each pattern is representative of a part of an English sentence. Our main objective of creating this taxonomy is having a list of English language grammatical patterns which are the most common patterns in different CREs. Furthermore, these patterns help us to specify the start and end points of the reference in a CRE. It is important to note that descriptions about Quantifiers like +, *, ?, | which are used in Figure 5.4 are provided in Tables 2.1 and 2.2.



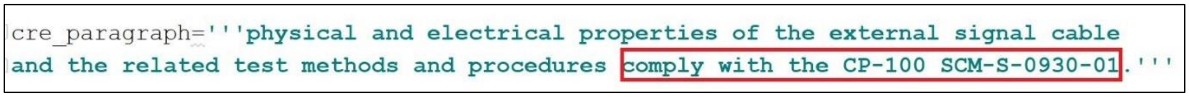
```
"""HasLeaf_Pattern:

{<VBP><IN><DT><NNP>+}
{<IN><NN><IN><NNP><NNS><,><JJ><NNP>+<,><NNP>+<,><NNP>+<,><NNP>+<,><NNP><CC><NNP>}
{<IN><NN><IN><NNP><,><NNP><,><NNP><CC><NNP>}
{<IN><NN><IN><NNP>+<,><NNP><CC><JJ><NNP>+}
{<IN><NN><IN><NNP><CC><NNP>}
{<IN><NN><IN><NNP><NNS|NN>}
{<IN><NN><IN><NNP><CD><CC><CD><IN><NNP>+}
{<IN><NN><IN><NNP>+<CD><,><NNP><IN><NNP>+<CC><NNP><NNPS><,><CC><NNP>+<CD><,><NNP><TO><NNP>+<IN><NNP>+<CC><NNP><VBZ>}
{<VBZ|NN|VBD><DT><NNS><IN><JJ|NN>+<CD|CC><DT>?<NNS>?<IN><JJ><CD><IN><NNP>+<CD>}
{<VB><IN><NNP>+<CD>+<\(><NNP>+<\)><,><NNP>+<CC><NNP>+<\(><CD><NN><\)>}
{<IN><NN><IN><NNP>+<,><NNP>+<IN><NNP>+<CC><IN><NN><IN><JJ><NN><NNS>}
{<IN><NN><IN><VBG><NNP>+<IN><NNP>+<\(><NNP>+<\)><NNP>+<IN><NNP>+}
{<IN><NN><IN><NNP>+<VBZ><VBN><JJ><NNS><CC><NNP>+<\(><NNP><\)>}
{<VBZ|NN|VBD><DT><NNS><IN><JJ><CD><IN><NNP>+<CD>}
{<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>}
{<IN><NN><IN><NNP><CD><CC><CD><IN><NNP>+}
{<IN><NN><IN><NNP>+<IN><NNP>+<IN><NNP>+}
{<NN><TO><DT><NNS><IN><NNP>+<\,><NNP>+}
{<IN><NN><IN><DT><NNP><NNP><CC>+<NNP>+}
{<VBG><TO><NNP><CD><\,><NNP>+<\#><NNP>}
{<VBG><TO><NNP>+<\,><NN>+<\(><CD><\)>}
{<IN><NN><IN><NNP><CD><IN><NNP>}
{<IN><DT><NNP>+<VBD><IN><NNP>+<IN><NNP>+<CC><NNP>+<,><NNP><CD>}
{<IN><DT><NN><NNS><IN><NNP><CD><NNP>+}
{<IN><DT><NNP>+<VBD><NNP>+<IN>*<NNP>+}
{<NNP><IN><DT><NNP><NNP><CC><NNP><NNP>}
{<NNP><IN><DT><NNS><IN><NNP>+<CD>}
{<NNP><TO><DT><NNP><NNP>}
{<NN><TO><DT><NNP><NNS>}
{<NNP><IN><NNP>+}
{<VB><IN><NNP>+<CD>+<\(><NNP>+<\)>}
{<VB><DT><NNP><CC><NNP>}
{<VB><IN><NNP>+}
{<VB|VBG><IN><DT><NNP>+}
```

Figure 5. 4: A taxonomy of common grammatical patterns

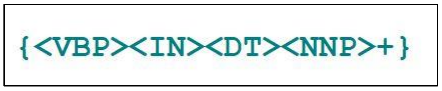
In the following there are three figures for finding a better understating form the usage of HasLeaf_Pattern taxonomy. In Figure 5.6, you can see a paragraph containing a reporting phrase/key-phrase followed by a cross-reference. Figure 5.7 shows one of the patterns of

HasLeaf_Pattern taxonomy, which can be matched with the selected red part in Figure 5.6. Figure 5.8 shows a corresponding sample from the tokens and their roles.



```
cre_paragraph='''physical and electrical properties of the external signal cable
and the related test methods and procedures comply with the CP-100 SCM-S-0930-01.'''
```

Figure 5. 5: A paragraph contains a reporting phrase and a cross-reference



```
{<VBP><IN><DT><NNP>+}
```

Figure 5. 6: A pattern which can be matched with the selected red part in Figure 5.5

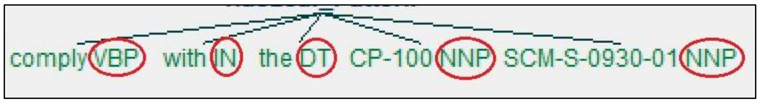


Figure 5. 7: A sample from tokens and their roles

Considering the provided examples, here we mention some important aspects of our patterns:

1. As shown in Figure 5.8, there are two <NNP> in this example. So, considering the functionality of "+" quantifier (Section 2.10.2 and Table 2.2) (interpreted as unlimited allowed; one required), the defined pattern shown in Figure 5.7 is matched with our example in Figure 5.6; Because we have two sequential <NNP> in this example. Therefore, this pattern will also be matched with other references that have the same pattern and have still sequential <NNP> roles. So, it is one of the advantages our patterns because, such structures help us to identify many numbers of references having the same pattern. However, this advantage sometimes causes a critical situation. This critical situation with its solution (applying APA properties (Section 4.2.3)) will elaborately provide in Section 5.5.l.

2. If the pattern in Figure 5.7 change to {<VBP><IN><DT>*<NNP>+}, it will also match with the following phrase:

“Comply with **the** CBC-100 S-0930-01”

Because here, the “*” quantifier after <DT> is interpreted as: the word “the” is not required (see Table 2.2). Therefore, we can see that by adding only one sign (*) in a correct place we can define a pattern that can be matched with many references.

3. As elaborated in Section 4.1, reporting phrases (Table 4.7) can be presented in different tenses/forms. For example, if in the red rectangle part of Figure 5.5 we have “complying with” instead of “comply with” (for example: “**complying** with the CP-100 SCM-S-0930-01”), then the pattern in Figure 5.6 cannot be matched with our example (Figure 5.5) anymore. Because tense of “complying” is <VBG>; interprets as “present participle”, and tense of “comply” is <VBP>; interprets as “verb, non-3rd person singular present”. To support both forms of the key-phrase, we can define two following patterns:

   ✓{<**VBP**><IN><DT>*<NNP>+}
   ✓{<**VBG**><IN><DT>*<NNP>+}

   But as you can see both of these patterns are exactly the same and the difference is just in the key-phrase part. So, in such situations we decide to get help metacharacters (Section 2.10.1) in order to reduce the number of the same patterns in our taxonomy. In this example the “|” metacharacter (see Table 2.1) interprets as “or” (matches either expression it separates) and can help us to combine the two above patterns. So, if we use the following pattern, both forms of key-phrases (“comply with” and “complying with”) would be covered:

   {<VB|VBG><IN><DT>*<NNP>+}

4. Some of the key-phrases can be presented with more than one proposition. For example, verb “confirm” can be presented with both “to” and “with”. Considering the Table 5.1, <IN> is for supporting all the common propositions including “with”, “as”, “in”, “on”, “of”, “at”, etc. So, by using <IN> after the key-phrases we can cover various propositions. <TO> is another tag which is specifically used for supporting the “to” proposition. Therefore, pattern <VB><IN|TO> can be matched with: any key-phrase that is followed by any kind of proposition. It is important to note that, some of the key-phrases don’t take

propositions in English language. Therefore, for defining a more general pattern which can cover the different mentioned formats of key-phrases we can define the following pattern:

<VB><IN|TO>*

5.3.1.3 Learning HasLeaf_Pattern Taxonomy

The purpose of creating HasLeaf_Pattern taxonomy is defining the determined patterns to the NLTK (Section 2.11) parser, and then asking parser to find all the sentences which have the same patterns. For this we use RegexpParser method from NLTK library which was elaborately described in Section 2.10. Therefore, at this step we feed this parser all the existing patterns in HasLeaf_Taxomony. From now on, RegexpParser knows the types of patterns that must be looked for.

5.3.2 Has_Whitelist

This step starts with a module called "Has_Whitelist". It is designed for checking if each paragraph contains any of the whitelist items (see Table 4.1) or not. This is a searching function for checking each paragraph context to see if it contains any of the whitelist items or not. The result of this search leads to two different conditions:

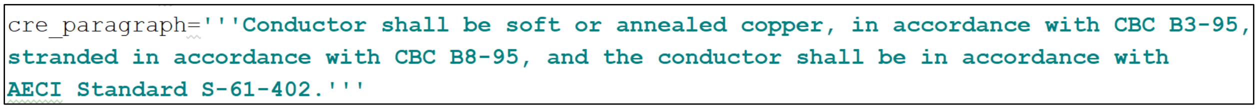
1. If the result does not contain whitelist items: In this case, it can have two different meanings in the algorithm:
    a. The algorithm should dismiss working on that paragraph because it is likely not a cross-reference expression.
    b. The whole paragraph might be a reference type NC (Section 4.2.2.3). This case should be done in future work.
2. If the result contains the whitelist items: In this case it means that the paragraph has passed the first validation step and is likely to be a cross-reference expression with type DC or IDC (refer to Sections 4.2.2.1 and 4.2.2.2, respectively). However, for recognizing the DC and IDC References types, having the whitelist items in a CRE does not cover all the conditions for making sure that paragraph is a cross-reference expression or not; it is just the first required condition. This is because, we may have a sentence that has one of the whitelist items but without the purpose of pointing any specific cross-references. Since in

the scope of our work required detection of DC reference types, and not the IDCs and NCs, this condition, and at this step the paragraph, is considered as a DC type. Working on IDC and NC types should be done in future work.

