NATURAL LANGUAGE PROCESSING-BASED AUTOMATED INFORMATION EXTRACTION FROM BUILDING CODES TO SUPPORT AUTOMATED COMPLIANCE CHECKING

by

Xiaorui Xue

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

Department of Construction Management Technology
West Lafayette, Indiana
August 2022

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Jiansong Zhang, Chair

Department of Constrution Management Technology

Dr. Yunfeng Chen

Department of Construction Management Technology

Dr. Luciana Debs

Department of Construction Management Technology

Dr. Yi Jiang

Department of Construction Management Technology

Dr. Zeljko M. Torbica

Department of Construction Management Technology

Approved by:

Dr. Kathryne A. Newton

To my family…

ACKNOWLEDGMENTS

I want to thank my major professor, Dr. Jiansong Zhang for his support and all his help with this PhD study. I also want to thank Dr. Yunfeng Chen, Dr. Luciana Debs, Dr. Yi Jiang, and Dr. Zeljko M. Torbica for serving in my committee. I also want to give thanks to members of our research lab, AutoIC lab. They provide me invaluable help and friendship during my PhD study. In addition, I want to thank my parents for their support in my life. Without their support, I could not have gone this far.

The author would like to thank the National Science Foundation (NSF). The material in this dissertation is based on work supported by the NSF under Grant No. 1827733. Any opinions, findings, and conclusions or recommendations expressed in this dissertation are those of the author and do not necessarily reflect the views of the NSF.

Some or all data, models, or code generated or used during this dissertation are available upon reasonable request.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Traditional manual code compliance checking process is a time-consuming, costly, and error-prone process that has many shortcomings (Zhang & El-Gohary, 2015). Therefore, automated code compliance checking systems have emerged as an alternative to traditional code compliance checking. However, computer software cannot directly process regulatory information in unstructured building code texts. To support automated code compliance checking, building codes need to be transformed to a computer-processable, structured format. In particular, the problem that most automated code compliance checking systems can only check a limited number of building code requirements stands out.

The transformation of building code requirements into a computer-processable, structured format is a natural language processing (NLP) task that requires highly accurate part-of-speech (POS) tagging results on building codes beyond the state of the art. To address this need, this dissertation research was conducted to provide a method to improve the performance of POS taggers by error-driven transformational rules that revise machine-tagged POS results. The proposed error-driven transformational rules fix errors in POS tagging results in two steps. First, error-driven transformational rules locate errors in POS tagging by their context. Second, error-driven transformational rules replace the erroneous POS tag with the correct POS tag that is stored in the rule. A dataset of POS tagged building codes, namely the Part-of-Speech Tagged Building Codes (PTBC) dataset (Xue & Zhang, 2019), was published in the Purdue University Research Repository (PURR). Testing on the dataset illustrated that the method corrected 71.00% of errors in POS tagging results for building codes. As a result, the POS tagging accuracy on building codes was increased from 89.13% to 96.85%.

This dissertation research was conducted to provide a new POS tagger that is tailored to building codes. The proposed POS tagger utilized neural network models and error-driven transformational rules. The neural network model contained a pre-trained model and one or more trainable neural layers. The neural network model was trained and fine-tuned on the PTBC (Xue & Zhang, 2019) dataset, which was published in the Purdue University Research Repository (PURR). In this dissertation research, a high-performance POS tagger for building codes using one bidirectional Long-short Term Memory (LSTM) Recurrent Neural Network (RNN) trainable layer, a BERT-Cased-Base pre-trained model, and 50 epochs of training was discovered. This

model achieved 91.89% precision without error-driven transformational rules and 95.11% precision with error-driven transformational rules, outperforming the otherwise most advanced POS tagger's 89.82% precision on building codes in the state of the art.

Other automated information extraction methods were also developed in this dissertation. Some automated code compliance checking systems represented building codes in logic clauses and used pattern matching-based rules to convert building codes from natural language text to logic clauses (Zhang & El-Gohary 2017). A ruleset expansion method that can expand the range of checkable building codes of such automated code compliance checking systems by expanding their pattern matching-based ruleset was developed in this dissertation research. The ruleset expansion method can guarantee: (1) the ruleset's backward compatibility with the building codes that the ruleset was already able to process, and (2) forward compatibility with building codes that the ruleset may need to process in the future. The ruleset expansion method was validated on Chapters 5 and 10 of the International Building Code 2015 (IBC 2015). The Chapter 10 of IBC 2015 was used as the training dataset and the Chapter 5 of the IBC 2015 was used as the testing dataset. A gold standard of logic clauses was published in the Logic Clause Representation of Building Codes (LCRBC) dataset (Xue & Zhang, 2021). Expanded pattern matching-based rules were published in the dissertation (Appendix A). The expanded ruleset increased the precision, recall, and f1-score of the logic clause generation at the predicate-level by 10.44%, 25.72%, and 18.02%, to 95.17%, 96.60%, and 95.88%, comparing to the baseline ruleset, respectively.

Most of the existing automated code compliance checking research focused on checking regulatory information that was stored in textual format in building code in text. However, a comprehensive automated code compliance checking process should be able to check regulatory information stored in other parts, such as, tables. Therefore, this dissertation research was conducted to provide a semi-automated information extraction and transformation method for tabular information processing in building codes. The proposed method can semi-automatically detect the layouts of tables and store the extracted information of a table in a database. Automated code compliance checking systems can then query the database for regulatory information in the corresponding table. The algorithm's initial implementation accurately processed 91.67 % of the tables in the testing dataset composed of tables in Chapter 10 of IBC

2015. After iterative upgrades, the updated method correctly processed all tables in the testing dataset.

1 INTRODUCTION

A portion of this chapter was previously published in:

Xue, X., Zhang, J. (2020). Building codes part-of-speech tagging performance improvement by error-driven transformational rules. *Journal of Computing in Civil Engineering*, *34*(5), 04020035. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000917

Xue, X., Zhang, J. (2021). Erratum for "Building codes part-of-speech tagging performance improvement by error-driven transformational rules" by Xiaorui Xue and Jiansong Zhang. *Journal of Computing in Civil Engineering*, *35*(1), 08220002. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000950

Xue, X., Zhang, J. (2021). Part-of-speech tagging of building codes empowered by deep learning and transformational rules. *Advanced Engineering Informatics*, *47*(January 2021), 101235. https://doi.org/10.1016/j.aei.2020.101235

Xue, X., Zhang, J. (2022). Regulatory information transformation ruleset expansion to support automated building code compliance checking. *Automation in Construction*, *138*(June 2022), 104230. https://doi.org/10.1016/j.autcon.2022.104230

Xue, X., Wu, J., Zhang, J. (2022). Semi-automated generation of logic rules for tabular information in building codes to support automated code compliance checking. *Journal of Computing in Civil Engineering*, *36*(1), 04021033. https://doi.org/10.1061/(ASCE)CP.1943-5487.0001000

Author Contribution

The authors confirmed contribution to the paper as follows:

Study conception and design: Xiaorui Xue, Jin Wu, Jiansong Zhang.

Data collection: Xiaorui Xue, Jin Wu, Jiansong Zhang.

Analysis and interpretation of results: Xiaorui Xue, Jin Wu, Jiansong Zhang.

Draft manuscript preparation: Xiaorui Xue, Jin Wu, Jiansong Zhang.

All authors reviewed the results and approved the final version of the manuscript.

Construction industry is regulated by a wide range of building codes. Code compliance checking to get the approval of a building permit is a crucial step prior to construction. However, traditional manual code compliance checking is time-consuming and expensive (Zhang & El-Gohary, 2016). Therefore, the demand to automate code compliance checking emerged. In order to achieve full automation in code compliance checking, regulatory information in building codes must be extracted and stored in a computer-processable and structured format to support automated code compliance checking.

Building codes need to be converted from unstructured natural language to a structured format that computers are able to process to support automated code compliance checking. Some automated code compliance checking systems relied on hiring domain experts to perform manual transformations (İlal & Günaydın, 2017). However, attempts to achieve automated code compliance checking this way, even with support from government, often ceased after intensive investment due to high cost of maintaining domain experts' efforts (Amor & Dimyadi, 2021). The automated transformation of building code requirements to a computable structured format is a natural language processing (NLP) task that requires highly accurate part-of-speech (POS) tagging on building codes. Part-of-speech taggers categorize words according to their syntactic functions in a sentence (Brill, 1992) and was frequently used as a basic step in NLP-based architecture, engineering, and construction (AEC) domain research and applications (Kwayu et al, 2019; Ren & Zhang, 2021; Zhang & El-Gohary, 2013). Existing POS taggers, however, do not provide sufficient accuracy on building codes, because performance of POS taggers deteriorates in out-of-domain text (Coden et al., 2005). To better support automated code compliance checking, the authors proposed the following Natural Language Processing (NLP)-

based methods to support automated Information Extraction (IE) from building codes in this chapter: (1) improving performance of part-of-speech tagging on building codes by error-fixing rules, (2) part-of-speech tagging of building code by deep learning and error-fixing rules, (3) generating logic clauses for tabular information in building codes, and (4) pattern matching-based transformational ruleset expansion method to increase the coverage of building code transformational rules.

This dissertation research was conducted to improve the performance of POS taggers by error-driven transformational rules that revise machine tagged POS results. The proposed method utilizes a syntactic and semantic rule-based, NLP approach combined with a structure that is inspired by transfer learning. In transfer learning, large models, which are usually trained on large body of texts on unsupervised tasks, are fine-tuned on small labeled datasets to increase performance on supervised tasks (Pan & Yang, 2009). This method generates a group of transformational rulesets, from simple ones to complex ones, that can convert machine taggers' tagging results to their corresponding human-labeled gold standard. The transformational rules utilize syntactic and semantic information of domain texts.

Automated building code compliance checking systems were under development for many years (Dimyadi & Amor, 2013). However, the excessive number of human inputs needed to convert building codes from natural language to computer understandable codes severely limited their range of applicable code requirements (İlal & Günaydın, 2017). To address this, automated code compliance checking systems need to enable an automated regulatory rules conversion. Accurate POS tagging of building code texts is crucial to support this conversion. Previous experiments showed that the state-of-the-art generic POS taggers did not perform well on building codes (Xue & Zhang, 2020). In view of that, this dissertation research was conducted to provide a new POS tagger tailored to building codes. It utilizes deep learning neural network model and error-driven transformational rules. The neural network model contains a pre-trained model and one or more trainable neural layers. The pre-trained model was fine-tuned on Part-of-speech Tagged Building Codes (PTBC), a POS tagged building codes dataset prepared in this dissertation research. The fine-tuning of pre-trained model allows the proposed POS tagger to reach high precision with a small amount of available training data.

One limitation of many existing automated code compliance checking systems/methods is their limited range of checkable building code requirements. To address that, the state of the art uses pattern matching-based rules to transform building code requirements to computable formats automatically, but the ruleset was developed and tested only on few chapters of building code requirements (Zhang & El-Gohary, 2016). An efficient ruleset expansion method is needed to enlarge its range of checkable building code requirements with low-cost and bring automated code compliance checking systems closer to full deployment. Expanding an existing regulatory information transformation ruleset requires less manual effort than developing a new ruleset. This dissertation research was conducted to provide a method that can expand the range of checkable code requirements of automated code compliance checking systems without significant manual effort. The proposed ruleset expansion method takes an iterative approach to ensure the generality and validity of new pattern matching-based rules and the quality of information transformation results.

Another limitation of the range of checkable building codes of many existing automated code compliance checking system is that they mostly focused on and were limited to automatically processing regulatory information that was stored in the text part of the codes. Nonetheless, a fully automated method for code compliance checking should be able to examine regulatory information in other parts of the textual document, such as in tables. This dissertation research was therefore conducted to provide a semiautomated information extraction and transformation approach for tabular regulatory information in building codes. The proposed method can detect table layouts semi-automatically and save extracted table information in a database. Automated code compliance checking systems can then query this database for regulatory information in related tables from building codes.

1.1 Research Question

This dissertation research was conducted to answer the research question "How to improve the automated processing of building codes to better support automated code compliance checking compared to the state of the art?"

The main research question was divided into two sub-questions:

1. How to improve the performance of POS tagging on building codes compared to the state of the art?
2. How to expand the range of checkable building code requirements that can be used in state-of-the-art automated code compliance checking systems?

1.2 Significance

The problem addressed in this dissertation research is the lack of full automation in building code compliance checking. Manual code compliance checking is time-consuming, costly and error-prone (Zhang & El-Gohary 2015). The average waiting time for obtaining building permit is more than two months with a minimum cost of hundreds of dollars (Xue & Zhang, 2020). Productivity of construction industry is also affected by a slow manual code compliance checking (Ding et al., 2006). Construction industry contributes 4.1% of US economy in 2018 and 2019 (Bureau of Economic Analysis, 2022). However, productivity of the construction industry has been in stagnation (Bureau of Labor Statistics, 2018). Automated code compliance checking can reduce errors and improve efficiency in code compliance checking (Zhong et al., 2012). Non-compliance in building design is expensive to fix and could lead to expensive penalty fines.

1.3 Purpose Statement

The overall purpose of this dissertation research is developing NLP-based automated information extraction methods to support automated building code compliance checking. For the four methods proposed in this dissertation, each of them has their own specific purpose as detailed as follows. The goal of the first two POS tagging methods is to improve the performance of POS taggers on building codes compared to the state of the art. The third and fourth methods, the ruleset expansion methods and the tabular information extraction, respectively, aim to increase the range of checkable building code requirements that can be used in state-of-the-art automated code compliance checking systems.

1.4 Dissertation Structure

This dissertation consists of six chapters (Table 0.1) and addressed two research questions (Figure 0.1). Chapter One (i.e., introduction chapter) introduces the motivation behind the research carried out in this dissertation research and two research questions addressed in this dissertation research. Chapter Two focuses on improving the accuracy of POS tagging of existing POS taggers on building codes by using error-driven transformational rules. Chapter Three then goes beyond the use of existing POS taggers by developing a new building code POS tagger that combines the use of error-driven transformational rules and a neural network model.

A highly accurate POS tagging is important for automated code compliance checking to achieve full automation because NLP is needed to extract and transform regulatory information from building codes automatically into a computable format and POS tagging is an important basic step in NLP. Zhang (2015) described the importance/challenge of the NLP as:

> "For the purpose of ACC, a successful information extraction does require correct understanding of the text source (i.e., textual regulatory documents). This need of a deep level of NLP makes the problem of automated information extraction for compliance checking purposes challenging." (p.11)

POS tagging is an essential first step of many (if not all) NLP processes. Previous works use generic POS tagger, whose performance is limited on building codes. To push for full automation, a construction domain specific POS tagger was developed in this dissertation research.

Chapter Four expands the range of checkable building code requirements of an automated code compliance checking system by providing a pattern matching-based ruleset expansion method to expand an existing pattern matching-based ruleset. The pattern matching-based ruleset utilizes POS tagging information of the building code. Chapter Five expands the range of checkable building code requirements from textual information to non-textual information by proposing a tabular information extraction method. Last but not least, Chapter Six, which is the conclusion chapter, discusses the overall conclusions of the dissertation research.

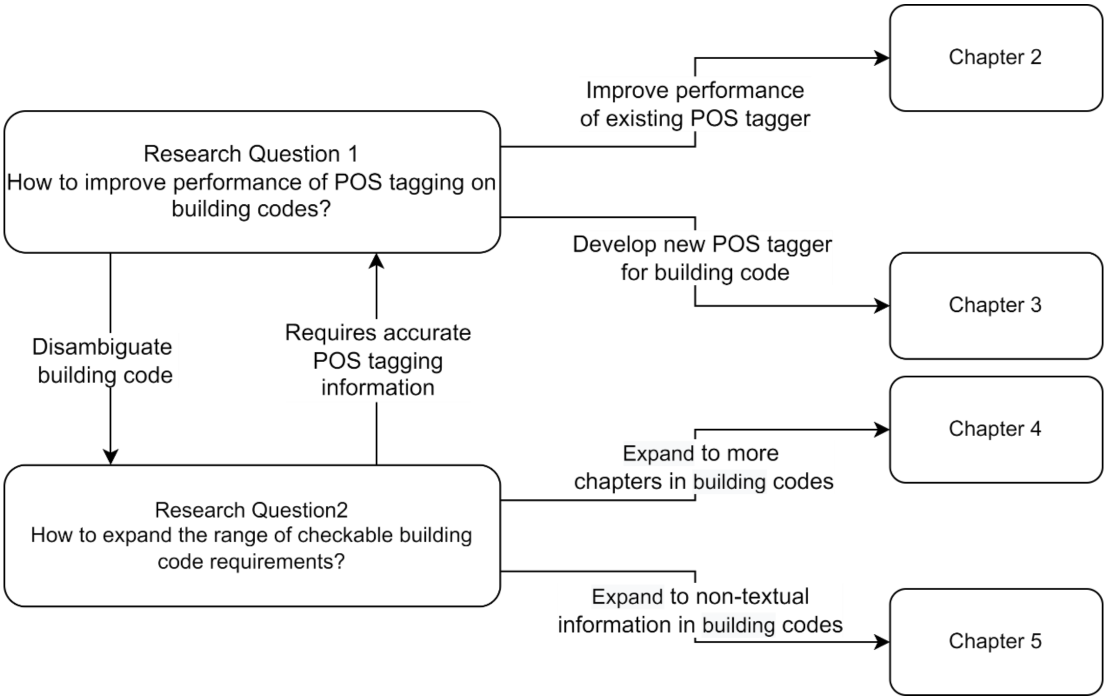

Chapter 2
Improve performance of existing POS tagger
Research Question 1
How to improve performance of POS tagging on building codes?
Develop new POS tagger for building code
Chapter 3
Disambiguate building code
Requires accurate POS tagging information
Chapter 4
Expand to more chapters in building codes
Research Question2
How to expand the range of checkable building code requirements?
Expand to non-textual information in building codes
Chapter 5

Figure 0.1. Relation between Published Chapters and Research Questions

Table 0.1. Chapters in the Dissertation

Chapter	Title
1	Introduction
2	Building codes part-of-speech tagging performance improvement by error-driven transformational rules
3	Part-of-speech tagging of building codes empowered by deep learning and transformational rules.
4	Regulatory information transformation ruleset expansion to support automated building code compliance checking.
5	Semi-automated generation of logic rules for tabular information in building codes to support automated code compliance checking.
6	Conclusion

1.5 Definitions

A group of concepts and terms are central to this dissertation research. To provide key information about the dissertation and facilitate understanding to the dissertation research, definitions of key concepts and terms are provided in this section. For terms and concepts that are unique to this dissertation research, operational definitions are provided.

Automated Code Compliance Checking System (ACCC) system is defined as “a software that does not modify a building design, but rather assesses a design on the basis of the configuration of objects, their relations or attributes” automatically (Eastman et al., 2009).

Building Code is defined as “a set of laws enacted by state, county and city governments to determine the required design and construction standards for home construction” (Findwell, 2020).

Industry Foundation Classes (IFC) is defined as “a standardized, digital description of the built asset industry.” (buildingSMART International, 2020). The IFC models are defined using the Standard for Exchange of Product (STEP) method. The IFC specification is drafted in the EXPRESS data modeling language. The IFC standard is registered as an international standard (ISO 16739-1:2008). The IFC standard is a vender-neutral, open and platform-independent standard. (buildingSMART International, 2020).