5.4 Pattern Recognition

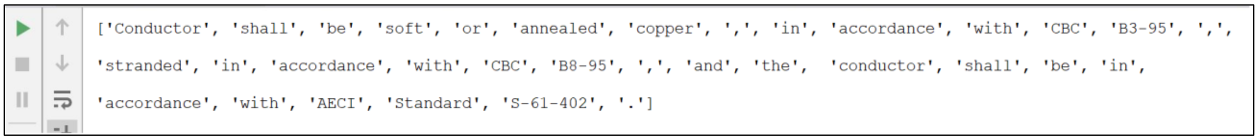
5.4.1 Tokenization

After passing the first validation step done in Has_Whitelist step (Section 5.3.2), the paragraph is sufficient enough to work on. So, we applied a tokenization (Section 2.9) function on that paragraph. Tokenization is the process of splitting a stream of a text into meaningful tokens (Kit, 1992). Here we tokenized each paragraph into separate words, because as we mentioned in Section 5.3.1.2, we need to specify the role of each token in a sentence. Therefore, in our work, tokenization is a preprocessing step, since in the following steps the grammatical or structural role of each token should be determined by POST (Section 2.7.1). For example, in Figure 5.9 we see a paragraph before tokenization and Figure 5.10 shows the result of tokenization applied on that paragraph.



```
cre_paragraph='''Conductor shall be soft or annealed copper, in accordance with CBC B3-95,
stranded in accordance with CBC B8-95, and the conductor shall be in accordance with
AECI Standard S-61-402.'''
```

Figure 5. 8: Paragraph before tokenization



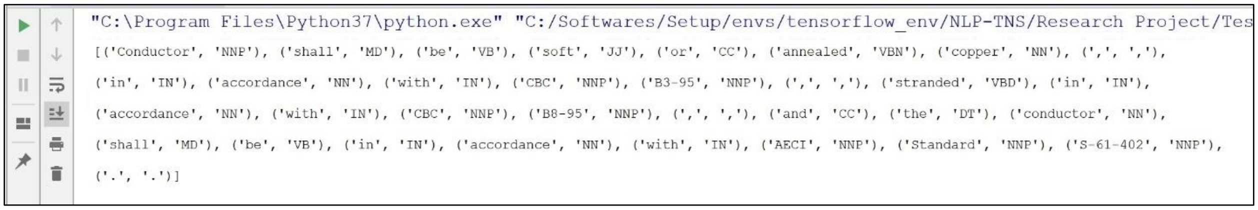
```
['Conductor', 'shall', 'be', 'soft', 'or', 'annealed', 'copper', ',', 'in', 'accordance', 'with', 'CBC', 'B3-95', ',',
'stranded', 'in', 'accordance', 'with', 'CBC', 'B8-95', ',', 'and', 'the',  'conductor', 'shall', 'be', 'in',
'accordance', 'with', 'AECI', 'Standard', 'S-61-402', '.']
```

Figure 5. 9: Paragraph after tokenization

5.4.2 Part of Speech Tagging

In Section 5.3.1.2, we described creating “HasLeaf_Pattern” taxonomy, which is designed for the RegexpParser (Section 2.10) to understand what patterns are supposed to look for (examples provided in Figures 5.6 and 5.7, respectively). Therefore, at this point we need a technique capable of specifying the grammatical role of the token of the paragraphs (example provided in Figure

5.8). Part of Speech Tagging is an NLP function from NLTK library which can do this task. Therefore, after applying POST, we will have an array of words which are tagged with a label that represent its grammatical role in the sentence. Figure 5.11 depicts the result after applying POST on top of the tokens.



```
"C:\Program Files\Python37\python.exe" "C:/Softwares/Setup/envs/tensorflow_env/NLP-TNS/Research Project/Tes
[('Conductor', 'NNP'), ('shall', 'MD'), ('be', 'VB'), ('soft', 'JJ'), ('or', 'CC'), ('annealed', 'VBN'), ('copper', 'NN'), (',', ','),
('in', 'IN'), ('accordance', 'NN'), ('with', 'IN'), ('CBC', 'NNP'), ('B3-95', 'NNP'), (',', ','), ('stranded', 'VBD'), ('in', 'IN'),
('accordance', 'NN'), ('with', 'IN'), ('CBC', 'NNP'), ('B8-95', 'NNP'), (',', ','), ('and', 'CC'), ('the', 'DT'), ('conductor', 'NN'),
('shall', 'MD'), ('be', 'VB'), ('in', 'IN'), ('accordance', 'NN'), ('with', 'IN'), ('AECI', 'NNP'), ('Standard', 'NNP'), ('S-61-402', 'NNP'),
('.', '.')]
```

Figure 5. 10: Part of Speech Tagging result over the tokens

5.4.3 Pattern Recognition Parser

As shown in Figure 5.11, when POST is applied on each paragraph to the number of tokens of the paragraphs, separate tuples are created. After performing POST each paragraph is ready to be gone under a pattern recognition parser. Here, we used RegexpParser (Section 2.10) from NLTK for doing this. RegexpParser parser is supposed to look for the similar patterns which are matched with the defined patterns shown in Figure 5.4. Therefore, if a paragraph has one or more patterns that are recognized by the RegexpParser, it must assign a “HasLeaf_Pattern” label to each identified pattern. We determined “HasLeaf_Pattern” name once we defined our taxonomy, shown on top of in Figure 5.4.

5.4.4 Generating Tree

The output of RegexpParser is always a tree which can have one or more than one leaf. Once the RegexpParser is applied on the tokenized paragraphs, if for example in that paragraph three patterns are recognized, then a tree will be shown by three leaves name “HasLeaf_Pattern”. For example, after tokenizing and applying POST (shown in Figure 5.10,5.11, respectively), we can apply RegexpParser. Figure 5.12 shows our sample paragraph that the key-points are highlighted with red color.

```
paragraph='''Conductor shall be soft or annealed copper, in accordance with CBC B3-95,
stranded in accordance with CBC B8-95, and the conductor shall be in accordance with
AECI Standard S-61-402.'''
```

Figure 5. 11: Cross-reference expression with three references before applying RegexpParser

Figure 5.13 shows that RegexpParser detected the three patterns in the paragraph.

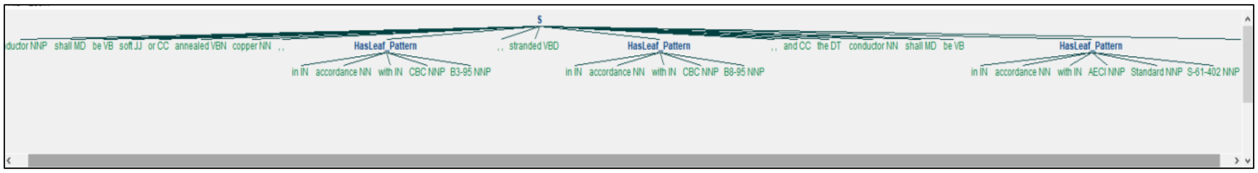
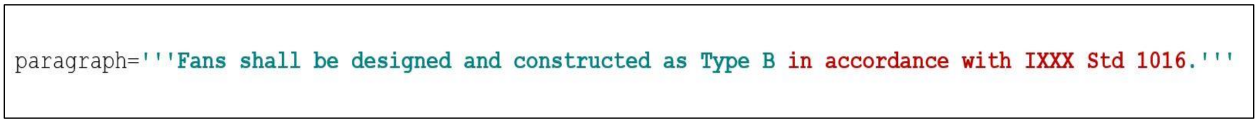


Figure 5. 12: A tree with three detected leaves after applying RegexpParser

Figure 5.14 shows a paragraph before applying the parser and Figure 5.15 shows that RegexpParser detects one pattern in the paragraph.



```
paragraph='''Fans shall be designed and constructed as Type B in accordance with IXXX Std 1016.'''
```

Figure 5. 13: Cross-reference expression with one reference before applying RegexpParser

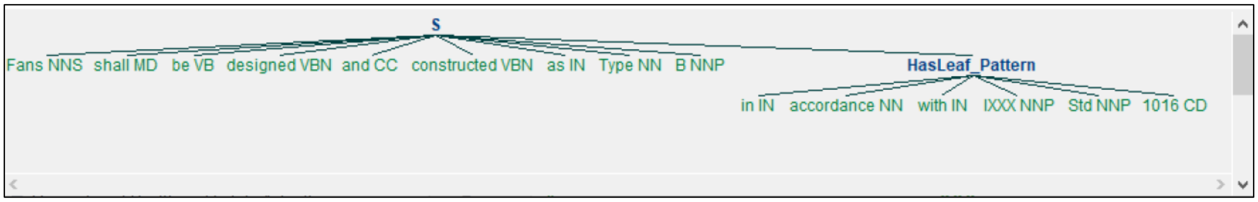


Figure 5. 14: A tree with one detected leaf after applying RegexpParser

5.4.5 Has_Whitelist

The previous step involved pattern recognition but in order to assure the validity of the identified leaves/patterns, we passed each leaf from a module named “is_a_valid_PatternReference”. We do this to search the whitelist items in the identified leaves. This is because, whitelist items are our cues that can partially assure us that the identified pattern is the true one. The output of this search may have the two following results:

5.4.5.1 Paragraph Dismissal

If “is_a_valid_PatternReference” module does not return the existence of one of whitelist items, in our current algorithm it means the identified pattern has the grammatical structure of a CRE, but likely doesn’t contain any reference. So, the algorithm dismisses that paragraph. This dismissed paragraph can be first tested under other conditions to see whether it has reference or not. This work can be done in future. At the moment we suppose that it probably doesn’t contain references.

5.4.5.2 Passing Validation

If “is_a_valid_PatternReference” module returns the existence of one of the whitelist items, it means the identified leaf passed the second validation step. So, all the context of the detected leaf should be entered the next step which is Reference Identifier.

5.5 Reference Identifier

The progress of the algorithm enters this step if the identified leaves pass the second validation from Section 5.4.5.2. At this step, we are facing one or more than one leaves. It is important to note that each leaf contains two parts:

1) Whitelist item: Since at this step we are very close to finding the cross-references, and we used whitelist items just as the indicators which can help us for approaching the references, whitelist items are now extra information and are no longer needed. Therefore, we can consider them as noise which should be dismissed from the rest of the process.

2) Probably the cross-reference: Normally at this step, in this part of the result we should have one or more than one reference. In order to make sure that this remaining part is a reference or not, all the value of this part should be passed to a module called “is_valid_reference”. This checks the “Has_APA_Properties” function which is described in the next step.

5.5.1 Has_APA_Properties

“Has_APA_Properties” is a module that we designed for checking the value of each identified leaf with the standard properties of APA in-text citations (Section 4.2.3) that are proposed by American Psychological Association. We defined these standards in our algorithm. These properties are:

1. References can consist of acronyms with all capital letters (example: In accordance with IXXX No. 85.)
2. References can consist of sequential words that each word starts with capital letters (example: in accordance with National Building Code)
3. References can consist of digits (example: in accordance with IXXX Std No. 85.)
4. References can consist of special characters like: “/, -, ., _”, etc. (example: given in NAC/ASC A23.1/A23.2-M).

Our reason for considering this step as the step of reference validation is because: sometimes in the identified reference, some extra tokens might be identified as the main components of the reference; for instance, word “the” has <DT> grammatical role (Determiner), and we have defined this grammatical role in the pattens of our “HasLeaf_Patterns” taxonomy (see the example in Figures 5.6, 5.7 and 5.8). Once the identified reference entails the token “the”, and then the reference enters to “Has_APA-Properties” module, the module removes the token “the” from the rest of the reference components. This is because, "the" doesn’t meet the APA properties; “the" doesn’t start with capital letter, also, it is not a word with all capital letters.

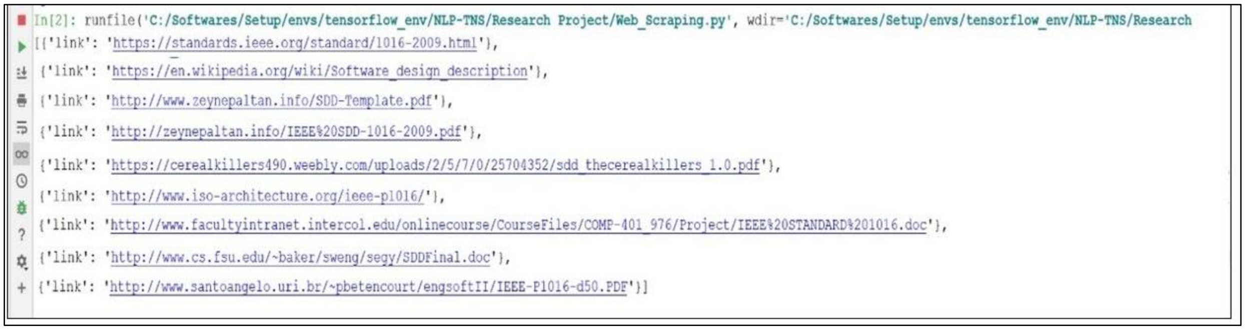
The other example that shows the functionality of “Has_APA-Properties” module is: for example, if a CRE contains the “CBC-100 CND-S-0930-01” reference, the pattern {<NNP>+} is matched with this reference. “NNP” refers as to any word which has the role of “noun” or “proper singular noun”. So, in this example, both “CBC-100” and “CND-S-0930-01” components are considered/labeled as “NNP” by POST. Therefore, <NNP>+, is matched with the above-mentioned reference. Here, the quantifier (+) after the <NNP> means, supporting unlimited number of sequential words which have “NNP” grammatical role in the sentence. Here the important note is: pattern {<NNP>+} is also matched with “Software Quality Assurance Plan” reference. Because all the components of this reference are also tagged as NNP by POST. That means, one {<NNP>+} pattern can support all references that have sequential NNP tokens. It is the positive side of our “HasLeaf_Pattern” taxonomy; because by only one pattern we can detect many numbers of references that have the same pattern. But it causes a problem in some situations; when parser considers the next <NNP> tokens as the reference component, where they are not actually the reference components. For example, suppose that there is an example regular word (written in small letters and have the NNP role) after this reference “CBC-100 CND-S-0930-01”. In such situation, the pattern identifier (RegExpress) cannot understand that the exampled word is not part of the reference. It considers that irrelevant word as one of the main components of the reference, because this word is again another NNP and it can be matched with <NNP>+ pattern. In such situations when the identified reference enters to “Has_APA-Properties” module, all the components of the identified reference must meet at least one of the properties of APA standards. Therefore, words that are not written with capital letters, are considered as extra identified tokens. So, such components would be removed from the identified reference. Finally, we would choose the correct reference components.

So, identified references are the input value for “Has_APA-Properties” module and this module works as a controller for removing/dismissing the extra identified reference components and choosing the standard identified components. Passing this step is the third and last validation step and then the identified reference will be considered as a real reference.

5.6 Web Scraping

5.6.1 Google Search API

Up to here, the process of the first level of reference identification over contractual document is done. At this step, we progress our searching process to identify related external references for each contractual requirement. It should be noted that we have a number of legal documents that are the actual resources of some of the identified references. At this step, our algorithm considers each identified reference as a keyword. If for the indented keyword, our algorithm doesn’t find any local resource among our available resources, then by using web scraping techniques (Section 2.12) the keyword is passed to google search engine API, in order to find a resource for the intended reference. For this, we use Request Python library which can send HTTP requests to the servers. Its functionality is described in Section 2.12.1. Here, we choose Google.com as the server, because it can supply the search results very quickly. At this step, each keyword (references which were identified in the first step of reference identification) should be received by Google search API. In this function not only we post the keyword, but we also determine the number of results that google should return. We set 10 results in our work. It means google should recommend and return 10 results for each reference. Although we always choose the first recommended result, we set it to 10 since we have a plan with the rest nine results which would be explained in Future Work, Section 10.2.2. Figure 5.16 Shows that the algorithm has passed the reference (the identified reference (IXXX Std 1016) in Figure 5.15) to google search engine and google returns 10 results.



```
In[2]: runfile('C:/Softwares/Setup/envs/tensorflow_env/NLP-TNS/Research Project/Web_Scraping.py', wdir='C:/Softwares/Setup/envs/tensorflow_env/NLP-TNS/Research
[{'link': 'https://standards.ieee.org/standard/1016-2009.html'},
 {'link': 'https://en.wikipedia.org/wiki/Software_design_description'},
 {'link': 'http://www.zeynepaltan.info/SDD-Template.pdf'},
 {'link': 'http://zeynepaltan.info/IEEE%20SDD-1016-2009.pdf'},
 {'link': 'https://cerealkillers490.weebly.com/uploads/2/5/7/0/25704352/sdd_thecerealkillers_1.0.pdf'},
 {'link': 'http://www.iso-architecture.org/ieee-p1016/'},
 {'link': 'http://www.facultyintranet.intercol.edu/onlinecourse/CourseFiles/COMP-401_976/Project/IEEE%20STANDARD%201016.doc'},
 {'link': 'http://www.cs.fsu.edu/~baker/sweng/segy/SDDFinal.doc'},
 {'link': 'http://www.santoangelo.uri.br/~pbetencourt/engsoftII/IEEE-P1016-d50.PDF'}]
```

Figure 5. 15: Google recommendation URLs for the identified reference; returned by code

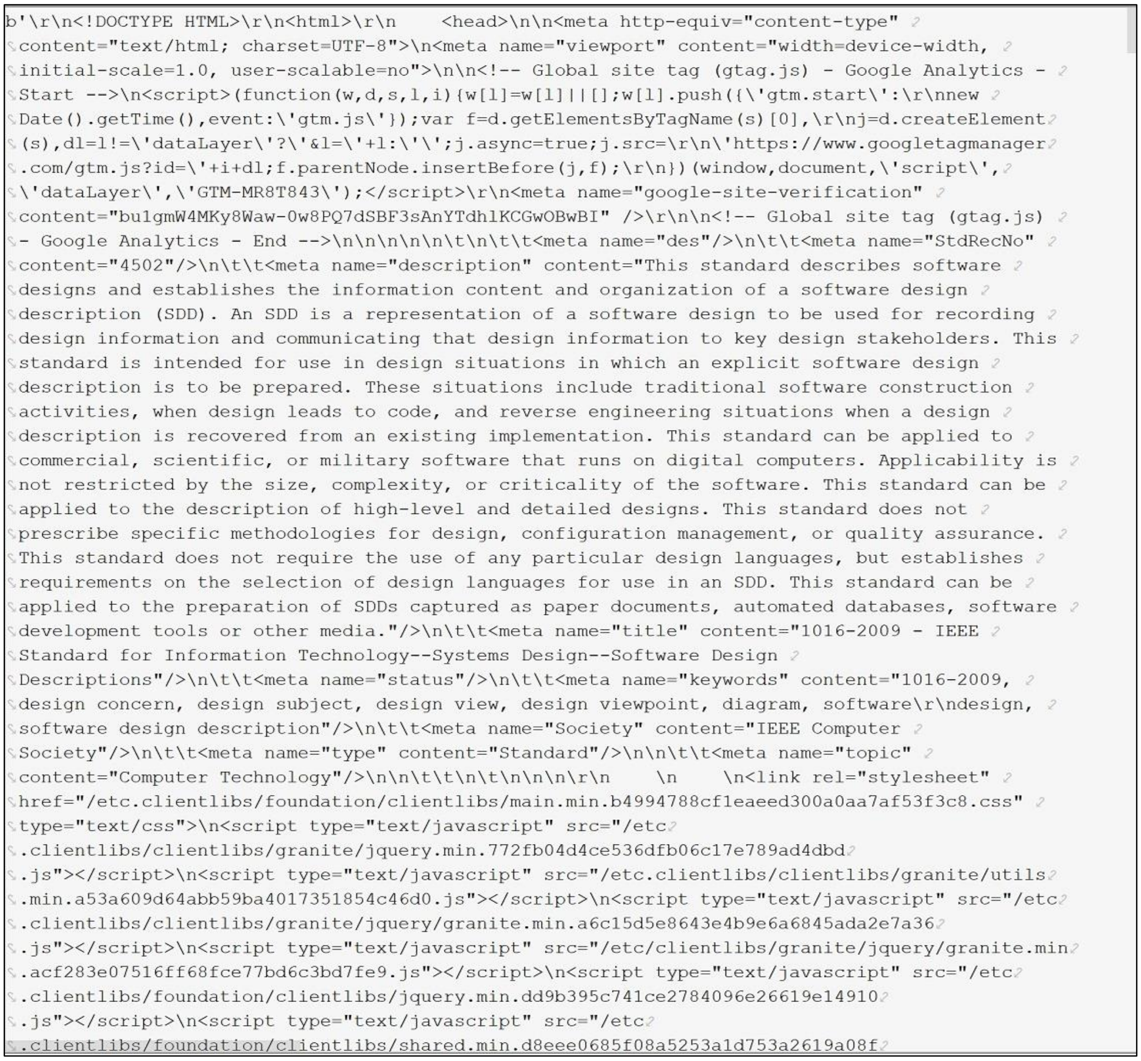
5.6.2 Choosing URL

From the 10 recommended results, we always chose the first one in our work and again by using the Request methods we open that page and read the HTML context. Here we optimistically suppose that most of the time, google is recommending the first result as the best and the closest resource to the keyword. Although we have an idea about how we can choose the best recommended URL among these 10 results, we discuss this in the future work section.

5.6.3 Extracting HTML Context

Since our mission in this thesis is identifying the textual references, this step is needed to first parse the HTML page context and then extract all the HTML tags which are specifically used for supporting and representing texts. For this, we get help from BeautifulSoup Python library which is capable of parsing data which has been already extracted from HTML documents through Requests library. The functionality of this library is described earlier in Section 2.11.2. After parsing this HTML context, we started finding such HTML tags as <a>, <p>, <h1><h2><h3>,

<span> which are respectively for representing URLS, paragraphs, titles and descriptions. So, the result of this step of HTML extraction is the whole textual context embedded in the mentioned tags. Therefore, at this step, again we have structured textual data which may contain references. So, we can apply all our algorithm processes on this new textual data; from step paragraph splitting (Section 5.2.3) to the current step of finding the references. This reference identification process can be repeated for each identified reference until we don't find any new reference for each reference. This is an idea that can be done in future work. In the work we defined a threshold with the value of 2, which determines the number of progressing steps of this process. This value can be changed to any other numbers. Figure 5.17 shows a view from the HTML context of the first URL shown in Figure 5.16.



```
b'\r\n<!DOCTYPE HTML>\r\n<html>\r\n    <head>\n\n<meta http-equiv="content-type"
content="text/html; charset=UTF-8">\n<meta name="viewport" content="width=device-width,
initial-scale=1.0, user-scalable=no">\n\n<!-- Global site tag (gtag.js) - Google Analytics -
Start -->\n<script>(function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({\'gtm.start\':\r\nnew
Date().getTime(),event:\'gtm.js\'});var f=d.getElementsByTagName(s)[0],\r\nj=d.createElement
(s),dl=l!=\'dataLayer\'?\'&l=\'+l:\'\';j.async=true;j.src=\r\n\'https://www.googletagmanager
.com/gtm.js?id=\'+i+dl;f.parentNode.insertBefore(j,f);\r\n})(window,document,\'script\',
\'dataLayer\',\'GTM-MR8T843\');</script>\r\n<meta name="google-site-verification"
content="bu1gmW4MKy8Waw-0w8PQ7dSBF3sAnYTdhlKCGwOBwBI" />\r\n\n<!-- Global site tag (gtag.js)
- Google Analytics - End -->\n\n\n\n\n\t\n\t\t<meta name="des"/>\n\t\t<meta name="StdRecNo"
content="4502"/>\n\t\t<meta name="description" content="This standard describes software
designs and establishes the information content and organization of a software design
description (SDD). An SDD is a representation of a software design to be used for recording
design information and communicating that design information to key design stakeholders. This
standard is intended for use in design situations in which an explicit software design
description is to be prepared. These situations include traditional software construction
activities, when design leads to code, and reverse engineering situations when a design
description is recovered from an existing implementation. This standard can be applied to
commercial, scientific, or military software that runs on digital computers. Applicability is
not restricted by the size, complexity, or criticality of the software. This standard can be
applied to the description of high-level and detailed designs. This standard does not
prescribe specific methodologies for design, configuration management, or quality assurance.
This standard does not require the use of any particular design languages, but establishes
requirements on the selection of design languages for use in an SDD. This standard can be
applied to the preparation of SDDs captured as paper documents, automated databases, software
development tools or other media."/>\n\t\t<meta name="title" content="1016-2009 - IEEE
Standard for Information Technology--Systems Design--Software Design
Descriptions"/>\n\t\t<meta name="status"/>\n\t\t<meta name="keywords" content="1016-2009,
design concern, design subject, design view, design viewpoint, diagram, software\r\ndesign,
software design description"/>\n\t\t<meta name="Society" content="IEEE Computer
Society"/>\n\t\t<meta name="type" content="Standard"/>\n\n\t\t<meta name="topic"
content="Computer Technology"/>\n\n\t\t\n\t\n\n\n\r\n    \n    \n<link rel="stylesheet"
href="/etc.clientlibs/foundation/clientlibs/main.min.b4994788cf1eaeed300a0aa7af53f3c8.css"
type="text/css">\n<script type="text/javascript" src="/etc
.clientlibs/clientlibs/granite/jquery.min.772fb04d4ce536dfb06c17e789ad4dbd
.js"></script>\n<script type="text/javascript" src="/etc.clientlibs/clientlibs/granite/utils
.min.a53a609d64abb59ba4017351854c46d0.js"></script>\n<script type="text/javascript" src="/etc
.clientlibs/clientlibs/granite/jquery/granite.min.a6c15d5e8643e4b9e6a6845ada2e7a36
.js"></script>\n<script type="text/javascript" src="/etc/clientlibs/granite/jquery/granite.min
.acf283e07516ff68fce77bd6c3bd7fe9.js"></script>\n<script type="text/javascript" src="/etc
.clientlibs/foundation/clientlibs/jquery.min.dd9b395c741ce2784096e26619e14910
.js"></script>\n<script type="text/javascript" src="/etc
.clientlibs/foundation/clientlibs/shared.min.d8eee0685f08a5253a1d753a2619a08f
```

Figure 5. 16: A view from HTML context of a third-party resource

5.7 Visualization

This step is for representing all of the identified references which are gathered through all the steps. As we discussed in the previous steps, at the first step we identified references from contractual requirement documents. Once the references of each contractual requirement are identified, then by using Texttable Python library we created a table for showing the identified references. After that, we should choose a resource for each identified reference. If the tool cannot find any local resource related to the reference, the intended reference is passed to google search engine. Google search API finds and opens a resource for that reference. Once we open the related HTML page, we again start reference identification in that page. This process should be repeated based on the value of threshold defined in our loop. Therefore, at this step we created a table with generating dynamic columns for each level of identification. For example, if our identification threshold is equal to 2, there would be a table with 2 columns of identified references. More comprehensive description is provided in chapter 6.