Building Information Modeling (BIM) is defined as *“a digital representation of physical and functional characteristics of a facility. A BIM is a shared knowledge resource for information about a facility forming a reliable basis for decisions during its life-cycle; defined as existing from earliest conception to demolition” by the National Building Information Model Standard Project Committee. The application of BIM promises the collaboration of stakeholders in different stage of a building construction project. (National Building Information Model Standard Project Committee, 2022). The BIM model of a building can be used to plan, design, construct, and operate the building (Azhar, 2011). In recent years, the term can also be used to refer to the process of creating digital representation of a built asset (Autodesk Company, 2022).*

Natural Language Processing (NLP) is defined as “the subfield of computer science concerned with using computational techniques to learn, understand, and produce human language content” (Hirschberg & Christopher, 2015). Natural Language Processing “in a wide sense to cover any type of computer manipulation of natural language. At one extreme, it could be as simple as counting word frequencies to compare different writing styles. At the other extreme, NLP involves “understanding” complete human utterances, at least to the extent of being able to give useful responses to them.” (Bird et al., 2009). Natural language processing algorithm can take a rule-based approach or used statistical models (Nadkarni et al., 2011).

Machine Learning (ML) is defined as a “field of study that gives computers the ability to learn without being explicitly programmed” (Samuel, 1959). Another popular definition of machine learning is “the study of computer algorithms that allow computer programs to automatically improve through experience” (Mitchell, 1997). Machine learning includes supervised algorithms that infer the underlying relationship between observed data and targeted value (label) and unsupervised algorithms that discover hidden patterns in unlabeled dataset (Awad & Khanna, 2015).

Deep Learning is defined as “computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction.” (LeCun et al., 2015). Deep learning algorithms use neural networks with a muti-layer structure (Ciregan et al., 2012).

Logic Clause is a representation that supports automated logic reasoning (Zhou, 1994).

Recurrent Neural Network (RNN) is a type of network that feed outputs of previous timesteps to the next step (Staudemeyer & Morris, 2019).

Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) is a type of RNN that has a cell state to control access to information in previous timesteps (Staudemeyer & Morris, 2019).

1.6 Assumptions

In this dissertation research, the following assumptions were made. First, different chapters of the same building code were drafted in a coherent style. Therefore, patterns that exist in one chapter of building code may also exist in other chapters of the same building code. Second, structures in building codes were well-defined (Jiang, 2012). Third, building design information has been comprehensively and accurately extracted from building design documents (i.e., Industrial Foundation Classes (IFC) files). The last assumption is that all needed building design information for code checking was provided in the building design model.

1.7 Limitations

This dissertation research has multiple limitations. First, the conversion of building code from natural language to a computer-processable format is not perfect yet. Manual refinement of conversion result is still needed. Second, this dissertation research focuses on compliance checking of International Building Code 2015. The range of checkable building codes tested in this dissertation research, although improved from the state of the art, is still not comprehensive. Third, the tabular information extraction method is semi-automated instead of fully automated.

1.8 Delimitations

The scope of this dissertation research is limited to building code in English. Building codes that are not in English are excluded from this dissertation research.

1.9 Permission to Republish

Contents of this dissertation are based on published papers. Permissions to republish them in this dissertation are obtained from their corresponding publisher (Appendix B).

Xue, X., Zhang, J. (2020). Building codes part-of-speech tagging performance improvement by error-driven transformational rules. *Journal of Computing in Civil Engineering*, *34*(5), 04020035. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000917

Xue, X., Zhang, J. (2021). Erratum for “Building codes part-of-speech tagging performance improvement by error-driven transformational rules” by Xiaorui Xue and Jiansong Zhang. *Journal of Computing in Civil Engineering*, *35*(1), 08220002. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000950

Xue, X., Zhang, J. (2021). Part-of-speech tagging of building codes empowered by deep learning and transformational rules. *Advanced Engineering Informatics*, *47*(January 2021), 101235. https://doi.org/10.1016/j.aei.2020.101235

Xue, X., Wu, J., Zhang, J. (2022). Semi-automated generation of logic rules for tabular information in building codes to support automated code compliance checking. *Journal of Computing in Civil Engineering*, *36*(1), 04021033. https://doi.org/10.1061/(ASCE)CP.1943-5487.0001000

Xue, X., Zhang, J. (2022). Regulatory information transformation ruleset expansion to support automated building code compliance checking. *Automation in Construction*, *138*(June 2022), 104230. https://doi.org/10.1016/j.autcon.2022.104230

Republished with permission of American Society of Civil Engineers, from Building codes part-of-speech tagging performance improvement by error-driven transformational rules, Xiaorui Xue, and Jiansong Zhang, 34, 2020; permission conveyed through Copyright Clearance Center,Inc.

Republished with permission of American Society of Civil Engineers, from Erratum for “Building codes part-of-speech tagging performance improvement by error-driven transformational rules” by Xiaorui Xue and Jiansong Zhang, Xiaorui Xue and Jiansong Zhang, 35, 2021; permission conveyed through Copyright Clearance Center,Inc.

Republished with permission of Elsevier Science &Technology Journals, from Part-of-speech tagging of building codes empowered by deep learning and transformational rules, Xiaorui Xue, and Jiansong Zhang, 47, 2022; permission conveyed through Copyright Clearance Center,Inc.

Republished with permission of Elsevier Science &Technology Journals, from Regulatory information transformation ruleset expansion to support automated building code compliance checking, Xiaorui Xue, and Jiansong Zhang, 138, 2022; permission conveyed through Copyright Clearance Center,Inc.

Republished with permission of American Society of Civil Engineers, from Semi-automated generation of logic rules for tabular information in building codes to support automated code compliance checking, Xiaorui Xue, Jin Wu, and Jiansong Zhang, 36, 2022; permission conveyed through Copyright Clearance Center,Inc.

2 BUILDING CODES PART-OF-SPEECH TAGGING PERFORMANCE IMPROVEMENT BY ERROR-DRIVEN TRANSFORMATIONAL RULES

Xiaorui Xue, S.M.ASCE[1]; Jiansong Zhang, Ph.D., A.M.ASCE[2]

A portion of this chapter was previously published by:

Xue, X., Zhang, J. (2020). Building codes part-of-speech tagging performance improvement by error-driven transformational rules. *Journal of Computing in Civil Engineering*, *34*(5), 04020035. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000917

Xue, X., Zhang, J. (2021). Erratum for "Building codes part-of-speech tagging performance improvement by error-driven transformational rules" by Xiaorui Xue and Jiansong Zhang. *Journal of Computing in Civil Engineering*, *35*(1), 08220002. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000950

Author Contributions

The authors confirmed contribution to the paper as follows:

1. Study conception and design: Xiaorui Xue, Jiansong Zhang.
2. Data collection: Xiaorui Xue, Jiansong Zhang.
3. Analysis and interpretation of results: Xiaorui Xue, Jiansong Zhang.
4. Draft manuscript preparation: Xiaorui Xue, Jiansong Zhang.
5. All authors reviewed the results and approved the final version of the manuscript.

2.1 Literature Review

2.1.1 Automated Code Compliance Checking

To address the increasing demand in building permits, many researchers and industry experts introduced new methods of code compliance checking. Their efforts focus on making code compliance checking paperless, automated and standardized. The structural design checking using decision table (Fenves, 1966) was one of the first efforts in this domain (İlal &

Günaydın, 2017), which led to many attempts to create expert systems for building codes in the 1980s (Dimyadi & Amor, 2013), such as the Standard Interface for Computer Aided Design (SICAD) (Lopez et al., 1989), the Standards Processing Expert (SPEX) (Delis & Delis, 1995; Garrett & Fenves, 1987), and the Design Prototypes (Gero, 1990). However, low performance and high maintenance cost of expert systems in the 1980s limited these attempts only to proofs of concepts with a lack of actual implementations. An expert system, which uses a vast body of domain-specific knowledge stored in a computer (Liao, 2005), has limitations such as high maintenance cost, difficulty in scalability, and the narrow range of applications (Chollet, 2017) These forerunners' efforts gave birth to more recent code compliance checking expert systems, such as BCAider and DesignCheck, in early 2000s (Dimyadi & Amor, 2013). In addition, there were expert systems that focused on building codes in a specific domain or a limited range of domains in 1980s and 1990s. For example, the Fire-Code Analyzer (Delis & Delis, 1995) focused on fire protection related codes in New Zealand, the Life Safety Code Advisor focused on National Fire Protection Association (NFPA) safety code in the U.S., and the TALLEX (Sabouni & Al-Mourad, 1997) focused on tall buildings in the United Arab Emirates (UAE).

The rise of building information modeling (BIM) since the 2000s provided new ways for performing many tasks in the AEC domain such as structural analysis (Ren & Zhang, 2020; Wong Chong et al., 2020; Wu et al., 2021), fire safety evaluation (Wang et al., 2020), building energy modeling (Li & Zhang, 2020; Li et al., 2022; Li & Zhang, 2021, Li et al., 2021), construction management (Akanbi et al., 2020; Zhang & Laddipeerla, 2018), construction automation (Brissi et al., 2021; Lacny & Zhang, 2022; Wong Chong & Zhang, 2021; Wong Chong et al., 2022) and civil infrastructure management (Akanbi & Zhang 2022; Guo et al., 2021). BIM also dramatically changed the way code compliance systems work by providing a reliable digital representation of buildings (Nguyen & Kim, 2011). For example, Solibri Model Checker (SMC) started as a BIM validation tool, and it obtained code compliance checking ability in its later updates (Eastman et al., 2009). Singapore government initiated the Construction and Real Estate Network (CORENET) project, which allows BIM models, instead of papers, to be submitted for plan review. The UK government started to require submissions of BIM for all public projects that are funded by the British Central Government from 2016 (UK BIM Task Group, 2016). The KbimCode in South Korea was capable of code compliance checking of BIM against building codes, but it needs manual efforts to convert building codes from natural language to a computer-processable format (Choi & Kim, 2017).

2.1.2 Information Extraction Systems

With BIM as a reliable digital representation of buildings, code compliance checking systems made great progress over the last two decades. However, they are still far from a wide real-world deployment. In many current automated code compliance systems, information extraction and information transformation rely on domain experts' manual efforts to convert building codes to a computer-processable format, such as decision tables (Tan et al., 2010), regulatory knowledge model (Dimyadi et al., 2016), and structured regulatory information rulesets (İlal & Günaydın, 2017).

Based on existing literature, current code compliance checking systems lack automated regulatory information extraction and transformation capabilities. By drafting building codes in computer-checkable logic clauses or rulesets instead of natural language, code compliance checking systems can bypass the needed information extraction and transformation step and achieve full automation in an alternative way. However, such a dramatic shift is not expected in a foreseeable future (Bell et al., 2009; Li et al., 2012). In addition, the large size of existing building codes creates further challenges in achieving such a transition. In the U.S., local jurisdictions usually apply customizations and modifications to standard codes published by the international code council (ICC), which further contribute to the complexity of the body of building codes. Automated information extraction and transformation are necessary for automated code compliance systems to function on existing as well as forthcoming building code versions. Some researchers proposed sematic analysis of building codes through deep learning for information extraction, but the extracted information failed to convert to checkable rules (Song et al., 2018). Pattern matching-based natural language processing method, on the other hand, can generate logic clauses through information extraction and transformation with a high accuracy (Li et al., 2016; Xu & Cai, 2020; Zhang & El-Gohary, 2015). The pattern matching-based method of Zhang and El-Gohary (2015) can convert natural language provisions to logic clauses, and their entire automated code compliance checking method reached a 98.7% recall and 87.6% precision in non-compliance detection (Zhang & El-Gohary, 2017). However, to enable real-world applications, the recall must be improved to 100%. The main sources of errors reported by Zhang and El-Gohary (2017) were of two types: limitations of the extraction and transformation rules, and limitations of the state-of-the-art POS taggers' performance on building codes. Reducing/eliminating such errors were expected to further improve the overall non-compliance detection performance. In this chapter, the authors focus on addressing the performance of the state-of-the-art POS taggers on building codes, because the extraction and transformation rules use the POS tagging information and therefore rely on its performance.

2.1.3 Part-of-Speech (POS)

A fully automated code compliance checking system could be an NLP-based system with an essential information extraction and transformation component. The information extraction and transformation component utilizes part-of-speech information as well as other syntactic/semantic information of building codes provisional sentences to convert building codes from natural language to computer-processable representations. POS tagging is about assigning the corresponding morphosyntactic category to each word in a sentence (Giménez & Marquez, 2004). As an early step of the discussed automated code compliance checking system, POS tagging will cascade errors into later steps of the system (Dell'Orletta, 2009) and jeopardize its final performance. An accurate POS tagging results of building codes is the foundation to support the high performance of the information extraction and transformation component and therefore the entire automated code compliance checking system.

POS categories of words are classes of words that share common features (Brill, 1992). In general, there are eight basic POS categories in English, namely, noun, pronoun, verb, adjective, adverb, preposition, conjunction and interjection (Butte College, 2016). However, a decent representation of text for further NLP analysis needs more than just eight POS tags. For example, singular noun and plural noun are usually separated into two different categories. Among the commonly used tagsets, Universal tagset has 12 tags (Petrov et al., 2012), Penn Treebank tagset has 36 tags (Marcus et al., 1993), and Brown tagset has 179 tags (Francis & Kucera, 1979). The authors decided to use Penn Treebank tagset (Table 0.1) because of its good balance between information richness and conciseness.

Table 0.1. POS Tags in the Penn Treebank Tagset

	Tag	Description
1	CC	Coordinating conjunction
2	CD	Cardinal number
3	DT	Determiner
4	EX	Existential *there*
5	FW	Foreign word
6	IN	Preposition or subordinating conjunction
7	JJ	Adjective
8	JJR	Adjective, comparative
9	JJS	Adjective, superlative
10	LS	List item marker
11	MD	Modal
12	NN	Noun, singular or mass
13	NNS	Noun, plural
14	NNP	Proper noun, singular
15	NNPS	Proper noun, plural
16	PDT	Predeterminer
17	POS	Possessive ending
18	PRP	Personal pronoun
19	PRP$	Possessive pronoun
20	RB	Adverb
21	RBR	Adverb, comparative
22	RBS	Adverb, superlative
23	RP	Particle
24	SYM	Symbol
25	TO	*to*
26	UH	Interjection
27	VB	Verb, base form
28	VBD	Verb, past tense
29	VBG	Verb, gerund or present participle
30	VBN	Verb, past participle
31	VBP	Verb, non-3rd person singular present
32	VBZ	Verb, 3rd person singular present
33	WDT	Wh-determiner
34	WP	Wh-pronoun
35	WP$	Possessive wh-pronoun
36	WRB	Wh-adverb

There are multiple ways to get a textual corpus POS tagged. Human annotators can complete this task with their knowledge in English and understanding of the text. However, the high cost, low speed and human inconsistency make it rarely used in real-word applications. This is especially the case if the POS tagging is to support automated extraction and transformation of code requirements for automated compliance checking. In contrast, POS tagging software, or POS taggers (will be called machine taggers hereafter) are usually used in NLP systems because of their fast tagging speed, low tagging cost, and free of human inconsistency. Machine taggers can tag a large amount of text in a short time without human interventions. The large amount of existing and upcoming building codes and frequent building codes updates require a machine POS tagging solution to support automated code compliance checking systems. POS taggers annotate texts according to rules or mathematical models. Correspondingly, there are two main types of machine POS taggers based on their corresponding annotation methodologies: rule-based POS taggers and machine learning POS taggers. These rules or models are either developed by humans or generated by algorithms.

2.1.3.1 Rule-based Part-of-Speech Tagger

Rule-based POS taggers decide POS tags of words based on a set of rules. Rules are designed to make POS tagging results of texts follow human-labeled results. These rules can be either hand-crafted by domain experts or generated by algorithms. Domain experts generate rules based on their understanding of English grammar and the text being tagged. Rules can also be generated by algorithms. POS taggers with hand-crafted rules are rarely used nowadays. They usually are not intended for practical use but rather for educational purposes. For example, Bird et al. (2009) introduced a rule-based tagger with hand-crafted rules for educational purpose. However, this tagger has a low accuracy and only slightly outperformed a baseline tagger that tagged all words as “NNS” (plural nouns) (Bird et al., 2009). Development of rule-based POS taggers stopped because they, even with thousands of hand-crafted rules, fail to reach a comparable accuracy to that of machine learning taggers. For example, the TAGGIT system contains more than 3,000 hand-crafted rules and reached a 77% accuracy on Brown corpus (Greene & Rubin, 1971), whereas the state-of-the-art machine taggers had an accuracy of 87.1% on Brown corpus which was much higher than the 77% accuracy achieved by TAGGIT (Li et al., 2012). However, rule-based POS tagger with algorithm-generated rules can achieve a higher accuracy than rule-based POS taggers with hand-crafted rules (Bird et al., 2009). For example, Brill (1992) developed the Brill tagger with algorithm-generated rules and claimed his tagger’s performance “on par with stochastic taggers.”

2.1.3.2 Machine Learning Part-of-Speech Tagger

Classification is one main task that machine learning was designed for. POS tagging is a type of classification task, i.e., classifying words into different POS categories according to its context and English grammar. Machine learning taggers are built by training machine learning models on corpus of English texts. Different machine learning models can be used such as support vector machines (SVM), decision tree, hidden Markov model (HMM), and neural network.

2.2 Methodology

The authors propose to use transformational rules to address errors in the tagging results of general POS taggers (i.e., machine taggers trained on general English texts) on building codes to increase their accuracy on POS tagging of building codes. Instead of training a new POS tagger from scratch, improving existing taggers can decrease the amount of annotated data needed, therefore save system development time and effort and potentially achieve higher POS tagging accuracy. The transformational rules are automatically generated by algorithms with no human intervention during the generation process execution.

In this chapter, the authors define errors in POS tagging as nonconformities between the machine-assigned POS tag of a word and that word's human-labeled tag. For example, machine taggers make a POS tagging error by tagging the word "can" in the phrase "a steel can," which is a noun, as an "auxiliary verb." Errors are further grouped into types. A type of error subsumes all appearances of a word in the textual data that have the same correct POS tag and are given the same incorrectly assigned POS tag by machine taggers. For example, for all occurrences of the word "can" as a noun, machine taggers may correctly tag them as a noun or incorrectly tag them as a modal verb or verb. For the occurrences that machine taggers incorrectly tagged the word "can" as a verb, it is one type of error. For the occurrences that machine taggers incorrectly tagged the word "can" as a modal verb, it is a different type of error. The proposed method focuses on decreasing the overall occurrence of errors, not specific types of errors. However, knowing possible types of errors is helpful to identify sources of errors. Furthermore, POS tagging errors in building codes textual data show a long-tail distribution. That is, a few types of errors happen many times and most types of errors only happen few times. In fact, for 1,758 types of 31,495 errors in the authors' data of POS tagged building code where errors were defined to be the difference between machine tagging results and manually created gold standard, the top 100 types occurred 20,338 times, which accounted for 64.58% of all errors (Xue & Zhang, 2020). The uneven distribution of errors implies that a small number of fixes may eliminate a large portion of errors.

2.2.1 Overview of the Method

The authors' proposed method divides textual data into two parts, training dataset and testing dataset. The proposed method has two main components, rule generation component, and rule application component. The rule generation component uses rule templates to generate transformational rules. For example, "If the word B after the word A is tagged as X and the word A is tagged as Y, then change the tag of the word A to Z" is a rule template. All rules that are generated by the same template form a ruleset. This method allows users to input their customized templates to generate customized rulesets. The authors provided sample rule templates in the experiment section. The rule generation component generates rules from simple rulesets to complex rulesets, from unigrams to n-grams, and from syntax to semantics. Before the development of each ruleset, the errors in the training set are collected and recorded. A process flowchart about error collection is shown in Figure 0.1. This process compares machine-generated tags of words and their corresponding human-labeled tags (from gold standard) in the training dataset, and records any word whose machine-generated tag is different from its human-labeled tag. If the machine-generated tag of the word "wood" is JJ (Adjective) and its human-labeled tag is NN (Noun), this method then records that the word "wood" is incorrectly tagged as JJ (Adjective) when it should be tagged as NN (Noun). This process is automatically and algorithmically performed by comparing the machine-assigned POS tag of a word and the human-labeled POS tag (from gold standard) of the same word, and recording any discrepancy between them for later steps of this method. After the error collection process, the rule generation process begins. The rule extraction component collects contextual information of errors in the training dataset and converts them to candidate transformational rules according to the template of that ruleset, and filters out unqualified rules. This is also automatically performed without human intervention. The proposed method will collect POS tags of words before and after the target word as the contextual information of the collected error. Before the extraction of the next ruleset, rules in the previous ruleset are applied to the training text. After the completion of rule development, all rulesets are applied to the testing dataset to evaluate the performance of the developed rules. The method also records remaining errors after each ruleset is applied to the testing dataset. The steps of this method are shown in Figure 0.2.