Chapter 6

6 Output of Executing the Solution Algorithm

This section is for representing all the final output of executing the proposed solution. It consists of two sub-sections including detailed output produced by the algorithm and statistical results signifying the quality of the proposed solution. In the first sub-section, there are two views from the output of our algorithm. The first view (Table 6.1) depicts the identified external references from the primary contractual document, and the second view (Table 6.2) depicts the relation between the first identified level of references with the second level identified from secondary resources including from the world wide web. The second sub-section of this chapter describes the summary of the cross-references gathered from one contractual document with 683 pages and 10345 paragraphs. It would also report a result form the number of the identified references of the first 100 secondary resources.

6.1 Detailed view produced by algorithm

The proposed algorithm produces two levels of references. In the first level, you can see a view from identified external references (see Table 6.1) from one contractual requirement document with 683 pages and 10345 paragraphs. In the second level, there is a list of secondary (or indirect) identified references (see Table 6.2) for each external reference found previously in the first level. Our final result is represented in Table 6.2. This figure depicts a table with three columns. The first column is named “Contractual Requirement Num”. It is for showing the number of each contractual requirement. Therefore, each row of this table is supposed to show a list of detected external references for each contractual requirement. In the second column of this table, which is named “Reference Level 1”, all the identified references from contractual requirements are listed. The third column which is named “Reference Level 2” shows all the identified secondary references. As shown in Table 6.2, the level of analysis is done up to two levels; however, in theory, there can be more indirect levels, as described in Sections 4.1. However, note that our algorithm doesn't produce the values of Tables 6.1 and 6.2 separately. It generates both results in one execution, shown in Table 6.2.

6.1.1 Identified external references from one contractual requirement document

In Table 6.1, the first level of identification is shown. Here, you can see all the identified external references from one contractual document with 683 pages and 10345 paragraphs. Our tool returns this step of the result in only 16 seconds and 72 milliseconds (0:0:16:72). Column "Contractual Requirement Num" is the representative of each contractual requirement. DC external references are detected by the algorithm and then are visualized in the column of "Reference Level 1". The values of this column are itemized by sequential numbers. For example, if one reference is detected for contractual requirement 2.5.2, only one reference with number one is shown in this column. If three references are detected for a contractual requirement, three references which are numbered from one to three are listed in this column. In addition, in the bottom of this column/per row you can see the total number of identified references row by row. In this section, we just show 10 identified references out of the total 667 identified results.

Table 6. 1: A view from output of our tool for identifying external references from one contractual requirement

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**

6.1.2 Identified external references from secondary resources

As described in Chapter 5, when the first level of references has been identified, each reference is then considered as a keyword and the exploration will go one step further. At this point, in order to find a resource for the intended keyword, first the algorithm searches for a local file/resource from the database. If it finds it, that resource would be chosen for the second level of reference identification process. If the algorithm doesn’t find any local file/resource, the keyword would be posted to Google search engine API. Reference identification process will then apply over the content of the web page recommended by the search engine. For showing this process, we create a table that can work as a map for representing the relationships between each contractual requirement and its related references. Table 6.2 is designed to show this map and the relations between its elements. In table 6.2, column “Contractual Requirement Num” is for showing the number of each contractual requirement. Column “Reference Level 1” shows the identified references for each contractual requirement. Column “Reference Level 2” is for representing all the identified references of the second step of reference identification. In this step, if the algorithm doesn’t find any reference, column “Reference Level 2” will be filled out with dashes. Therefore, this table is considered as a map and at each row contractual requirement number points to its related references located in column “Reference Level 1”. Also, the first reference in column “Reference Level 1” points to the references located in column “Reference Level 2”.

Table 6. 2: Final output of our tool for identifying external references from one contractual requirement and extracting references from the secondary resources

Contractual Requirement Num	Reference Level 1	Reference Level 2
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	1- Clarification Labor Code 1500 2-Documet 00700.8.90 3- SPPO 380
CR 4.1.1	1-Document 00500 SPECIAL CONDITONS 2-CAN/ACA-A6/A362 Portland Type	1- ICA T1.1/T1.1R 2- CCRRN National Building 3- ARS 41-GP-34M Type IV
CR 2.6.1.4	1- CBC 41-GP-34M Type III	1- Food Safety Quality Act. 2- CBC/ASC-G30.18 3- CBC/ASC-G30.19 4- SPPO 790
CR 2.7.1	1- National Building Code 2- XAXM Manual Standard Practice	1- SCA Standard Z91 2- ISNA/EMSA B16.26

6.2 Statistical results

In this section, we first summarize the result of our approach over one contractual document with 683 pages in Table 6.3. Then, we will provide a report form the number of identified references of the first 100 secondary resources.

Table 6. 3: Summary of our study over external cross-references in one contractual requirement document

Important elements	**values**	**Percentage**	**Description**
Number of pages:	683	-	N/A
Number of paragraphs:	10345	-	N/A
Total number of all types of existed external references (calculated manually):	802	-	It includes all the following references types: DC references, IDC references, NC references, and References don't meet APA properties (Sections 4.2.2.1, 4.2.2.2, 4.2.2.3 and 4.2.3, respectively)
Total number of DC references (calculated manually):	667	83% of all the external references are DC references	This is total number of references that were supposed to be identified automatically in this thesis (elaborates in Section 4.2.2.1).
Number of IDC external references:	15	1.8% of all the external references are IDC references	Identifying IDC external references are supposed to be done in future work (elaborates in Sections 4.2.2.2 and 10.2.1).
Number of NC external References:	112	13.96% of all the external references are NC references	Identifying NC external references are supposed to be done in future work (elaborates in Sections 4.2.2.3 and 10.2.1).
Number of references meet APA properties:	8	0.99% of all the external references don't meet	Figure 4.6 shows an example of external reference doesn't meet APA properties.

		APA properties	
Number of detected DC external references through the automated tool	641	96% of all the existed DC references are detected by our automated tool	N/A

Other detailed information including precision, recall and F-measure is provided in Quality of algorithm chapter (Chapter 7).

Additionally, it is important to note that, our tool succeeded to find 2464 references from different secondary resources, for the first 130 references (20% of all the identified references) which are identified from contractual requirement document.

Chapter 7

7 Quality of algorithm

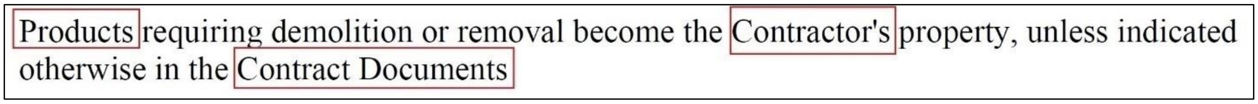
In this chapter, first we explain about confusion matrix (Section 2.14) which helped us to evaluate the quality of our approach. The chapter is then continued with the required values of confusion matrix in our work including: Observation (Section 2.14.1), Positive and Negative Observations (Section 2.14.2), True Positive (Section 2.14.3), True Negative (Section 2.14.4), False Positive (Section 2.14.5), False Negative (Section 2.14.6), Recall (Section 2.14.7), Precision (Section 2.14.8) and F-Measure (Section 2.14.9). Finally, this chapter would be terminated with providing a summary of the calculated values.

7.1 Confusion Matrix

In order to evaluate the accuracy of our results, we use confusion matrix which is a table with specific elements and used to describe the performance of a classification model (Section 2.13). It is important to note that, here by using confusion matrix we are testing the result of our proposed approach. The dataset (contractual document) that we are working on consists of raw textual data. This data is not classified/labeled and we don’t have train set and test set (Science, 2013). Therefore, we had to do the classification manually (further description is provided in Section 7.1.2). The values of the main elements of confusion matrix in our work are provided in the following:

7.1.1 Observation

In our work, a word or sequential of words with all capital letters or a word or sequential of words starting with capital letters (camel case format), is an observation. For example, Figure 7.1 shows two words “Products”, “Contractor” that start with capital letters. Also, “Contract Documents” shows sequential words that each word starts with capital letter (camel case format). All of these three samples have the conditions of our observations.



Products requiring demolition or removal become the Contractor's property, unless indicated otherwise in the Contract Documents

Figure 7. 1: Example (1) of observation

The other examples of our observations are shown in Figure 7.2. Here, two words “RSS”,” USRC” that are shown with all capital letters have the condition of our observation.

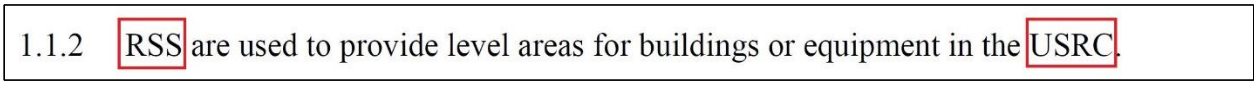

1.1.2 RSS are used to provide level areas for buildings or equipment in the USRC.

Figure 7. 2: Example (2) of observation

The reason for choosing such conditions as the conditions of our observation is because, we have the same conditions for passing the final validation step to accept a word or sequence of words as a reference (elaborated in Section 4.2.3).

In one contractual requirement document with 683 pages, we found 17,025 observations (calculated manually).

7.1.2 Positive and Negative Observations

In our work, observations that are references are considered as positive observations, and observations that are not references are considered as negative observations (Section 2.14.2).

Since our dataset (contractual document) consists of raw textual data, and positive and negative observations are not labeled, we had to manually determine the essence of the positive and negative observations. Therefore, to calculate the accuracy of our work, we manually labeled our sample data and tested our algorithm on 3,405 observations. This number is equal to 20% of the total observations; the rate/size which is recommended for validation of test sets in data science problems (Science, 2013).

7.1.3 True Positive (TP)

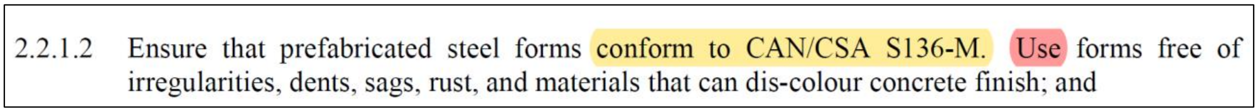
In our work, TP (Section 2.14.3) is for observations which are references, and our algorithm has recognized them correctly. TP is equal to 258 in our work.

7.1.4 True Negative (TN)

In our work, TN (Section 2.14.4) is for observations which are not references, and our algorithm correctly hasn’t recognized them as references. TN is equal to 3,104 in our work.

7.1.5 False Positive (FP)

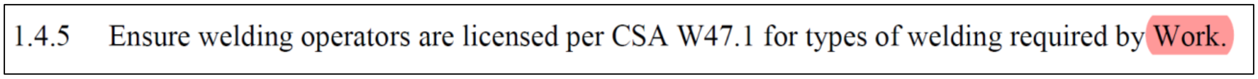
In our work, FP (Section 2.14.5) is for observations which are not references, but our algorithm has recognized them as references. FP is equal to 2 in our work. Figure 7.3, depicts an example from one of the FPs. It shows that the reference which is detected by the algorithm is “CAN/CSA S136-M. Use”. However, the word “Use” is extra and it is not part of the reference. We believe adding sentence splitting in our future-work algorithm (after paragraph splitting) should eliminate this problem.



2.2.1.2 Ensure that prefabricated steel forms conform to CAN/CSA S136-M. Use forms free of irregularities, dents, sags, rust, and materials that can dis-colour concrete finish; and

Figure 7. 3: FP Example 1

Figure 7.4 shows the other false detection of our algorithm which choose the word “Work” as a reference, however it is just a simple word which is represented in capital.



1.4.5 Ensure welding operators are licensed per CSA W47.1 for types of welding required by Work.

Figure 7. 4: FP Example 2

7.1.6 False Negative (FN)

In our work, FN (Section 2.14.6) is for observations which are references, but our algorithm hasn’t recognized them as references. FN is equal to 41 in our work. It is important to note that, only a small portion of these FNs relates to the references that were supposed to be identified by our current algorithm. In other words, these 41 FNs belong to five different categories of external references, and only the first category was supposed to be covered in the current work. Table 7.1 shows a summary of different FNs in our test data.

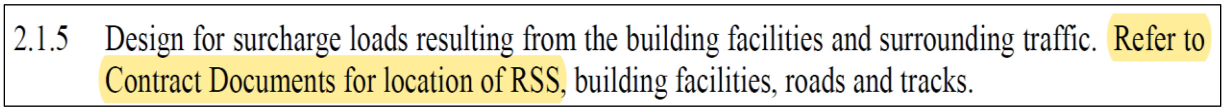
Table 7. 1: Summary of all types of FNs

Total number of FNs	Number of FNs for DC references (current work)	Number of FNs for DC references don’t use reporting phrase (Future work)	Number of FNs for NC references (Future work)	Number of FNs for IDC References (Future work)	Number of FNs for DC references don’t meet APA properties
41(13%)	3 (0.95%)	7 (2.21%)	25 (7.92.%)	3 (0.95%)	3 (0.95%)

Following items provide more information about the details of our FNs:

1. **References which are in the category of Direct Cue (DC); they have reporting phrases, but the reference part is not recognized by the algorithm**:

For example, in Figure 7.5, “Contract Documents” is an external reference and the verb “Refer to” is a reporting phrase, which comes before the reference. This verb is defined in Whitelist. The required English grammatical pattern for supporting this CRE is also defined in HasLeaf-Pattern taxonomy. But it is not detected by the algorithm! Form all 41 FNs, only 3 (0.95%) of them are categorized in this group and all of them are using “Refer to” as the key-phrase for referring to the reference.

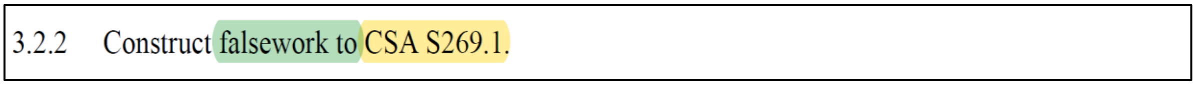


2.1.5 Design for surcharge loads resulting from the building facilities and surrounding traffic. Refer to Contract Documents for location of RSS, building facilities, roads and tracks.

Figure 7. 5: FN Example – DC reference model

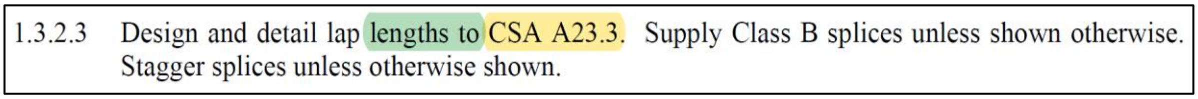
2. **References which are in the category of Direct Cue (DC), but the verb which is used before the references is not a reporting phrase:**

Figures 7.6, 7.7, 7.8, 7.9 and 7.10 show these types of references. All the following five figures depict five phrases, including: “falsework to”, “length to”, “joint to”, “element to”, “license”, which are coming exactly before the references but they don’t lie in the category of reporting phrases. So, the reason that our algorithm couldn’t detect them is because, these phrases are not defined in Whitelist. Form all 41 FNs, 7 (2.21%) of them are categorized in this group.



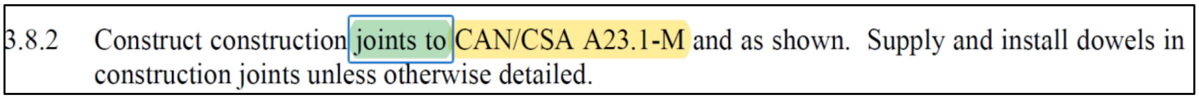
3.2.2 Construct falsework to CSA S269.1.

Figure 7. 6: FN Example 1 – DC reference model without reporting phrase



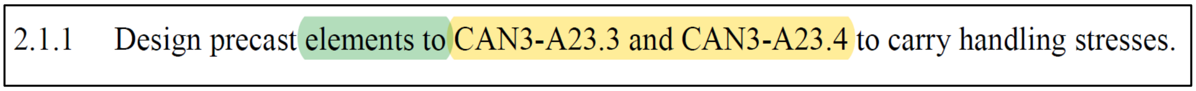
1.3.2.3 Design and detail lap lengths to CSA A23.3. Supply Class B splices unless shown otherwise. Stagger splices unless otherwise shown.

Figure 7. 7: FN Example 2 – DC reference model without reporting phrase



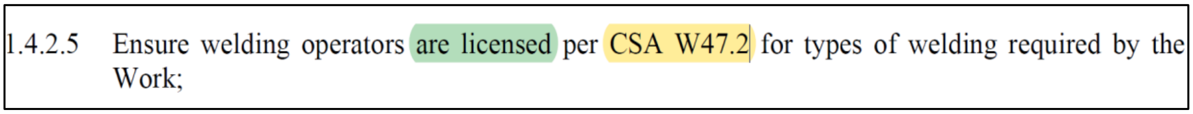
3.8.2 Construct construction joints to CAN/CSA A23.1-M and as shown. Supply and install dowels in construction joints unless otherwise detailed.

Figure 7. 8: FN Example 3 – DC reference model without reporting phrase



2.1.1 Design precast elements to CAN3-A23.3 and CAN3-A23.4 to carry handling stresses.

Figure 7. 9: FN Example 4 – DC reference model without reporting phrase



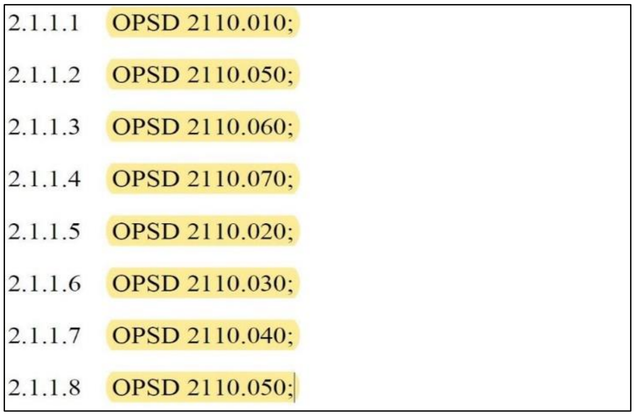
1.4.2.5 Ensure welding operators are licensed per CSA W47.2 for types of welding required by the Work;

Figure 7. 10: FN Example 5 – DC reference model without reporting phrase

It is important to note that, sentence type in the examples provided in Figures 7.6, 7.7, 7.8 and 7.9 is imperative and the mood of the verb is imperative. This information may help to identify these references as well in our future work.

3. **References which are in the category of No Cue (NC):**

Figure 7.11, depicts samples of these references which are in our test dataset, and our algorithm could not detect them. As it is shown, these types of references represent as a title, without any reporting phrase or any other indicator before or after them. From all 41 FNs, 25 (7.92.%) of them are categorized in this group. As it was mentioned earlier, recognizing NC references was not in the scope of this thesis and can be done in future work.



2.1.1.1 OPSD 2110.010;

2.1.1.2 OPSD 2110.050;

2.1.1.3 OPSD 2110.060;

2.1.1.4 OPSD 2110.070;

2.1.1.5 OPSD 2110.020;

2.1.1.6 OPSD 2110.030;

2.1.1.7 OPSD 2110.040;

2.1.1.8 OPSD 2110.050;

Figure 7. 11: FN Example – NC reference model

4. **References which are in the category of InDirect Cue (IDC):**

Form all 41 FNs, 3 (0.95%) of them are categorized in this group. Again, identifying this group was not in the scope of this thesis. Further information regarding this group is provided in Section 4.2.2.2.

5. **External references which don't meet APA properties:**

There are other types of external references which are shown in small letters. As we described earlier (Section 4.2.3), we suppose that all the references must meet APA properties. So, our algorithm can't recognize references which are written in small letters. We found 3 (0.95%) numbers of this type of reference in our test data.

7.1.7 Accuracy

In confusion matrix, the following formula is used for calculating the accuracy.

Accuracy= (TP+TN) / (TP+TN+FP+FN) (GeeksforGeeks, 2013)

Therefore, when we fill out the accuracy formula with all the above values, the accuracy would be:

Accuracy= (258+3104) / (258+3104 +2+41)

Accuracy= 99%

7.1.8 Recall

Recall (Section 2.13.7) is another element which can be calculated from the confusion matrix elements. The following formula is used for calculation the recall:

Recall= TP / (TP+FN) (Mastery, 2019)

Recall= 258 / (258+ 41)

Recall= 0.86

7.1.9 Precision

In order to calculate the value of precision (Section 2.13.8) we should use the following formula:

Precision = TP / (TP+FP) (Mastery, 2019)

Precision = 258 / (258+ 2)

Precision = 0.99

7.1.10 F-Measure

F-Measure (Section 2.13.9) is the other factor which is calculated the values of precision and recall. In order to calculate the value of F-Measure we should use the following formula:

F-Measure= 2* Recall*Precision / (Recall + Precision) (Mastery, 2019)

F-Measure= 2*0.86*0.99 / (0.86 + 0.99)

F-Measure= 0.92

7.1.11 Summary of rates computed from confusion matrix

Table 7.2 provides the summary of important values which are computed from the concepts and formulas of confusion matrix.

Table 7. 2: Summary of values computed from confusion matrix

Element title	**Value**
Total number of observations	17,025
Number of observations in test dataset	3,405
TP (True Positive) observations	258
TN (True Negative) observations	3,104
FP (False Positive) observations	2
FN (False Negative) observations	41
Accuracy	99%
Recall	86%
Precision	99%
F-Measure	92%

Chapter 8

8 Comparison with Related Work

In this chapter, we explain how our work compares with related work. In Section 8.1, we summarize the differences between our work and related work. Table 8.1 captures these differences in a succinct manner. The rest of the chapter then gives more details about each of these differences: Text-Type, Type of cross-references, Phrase-Types, Source of Phrases, CRE-format, Determining the scope of the references, Cross-reference validation, Access to third party resources, Document structure, and Overall approach.

8.1 Summary of comparison with related work

In Table 8.1, the "Subject" column shows the titles of the differences between the previous approaches with our approach. In the "Related Work" column, a summary from each previous work for each "Subject" is provided. In the "Our Approach" column, our work for each "Subject" is summarized. A cell in this table that is indicated as "Non-Existent" implies that the corresponding "Subject" is not dealt with in any related works and, hence, is unique to our approach.