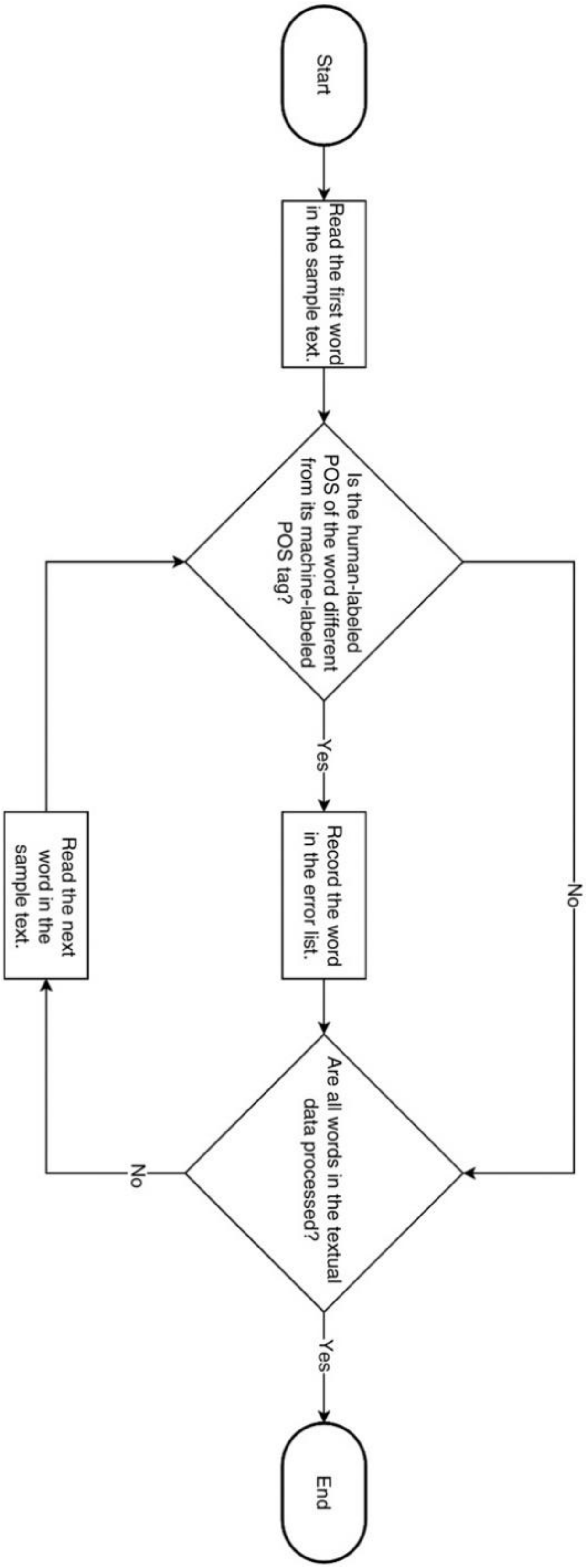
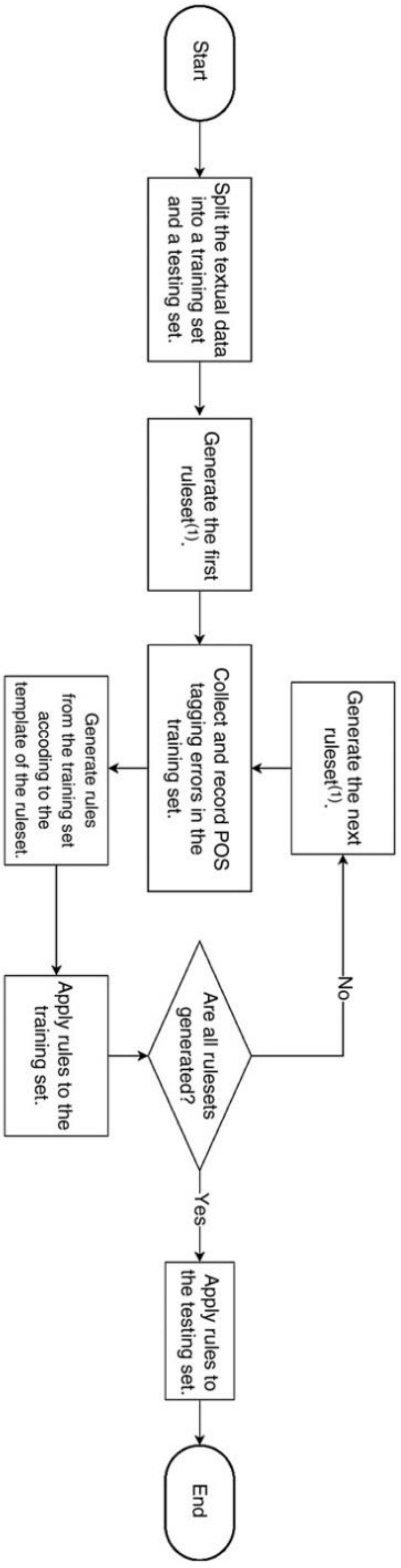

Start
Read the first word in the sample text.
Is the human-labeled POS of the word different from its machine-labeled POS tag?
Yes
No
Record the word in the error list.
Are all words in the textual data processed?
No
Read the next word in the sample text.
Yes
End

Figure 0.1. Error Collection Process


Start
Split the textual data into a training set and a testing set.
Generate the first ruleset(1).
Collect and record POS tagging errors in the training set.
Generate the next ruleset(1).
Generate rules from the training set accoding to the template of the ruleset.
Apply rules to the training set.
Are all rulesets generated?
No
Yes
Apply rules to the testing set.
End

(1) This method used templates to generate rules and all rules that are generated by the same template form a rule set.

Figure 0.2. Proposed Method

2.2.2 Description of Transformational Rules

The transformational rules fix POS tagging errors in the textual data. The POS tagging errors in the textual data are gathered by comparing machine tagging results of the textual data to the human-labeled gold standard. The rules store the word it matches and its contextual information, including semantic information (e.g., the word before the target word is “egress”) and syntax information [e.g., the POS tag of the word before the target word is NN (noun)]. The proposed method utilizes two types of rulesets: n-gram rulesets that consider n-grams information of words and remaining error rulesets that consider remaining errors in the text. Rules in the N-grams rulesets also need to meet the rule acceptance criterion, which states that rules should be risk-controlled in introducing new errors in the training set.

2.2.2.1 N-grams Rulesets

N-gram rulesets are developed through the contextual information of errors in the training data. This chapter does not differentiate bigram rules from n-gram rules. The authors treated them unanimously as n-gram rules. For example, “If machine tagger tags the word before ‘pedestrian’ as a noun and tags the token ‘pedestrian’ as an adjective, change POS tag of that prior word to adjective” is an n-gram rule. Each N-gram rule represents a context in which a word only has one possible correct POS tag. The context may include the word itself, the machine-assigned POS tag of the word, and machine-assigned POS tags of the word before and after a word.

2.2.2.2 Remaining Error Rulesets

After all n-gram rulesets are applied to the training data, a special ruleset is generated by fixing the n most common errors remaining in the training data. The choice of n is arbitrary. This special ruleset is special because the generation of rules in this set needs information from the entire training dataset whereas the generation of n-gram rulesets only need information from one sentence. The rule of thumb is that the user can choose a larger n when there are more errors in the training dataset compared to when there are less errors. Different values of n can be tested to optimize the performance.

2.2.2.3 Rule Acceptance Test

To reduce the potential negative effects of transformational rules on the downstream tasks of the automated code compliance checking system, n-gram rules should be risk controlled in introducing new errors to the textual data. The rule acceptance test ensures an n-gram rule should be at a low risk in introducing new errors by making sure that transformational rules only replace the POS tag of a word with a more commonly used POS tag of the word in the context described in the rule. If a rule replaces the POS tag of a word from a more commonly used one to a less commonly used one, the rule is prone to introducing new errors and therefore will be dropped. Although the introduction of errors could have negative impact on the downstream tasks of the automated code compliance checking system, it is mathematically true that a rule that fixes more errors than it introduces can increase the level of accuracy. The increase in POS tagging accuracy may enhance the performance of downstream tasks of the automated code compliance checking system and drive the entire system closer to the 100% recall goal. Therefore, if a rule replaces a rarely used POS tag of a word with a commonly used POS tag of the word, the risk of it introducing errors is low. The rule meets the rule acceptance criterion and will be kept in the ruleset. This strict requirement may limit the number of transformational rules generated, but it ensures a steady improvement of the quality of extracted rules and the rulesets' performance. Calculating the accuracy of POS tagging before and after a rule is applied is a possible solution. However, a rule may overfit the training dataset and its ability to increase the accuracy in the training dataset does not necessarily lead to the same effect on the testing text. Instead, selecting a more commonly used POS tag for replacement can leverage syntactic information in the rule generation process to alleviate overfitting.

2.2.3 Rule Generation

The rule generation processes for each ruleset are similar. A general description of the rule generation procedure is shown in

Figure 0.3. For each ruleset, the rule generation component collects contextual information of all errors and their corresponding human-labeled tags in the training dataset. In the second step, this component coverts collected information of each error into candidate rules by deleting unnecessary contextual information. For example, if a rule only considers the POS tag of the

word before the target word, then only the target word, POS tag of the target word, and POS tag of the word before the target word will be kept and everything else in the target word’s context will be deleted.

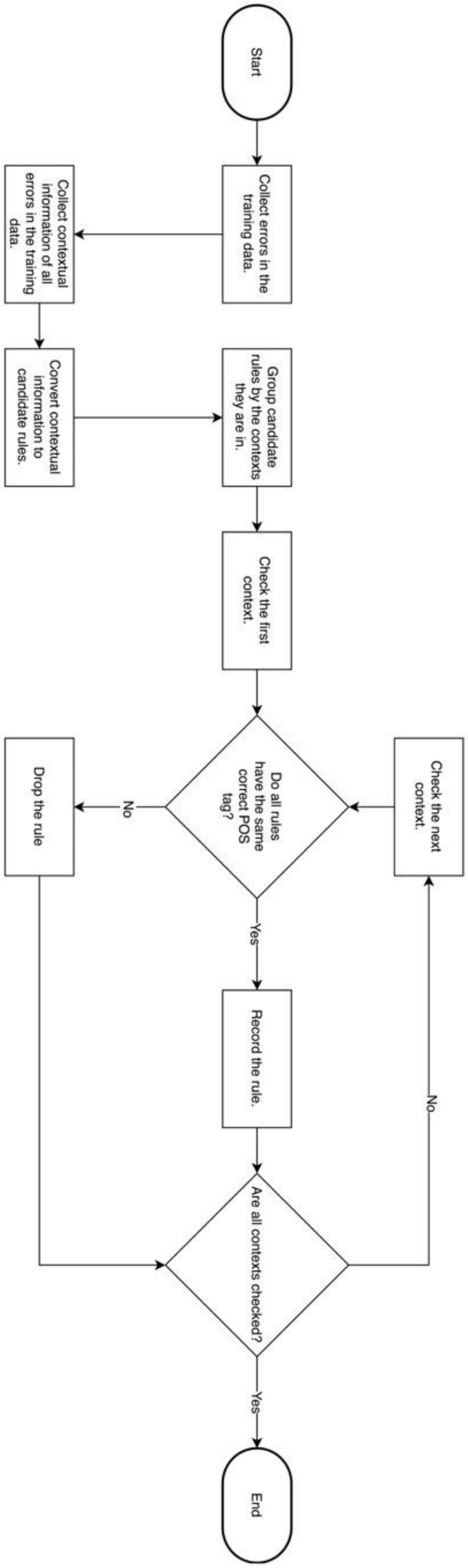

Start
Collect errors in the training data.
Collect contextual information of all errors in the training data.
Convert contextual information to candidate rules.
Group candidate rules by the contexts they are in.
Check the first context.
Do all rules have the same correct POS tag?
No
Drop the rule
Yes
Record the rule.
Are all contexts checked?
No
Check the next context.
Yes
End

Figure 0.3. Rule Generation Process

After that, all candidate rules need to undergo the rule acceptance test. This test clarifies the ambiguities in the textual data. One main challenge in POS tagging is that the same word may have different POS tags in different contexts. This test can ensure that a rule changes the POS tag of the target word to a POS tag that has a low risk to be incorrect.

There are two scenarios that may occur in the generation of rules: (1) a rule replaces a word's POS tag with the word's more commonly used POS tag in the context, or (2) a rule does not replace a word's POS tag with the word's more commonly used POS tag in the context. In the first scenario, this method will generate one candidate rule to fix all occurrences of this type of error. The candidate rule can pass the rule acceptance test and be included. In the second scenario, however, this indicates that the POS tag replacement was inappropriate. The rule acceptance test will prevent such candidate rules from being used, by comparing the POS tag that the rule uses to replace the machine-generated POS tag of a word with the word's commonly used POS tags in the gold standard. There is less risk in the first scenario than in the second scenario. Replacing the machine generated tag of a word with the word's more commonly used POS tag in the gold standard has low risk in introducing new errors. For example, the word "accordance" has the most commonly used POS tag NN and a rarely used POS tag IN in the gold standard and it is more likely to fix an error by replacing the machine generated POS tag with NN than with IN. In the second scenario, however, there is a high risk. For example, replacing the tag of "accordance" to IN is more likely to introduce error than replacing it with NN. This indicates the POS tag in this case is inappropriate and the method will not accept the rule in this scenario. Table 0.2 shows some example sentences and candidate rules with regard to the above discussed scenarios. In this chapter, the authors adopted the widely used Penn Treebank POS tagset, which consists of 36 tags.

Table 0.2. Candidate Rules with High Risk and Low Risk

Scenario	Sentence	Candidate Rule
High Risk	Each portion of a building shall be individually classified *(Manual tag: VBD, Machine tag: VBN)* in *(IN)* accordance with Section 302.1.	If the word that is one position after the word "classified" is tagged as IN and the word "classified" is tagged as VBN, then change the tag of the word "classified" to VBD.
	Handrails within dwelling units are permitted to be interrupted *(Manual tag: VBD, Machine tag: VBN)* by *(IN)* a newel post at a turn or landing.	If the word that is one position after the word "interrupted" is tagged as IN and the word "interrupted" is tagged as VBN, then change the tag of the word "interrupted" to VBD.
Low Risk	The face of an exit sign illuminated *(Manual tag: VBN, Machine tag: VBD)* from (IN) an external source shall have an intensity of not less than 5 footcandles.	If the word that is one position after the word "illuminated" is tagged as IN and the word "illuminated" is tagged as VBD, then change the tag of the word "illuminated" to VBN.
	Clear openings of doorways with swinging *(Manual tag: VBG, Machine tag: JJ)* doors *(NNS)* shall be measured between the face of the door and the stop, with the door open 90 degrees.	If the word that is one position after the word "swinging" is tagged as NNS and the word "swinging" is tagged as JJ, then change the tag of the word "swinging" to VBG.

The decrease in the total number of errors only indicates that a rule solved more errors than it introduced in the training dataset. It cannot ensure that a rule is general enough to have the same effect on the testing text. To alleviate potential overfitting, additional syntactic information about the building codes and English grammar is used in the rule generation. The syntactic information helps prevent adding a rule that is likely to introduce more errors than it fixes.

2.2.4 Rule Application

In the rule application process (Figure 0.4), the rule application component will apply transformational rules to the textual data and fix POS tagging errors. For each rule, the rule application component will search through the entire text and look for words whose contextual information matches that rule's conditions. If a word's contextual information was found to match that rule's conditions, the rule application component will replace the machine-generated tag of that word with the predefined tag in the rule. After the generation of each ruleset, the developed ruleset is applied to the training dataset to prevent the rule application component from developing different rules that essentially fix the same error. After the generation of all rulesets, the rulesets are applied to the testing dataset as a whole.

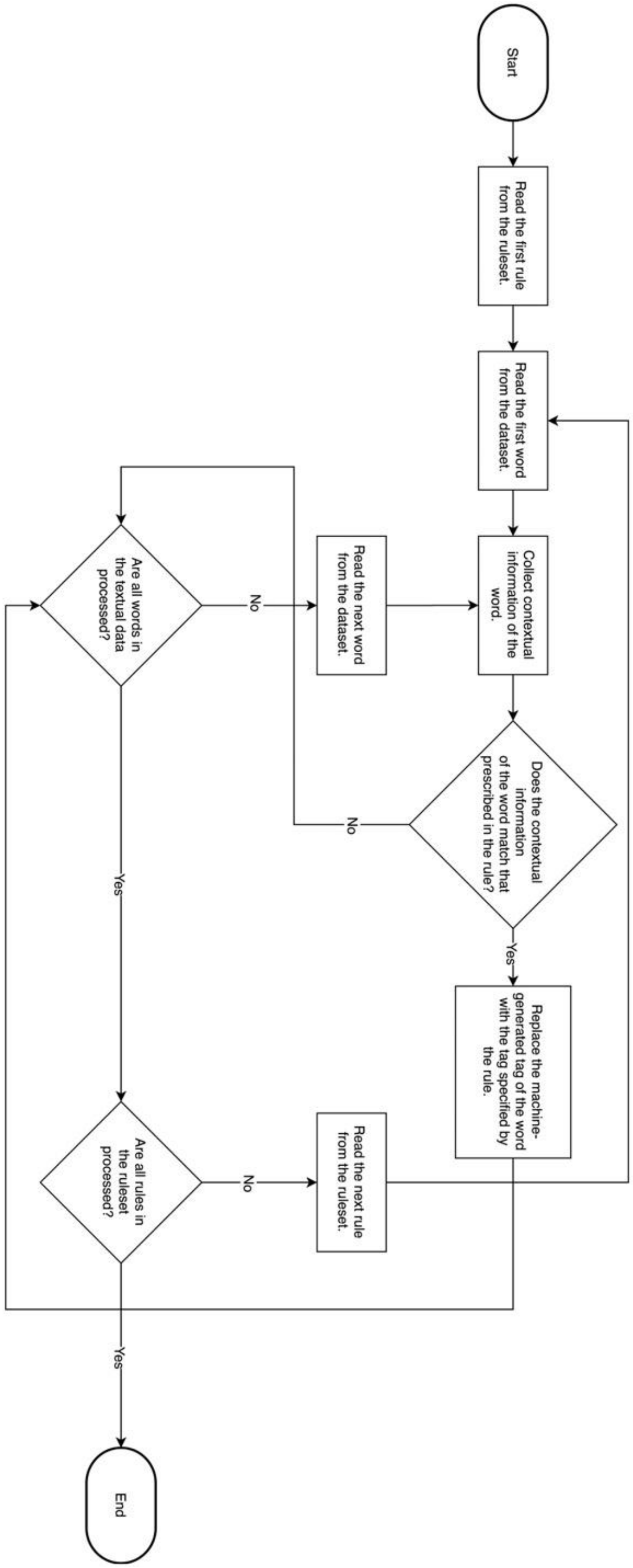

Start
Read the first rule from the ruleset.
Read the first word from the dataset.
Collect contextual information of the word.
Does the contextual information of the word match that prescribed in the rule?
Yes
Replace the machine-generated tag of the word with the tag specified by the rule.
No
Are all words in the textual data processed?
No
Read the next word from the dataset.
Yes
Are all rules in the ruleset processed?
No
Read the next rule from the ruleset.
Yes
End

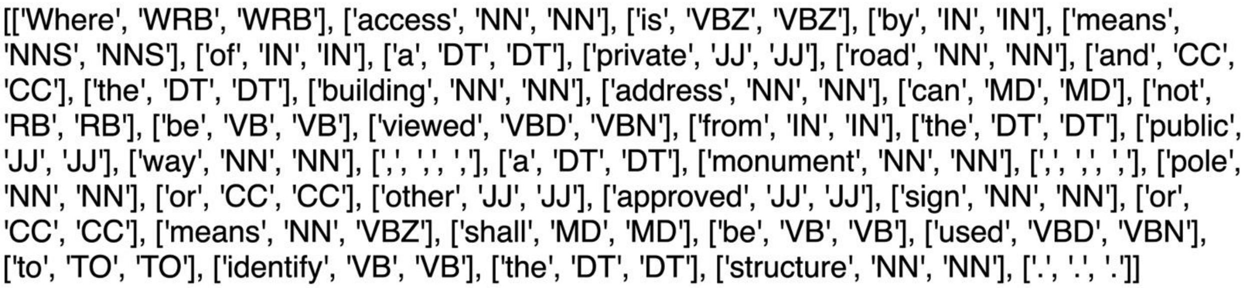
Figure 0.4. Rule Application Process

2.3 Experiment

To test the performance of the proposed method on domain-specific data, the authors applied this method to the POS tagged building codes (PTBC) dataset (Xue & Zhang, 2019). It contains 1,522 POS tagged sentences from Chapters 5 and 10 of the 2015 International Building Codes (IBC). For each tagged sentence, the dataset provides machine-generated and human-labeled POS tags of every token. In the formation of the PTBC dataset, the authors collected textual data by obtaining the Portable Document Format (PDF) version of 2015 IBC and manually extracted building code text from Chapters 5 and 10. A group of seven state-of-the-art machine taggers POS tagged the extracted texts. The seven selected POS taggers were: (1) the NLTK tagger (Loper & Bird, 2002), (2) the spaCy tagger (Explosion AI, 2017), (3) the Standford coreNLP tagger (Manning et al., 2014), (4) A Nearly-New Information Extraction System (ANNIE) tagger in the General Architecture for Text Engineering (GATE) tool (Cunningham, 2002), (5) the Apache OpenNLP tagger (Kottmann et al., 2011), (6) the TreeTagger (Schmid et al., 2007), and (7) the RNNTagger (Schmid, 2019). These taggers were chosen because they have high accuracy, are easy to use, and freely available. The most commonly chosen tag of each word in the extracted text by all the seven taggers formed the machine tagging results. The authors selected the Penn Treebank POS tagset because it was commonly used in various domains for NLP tasks and it is balanced between conciseness and informational richness. Five graduate students labeled textual data without access to others' tagging results. All of them have proficiency in English and building domain knowledge to complete the tagging task, which ensures the quality of the textual data annotation. The mostly commonly chosen tag by them formed the gold standard of POS tagging of the textual data, with an inter-annotator agreement of 0.91.

The PTBC dataset was split into the training data, which contains 80% of the original dataset, and the testing data, which contains the remaining 20% of the original dataset. In the experiment, text is stored in lists of tuples (Figure 0.5). Each sentence is a list of tuples and each tuple in the list stores the word itself, human generated tag of the word, and machine generated tag of the word. In this experiment, the authors used possible combinations of contextual information of mistakenly tagged words in the textual data, to generate templates that rule generation component can use to extract rules. In total, fourteen templates were used in the experiment. They are listed in

Table 0.3. The rule generation component extracted rules in the same order.



```
[['Where', 'WRB', 'WRB'], ['access', 'NN', 'NN'], ['is', 'VBZ', 'VBZ'], ['by', 'IN', 'IN'], ['means',
'NNS', 'NNS'], ['of', 'IN', 'IN'], ['a', 'DT', 'DT'], ['private', 'JJ', 'JJ'], ['road', 'NN', 'NN'], ['and', 'CC',
'CC'], ['the', 'DT', 'DT'], ['building', 'NN', 'NN'], ['address', 'NN', 'NN'], ['can', 'MD', 'MD'], ['not',
'RB', 'RB'], ['be', 'VB', 'VB'], ['viewed', 'VBD', 'VBN'], ['from', 'IN', 'IN'], ['the', 'DT', 'DT'], ['public',
'JJ', 'JJ'], ['way', 'NN', 'NN'], [',', ',', ','], ['a', 'DT', 'DT'], ['monument', 'NN', 'NN'], [',', ',', ','], ['pole',
'NN', 'NN'], ['or', 'CC', 'CC'], ['other', 'JJ', 'JJ'], ['approved', 'JJ', 'JJ'], ['sign', 'NN', 'NN'], ['or',
'CC', 'CC'], ['means', 'NN', 'VBZ'], ['shall', 'MD', 'MD'], ['be', 'VB', 'VB'], ['used', 'VBD', 'VBN'],
['to', 'TO', 'TO'], ['identify', 'VB', 'VB'], ['the', 'DT', 'DT'], ['structure', 'NN', 'NN'], ['.', '.', '.']]
```

Figure 0.5. Textual Data Example

Table 0.3. Transformational Rulesets in the Experiment

Ruleset	Description
1	If the word A is tagged as X, then change the tag X to Y.
2	If the word that is one position before the word A is tagged as X and the word A is tagged as Y, then change the tag of the word A to Z.
3	If the word that is one position after the word A is tagged as X and the word A is tagged as Y, then change the tag of the word A to Z.
4	If the word that is one position before the word A is word B and the word A is tagged as X, then change the tag of the word A to Y.
5	If the word that is one position after the word A is word B and the word A is tagged as X, then change tag of the word A to Y.
6	If the word that is one position after the word A is tagged as X and the tag of the word that is two positions after word A is Y and the word A is tagged as Z, then change the tag of the word to W.
7	If the word that is one position after the word A is tagged as X and the tag of the word that is two positions after word A is Y and the word A is tagged as Z, then change the tag of the word A to W.
8	If the word one position before the word A is B, the word two positions before the word A is C, and the word A is tagged as X, then change the tag of word A to Y.
9	If the word one position after the word A is B, the word two positions after the word A is C, and the word A is tagged as X, then change the tag of the word A to Y.
10	If the tag of the word that is two positions after word A is X and the word is tagged as Y, then change the tag of the word A to Z.
11	If the tag of the word that is two positions before word A is X and the word is tagged as Y, then change the tag of the word A to Z.
12	If the word that is two positions after the word A is B and the word A is tagged as X, then change the tag of the word A to Y.
13	If the word that is two positions before the word A is B and the word A is tagged as X, then change the tag of the word A to Y.
14	Fix five most common errors remaining in the training set.

This method was also tested on the freely accessible portion of the Penn Treebank Corpus in the Natural Language Toolkit (NLTK) to further evaluate the applicability of the proposed method. The authors used the NLTK tagger to tag the text and collected the machine tagging results. Gold standard POS tags of the available text provided by the Penn Treebank Corpus served as the target of transformation. This comparative experiment was conducted in the same way as the previous experiment on PTBC data.

2.4 Results

In total, on the PTBC data, 671 rules were generated in 14 rulesets. All extracted rules, when combined, fixed 2,097 out of 3,013 errors in the training dataset and 656 out of 924 errors in the testing dataset. They increased the tagging accuracy in the training dataset from 90.49% to 97.11% and that in the testing dataset from 89.13% to 96.85%. This 96.85% accuracy in testing dataset is comparable to the performance of the state-of-the-art POS taggers on general English corpus. The first three rulesets, which used contexts represented by: (1) the target word itself, (2) POS tag of the word two positions before the target word, and (3) POS tag of the word two positions after the target word, contained 616 rules (91.80% of all rules). In total, these first three rulesets fixed 2,042 errors (67.77% of errors) in the training dataset and 553 errors (59.85% of errors) in the testing dataset.

Accuracy of POS tagging both in the training dataset and in the testing dataset increased after application of the transformation rules. Before application of any transformational rules, the training dataset had an accuracy of 90.49% and the testing dataset had an accuracy of 89.13%. After all rulesets were applied, the training dataset achieved an accuracy of 97.11% and the testing dataset achieved an accuracy of 96.85%. The overall reduction of errors in the training set was 69.60% and that in the testing set was 71.00%. The most significant increase in accuracy happened after the application of the first and second rulesets. After the first ruleset was applied, accuracy in the training dataset increased from 90.49% to 95.97% and that in the testing dataset increased from 89.13% to 93.90%. After the second ruleset was applied, accuracy in the training dataset increased from 95.97% to 96.61% and that in the testing dataset increased from 93.90% to 95.20%. The number of errors and accuracy after application of each ruleset is provided in Table 0.4.

Table 0.4. POS Tagging Accuracy After Applying Each Ruleset

Ruleset	Training Dataset		Testing Dataset	
	Number of Errors	Accuracy	Number of Errors	Accuracy
1	1277	95.97%	518	93.90%
2	1073	96.61%	408	95.20%
3	971	96.93%	371	95.63%
4	928	97.07%	355	95.82%
5	926	97.08%	355	95.82%
6	918	97.10%	355	95.82%
7	918	97.10%	355	95.82%
8	914	97.11%	353	95.85%
9	902	97.15%	347	95.92%
10	899	97.16%	347	95.92%
11	899	97.16%	347	95.92%
12	899	97.16%	347	95.92%
13	899	97.16%	347	95.92%
14	916	97.11%	268	96.85%

The authors recorded the number of errors each rule fixed to evaluate effectiveness of the generated rules. Ten rules that fixed the most errors fixed 30.47% errors in the training dataset and 23.70% errors in the testing dataset, respectively. This distribution confirms the authors' prediction that a small group of rules can fix a large number of errors. Eight out of ten most frequently applied rules in the training dataset are unigram rules, and that in the testing dataset is also eight out of ten. This distribution shows that simple rules are more frequently applied than complex rules. It may not be necessary to generate over-complex rules for increasing POS tagging accuracy.

In the development of this method, the authors attempted to lemmatize word in text before the generation of transformational rules. The authors assumed that mapping multiple words to their common lemmatized form would improve the coverage of error cases. However, this generalization did not improve the performance and therefore the authors abandoned this technique. Word lemmatization didn't change the number of extracted rules in all rulesets. It is possible that mapping multiple forms of a word to one may have harmed the diversity of contextual information representation. The authors concluded that word lemmatization did not bring benefit to the proposed method.

This chapter also included a comparative study that applied the proposed method to improving NLTK POS tagger's performance on Penn Treebank Corpus. This cross-comparison provides a useful benchmark for other researchers to compare this method's performance on general English text. In the processing of the Penn Treebank Corpus, the authors noticed that a non-negligible amount of words in Penn Treebank Corpus, which do not belong to any Penn Treebank POS tagset, were tagged as '-none-'. Pre-processing Penn Treebank Corpus is a possible way to eliminate this '-none-' tag. However, solving this issue is out of the scope of this chapter. The authors decided to use the Penn Treebank Corpus and the NLTK tagging results in this method as is. The authors divided the Penn Treebank Corpus into a training dataset and a testing dataset with an 80/20 split. NLTK tagger tagged 89.28% of words in the training dataset correctly and 89.37% of words in the testing dataset correctly. The proposed method then increased the accuracy of NLTK tagger to 91.58% on the training dataset and to 91.91% in the testing dataset. This increase in accuracy indicates that the proposed method has the ability of improving the POS tagging accuracy of general English text as well (in addition to building codes).

2.5 Discussion

Comparing to previous rule-based POS tagger that used hand crafted rules (Bird, 2009), the proposed error-driven transformational rules are automatically generated by algorithms. Previous rule-based POS tagger that used automated generated rules is not for domain-specific text (Brill, 1992). The error-driven transformational rules are applicable to domain-specific text, such as building codes. Existing machine learning POS taggers require a significant amount of training data (Giménez, 2004; Brants, 2000). The proposed error-driven transformational rules can be trained on a limited amount of training data.

Due to the specific type of texts covered in the chapter, the authors suggest that error-driven transformational rules should only be applied to texts that are in the target domain. A major potential risk is that transformational rules may introduce errors to the tagging results. This risk is controlled by the rule acceptance test. This constraint can push the machine labeled result unidirectionally to the human labeled result.

Research interests of the authors require the use of the PTBC dataset (Xue & Zhang, 2020), which is a new dataset and not used by other research currently. This method may overfit this particular dataset and lacks the ability to boost tagging accuracy of POS taggers, which are trained on general English, on general English. The authors conducted a comparative study to address this concern. Specifically, this method was used to boost the performance of Natural Language Toolkit (NLTK) tagger on the part of Penn Treebank Corpus that were readily available in NLTK (Loper & Bird, 2002).

This method does not address unknown words. It requires a word to exist in the training dataset to generate transformational rules for it. This limitation, however, should not significantly influence the performance of the error-driven transformational rules, because generated rules are only to be applied to the text in the target domain (e.g., building codes), in which the rate of unknow words is expected to be low. The stringent format of the error-driven transformational rules in the proposed method, while effectively induced rules to improve POS tagging results, may introduce counter-intuitive tagging results. To alleviate that, future work may look into different representations of the fixes (e.g., tokens' roles) in addition to their original POS tags. In addition, the authors only tested the method on the commonly adopted Penn Tree bank tag set, how this method will perform when using other tag set will need to be investigated in the future work.

2.6 Contributions to the Body of Knowledge

This chapter research was conducted to present a new way to obtain domain-specific English texts POS tagged accurately when there is no POS tagger trained on text in that domain by error-driven transformational rules. The proposed method can alleviate problems such as, (1) the lack of POS taggers that are trained on domain-specific English texts, (2) the performance drops of general POS taggers on domain-specific texts, and (3) the high cost of developing a large domain-specific corpus needed in training domain-specific POS taggers.

First and foremost, this method provides a possible way for future researchers to get reliable POS tagged text in a selected domain without the need of a specialized POS tagger. The authors discovered that simple unigram and bigram rules resolved most errors. Word lemmatization did not bring observable benefit to this method. For future application of this method, development time could be saved by avoiding over-complicated rulesets and word lemmatization.

Secondly, this chapter research was conducted to prove that it is possible to boost the performance of POS taggers that are trained on general English texts on domain-specific English texts with a small set of algorithmically generated rules. The authors used building codes as an example. These rules can increase the accuracy of POS taggers on building codes from 89.13% to 96.85% with 671 rules. This significant improvement is achieved by using a small set of labeled data. The fact that all rulesets transform machine-generated POS tags of words unidirectionally to their human-annotated tags proved the validity of the rule acceptance criterion. In addition, the increase in the accuracy in the testing dataset after the application of the last ruleset supports its exemption from the rule acceptance criterion.

Thirdly, the rules generated in this chapter research can be used to increase the accuracy of POS tagging results on building codes. If interested researchers use one of the POS taggers tested, they can directly apply the developed rulesets to improve the POS tagging results/performance on building codes. The potential risk of introducing more errors were alleviated by the constraint applied when the rules were derived. This method does not need experts to generate new rules to be adapted to new domains, but it needs experts to annotate some training data as gold standard. Last but not least, this method is also applicable to general English. With a small amount of human-labeled data, it can boost the accuracy of POS taggers that are trained on general English, on general English.

2.7 Conclusion

This chapter research presented a new method to increase the accuracy of POS taggers, that were trained on general English texts, on building codes by using error-driven transformational rules. The authors developed an algorithm to generate these rules and tested the algorithm on the PTBC dataset (Xue & Zhang, 2019). The experiment shows this method can increase the POS tagging accuracy on building codes from 89.13% to 96.85%. A comparative test on NLTK and Penn Treebank Corpus shows that the proposed method can also increase the POS tagging accuracy on general English texts.

2.8 Acknowledgement

The authors would like to thank the National Science Foundation (NSF). The material in this chapter is based on work supported by the NSF under Grant No. 1827733. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

3 PART-OF-SPEECH TAGGING OF BUILDING CODES EMPOWERED BY DEEP LEARNING AND TRANSFORMATIONAL RULES

Xiaorui Xue, S.M.ASCE[1]; Jiansong Zhang, Ph.D., A.M.ASCE[2]

A portion of this chapter was previously published by:

Xue, X., Zhang, J. (2021). Part-of-speech tagging of building codes empowered by deep learning and transformational rules. *Advanced Engineering Informatics*, *47*(January 2021), 101235. https://doi.org/10.1016/j.aei.2020.101235

Author Contributions

The authors confirmed contribution to the paper as follows:

Study conception and design: Xiaorui Xue, Jiansong Zhang.

Data collection: Xiaorui Xue, Jiansong Zhang.

Analysis and interpretation of results: Xiaorui Xue, Jiansong Zhang.

Draft manuscript preparation: Xiaorui Xue, Jiansong Zhang.

All authors reviewed the results and approved the final version of the manuscript.

3.1 Literature Review

3.1.1 Part-of-Speech (POS)

A word’s POS category provides its syntactic information in a sentence (Abzianidze & Bos, 2017). In English, there are eight main POS categories: (1) noun, (2) verb, (3) adjective, (4) adverb, (5) pronoun, (6) preposition, (7) conjunction, and (8) interjection. POS taggers are systems that automatically assign POS categories to words according to their contextual information in a sentence (Schmid, 1994). POS taggers have a variety of applications in the AEC domain. For example, Lee et al. (2020) POS tagged construction contracts to identify missed contract conditions from the perspective of contractors. However, the reliance on manual feature extraction and manual rule generation creates challenges in large scale applications. Hassan and Le (2020) used POS tagging to spot contractual requirements from construction contract documents. However, the Support Vector Machines (SVM) algorithm used to identify contractual requirements relies on manual feature engineering and may raise the concern of overfitting. Zhou and El-Gohary (2018) utilized POS tagging information to match design requirements in energy codes to their corresponding objects in building information models (BIMs). The matching process takes a four-step approach: First, POS tagging information and other contextual information of design requirements and BIM objects are collected; Second, the Word2vec algorithm calculates the vectors of BIM objects and design requirements; Third, vector similarity algorithm calculates the vector similarity between BIM objects and design requirements; Fourth, a match is claimed if the vector similarity between a BIM object and a design requirement is higher than a predefined threshold, which was set arbitrarily to obtain the highest precision and recall empirically. In this four-step approach, errors could accumulate in each step, and the concern of overfitting also presents. Therefore, the authors suggest an end-to-end method that does not rely on manually generated rules or features. Neural network models could meet the above requirements (Wang et al., 2019).

A simple deep learning model without man-made task-specific features can outperform most state-of-the-art non-deep learning models even with cherry-picked features, in a wide range of NLP tasks such as part-of-speech tagging, chunking, named entity recognition, and semantic role labeling (Collobert et al., 2011). For example, Marques and Lopes (2001) utilized a simple feed-forward model to decrease the amount of data needed to train a POS tagger. Yu et al. (2017) used two Convolutional Neural Network (CNN) models to capture morphological information of character-level n-grams and contextual information of word-level n-grams, which outperformed simple feed-forward model. Recent developments in deep learning indicated that RNN is the “to-go” solution for NLP tasks (Chollet, 2017). Pre-trained models were pre-trained on a large body of text with unsupervised tasks, such as, predicting the next word given all preceding words and predict if two sentences are from the same article (Devlin et al., 2018). The use of general pre-trained models helped boost the performance of domain-specific NLP tasks in biology (Lee et al., 2020), finance, and law (He et al., 2020). It also reduced the amount of labeled data needed when applying deep learning in domain-specific tasks (Tai et al., 2020; Xue et al., 2022).

3.1.2 Error-driven Transformational Rules

Error-driven transformational rules are introduced to boost POS taggers' accuracy (Raghavan et al., 2010; Xue & Zhang 2020). The rules are designed to transform the machine-generated POS tag of a word to its human-labeled gold standard. When the rule generation algorithm spots a difference between machine-generated POS tags and the human-labeled gold standard, it records the difference as an error and uses the context of the error (i.e., words and POS tags of words around the word) to generate a rule to fix the error. The generation of rules is automated. Rules are reusable once generated. Rules may have the risk to introduce new errors. The rule generation algorithm controls this risk by dropping rules that have a high risk of introducing errors.