Table 8. 1: Summary of comparison with related work

Row #	Subject	**Related Work**	**Our Approach**
1	Text-Type	• (Tran *et al.*, 2014): Japanese national pension *law* document • (Adedjouma *et al.*, 2014): Luxembourgish Income Tax *law* document • (Sannier *et al.*, 2016): Luxembourgish *law* documents	Project contract (lists high-level requirements) (see Sections 2.1.1 and 4.1)
2	Type of cross-references	• (Tran *et al.*, 2014) and (Adedjouma *et al.*, 2014) : Cross-references *internal* to the document (see Section 2.4.2)	Cross-references *external* to the contract (to regulatory documents, standards,

			and the web) (see Section 2.4.2)
3	Phrase-Types	(Sannier *et al*., 2016): Classifying phrases into the 11 groups provided in Table 3.2 (Compliance, Constraint, Definition, Delegation, Exception, Refinement, General Amendment, Amendment by Addition, Amendment by Deletion, Amendment by Resignation, Amendment by Replacement)	Using a huge diversity of reporting phrases (see Table 4.7)
4	Source of Phrases	(Sannier *et al*., 2016): Two legal documents	Established English literature (see Table 4.7)
5	Cross-Reference Expression (CRE) format	(Adedjouma *et al*., 2014): CREs are categorized as explicit and implicit (which are all about how cross-references are written in a CRE) (see Table 3.1)	CREs are categorized into Direct Cue, InDirect Cue and No Cue (which are all about the position/status of reporting phrases in a CRE) (see Sections 4.2.2.1, 4.2.2.2 and 4.2.2.3)
6	Determining the boundaries of the references	• (Tran *et al.*, 2014): Defining different notations (B_M, I_M, E_M, O) for tagging different parts of the references • (Adedjouma *et al*., 2014): Converting a non-markup format into a markup format (XML) and then creating patterns based on explicit and implicit CREs (see Table 3.1)	Creating “Has_Leaf_Pattern” taxonomy (see Figure 5.4) with Part of Speech Tagging
7	Validation of Cross-references	Non-Existent	The identified references must satisfy APA properties (see Sections 4.2.3 and 5.5.1)

8	Access to third party resources	Non-Existent	Using web scraping techniques (see Section 5.6)
9	Document structure	(Tran *et al.*, 2014) and (Adedjouma *et al*., 2014) and (Sannier *et al*., 2016): Legal documents consist of book, title, chapter, section, sub-section, paragraph (see Section 2.1.3).	Project Contract documents consist of sections, subsections, sequential paragraphs (sections and sub-sections are not important for identifying external references) (see Sections 2.1.1 and 4.1)
10	Overall approach	• (Tran *et al*., 2014): Mention detection and mention splitting, Mention Classification, Position Recognition, Antecedent candidate extraction, Antecedent Determination • (Adedjouma *et al*., 2014): Define schema for structure of legal text, transform into markup text, Resolve cross-references, Visualization & Analysis	Structuring Data, pattern Setting, pattern recognition, Reference identification, Web scraping, Visualization (see Figure 5.1)

8.2 Text-Type

With reference to Table 8.1, Text-Type refers to the type of documents processed. Whereas our study investigates a contract document (Sections 2.1.1 and 4.1); the related works (Tran *et al.*, 2014) ,(Adedjouma *et al*., 2014) and (Sannier *et al.*, 2016) investigate legal documents (Section 2.1.3).

8.3 Type of cross-references

The type of cross-references is one of the most noteworthy differences between our work with the previous work. (Tran *et al*., 2014) and (Adedjouma *et al*., 2014) worked only on extracting cross-references internal to a given document and didn't work on extracting external ones referring to the existing external documents (Maxwell *et al.*, 2012). For dealing with the internal cross-references all the elements of document including headers, footers, titles, chapters, sections, sub-sections, paragraphs and idents (more details is provided in (Adedjouma, et al, 2014)) may refer to each other. Therefore, any automated solution approach for detecting internal references needs to determine the essence of such entities. For example, the automated approach should identify the position of the referred "paragraph" or the referred "section" in the current document. Therefore, for identifying internal references "paragraph", "section", "subsection" and so forth are static and certain terms which must be considered in the solution approach. In contrast, in external references, references refer to the documents existing out of the current document. Therefore, unlike internal references, external references don't consist of static and certain terms including "paragraph", "section", "subsection", etc. Therefore, for identifying external references we are facing various unpredicted combinations of alphabets, numbers, special characters and words (see Section 4.2.1). Therefore, the outcome of identifying internal references is not comparable with the outcome of identifying external references.

8.4 Phrase-Types

As elaborated in Chapter 3, (Sannier *et al*., 2016) created a taxonomy (see Table 3.2) of 11 phrase-categorizations including: Compliance, Constraint, Definition, Delegation, Exception, Refinement, General Amendment, Amendment by Addition, Amendment by Deletion, Amendment by Resignation, and Amendment by Replacement. In contrast, we created a list of "Reporting Phrases" (see Table 4.7). These phrases refer to the relevant target information. This source of information can be references, important ideas, discoveries or writings of experts in a special field of study (Sydney, 2019). Therefore, such phrases can help us identify to the references and they don't limit our algorithm to the phrase-categorization used in (Sannier *et al*., 2016). However, reporting phrases cannot cover all the phrases that appear before the references. For example, "Falsework to", "Lengths to", "joints to" are phrases that appear before references, but they are not reporting phrases (see Figures 7.6, 7.7, 7.8, 7.9 and 7.10)

8.5 Source of Phrases

In (Sannier *et al*., 2016), the authors identified their phrases (see Table 3.2) from two legislative documents. In contrast, we identified reporting phrases from different established English literature resources (AUSB, 2017)(Toronto, 2005) (Centre and Guide, 2014)(EAP Foundaton, 2019) (University, 2007).

8.6 CRE-format

As elaborated in Chapter 3, the authors of (Adedjouma *et al*., 2014), categorized CREs into two groups: explicit CREs and implicit CREs. This categorization is based on how cross-references are expressed in a CRE (see Table 3.1). In their work, explicit CREs consist of some predefined terms and patterns which compose part of a reference. For example: "article 102" is an explicit reference consisting of a predefined term ("article") and a number-pattern ("102"). On the other hand, if a CRE consist of "*following paragraph*", it means the current CRE is implicitly referring to a reference which is cited/located in the *following paragraph* of the CRE. This implicit CRE consist of two predefined terms ("*following*" and "*paragraph*") (see Table 3.1). In contrast, in our thesis we propose different classifications for CREs. It includes Direct Cue, InDirect Cue and No Cue (see Sections 4.2.1, 4.2.2 and 4.2.3). In this classification, our focus is on the permutations of specific key-phrases (see Table 4.7) which consider as the cues for identifying a reference. Sometimes these key-phrases appear prior to a reference (Direct Cue), e.g., "Designed and certified for 85 dBA maximum noise level when measured **in accordance with** IXXX No. 85.". Sometimes key-phrases appear after a part of a reference, and the remaining of the reference appears prior to the key-phrase (InDirect Cue), e.g., "Cooper E90 loading **in accordance with** AREAAA;". In this case we have the cue but it is presented after only a part of the reference. So, the rest of the reference is not identifiable in our current work. Sometimes CREs don't have any auxiliary cue. In such cases there is no key-phrase and all the CRE is a reference (No Cue), e.g., "DSPO 2110.050".

8.7 Determining the boundaries of the references

In (Tran *et al.*, 2014), a cross-reference is referred to as "mention". The starting and ending scope of a reference is determined by defining different notations for different parts of the reference. These notations include the following items:

- B_M (Begin of Mention): The first element of a mention is tagged with B_M
- I_M (Inner of Mention): The remaining elements of the mention are tagged with I_M
- E_M (End of Mention): The last element of the mention is tagged with E_M
- O (Others): All elements outside the mention are tagged with O

Applied approach for determining the scope of the reference is different in (Adedjouma *et al.*, 2014). They first generate a hierarchical structure for showing relation between different parts and converting a non-markup format into a markup format (XML). Then, based on the formats of explicit and implicit CREs (see Table 3.1), they create some predefined terms and patterns. For example: "art 106" is an explicit reference consisting of a predefined term ("art") and a number-pattern ("106"). Once these patterns are interpreted the boundaries of the references are determined.

In our thesis, by creating HasLeaf_Pattern taxonomy (see Figure 5.4), we use a new approach for finding the boundaries of the references. By using POS tagging (Section 2.7.1) we generate a taxonomy containing various grammatical structures. For example, if a CRE contains the reference "Electrical Code", then the pattern {<JJ><NNP>} is matched with the reference. "JJ" refers as to any word which has the role of "adjective". "NNP" refers as to any word which has the role of "noun" or "proper singular noun". By using this approach, the starting and ending points of the reference parts can be identified without any predefined values.

8.8 Cross-references validation

Sometimes the patterns in Has_Leaf_Pattern taxonomy causes that RegExpress parser detect extra words/tokens and consider those words as the boundary of the reference. Therefore, we design cross-reference validation step for verifying the tokens of our detected references. This step is for ensuring that the identified references are valid. For this purpose, we define APA standard properties (For example: references can consist of acronyms with all capital letters (example: In accordance with IXXX No. 85.), or references can consist of sequential words that each word starts with capital letters (example: in accordance with National Building Code), or references can consist of digits (example: in accordance with IXXX Std No. 85.), or references can consist of special characters like: “/, -, ., _”, etc. (example: given in NAC/ASC A23.1/A23.2-M)). The tokens of our identified references should pass one these properties to be considered as true token of the reference. This is the last validation step in our work not found in other works. Further explanation is provided in Sections 4.2.3 and 5.5.1.

8.9 Access to third party resources

Our approach involves creating a chain of references, across a set of target documents for each contractual requirement. This is a new idea, of which we have implemented up to two levels of indirection thus far. In addition, we use web scrappers to access third party resources not available locally. In comparison to other works, they did not need to do such indirections because they identify references internal to one legal document unlike in our situation where there can be a network of project documents.

8.10 Document structure

As mentioned in Chapter 3, the main structure of the legal documents processed in previous works (Tran *et al.*, 2014) and (Adedjouma *et al.*, 2014) and (Sannier *et al*., 2016), consists of header, footer, book, title, chapter, section, sub-section, paragraph, etc. In contrast, the main structure of the contractual requirement document in our investigation consists of headers, footers, sections, sub-sections and paragraphs. Headers, footers, sections and sub-sections do not contain any external references; only paragraphs do.

8.11 Overall approach

The six-step algorithm in our overall approach (see Figure 5.1) to identifying external references is totally different from the way internal references are identified in previous work. Our algorithm contains the following steps:

- Structuring Data
- Pattern Setting
- Pattern Recognition
- Reference Identification
- Web Scraping
- Visualization

In contrast, in (Tran *et al.*, 2014), the key steps for recognizing internal references are:

- Mention detection and mention splitting
- Mention Classification, Position Recognition
- Antecedent candidate extraction
- Antecedent Determination

Likewise, in (Adedjouma *et al.*, 2014) the key steps for recognizing internal references are:

- Define schema for structure of legal text
- transform into markup text
- Resolve cross-references
- Visualization & Analysis

In summary, one can clearly note from this chapter the radically different approach we have taken in recognizing cross-references in contractual documents in software projects.

Chapter 9

9 Discussion: Anticipated impact of the results

In this chapter, we discuss about the anticipated impact of the research results in.

Our solution approach adds to the scientific body of knowledge other researchers can tap into for further experimentation and improvement.

Our tabulated result enables easier and more structured exploration between contractual documents and other provisions. Therefore it provides a proper basis for traceability (Ghanavati *et al.*, 2014).

The output of our tool (Table 6.2), represents the high-level type (Table 6.2) and detailed type of views (the content of each row in Table 6.2) of cross-references among documents. From a software engineering perspective, our result on automatic identification of cross-references across a set of external documents and web resources could possibly be helpful in other domains as well. For example, in engineering medical systems, there are regulatory requirements (e.g., HIPAA (Health Insurance Portability and Accountability Act (Office for Civil Rights, 2003)) that a system needs to comply with. Here, automatic identification of cross-references for extracting legal requirements from related documents can generally be highly accurate and save an enormous amount of development, management, and auditing effort. Similarly, in financial systems, transactions must comply with the Sarbanes-Oxley Act (Of, 2015).

Chapter 10

10 Conclusion and Future Work

In this chapter, we first provide a conclusion for this thesis, and then the ongoing future work that we intend to address the shortcomings of our work would be explained.