3.1.3 Recurrent Neural Network

Like any machine learning models including classic ones such as Naïve Bayes, Decision Tree, Support Vector Machines (SVMs), Random Forest (Cao et al., 2020; Wu et al., 2022; Zhang et al., 2016), neural networks predict categories of given inputs. In the context of POS tagging, neural networks predict POS categories of each word in a given input text, according to the word itself and its context (Figure 0.1). Neural networks learn a relationship between words and POS tags during their training and use this relationship to predict POS tags of words during their application. Traditional neural networks consider all words in a sentence to be independent from each other and do not consider words surrounding them in this prediction task. In contrast, Recurrent Neural Network (RNN) keeps a vector that represents other words in the sentence (which is called hidden state) and considers them in the prediction task. RNN processes sequential information by taking elements in the sequence one by one while maintaining a representation of all information it has seen so far (Chollet, 2017). RNN is able to process sentences with arbitrary length (Tang et al., 2015). The way that RNN processes sequential information gives it the ability to capture semantic meaning of a word based on words before/after it in the sentences (Young et al., 2018). For example, it is able to differentiate the meaning of the word "bank" in the phrase "river bank" and "blood bank". The sequential nature of RNN makes it widely adopted in many subfields of NLP, such as: (1) information extraction (Bhutani et al., 2019; Rao & Ke 2018), (2) machine translation (Barone et al., 2017; Vaswani et al., 2018), (3) speech recognition (Chan et al., 2016; Karita et al., 2019), (4) POS tagging (Plank et al., 2016; Shao et al., 2017), and (5) sentiment analysis (Agarwal et al., 2019; Baktha & Tripathy, 2017). There is also an RNN encoder-decoder model which has a high accuracy in sequence-to-sequence tasks (Cho et al., 2014). In this structure, the encoder is an RNN model that converts a variable-length sequence to a fixed-length vector representation and the decoder is another RNN model that converts the fixed-length representation to a variable-length sequence. Neural network models are deterministic when applied (i.e., in making predictions). One neural network model makes the same prediction result with the same input.

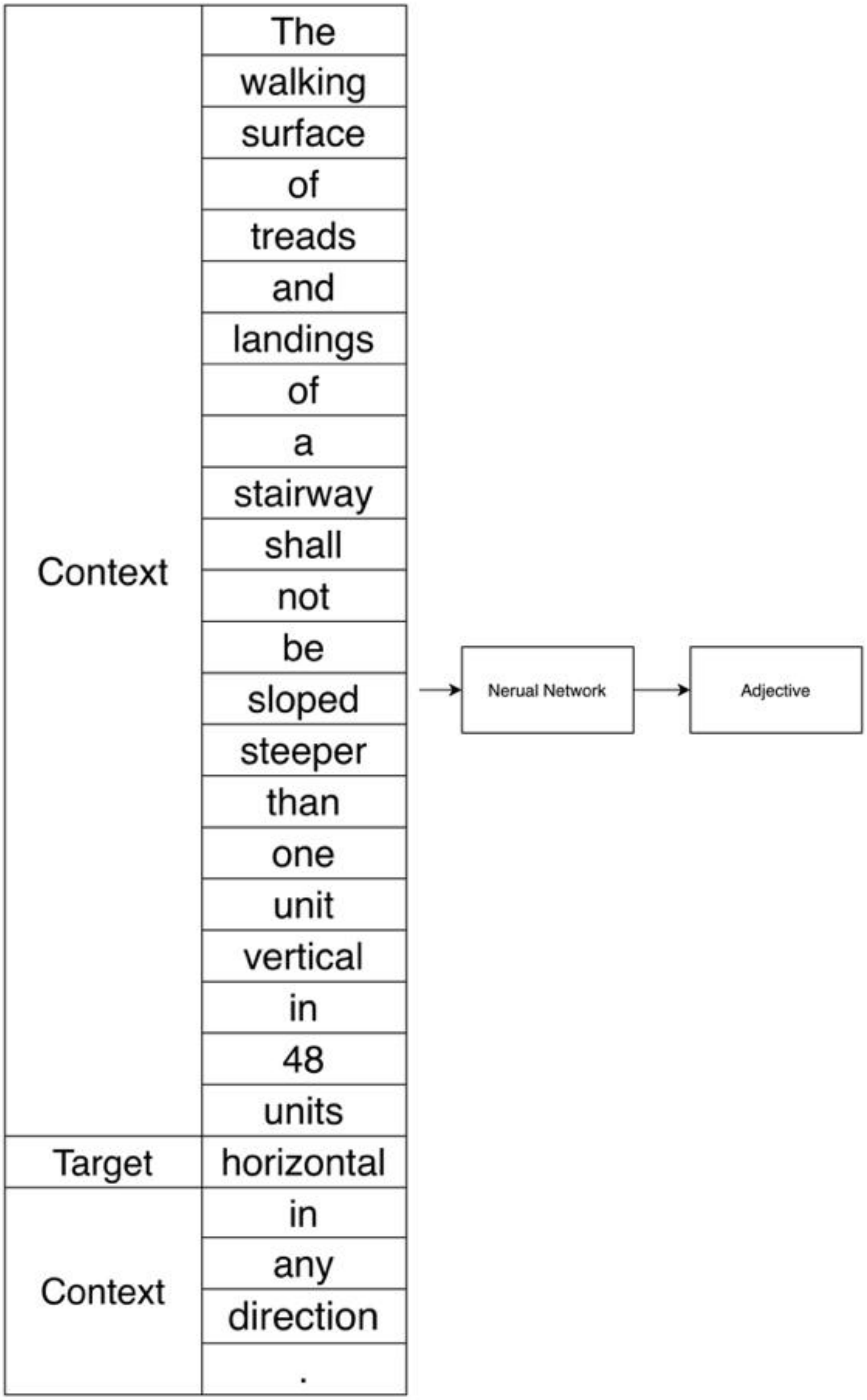


Figure 0.1. Example Application of a Neural Network POS Tagger

3.1.3.1 Simple RNN

A simple RNN keeps a hidden state that represents all previous words in the sentence. Therefore, the hidden state allows the simple RNN to take into consideration all words before the target word in POS tagging. A simple RNN contains an input layer *x*, a hidden layer h, and an output layer *y* (Elman, 1990). The hidden layer has weight W_h and a bias vector b_h. The input layer has a weight W_i. The output layer has a weight W_o and a bias vector b_o. In time step *t* of the training, the input to the RNN is denoted as x_t, the hidden state is denoted as h_t, and the output is denoted as Y_t. The hidden state at the time step *t* (i. e., h_t) is the sum of: (a) the input of current step x_t multiples the weight of the input layer W_i, (b) the hidden state of the last time step h_{t-1} multiplies its weight W_h, and (c) the bias vector of hidden layers b_h, after some non-linear transformation [Eq. (1)].

$$h_t = f(W_i x_t + \; W_h h_{t-1} + b_h) \tag{1}$$

The output at the time step *t* ($i.e., Y_t$) is the sum of: (a) the weights of output layer W_o multiples the hidden state at this time step h_t, and (b) the bias vector of output layer b_o [Eq. (2)].

$$Y_t = g(W_o h_t + b_o) \tag{2}$$

In Eqs. (1) and (2), *f* and g are activation functions that perform non-linear transformations. Some commonly used activation functions include sigmoid, Tanh, and Rectified Linear Unit (ReLU) (Glorot et al., 2011; Nwankpa et al., 2018).

Simple RNN suffers from the vanishing gradient problem (Hochreiter, 1998). The hidden state of a word is influenced more by words near it than words far away. In other words, simple RNN does not have a “long-term memory”. This problem makes simple RNN difficult to train and hard to capture long-term dependencies in a sentence. The long-term dependencies between words are important in POS tagging. Many variations of simple RNN were therefore developed to solve this problem.

3.1.3.2 Long Short-Term Memory

Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) alleviates the vanishing gradient problem by having a forget gate layer to decide which words to “remember” and which words to “forget”. It has a cell state to keep long-term dependencies, so it has “long-term memory”. The cell state allows LSTM-RNN to use long-term dependencies in POS tagging.

LSTM-RNN has an additional forget gate layer f to decide which information to keep or abandon, and a cell state C to capture long-term dependencies (Sak et al., 2014). The weight of the forget gate layer is W_f and its bias vector is b_f. The cell state has a weight W_C and a bias vector b_C. LSTM-RNN also has an input layer x. The input layer has a weight W_i and a bias vector b_i. The output layer has a weight W_o and a bias vector b_o. In time step t of the training, the input to the RNN is denoted as x_t, the hidden state is denoted as h_t, the output is denoted as Y_t, and the cell state is denoted as C_t, the value to update is denoted as i_t. Input to the neural network is first fed into the forget gate layer. The forget gate layer generates a vector f_t to represent the amount of information to keep, and f_t is calculated by Eq. (3):

$$f_t = \sigma\left(W_f * [h_{t-1}, x_t] + \ b_f\right) \tag{3}$$

where σ is the sigmoid function.

Then, the input layer calculates the candidate cell state by Eq. (4) and Eq. (5):

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \tag{4}$$

$$\widetilde{C_t} = \tanh(W_C * [h_{t-1}, x_t] + b_C) \tag{5}$$

Then, the cell state C_t is calculated by Eq. (6):

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \tag{6}$$

After that, the output layer Y_t and hidden state h_t are calculated by Eq. (7) and Eq. (8), respectively:

$$Y_t = \ \sigma(W_o * [h_{t-1}, x_t] + \ b_o) \tag{7}$$

$$h_t = Y_t * \tanh(C_t) \tag{8}$$

There is also a bi-directional variant of LSTM, which can capture information in a sequence from both directions. Simple RNN and LSTM-RNN have one hidden state that represents all words before the target word. Bi-directional LSTM-RNN additionally has an extra hidden state that represents all words after the target word. Therefore, simple RNN and LSTM RNN predict the POS tag of the target word solely by words before it, whereas bi-directional LSTM RNN predicts POS tag of the target word by the words both before and after it.

3.1.3.3 Gated Recurrent Unit

Gated Recurrent Unit (GRU) (Chung et al., 2014) is another way to address the vanishing gradient problem. It does not have a forget gate to control the flow of information, so it can access the entire hidden state. It has an update gate U and a reset gate R. The weight of the update gate is W_U, the weight of the reset gate is W_R, and the weight of the output layer is W_o. At time step *t*, the cell state of the update gate, reset state, and the hidden state are U_t, R_t, and h_t, respectively. GRU is calculated using Eqs. (9), (10), (11), and (12):

$$U_t = \sigma\left(W_U * X_t + W_{U,t-1} * h_{t-1}\right) \quad (9)$$

$$R_t = \sigma\left(W_R * X_t + W_{R,t-1} * h_{t-1}\right) \quad (10)$$

$$h'_t = tanh\left(W_o + R_t * W_{U,t-1} * h_{t-1}\right) \quad (11)$$

$$h_t = U_t * h_{t-1} + (1 - U_t) * h'_t \quad (12)$$

GRU can take long-term dependencies of words into the POS tagging task by accessing hidden states of every word in a sentence. There is also a bi-directional variant of GRU, which can use words both before and after a target word to predict its POS category.

3.1.3.4 Attention Mechanism

Attention mechanism can capture long-term dependencies with arbitrary lengths by calculating attention scores between all words in two sequences and feed the attention scores to a RNN (Hu, 2019). Therefore, it does not suffer from the vanishing gradient problem. LSTM RNN and GRU still suffer from the vanishing gradient problem when the dependencies are long enough. The attention mechanism predicts the POS tag of a word with its long-term dependencies. Attention mechanism shares the same encoder-decoder structure with the encoder-decoder RNN. The structure of attention mechanism brings its successful application in many sequence-to-sequence (Seq2Seq) tasks such as: (1) machine translation (Firat et al., 2016), (2) question-and-answering (Lu et al., 2016), and (3) text entailment (Rocktäschel et al., 2015). The attention mechanism allows the decoder to access hidden states of the encoder to track back the input sequence (Bahdanau et al., 2015). There are many variants of attention mechanisms. For example, global attention focuses on all words in the input including each target word, while local attention only focuses on words in a certain range (Luong et al., 2015). Two-way attention allows bi-directional attention between the source and target (Rocktäschel et al., 2015). This property of two-way attention makes it successful in non-sequence-to-sequence tasks as well, such as sentiment analysis (Ambartsoumian & Popowich, 2018).

3.1.3.5 Transformer

Transformer has a similar encoder-decoder structure as the attention mechanism, but it does not have an RNN (Vaswani et al., 2017). Transformer, like attention mechanism, can capture dependencies in any length. With fewer parameters than the attention mechanism, it is more resistant to overfitting. Therefore, transformer can make POS taggers more generalizable. The encoder and decoder of the transformer are stacks of multi-head attention layers and feed-forward layers with some add-and-normal layers. The multi-head attention is the concatenation of multiple self-attention matrices. The multi-head attention is used to capture different dependencies in a sentence. The first step to calculate the self-attention *Z* is to calculate: the Query *Q*, Key *K*, and Value *V* matrices with the embedding matrix *X*, the weight of Query W_Q, the weight of Key W_k, and the weight of Value W_V [Eqs. (13) to (15)].

$$Q = X * W_Q \tag{13}$$

$$K = X * W_k \tag{14}$$

$$V = X * W_V \tag{15}$$

Then, the self-attention matrix, or one head of the multi-head attention, is calculated by Eq. (16):

$$Z = softmax\left(\frac{Q * K^T}{\sqrt{d_k}}\right) * V \tag{16}$$

where d_k is the dimension of Key.

After that, multiple self-attention matrices are concatenated together to form a multi-head attention matrix Z_{multi} [Eq. (17)]. The multi-head attention is then multiplied to a weight matrix W_o to get a new attention matrix Z_{new} that captures information from all attention heads [Eq. (18)]. W_o is trained with the matrix Z_{multi}.

$$Z_{multi} = [Z_i, \dots Z_n] \tag{17}$$

$$Z_{new} = W_o * Z_{multi} \tag{18}$$

3.1.3.6 BERT

Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) is a language representation model of the transformer. This model was pre-trained on the BooksCorpus (Zhu et al., 2015) and the English Wikipedia data. Through pre-training, BERT introduces knowledge about general English into the POS tagger. Knowledge about general English is helpful to increase the POS tagger's performance on building codes, because these building codes are written in English. BERT is trained to predict masked words in a sentence and decide if the second sentence in a pair of sentences is actually the sentence after the selected sentence in the training text or just a randomly selected sentence. The BERT model achieved the state-of-the-art performance in 11 NLP tasks with fine-tuning. Information of the different available versions of BERT is provided in

Table 0.1. Large models have more layers, larger hidden states, more heads, and more parameters than base models. The fine-tuning of pre-trained models allows the neural network model to reach high accuracy on a small dataset (Zhang et al., 2021).

Table 0.1. Available Versions of BERT

Cased	Size	Number of Layers	Size of Hidden State	Number of Heads	Number of Parameters	Comments
Uncased	Large	24	1024	16	340M	Mask the same word.
Cased	Large	24	1024	16	340M	Mask the same word.
Uncased	Base	12	768	12	110M	
Uncased	Large	24	1024	16	340M	
Cased	Base	12	768	12	110M	
Cased	Large	24	1024	16	340M	
Cased	Base	12	768	12	110M	Trained on 104 Languages
Uncased	Base	12	768	12	110M	Trained on 102 Languages
N/A	Base	12	768	12	110M	Trained on Chinese

3.2 Methodology

To develop a POS tagger tailored to building codes, the authors combined the use of multiple state-of-the-art techniques such as error-driven transformational rules, recurrent neural networks, dropout layers, and pretrained models. At the core, the proposed POS tagger has two main components, a neural network model and a set of error-driven transformational rules. The neural network model initially predicts the POS tag of a word. The error-driven transformational rules fix errors made by the neural network model. The neural network model has a pre-trained model and multiple trainable layers (i.e., bi-directional LSTM-RNN layer, GRU layer, dropout layer, and TimeDistribute layer). The pre-trained model brings the general linguistic knowledge (i.e., English grammar) into the POS tagger. The authors fine-tuned the pre-trained model on a dataset of building codes to customize the pre-trained model with AEC domain knowledge. The bi-directional LSTM-RNN layer and GRU layer capture task-specific information (i.e., how building codes were drafted, and AEC terminologies). The dropout layer alleviates overfitting. The TimeDistribute layer Outputs the result. A POS tagger search strategy was proposed in this chapter's research to efficiently search for a well-performing POS tagger configuration.

3.2.1 POS Tagger Architecture

The architecture of the proposed POS tagger is shown in Figure 0.2, which illustrates: (1) an overview of the POS tagger components, and (2) how information flows between components. The inputted building codes are firstly tagged by the neural network model and afterwards processed by the error-driven transformational rules to fix errors made by the neural network model. The neural network model has two parts, a pre-trained model and additional trainable layers. The pre-trained model uses existing models published by other researchers or commercial/non-profit organizations. These were trained on large bodies of corpus. Many widely used pre-trained models can be inserted here such as Open AI GPT-2 (Radford et al., 2019), BERT (Devlin et al., 2018), and ELMO (Peters et al., 2018). This design allows the comparison between different pre-trained models in this context and the selection of the best-performing model. Weights of the pre-trained model were locked, which made them untrainable in the current context. The untrainable nature of the pre-trained models preserves the cross-domain, cross-application and cross-task information they collected in the original training process. On top of the pre-trained models, there are trainable layers. Weights of trainable layers will be updated in the training process, allowing trainable layers to capture the domain-specific, application-specific, and task-specific information in building code POS tagging. The architecture of this model allows substitution and therefore comparison between different types of neural network layers. The error-driven transformational rules are designed to correct errors of a neural network model.

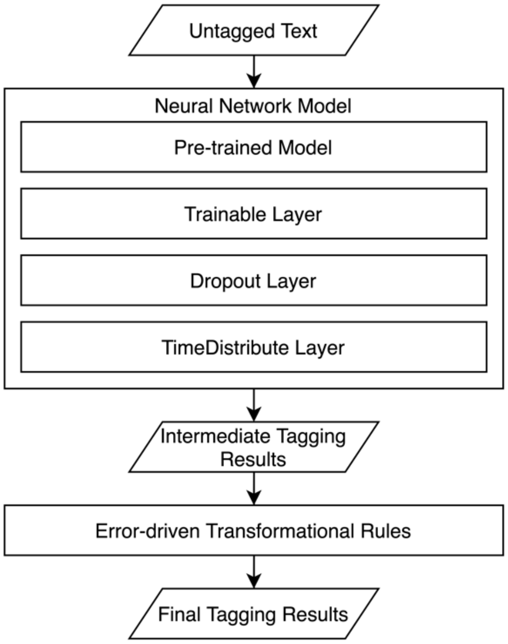


Figure 0.2. The Architecture of the Proposed POS Tagger

3.2.2 POS Tagger Search Strategy

Grid search is the most comprehensive way to find the optimal combination of pre-trained models, trainable layers and the number of training epochs by exhaustively searching every possible combination. A global grid search is inefficient, however, because many combinations that are unlikely optimal will be attempted. The authors developed a three-step searching strategy (Figure 0.3) that can reduce the time in finding the optimal combination by ruling out combinations that have low probabilities of being optimal. The first step of this search strategy is finding the best performing combination of epochs of training and trainable layers by attempting all possible combinations of them while replacing the pre-trained model with a random number embedding layer. Because the pre-trained model has been replaced with a random number embedding layer to save training time, grid search is made possible and efficient. An embedding layer converts text strings to vectors of numbers based on the context of the text string and the nature of the embedding layer (e.g., the algorithm used in the layer and the size of the output vector). The pre-trained models will be used to instantiate the embedding layer later in the proposed method. A random number embedding layer is a type of embedding layer that directly maps words to vectors of the random numbers without considering the words' context. It is much smaller and simpler than the pre-trained models and requires significantly less time to train. In this step, the authors intend to find a well performing combination of epochs of training and trainable layers in a short timeframe, so the random number embedding layer is used to help achieve that. In the second step, the random number embedding layer is substituted with different pre-trained models in the locally best-performing combination of number of epochs and trainable layers that was identified in the first step. This step is aimed to find a well performing pre-trained model. In the last step, the authors increase the number of trainable layers until the accuracy of the POS tagger stops increasing, to identify the optimal number of trainable layers. The selection of the hyper-parameters ceases when the authors cannot increase the performance of the model further in a meaningful way or if the performance is already satisfactory.

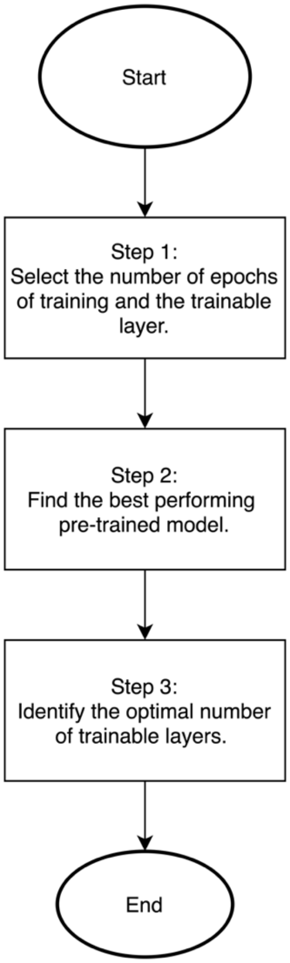

Start
Step 1:
Select the number of epochs of training and the trainable layer.
Step 2:
Find the best performing pre-trained model.
Step 3:
Identify the optimal number of trainable layers.
End

Figure 0.3. The Three-step Approach for Efficient Grid Search

3.3 Experiment

3.3.1 Textual Data

The proposed POS tagger was trained on the POS tagged building codes (PTBC) dataset (Xue & Zhang, 2019), a dataset that consists of 1,522 POS tagged sentences in Chapters 5 and 10 of the 2015 International Building Code (IBC). In total, the PTBC dataset has 39,875 tokens. A token is the smallest unit in POS tagging, such as a word or a punctuation. For example, the word “means” and the period are two tokens in the sentence “The means of egress shall have a ceiling height of not less than 7 feet 6 inches.”, which has 18 tokens in total. The split of the dataset into training, validation, and testing data is shown in Figure 0.4: 40% of the dataset as training data, 10% of the dataset as validation data, and 50% of the dataset as testing data. Furthermore, the first 90% of the testing data was further used as the training data of the error-driven transformation rules, which was then tested on the rest of the data. Seven state-of-the-art machine taggers were used to tag the textual data, including: (1) the NLTK tagger (Loper & Bird, 2002), (2) the spaCy tagger (Explosion AI, 2017), (3) the Standford coreNLP tagger (Manning et al., 2014), (4) A Nearly-New Information Extraction System (ANNIE) tagger in the General Architecture for Text Engineering (GATE) tool (Cunningham, 2002), (5) the Apache OpenNLP tagger (Kottmann et al., 2011), (6) the TreeTagger (Schmid, 1994), and (7) the RNNTagger (Schmid, 2019; Schmid, 1994). The seven machine taggers were selected because of their high-accuracy, ease of use, and free availability. The most commonly chosen POS tag of words by the machine taggers formed the machine-tagged result. Five human annotators then independently POS tagged the textual data and the most commonly seen tag was chosen for each word. All human annotators are proficient in English and have sufficient background knowledge to understand building codes. POS tags of words by the human annotators formed the gold standard. In both the machine-tagged result and the gold standard, the most commonly chosen POS tag is selected by the highest count, meaning that the POS tag that is selected by the most machine taggers or human annotators is selected. For example, if four machine taggers tag the word “doorways” as Plural Noun (NNS), one machine tagger tags the word as 3rd person singular present verb (VBZ). The most commonly chosen POS tag of the word “doorways” is selected to be Plural Noun (NNS), in the machine-tagged result. If there is a tie, the authors break the tie by selecting the tag deemed most appropriate. In the generation of the gold standard, the authors developed a new labeling method in which human annotators address the differences between tagging results of different machine taggers. If all machine taggers tag a word identically, human annotators do not need to change the tag by machine taggers. For words that different machine taggers select different POS tags, human annotators are presented with all tags assigned by machine taggers as options to select from. To account for the risk that a word is not correctly tagged by any machine taggers, human annotators are allowed to assign a POS tag outside the provided tags as well. Human annotators also can change the POS tag of words that machine taggers reached a consensus on. Such changes will need to be discussed and get consensus from all human annotators (Xue & Zhang, 2020). The human annotators' tagging results reached an initial inter-annotator agreement of 0.91, which ensured the quality of the gold standard. The dataset contains the POS tags given by all seven machine POS taggers and five human annotators, the most commonly chosen tag by machine POS taggers and human annotators. In this experiment, the proposed POS tagger was trained to tag the textual data as closely as possible to the most commonly chosen tag by human annotators (Figure 0.5).

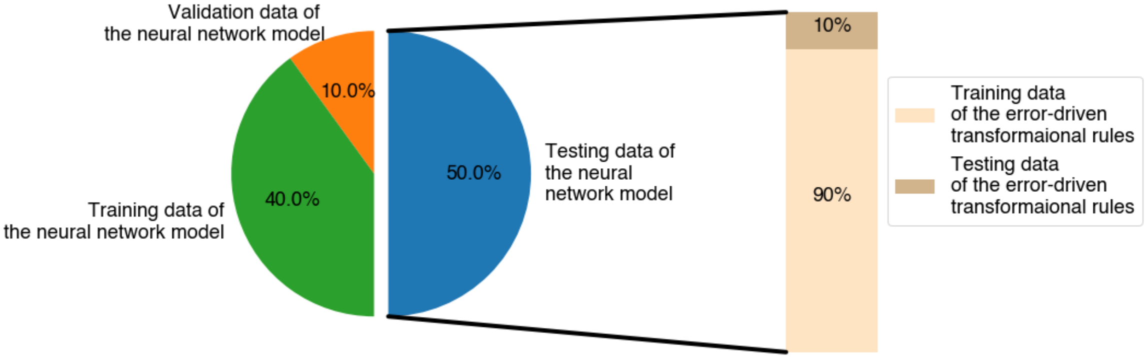


Figure 0.4. Split of Training, Validation, and Testing Data

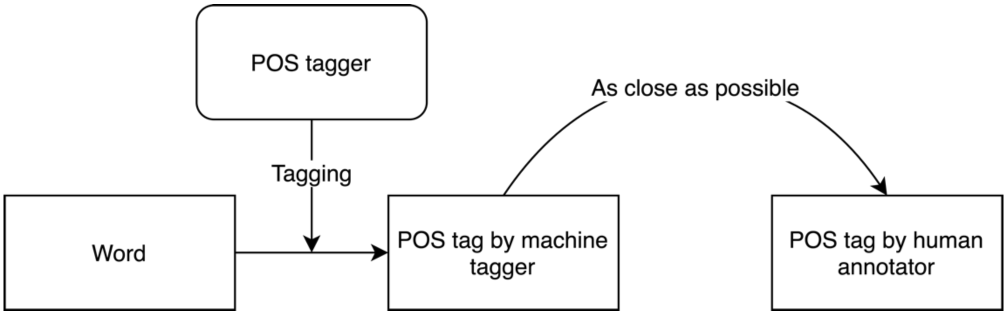


Figure 0.5. POS Tagger Goal

3.3.2 Step 1: Select the Number of Epochs of Training and the Trainable Layer

There were two types of trainable layers studied by the authors in this chapter: (1) bidirectional LSTM, and (2) bidirectional GRU. The number of epochs of training cannot be predicted before training (Chollet, 2017). The authors decided to train the model 15 epochs and 50 epochs (arbitrarily selected numbers) to analyze the impact of epochs of training on the performance of the model. The trainable layers were layers of bidirectional LSTM or bidirectional GRU. The size of trainable layers was 128. Between trainable layers, there were dropout layers with a dropout rate of 0.4. The authors selected hyper-parameters such as epochs of training, trainable layer size, and dropout rate based on past experience in deep learning. Neural network models with these hyper-parameters generally perform well on a wide range of tasks. Although it is possible to do a more thorough search on hyper-parameters, it is out of the scope of the research of this chapter. The random number embedding layer significantly saved the training time and allowed grid search in this step. The authors attempted four possible combinations (Figure 0.6): (1) one layer of bidirectional GRU model that was trained 15 epochs, (2) one layer of bidirectional GRU model that was trained 50 epochs, (3) one layer of bidirectional LSTM model that was trained 15 epochs, and (4) one layer of bidirectional LSTM model that was trained 50 epochs.

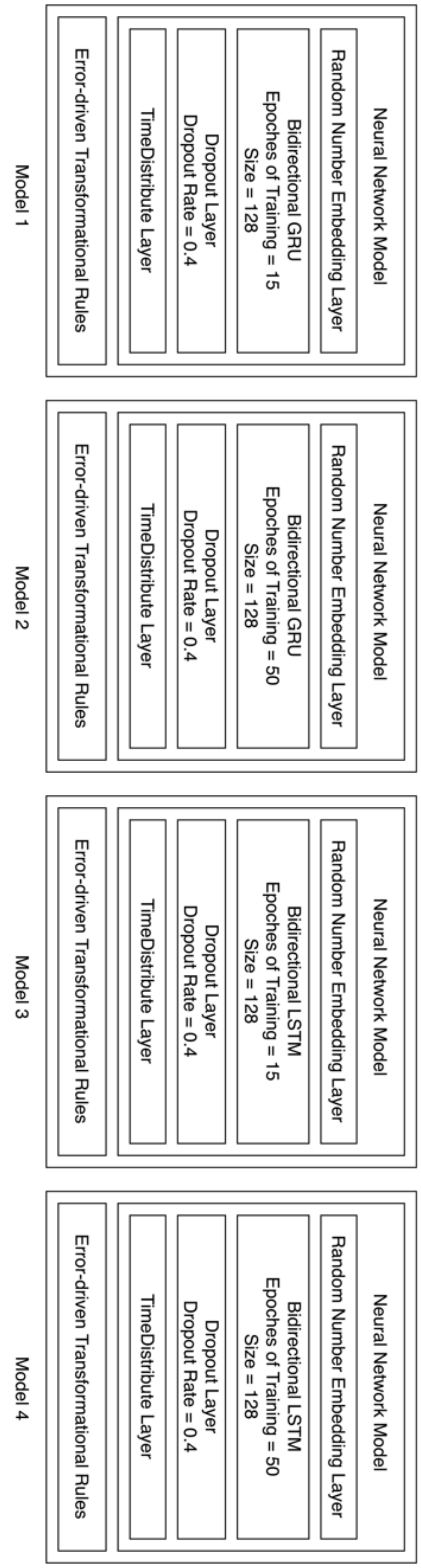


Figure 0.6. Models Trained in Step 1

3.3.3 Step 2: Search a Well-performing Pre-trained Model

Although there were multiple potentially well-performing pre-trained models available, the authors selected BERT, which had achieved the state-of-the-art performance on multiple NLP tasks with little fine-tuning needs (Devlin et al., 2018). The authors tested the eight available versions of BERT: (1) BERT-Large, Uncased (Whole Word Masking), (2) BERT-Large, Cased (Whole Word Masking), (3) BERT-Base, Uncased, (4) BERT-Large, Uncased, (5) BERT-Base, Cased, (6) BERT-Large, Cased, (7) BERT-Base, Multilingual Cased, and (8) BERT-Base, Multilingual Uncased. Therefore, eight models were trained in this step, corresponding to the eight versions of BERT (Figure 0.7). All of them shared the same trainable layers and were trained the same number of epochs.

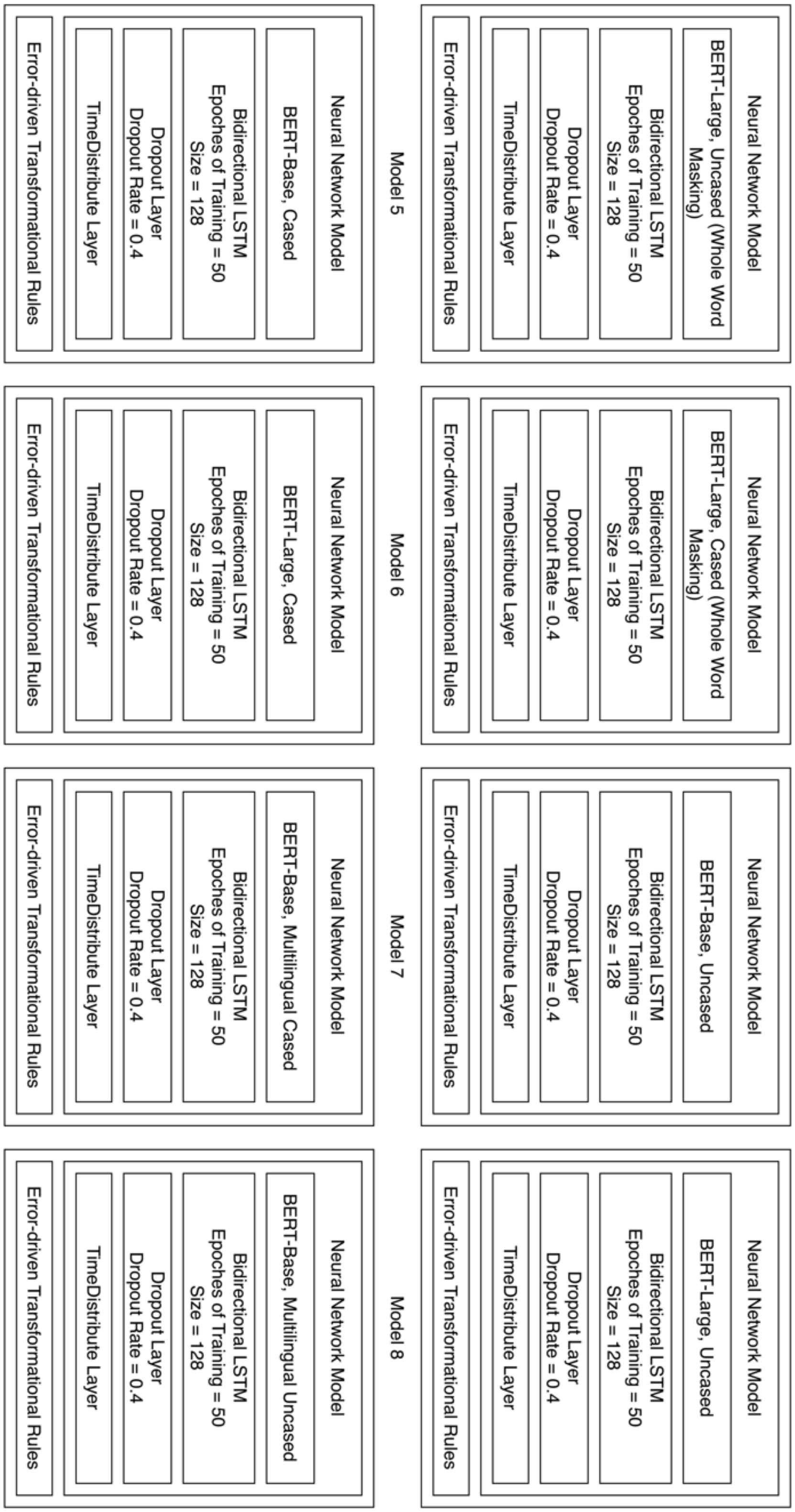


Figure 0.7. Models Trained in Step 2

3.3.4 Step 3: Search the Optimal Number of Trainable Layers

Stacking multiple trainable layers could possibly achieve higher precision by capturing more features in the textual data. However, too many trainable layers may lead to overfitting. To find the optimal number of trainable layers, the authors decided to increase the number of trainable layers and dropout layers until the precision stops increasing. There were two models trained in this step: Model 13, which has two bidirectional LSTM layers and Model 14, which has three bidirectional LSTM layers (Figure 0.8).

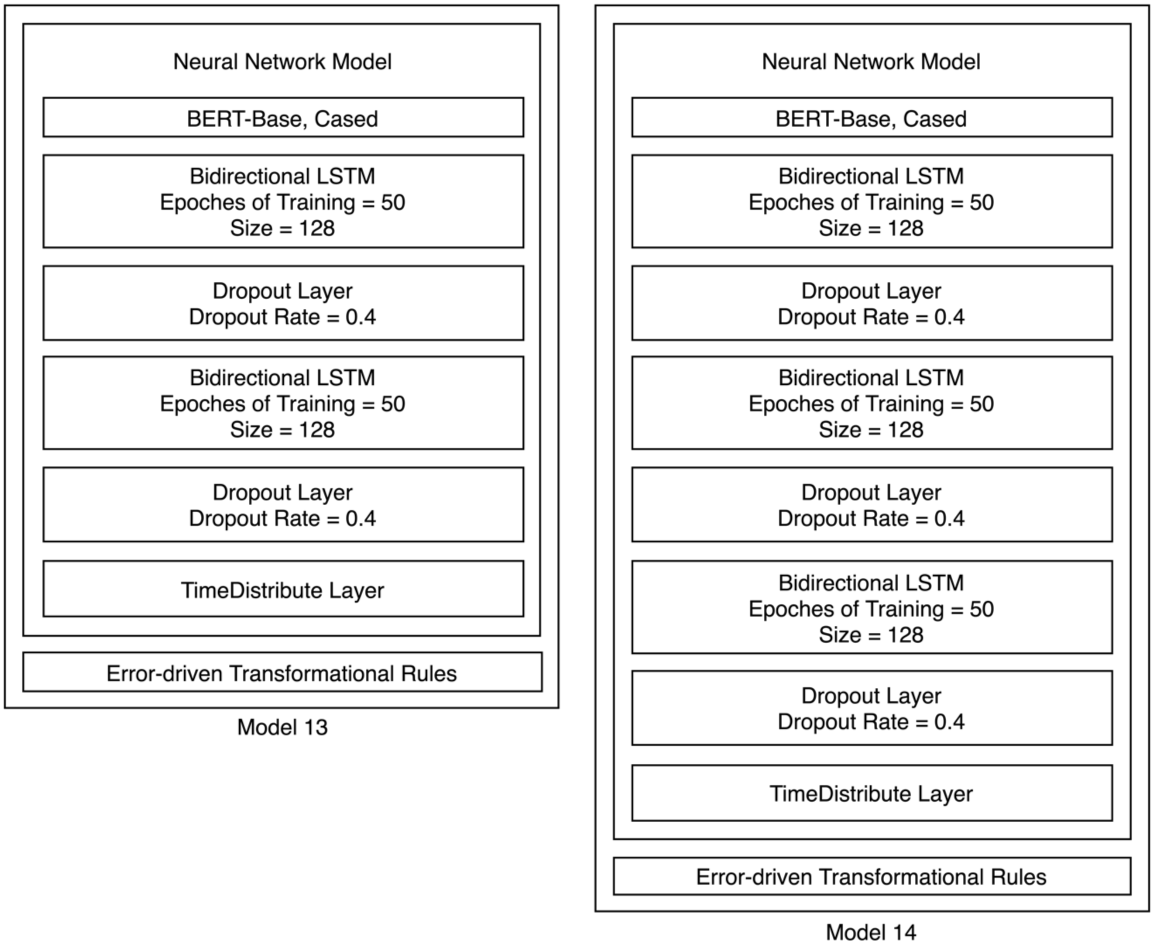


Figure 0.8. Two Models Trained in Step 3

3.4 Result

To find a well-performing combination of epochs of training, pre-trained models, and trainable layers to use in the POS tagger, the authors trained 14 models (Table 0.2). The best-performing POS tagger had a combination of one bi-directional LSTM trainable layer, BERT_Cased_Base pre-trained model, and was trained for 50 epochs. This model (Model 9 in Table 0.2) reached the highest accuracy after applying transformational rules. The optimization of the deep learning component of this POS tagger is out of the scope of the research of this chapter, which may be pursued in future research.