10.1 Conclusion

In this thesis, we implemented an approach for automatically identifying external references from a contract. We classified external references into three groups based on their differing formats: Direct Cue (DC) references, Indirect Cue (DC) references and No Cue (NC) references (see Sections 4.2.2.1, 4.2.2.2, 4.2.2.3). The format differences required distinct approaches for automatically identifying the references. In the case study contract with 683 pages and 10345 paragraphs, we identified 667 DC references (83% of the total external references), 15 IDC references (1.8%) and 112 NC references (13.96%). As described in Section 4.2.2.1, this thesis focuses on DC references.

As data preparation, we created two taxonomies: “whitelist” and ‘Hasleaf_Pattern”. The first one is a list consists of a number of “reporting phrases” that precede cross-references in the contract. They help in the identification of the cross-references. The second taxonomy consists of patterns that aid in finding the staring and end points of references.

By utilizing POST (see Section 2.7.1), RegExpress parser (see Section 2.10) and the mentioned taxonomies, we have created a tool (see Section 6.1.1), that can identify DC references from contracts with an average F-measure of 92% (average recall of 86% and an average precision of 99%). For cross-references with target documents not available locally, the tool searches the world wide web using Web Scraping techniques (see Section 6.1.2). With the target resource determined, the tool attempts to find second level references. Currently, the tool is limited to two levels of reference identification. This tabulated reference shows the relations between the references in the contract and the target resources with domain information.

This tabulated information can be used by various stakeholders including: project managers for scoping the effort and time for compliance analysis; analysts for eliciting project requirements;

testers for creating test cases, and others. The case study contract of 683 pages and 10345 paragraphs was processed for cross-references by the tool in approx. 17 seconds; manually identifying these references would take a number of days, thus saving an enormous amount of time and effort, not to mention the quality of the work. From this research, we conclude that it is indeed possible to identify cross-references that target external documents including the web.

10.2 Future Work

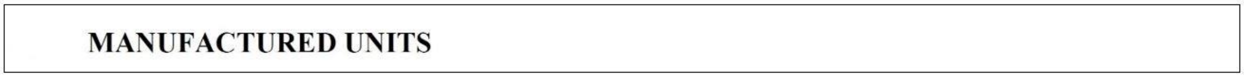
While the research work accomplished is encouraging, it is far from complete. Below, we describe example further work that can be pursued.

10.2.1 Identifying IDC and NC References

Recall that in Sections 4.2.2.1, 4.2.2.2, 4.2.2.3, we identified three types of references: DC, IDC and NC. The latter two were out of the scope of this thesis due to time constraints.

For IDC references our current approach can already identify the part of the reference which is coming after the reporting phrase. For finding the other part of the reference, more investigation should be done.

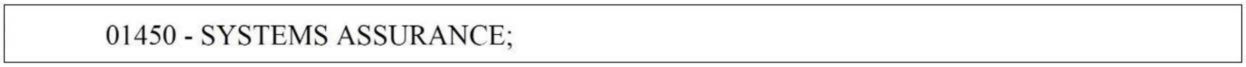
As for NC references, it is important to note that if we consider each NC reference as a paragraph, then we can say that the format of DC references is very similar to the format of section and sub-section titles. Figure 10.1 shows an example of a section title, which is always written in bold in contractual requirements:



MANUFACTURED UNITS

Figure 10. 1: An example forms a section title

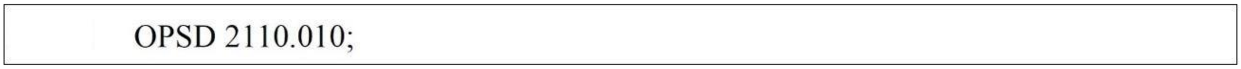
Figure 10.2 shows an example from a sub-section title



01450 - SYSTEMS ASSURANCE;

Figure 10. 2: An example from a subsection title

Figure 10.3 shows an example from a DC reference



OPSD 2110.010;

Figure 10. 3: An example from a DC reference

As shown in Figures 10.1,10.2,10.3, all of them consists of the combination of capital words and numbers. Therefore, NC references can be easily identified if all the words of a paragraphs meet APA properties (Section 4.2.3). But the important problem is with how the algorithm can distinguish a paragraph with a section or sub-section. It is an important issue because, if the algorithm doesn't be capable of determining the essence of the paragraphs/CREs, then it is likely that all the sections and sub-sections will be identified as NC references. For solving this issue, we believe by helping the applied approach in (Adedjouma *et al*., 2014) this problem will be mitigated. As elaborated in Chapter 3, they used a class diagram including the most important elements of a legal document, including: book, title, chapter, section, sub-section, paragraph, etc. We can use the same approach for distinguishing the paragraphs with sections and sub-sections. And then if each paragraph meets APA properties, it is likely to be a DC reference. This is our current idea; however, this idea can be improved after more investigation.

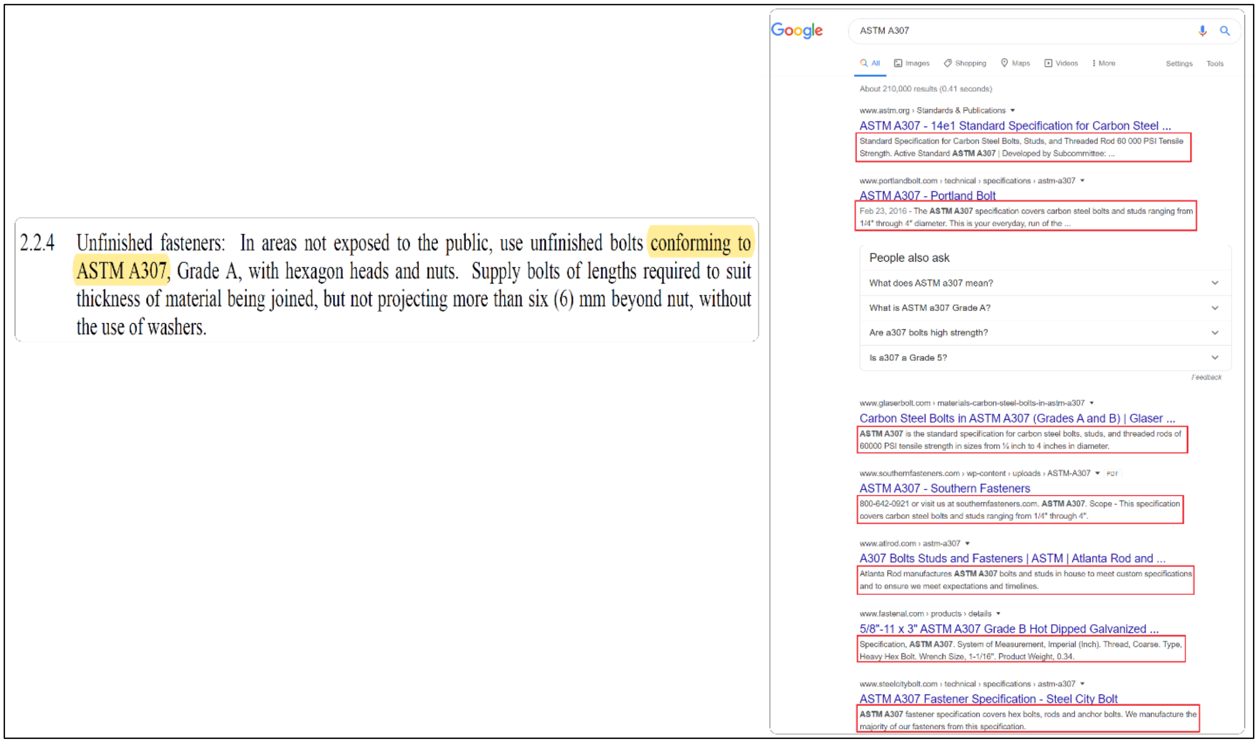
10.2.2 Choosing the Most Related URL

As we discussed in Section 5.6.2, in our current work when the identified references are posted to google search API, Google recommends 10 URLs and the algorithm chooses the first one as the related source. However, sometimes the first recommended URL is not the best source. We thus need a way to identify the most relevant URL based on, for example, certain against which the content from the recommended URLs can be compared. This is important because directing into the target resource will dictate the quality of information that can be elicited as project requirements.

At the moment our idea for mitigating this issue is using topic modeling technique (Duan and Zeng, 2013), which is mostly used for extracting the hidden topics from large volumes of text (Duan and Zeng, 2013). In Figure 10.4, in the left-side we depict a CRE (contains the "ASTM

A307" reference) extracted from contractual document, and in the right-side of the figure google recommendation URLs are shown. Each URL has a "description" (red rectangles in Figure 10.4). These "descriptions" can be identified by webs carpers, and in our current code we have access to these "descriptions". The basic idea is doing the following steps:

1. Doing one of the following approaches i or ii:
   1. applying topic modeling technique on each "description" and generating a topic for each one.
   2. Opening each URL and then applying topic modeling technique on all the context of each web page.
2. Applying topic modeling on the CRE (the left image in Figure 10.4), and generating a topic for the CRE.
3. Comparing (semantical comparison or string comparison, etc.) all the generated topics produced in step 1 with the generated topic produced in in step 2 and Choosing the most related/similar one.



2.2.4 Unfinished fasteners: In areas not exposed to the public, use unfinished bolts conforming to ASTM A307, Grade A, with hexagon heads and nuts. Supply bolts of lengths required to suit thickness of material being joined, but not projecting more than six (6) mm beyond nut, without the use of washers.

About 210,000 results (0.41 seconds)

www.astm.org › Standards & Publications
ASTM A307 - 14e1 Standard Specification for Carbon Steel ...
Standard Specification for Carbon Steel Bolts, Studs, and Threaded Rod 60 000 PSI Tensile Strength. Active Standard **ASTM A307** | Developed by Subcommittee: ...

www.portlandbolt.com › technical › specifications › astm-a307
ASTM A307 - Portland Bolt
Feb 23, 2016 - The **ASTM A307** specification covers carbon steel bolts and studs ranging from 1/4" through 4" diameter. This is your everyday, run of the ...

People also ask
What does ASTM a307 mean?
What is ASTM a307 Grade A?
Are a307 bolts high strength?
Is a307 a Grade 5?

www.glaserbolt.com › materials-carbon-steel-bolts-in-astm-a307
Carbon Steel Bolts in ASTM A307 (Grades A and B) | Glaser ...
ASTM A307 is the standard specification for carbon steel bolts, studs, and threaded rods of 60000 PSI tensile strength in sizes from ¼ inch to 4 inches in diameter.

www.southernfasteners.com › wp-content › uploads › ASTM-A307 PDF
ASTM A307 - Southern Fasteners
800-642-0921 or visit us at southernfasteners.com. **ASTM A307**. Scope - This specification covers carbon steel bolts and studs ranging from 1/4" through 4".

www.atlrod.com › astm-a307
A307 Bolts Studs and Fasteners | ASTM | Atlanta Rod and ...
Atlanta Rod manufactures **ASTM A307** bolts and studs in house to meet custom specifications and to ensure we meet expectations and timelines.

www.fastenal.com › products › details
5/8"-11 x 3" ASTM A307 Grade B Hot Dipped Galvanized ...
Specification, **ASTM A307**. System of Measurement, Imperial (Inch). Thread, Coarse. Type, Heavy Hex Bolt. Wrench Size, 1-1/16". Product Weight, 0.34.

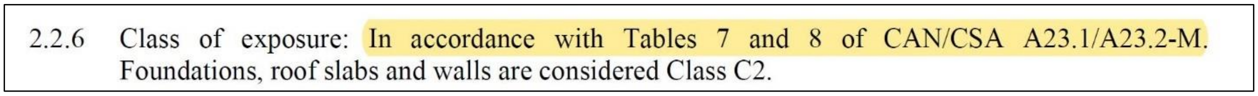
www.steelcitybolt.com › technical › specifications › astm-a307
ASTM A307 Fastener Specification - Steel City Bolt
ASTM A307 fastener specification covers hex bolts, rods and anchor bolts. We manufacture the majority of our fasteners from this specification.

Figure 10. 4: Google recommendation URLs for a reference

Important to note that, this is a basic idea for now and more investigation should be done on that.