Table 0.2. Summary of the Performance of Models

Model	Before Applying Rules			After Applying Rules		
	Precision	Recall	F1-score	Precision	Recall	F1-score
1	39.02%	17.91%	19.88%	61.59%	51.94%	43.71%
2	89.67%	87.65%	88.14%	93.68%	93.78%	93.64%
3	36.45%	17.41%	20.37%	61.82%	49.93%	43.62%
4	90.15%	87.76%	88.34%	93.53%	93.44%	93.41%
5	90.57%	88.60%	88.87%	94.98%	94.99%	94.88%
6	91.06%	88.64%	89.01%	94.73%	94.75%	94.63%
7	90.40%	88.37%	88.68%	94.16%	94.32%	94.14%
8	89.29%	87.24%	87.60%	93.50%	93.70%	93.49%
9	**91.89%**	**89.71%**	**90.06%**	**95.11%**	**95.42%**	**95.20%**
10	91.49%	89.32%	89.78%	94.50%	94.70%	94.51%
11	89.70%	87.56%	87.80%	94.23%	94.56%	94.33%
12	87.84%	85.92%	86.12%	93.31%	93.03%	93.04%
13	91.81%	89.81%	90.19%	95.04%	95.32%	95.08%
14	91.43%	89.82%	90.07%	94.64%	94.89%	94.70%

3.4.1 Step 1 Result: Epochs of Training and Trainable Layers Combination

Figure 0.9 demonstrates the influence of the trainable layer and the epochs of training on the accuracy of POS tagging. For both trainable layers, increasing the number of epochs can increase the precision. However, when the number of epochs was 15, the precision of the bi-directional LSTM model was lower than that of the bi-directional GRU model. When the number of epochs was 50, the precision of the bi-directional LSTM surpassed that of the bi-directional GRU model. This shows that the optimal number of epochs for different pre-trained models could be different.

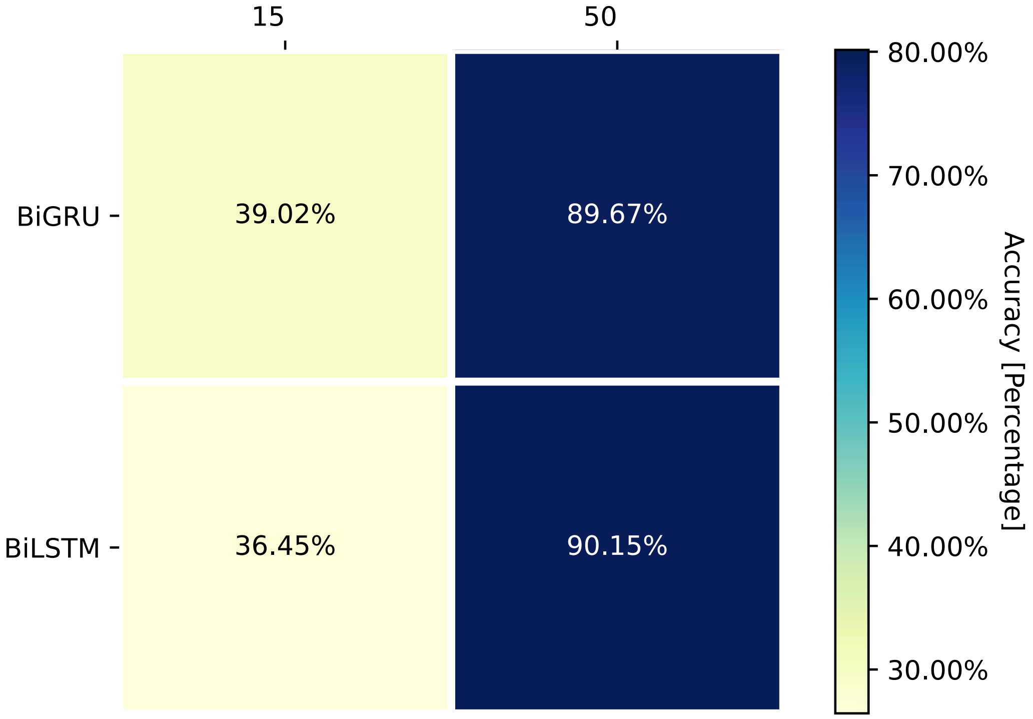


Figure 0.9. Influence of Epochs of Training and Trainable Layers to Precision

3.4.2 Step 2 Result: The Best-performing Pre-trained Model

The precision, recall, and F1-score of models with different pre-trained models are shown in Figure 0.10. All models trained in this step share the same trainable layer and the same number of epochs of training (50). The BERT_Base_Cased model achieved the highest precision, recall and F1-score. The average precision for models with cased models is 91.04% and that for models with uncased models is 89.53% (Figure 0.10). It shows cased information is useful in the POS tagging of building codes. The average precision for models with large models is 90.60% and that for models with base models (excluding multilingual models) is 91.15%. The two multilingual models were excluded in the comparison because there is no large multilingual model and the current POS tagging task is not multilingual. It may be counterintuitive because larger models generally achieve higher accuracy than smaller models. The authors suggest that more training data may be needed to release the full potential of large pre-trained models.

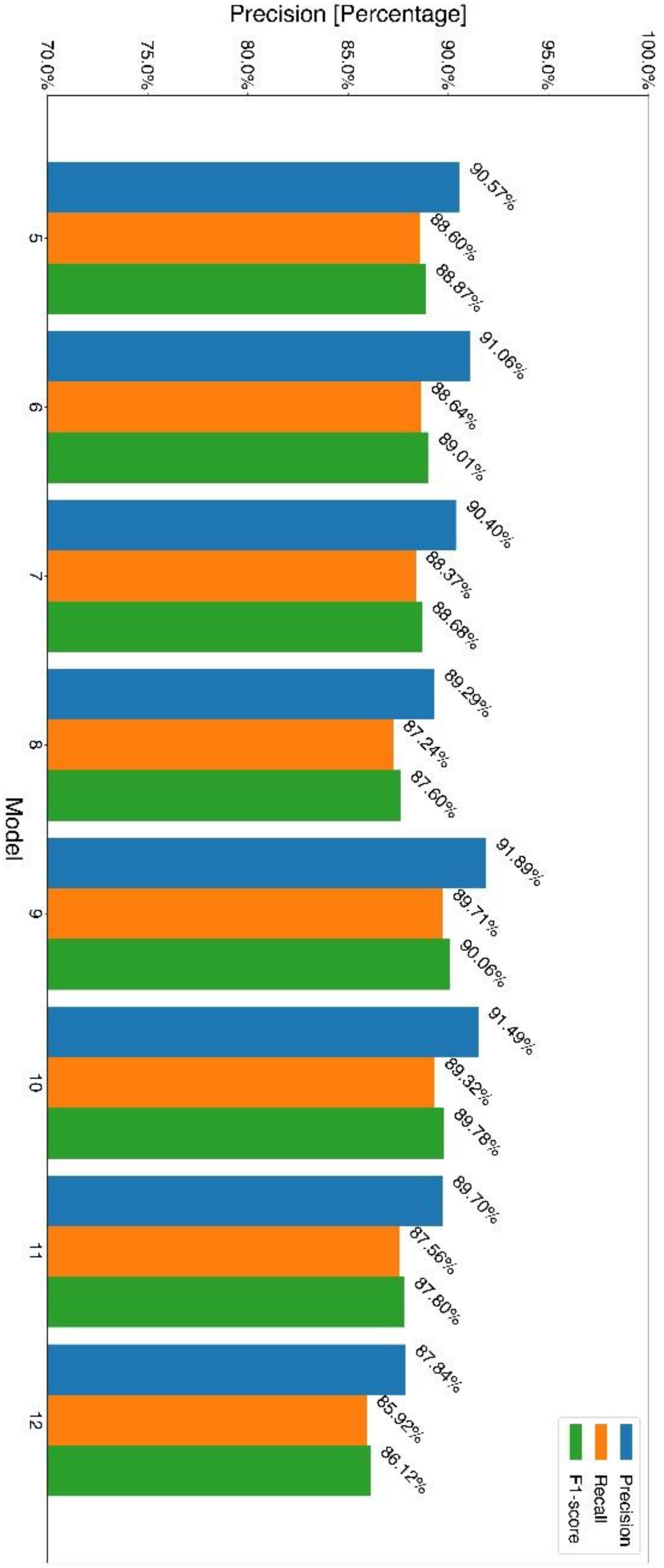


Figure 0.10. Precision, Recall and F1-score of Models with Different Pre-trained Models

3.4.3 Step 3 Result: The Optimal Number of Trainable Layers

After the best-performing pre-trained model was identified, the authors started to identify the optimal number of trainable layers. Result of this attempt is illustrated in Table 0.3. The model with one layer of bidirectional LSTM reached the highest precision. Precision of models decreases as the number of layers increases. The authors concluded that more data is needed to leverage the power of additional trainable layers.

Table 0.3. Number of Trainable Layers vs. Precision

Layers of Trainable Layers	Precision
1	91.49%
2	89.79%
3	87.84%

3.4.3.1 Effectiveness of Error-driven Transformational Rules.

This chapter's research also confirmed the effectiveness of error-driven transformational rules (Figure 0.11). The average precision after applying transformational rules is 94.57%. Although the precision before applying transformational rules varied with pre-trained models and trainable layers, the precision after applying the transformational rules all increased. Moreover, POS taggers with higher pre-rule-application precision will also have a higher post-rule-application precision. The transformational rules increase the precision of POS tagger by a margin of 4.02%. The average training accuracy and testing accuracy of all models that use pre-trained models are 95.45% and 94.57%, respectively. The average training accuracy of the models was only 0.88% higher than their average testing accuracy (

Figure 0.12), which alleviated overfitting concerns. The authors also compared the performance of the proposed tagger against the performance of other state-of the-art POS taggers on the PTBC dataset (Xue & Zhang, 2020) (Figure 0.13).

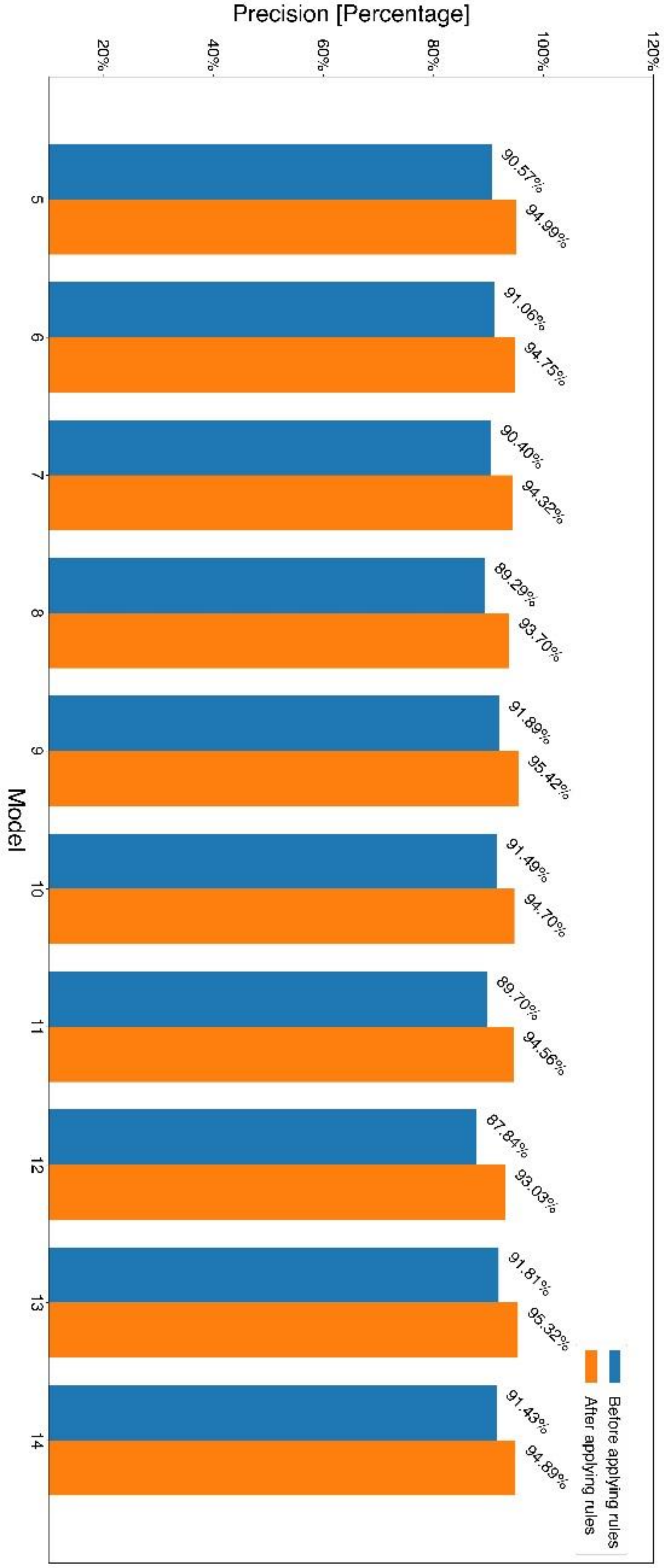


Figure 0.11. Precision of each Model before and after Applying Transformational Rules

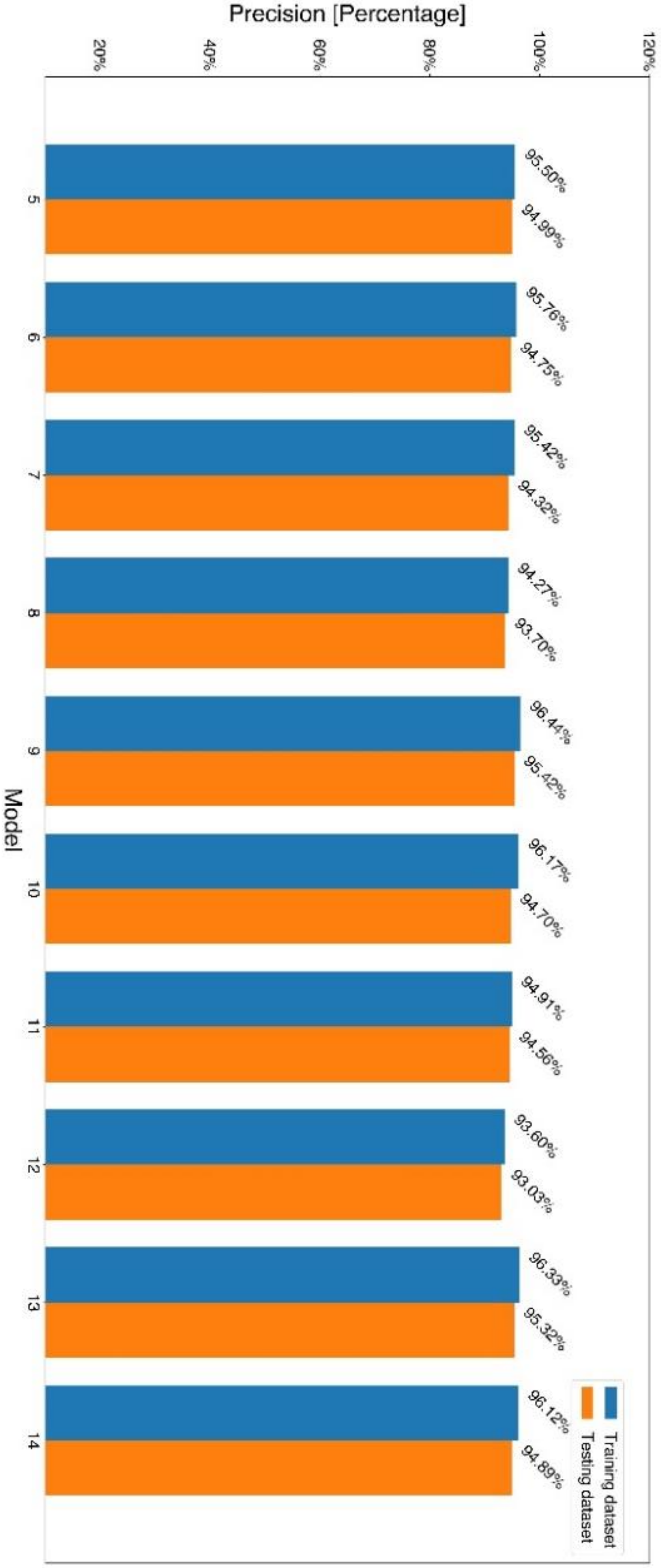


Figure 0.12. Training and Testing Accuracy of Models

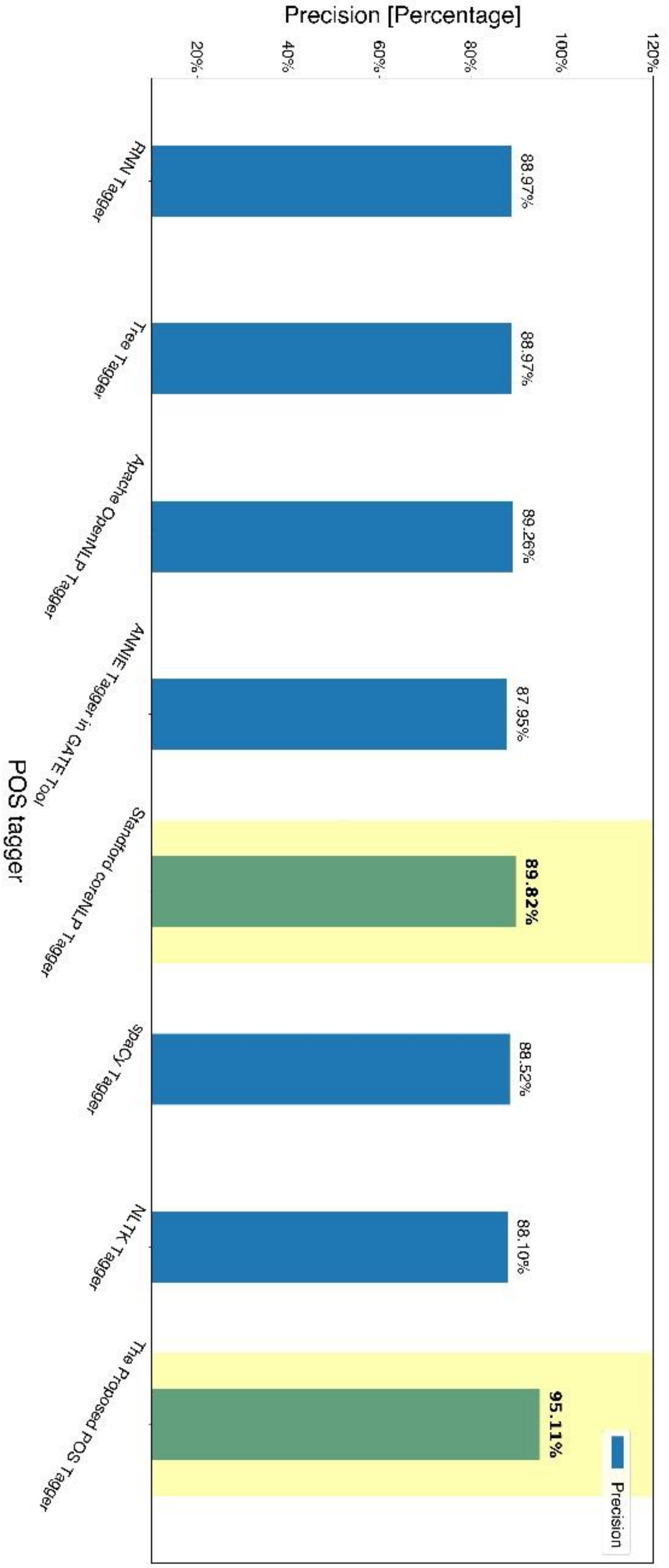


Figure 0.13. Comparison with State-of-the-art POS Tagger

3.4.3.2 Effectiveness of GRU

The bi-directional GRU model without BERT can achieve a precision that is comparable to bi-directional LSTM model that is enhanced by BERT. A significant amount of training time can be saved if there is no pre-trained model to fine-tune. The hardware requirement to fine-tune pre-trained models is also significantly higher than that of the random embedding layer. Directly using the bi-directional GRU model can save training time and cut hardware investment while the compromise on the precision of the POS tagger is within an acceptable range.