10.2.3 Finding Related Context

Some of the identified references refer to a specific part of the target resource, not the whole document or resource. For example, an identified reference may refer to a certain paragraph, a figure, a table, a sentence, a class, etc. In such a situation, there is need to find all the second-order references that might exist in the entire target document or resource; simply identifying the second-order references in the specific part of the target document/resource referenced suffices. Figure 10.5 depicts an example. Currently, our tool does not handle such cases.



2.2.6 Class of exposure: In accordance with Tables 7 and 8 of CAN/CSA A23.1/A23.2-M. Foundations, roof slabs and walls are considered Class C2.

Figure 10. 5: Example of an external cross-reference referring to a table

References

Adedjouma, M., Sabetzadeh, M. and Briand, L. C. (2014)
“Automated detection and resolution of legal cross references: Approach and a study of Luxembourg’s legislation,” in *2014 IEEE 22nd International Requirements Engineering Conference, RE 2014 - Proceedings*, pp. 63–72. doi: 10.1109/RE.2014.6912248.

Antón, P. N. O. and A. I. (2007)
“Addressing Legal Requirements in Requirements Engineering,” *Proceedings - 15th IEEE International Requirements Engineering Conference, RE 2007*, pp. 379–380. doi: 10.1109/RE.2007.65.

Association, A. P. (2019a)
American Psychological Association. Available at: https://apastyle.apa.org/about-apa-style.

Association, A. P. (2019b)
APA Style. Available at: https://apastyle.apa.org/style-grammar-guidelines/citations/index.

AUSB (2017)
“APA Signal Phrases for Quotes/Paraphrases.” Available at: https://www.antioch.edu/santa-barbara/wp-content/uploads/sites/4/2017/02/APA-Signal-Phrases-for-Quotes-and-Paraphrases.pdf.

Authority, H. & S. (2020)
Healthy, safe and productive lives and enterprises. Available at: https://www.hsa.ie/eng/Vehicles_at_Work/Work_Related_Vehicle_Safety/Legal_Requirements/.

Breaux, T. D. (2009)
Legal Requirements Acquisition for the Specification of Legally Compliant Information Systems. North Carolina State University.

Breaux, T. D. and Antón, A. I. (2008)
“Analyzing regulatory rules for privacy and security requirements,” *IEEE Transactions on Software Engineering*, 34(1), pp. 5–20. doi: 10.1109/TSE.2007.70746.

Brian Berenbach, Ren-Yi Lo, B. S. (2010)
"Contract-based requirements engineering," *2010 3rd International Workshop on Requirements Engineering and Law, RELAW 2010*, (October 2010), pp. 27–33. doi: 10.1109/RELAW.2010.5625354.

Centre, W. and Guide, L. (2014)
"Verbs for Reporting Writing Centre Learning Guide." Available at: https://www.adelaide.edu.au/writingcentre/sites/default/files/docs/learningguide-verbsforreporting.pdf.

Committee, P. A. S. (2001)
Standard for Information Technology — Portable Operating System Interface (POSIX ®) Technical, Group. doi: 10.1109/IEEESTD.2004.94442.

David Ascer, M. L. (2004)
Learning Python. Available at: https://books.google.ca/books/about/Learning_Python.html?id=ftA0yk1Z92wC&printsec=frontcover&source=kp_read_button&redir_esc=y#v=onepage&q&f=false.

Davis, J. and Goadrich, M. (2006)
"The relationship between precision-recall and ROC curves," *ACM International Conference Proceeding Series*, 148, pp. 233–240. doi: 10.1145/1143844.1143874.

Duan, J. and Zeng, J. (2013)
"Web objectionable text content detection using topic modeling technique," *Expert Systems with Applications*. Elsevier Ltd, 40(15), pp. 6094–6104. doi: 10.1016/j.eswa.2013.05.032.

EAP Foundaton (2019)
"Reporting verbs." Available at: https://www.eapfoundation.com/writing/references/reporting/.

Eisenstein, J. (2019)
Introduction to Natural Language Processing. Available at: https://books.google.ca/books?hl=en&lr=&id=72yuDwAAQBAJ&oi=fnd&pg=PR5&dq=natural+language+processing+2019&ots=gVbNY2-kp2&sig=jf32aGmiIBbrU9C3ezcZ1dEnwlg#v=onepage&q=natural language processing 2019&f=false.

Friedl, J. E. F. (2009)
Mastering Regular Expressions, 3e (O'Reilly, 2006). Available at: http://xlb.es/Mastering Regular Expressions (Friedl-2006).pdf.

Ghanavati, S. *et al.* (2014)
"Goal-oriented compliance with multiple regulations," *2014 IEEE 22nd International Requirements Engineering Conference, RE 2014 - Proceedings*. IEEE, (i), pp. 73–82. doi: 10.1109/RE.2014.6912249.

Government, C. (2019)
Environment Canada. Available at: https://www.canada.ca/en/health-canada/services/chemical-substances/canada-approach-chemicals/canadian-environmental-protection-act-1999.html.

Hamdaqa, M. and Hamou-Lhadj, A. (2011)
"An approach based on citation analysis to support effective handling of regulatory compliance," *Future Generation Computer Systems*. Elsevier B.V., 27(4), pp. 395–410. doi: 10.1016/j.future.2010.09.007.

Hamou-lhadj, A. (2010)
Regulatory Compliance and its Impact on Software Development. Concordia University.

Hancock, E. (1968)
Pattern Recognition. Available at: https://www.journals.elsevier.com/pattern-recognition.

Hardeniya, N. (2015)
NLTK Essentials. Available at: https://books.google.ca/books?hl=en&lr=&id=NDlECgAAQBAJ&oi=fnd&pg=PP1&dq=NLTK+Library&ots=dCgq1d61TC&sig=MNqZmCjgLkYjkSH1_DfxzdWcoM4&redir_esc=y#v=onepage&q=NLTK&f=false.

Ingolfo, S. *et al.* (2013)
"Arguing regulatory compliance of software requirements," *Data and Knowledge Engineering*. Elsevier B.V., 87, pp. 279–296. doi: 10.1016/j.datak.2012.12.004.

Kit, J. J. W. & C. (1992)
"TOKENIZATION AS THE IINIITIAL PHASE IIN NLP," *Japanese Society of Biofeedback Research*, 19, pp. 709–715. doi: 10.20595/jjbf.19.0_3.

Krzystof Jajuga, Andrzej Sokolowski, H.-H. B. (2002) *Classification, Clustering, and Data Analysis: Recent Advances and Applications*. Available at: https://books.google.ca/books?hl=en&lr=&id=0YrsCAAAQBAJ&oi=fnd&pg=PR5&dq=Classification,+Clustering,+and+Data+Analysis:+Recent+Advances+and+Applications&ots=cR3M5QJb3K&sig=9GpzggH3qmJTBxkxatKXKM9gAdU#v=onepage&q=Classification%2C Clustering%2C and Data An.

Maratea, A., Petrosino, A. and Manzo, M. (2014) "Adjusted F-measure and kernel scaling for imbalanced data learning," *Information Sciences*. Elsevier Inc., 257, pp. 331–341. doi: 10.1016/j.ins.2013.04.016.

Maxwell, J. C. *et al.* (2012) "A legal cross-references taxonomy for reasoning about compliance requirements," *Requirements Engineering*, 17(2), pp. 99–115. doi: 10.1007/s00766-012-0152-5.

Maxwell, J. C., Antón, A. I. and Swire, P. (2011) "A legal cross-references taxonomy for identifying conflicting software requirements," *Proceedings of the 2011 IEEE 19th International Requirements Engineering Conference, RE 2011*, pp. 197–206. doi: 10.1109/RE.2011.6051647.

Mitchell, R. (2018) *Web Scraping with Python: Collecting More Data from the Modern Web*. Available at: https://yanfei.site/docs/dpsa/references/PyWebScrapingBook.pdf.

Navigli, R. (2009) "Word sense disambiguation: A survey," *ACM Computing Surveys*, 41(2). doi: 10.1145/1459352.1459355.

Nekvi, M. R. I. and Madhavji, N. H. (2014) "Impediments to regulatory compliance of requirements in contractual systems engineering projects: A case study," *ACM Transactions on Management Information Systems*, 5(3). doi: 10.1145/2629432.

Nicolas Sannier, Morayo Adedjouma, Mehrdad Sabetzadeh, L. B. (2016) "Automated Classification of Legal Cross References Based on Semantic Intent Nicolas," in *IEEE Software*, pp. 86–91. doi: 10.1109/MS.2011.81.

Of, A. (2015)
“Sarbanes-Oxley Act of 2002,” *The Complete CPA Reference*, pp. 685–687. doi: 10.1002/9781119204121.ch13.

Office for Civil Rights, U. S. D. of H. & H. S. (2003)
“OCR PRIVACY BRIEF SUMMARY OF THE HIPAA PRIVACY RULE HIPAA Compliance Assistance,” *Summary of HIPAA Privacy Rule*, p. 23. doi: 10.1016/j.chroma.2005.11.119.

Oxford University (2019)
Oxford English Dictionary. Available at: https://www.oed.com/search?searchType=dictionary&q=Cross+Reference&_searchBtn=Search.

PennState Library University (2020)
APA Quick Citation Guide. Available at: https://guides.libraries.psu.edu/apaquickguide/intext.

Sannier, N. *et al.* (2017)
“An automated framework for detection and resolution of cross references in legal texts,” *Requirements Engineering*, 22(2), pp. 215–237. doi: 10.1007/s00766-015-0241-3.

Dos Santos, C. N. and Zadrozny, B. (2014)
“Learning character-level representations for part-of-speech tagging,” *31st International Conference on Machine Learning, ICML 2014*, 5(2011), pp. 3830–3838.

Steven Bird, Ewan Klein, and E. L. (2009)
Natural Language Processing with Python. Available at: http://www.nltk.org/book/.

Sun, X. *et al.* (2019)
“Towards easier and faster sequence labeling for natural language processing: A search-based probabilistic online learning framework (SAPO),” *Information Sciences*. Elsevier Inc., 478, pp. 303–317. doi: 10.1016/j.ins.2018.11.025.

Sydney, U. (2019)
Introducing Quotations and Paraphrases. Available at: https://student.unsw.edu.au/introducing-quotations-and-paraphrases.

Toronto, U. of (2005)
"Verbs For Citing Sources," (2003). Available at: https://www.utsc.utoronto.ca/ccds/sites/utsc.utoronto.ca.ccds/files/5.pdf.

Tran, O. T. *et al.* (2014)
"Automated reference resolution in legal texts," *Artificial Intelligence and Law*, 22(1), pp. 29–60. doi: 10.1007/s10506-013-9149-8.

University, M. (2018)
Citing and referencing: In-text citations. Available at: https://guides.lib.monash.edu/citing-referencing/APA-In-text.

University, T. R. (2007)
"Reporting Words / Phrases," p. 2007. Available at: https://www.tru.ca/__shared/assets/Reporting_Phrases30249.pdf.

Wikipedia (2019)
Unit of observation. Available at: https://en.m.wikipedia.org/wiki/Unit_of_observation.

Wu, Y. C., Lee, Y. S. and Yang, J. C. (2008)
"Robust and efficient multiclass SVM models for phrase pattern recognition," *Pattern Recognition*, 41(9), pp. 2874–2889. doi: 10.1016/j.patcog.2008.02.010.

Curriculum Vitae

Name: Elham Rahmani

Post-Secondary Education and Degrees:

Shiraz University
Shiraz, IRAN
2002-2004
Associate of Science in Computer Software

Shiraz Azad University
Shiraz, IRAN
2007-2009
Bachelor of Science in Computer Software

University of Western Ontario
London, Ontario, Canada
2018-2020
Master of Science in Computer Science (candidate)

Related Work Experience:

Graduate Research Assistant & Teaching Assistant (Computer Science)
University of Western Ontario
London, Ontario, Canada
2018-2020

Software Architect
Amel System Company
Shiraz, IRAN
2012-2017

Senior Software Developer & System Analyst
South Information Technology Company (SITCO)
Shiraz, IRAN
2010-2012

Software Developer
Parseh Smart Researchers Company
Shiraz, IRAN
2009-2010

Teacher (Computer Programming)
Tohid Technical High School
Shiraz, IRAN
2004-2006