3.4.3.3 Tagging Example

To validate this POS tagger, the authors compared the POS tagging result of this POS tagger to a baseline tagger which is a state-of-the-art generic POS tagger. As an example, the baseline tagger incorrectly labeled “horizonal” as a noun. This error may lead to incorrect extraction of embedded engineering knowledge in building codes. In contrast, the proposed POS tagger correctly labeled the word as an adjective. The automated code compliance checking system has a better chance to correctly extract the embedded engineering knowledge in the building codes by the proposed POS tagger, compared to the state-of-the-art generic POS taggers.

3.4.3.4 Impact of Data Split Scenarios

To analyze the impact of different training/testing data split scenarios on the precision, recall, and f1-score, the authors reported the precision, recall, and f1-score of the proposed POS tagger on two other training/testing split methods. The second training/testing split method is using: (1) 60% of the entire dataset as the training dataset of the neural network model, (2) 20% of the entire dataset as the validation dataset of the neural network model, (3) 20% of the entire dataset as the testing dataset of the neural network model, (4) 80% of the entire dataset as the training dataset of the error-driven transformational rules, and (5) 20% of the entire dataset as the testing dataset of the error-driven transformational rules (Table 0.4). The third training/testing split method is using: (1) 60% of the entire dataset as the training dataset of the neural network model, (2) 20% of the entire dataset as the validation dataset of the neural network model, (3) 20% of the entre dataset as the testing dataset of the neural network model, (4) 90% of the testing dataset of the neural network model as the training dataset of error-driven transformational rules, and (5) 10% of the testing dataset of the neural network model as the testing dataset of error-driven transformational rules (

Table 0.5). Results in all training/testing split scenarios showed consistency in: (1) the improvements of performance when using error-driven transformational rules, and (2) the improvement of performance over the state of the art.

Table 0.4. Results of Second Training/Testing Split Method

Model	Before Applying Rules			After Applying Rules		
	Precision	Recall	F1-score	Precision	Recall	F1-score
1	91.15%	89.39%	89.95%	93.10%	92.80%	92.82%
2	92.86%	91.21%	91.72%	94.82%	94.60%	94.64%
3	77.80%	72.13%	71.64%	83.58%	85.35%	83.37%
4	92.98%	91.20%	91.76%	94.62%	94.25%	94.31%
5	91.97%	90.30%	90.76%	96.04%	95.84%	95.56%
6	92.26%	90.28%	90.84%	96.25%	96.22%	95.99%
7	91.93%	90.32%	90.70%	96.00%	95.94%	95.65%
8	90.49%	89.28%	89.49%	95.85%	95.67%	95.37%
9	**93.18%**	**91.82%**	**92.18%**	**96.43%**	**96.35%**	**96.08%**
10	92.58%	91.17%	91.51%	96.31%	96.27%	96.00%
11	91.70%	89.90%	90.40%	95.79%	95.77%	95.44%
12	89.56%	87.93%	88.28%	95.04%	95.02%	94.70%
13	93.02%	91.65%	92.01%	96.40%	96.22%	95.94%
14	92.90%	91.77%	92.00%	96.83%	96.62%	96.28%

Table 0.5. Results of Third Training/Testing Split Method

Model	Before Applying Rules			After Applying Rules		
	Precision	Recall	F1-score	Precision	Recall	F1-score
1	91.17%	89.86%	90.23%	92.48%	92.32%	92.25%
2	92.83%	90.59%	91.27%	93.60%	93.19%	93.32%
3	77.91%	69.31%	69.47%	80.81%	80.24%	78.11%
4	92.88%	90.65%	91.34%	93.25%	92.97%	93.03%
5	92.07%	90.49%	90.90%	95.11%	94.71%	94.85%
6	92.06%	90.01%	90.61%	94.61%	94.27%	94.32%
7	91.62%	90.17%	90.43%	93.18%	92.62%	92.79%
8	90.79%	89.28%	89.61%	93.87%	93.50%	93.59%
9	**93.23%**	**91.47%**	**91.96%**	**96.12%**	**95.70%**	**95.84%**
10	92.25%	90.82%	91.20%	94.73%	94.49%	94.55%
11	91.90%	90.14%	90.51%	95.26%	94.93%	95.06%
12	90.31%	88.79%	89.29%	93.07%	92.62%	92.70%
13	92.83%	91.12%	91.49%	95.99%	95.48%	95.65%
14	92.73%	91.30%	91.60%	95.51%	95.26%	95.32%

3.5 Discussion

Previous POS taggers either took a rule-based approach (Bird, 2009) or used machine learning or deep learning algorithm exclusively (Giménez, 2004). The proposed POS tagger combine the advantage of rule-based approach and machine learning algorithm. One main limitation of the proposed POS tagger is acknowledged: the POS tagger still is not error-free. In spite of its improvement over the state of the art, this POS tagger may still not be accurate enough to support an error-free extraction of embedded engineering knowledge in building codes. Errors in POS tagging may have negative effect on the performance of NLP-based automated building code compliance checking systems that leverage it. The authors suggest that research to further increase the accuracy of POS taggers is still needed. The authors also plan to develop automated code compliance checking systems that have the robustness to tolerate a small amount of POS tagging errors.

3.6 Contributions to the Body of Knowledge

The research of this chapter has contributions in both theory and practice of POS tagging. Theoretically, it has two main contributions to the body of knowledge. First, it provides a deep-learning and rule-based hybrid method to enhance performance of POS taggers on domain-specific texts. The combination of deep learning neural network models and error-fixing transformational rules makes the proposed POS tagger outperform the state-of-the-art POS taggers with limited amount of training data. Many current state-of-the-art POS taggers were trained on the Penn Treebank (PTB) corpora which has 2,499 articles (each article contains tens, if not hundreds, of sentences). This POS tagger was trained on a dataset of only 1,522 sentences. Second, the research of this chapter shows the potential of deep learning in automated building code information extraction. The promising results of deep learning on the POS tagging of building codes paved the way to more applications of deep learning in automated building code compliance checking and engineering tasks in the AEC domain in general. In practice, the impact of this chapter's research on the AEC domain could be profound. It provides a more accurate POS tagger for building codes comparing to the state of the art, which will help automated code compliance checking systems to check more building code requirements automatically. The extension of checkable building code requirements could bring automated code compliance checking systems one step closer to a wide real-world deployment.

3.7 Conclusion

The ability to provide accurate POS tagging results of building codes paves the way to automated regulatory information extraction and widens the possible range of applicable code requirements of automated code compliance checking systems. The authors proposed a new POS tagger to support such systems. This is the first POS tagger that is tailored to building codes. The POS tagger gained information on general English by incorporating pre-trained deep learning models and captured AEC domain specific knowledge by fine-tuning on a domain-specific corpus. The POS tagger directly maps inputted words to POS tags without feature engineering. This nature of deep learning allows future domain experts to enhance the performance of this POS tagger by directly leveraging more training data. The experiment showed that the bi-directional GRU model without pre-trained models can reach a high precision that is comparable to the precision of the bi-directional LSTM models with pre-trained models. Using bi-directional GRU model can save time and cost to train a POS tagger, without significantly compromising precision. Although more training data may help unleash the full potential of pre-trained models and further improve performance, the authors were able to achieve a 95.11% precision using one bi-directional LSTM trainable layer and BERT_Cased_Base pre-trained model in combination with error-driven transformational rules, which significantly increased over the state-of-the-art.

3.8Acknowledgement

The authors would like to thank the National Science Foundation (NSF). The material in this chapter is based on work supported by the NSF under Grant No. 1827733. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

4 REGULATORY INFORMATION TRANSFORMATION RULESET EXPANSION TO SUPPORT AUTOMATED BUILDING CODE COMPLIANCE CHECKING

Xiaorui Xue, S.M.ASCE[1]; Jiansong Zhang, Ph.D., A.M.ASCE[2]

A portion of this chapter was previously published by:

Xue, X., Zhang, J. (2022). Regulatory information transformation ruleset expansion to support automated building code compliance checking. *Automation in Construction, 138*(June 2022), 104230. https://doi.org/10.1016/j.autcon.2022.104230

Author Contributions

The authors confirmed contribution to the paper as follows:

Study conception and design: Xiaorui Xue, Jiansong Zhang.

Data collection: Xiaorui Xue, Jiansong Zhang.

Analysis and interpretation of results: Xiaorui Xue, Jiansong Zhang.

Draft manuscript preparation: Xiaorui Xue, Jiansong Zhang.

All authors reviewed the results and approved the final version of the manuscript.

4.1 Literature Review

4.1.1 Natural Language Processing

Chowdhury defines natural language processing (NLP) as “an area of research and application that explores how computers can be used to understand and manipulate natural language text or speech to do useful things” (Chowdhury, 2003). NLP includes a wide range of tasks, such as (1) information retrieval (Raghavan et al., 2010), (2) information extraction (Cowie & Lehnert, 1996), (3) text classification (Zhang et al., 2015), (4) text generation

(McKeown, 1985), (5) text summarization (Nenkova & McKeown, 2012), (6) question answering (Soares & Parreiras, 2020), (7) machine translation (Koehn, 2009), and (8) speech recognition (Povey et al., 2011). There are two main approaches to accomplishing NLP tasks: the rule-based approach and the machine learning-based approach (Gali et al., 2008). Rule-based NLP systems may require manual effort in rule generation, but usually outperform machine learning-based NLP systems in a specific task or in a specific domain (Crowston et al., 2010). Machine learning-based NLP systems can be further classified into "shallow" learning systems and "deep" learning systems based on the types of machine learning models they use. "Shallow" learning systems use traditional machine learning algorithms, such as support vector machines (SVMs) or decision trees, and require manual feature engineering. Deep learning systems use neural networks and do not require manual feature engineering (Chollet, 2017). There is no lack of efforts to use NLP in the AEC domain. For example, Tixier et al. (2016) used NLP to extract the reasons for accidents from construction injury reports. Lin et al. (2013) used NLP technologies to extract information from BIM. ACC research also uses NLP techniques to process building codes, for matching between concepts in building codes and concepts in BIM (Zhang & El-Gohary, 2019), and for converting building codes to logic clauses that support automated reasoning (Zhang & El-Gohary, 2016).

4.1.2 Part-of-Speech

Part-of-speech (POS) of a word represents its lexical and syntactic function in a sentence (Barzilay & Elhadad, 1999). English words have eight basic POS categories: (1) noun, (2) verb, (3) adjective, (4) adverb, (5) pronoun, (6) preposition, (7) conjunction, and (8) interjection (Butte College, 2016). The same word may have different POS categories in different contexts. For example, the word "run" could be a verb in its simple present tense or past perfect tense depending on the context. In NLP systems, words are categorized into more specific POS categories to represent text more informatively. For example, the Penn Treebank Corpus classifies words into 36 POS categories (Marcus et al., 1993) and the Brown corpus has 179 POS categories (Francis & Kucera, 1979). In the development of the Penn Treebank Corpus and the Brown Corpus above, human annotators manually assigned words to different POS categories according to their understanding of the English language and the contexts of the words. POS tagging software, which is commonly called "POS taggers," could replace annotators' manual effort in this task. POS taggers automatically determine the POS category of a word using its contextual information in an algorithmic manner (Schmid, 1994). POS taggers began with rule-based taggers that used a set of rules to determine the POS categories of words. These rules can be compiled by experts (Bird et al., 2009) or extracted from text algorithmically (Brill, 1992). With the development and integration of machine learning, POS taggers shifted to the use of statistical models. For example, Giménez and Marquez (2004) used one SVMs model to determine POS categories of known words and another SVMs model to predict those of unknown words. Brants (2000) developed a POS tagger which uses Hidden Markov Models (HMM) to capture dependencies among words and determine the POS categories of words by their inter-dependencies. Plank et al. (2016) proposed the use of bi-directional neural networks to accomplish multilingual POS tagging. POS tagging is an important early step of many NLP systems (Giménez & Marquez, 2004).

4.1.3 Ontology

Ontology is the explicit and formal description of knowledge through relationships among concepts in a domain (Gruber, 1993). In 1999, the World Wide Web Consortium (W3C) first developed the Resource Description Framework (RDF) language for ontology (Brickley et al., 1999). Then, it collaborated with the Defense Advanced Research Projects Agency (DARPA) to extend the RDF into a more expressive DARPA Agent Markup Language (DAML) (Hendler & McGuinness, 2000; Mcguinness et al., 2002). After that, many ontologies emerged, either for a specific domain (e.g., medical) (Amos et al., 2020) or for general-purpose (Hepp, 2008). Ontology is used to: (1) analyze and reuse domain knowledge, (2) share structured domain knowledge among people and software, (3) specify domain assumptions, and (4) distinguish domain knowledge from operational knowledge (Noy & McGuinness, 2001).

4.1.4 Text Similarity Measurements

Text similarity is an important benchmark in NLP. There are many ways to measure the similarity between text strings (Gomaa & Fahmy, 2013). Text similarity can be measured by comparing words or characters in text strings. For example, the Levenshtein Distance (Levenshtein, 1966; Su et al., 2008) measures the minimum number of single character transformations needed to convert one string to another. In Levenshtein Distance, 0 means two strings are identical, and the larger it is, the less similar the two strings are, with no strict upper bound. The Jaccard Distance, on the other hand, measures the number of items shared by two sets (Kosub, 2019). In the Jaccard distance, 0 means two sets are identical, and 1 means two sets share no common items. The Jaro Winkler Similarity (Winkler, 1990) is an extension of the Levenshtein Distance. By normalizing the Levenshtein Distance with the length of the text string, the Jaro Winkler Similarity ranges from 0 to 1. In the Jaro Winkler Similarity, 0 means two text strings are completely different and 1 means two text strings are the same.

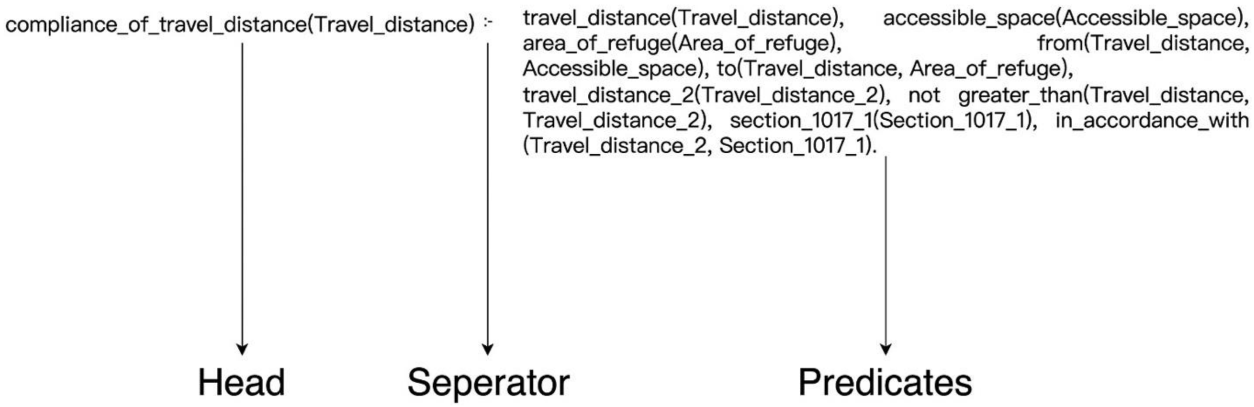
The inability to measure similarity between word meanings is one limitation of measuring text similarity at the word and character levels. One potential solution to the problem is representing words (and their contexts) as vectors in high-dimensional spaces. Popular text vectorization techniques include, for example, Word2Vec (Mikolov et al., 2013), FastText (Joulin et al., 2017), and Glove (Pennington et al., 2014). The distance between meanings of two words and their contexts can be measured by the cosine distance between the two vectors.

4.2 Methodology

The proposed method expands the range of processable building code requirements by adding new pattern matching-based rules to an existing ruleset. The pattern matching-based rules capture regulatory information in building codes and convert the captured information to logic clauses. The pattern matching-based rules consider both syntactic information, which is provided by POS tags (e.g., the word “height” in the phrase “building height” is a noun because it has a POS tag “NN”), and sematic information, which is provided by an ontology (e.g., the phrase “less than” after an attribute and before a value means that the attribute value must be smaller than the specified value, and the phrase “minimum clearance” means the attribute “clearance” must be greater than or equal to a specified value). For example, the pattern “subject, conjunction, subject” can be used to extract the regulatory information of which two subjects are in equivalent status. The conjunction (i.e., and, or) is a label by the POS tagger. Subjects, which were labeled as noun by the POS tagger, were further labeled as subjects after feature enhancement by the ontology. The pattern matching-based rules regulate how this method extracts building code requirements and converts them to logic causes. The logic clauses represent building code requirements in a strict horn clause (HC) format in B-Prolog syntax to avoid ambiguity in natural language and facilitate automated reasoning by the logic reasoner.

Each logic clause has a left-hand side and a right-hand side, separated by the delimiter ":-". The left-hand side is the head of the logic clause that represents the subject of compliance checking in the logic clause, i.e., the building design component that this code requirement governs. The subject of compliance can be an entire building, a component of a building, a certain attribute of a building, or an attribute of a building component. The predicates on the right-hand side of the delimiter ":-" (i.e., in the body of the logic clause) are conditions that the subject of compliance need to meet to comply with the building code requirement. This logic clause indicates that if all predicates on the right-hand side of the delimiter ":-" are evaluated to True, then the predicate on the left-hand side of the delimiter ":-" will also be evaluated to True. In the context of this dissertation research, it means that if the subject of compliance meets all the conditions of the corresponding building code requirement, it is then considered to be compliant with the building code requirement. In the logic clauses, the conjunction relation (i.e., AND) is represented as a comma "," and the disjunction relation (i.e., OR) is represented as a semicolon ";".

One manually generated logic clause example as part of the gold standard (see details in Section 5.2.2 - Gold Standard Generation) is provided in Figure 0.1. The "*Travel_distance*" in is the subject of compliance and the predicates on the right-hand side describe the building code requirement that the "*Travel_distance*" need to comply with. Each predicate describes one condition that the subject needs to satisfy to comply with in the building code requirement described in the logic clause. If all of the predicates on the right-hand side are evaluated to true, the ACC system will then determine the building design to be in compliance with the corresponding building code requirement. More specifically, the predicates "*from(Travel_distance, Accessible_space), to(Travel_distance, Area_of_refuge)*" describe that the travel distance is measured from the accessible space to the refuge area. The predicate "*in_accordance_with(Travel_distance_2, section_1017_1)*" describes that the travel distance is specified in Section 1017.1 of the IBC 2015. The predicates "not *greater_than(Travel_distance, Travel_distance_2*)" require the travel distance from the accessible space to the area of refuge to be no greater than the travel distance specified in Section 1017.1 of the IBC 2015. Other predicates in the logic clause are required by the strict HC format in B-Prolog syntax for this logic rule to execute. Overall, this logic clause describes the building code requirement that the maximum travel distance from an accessible space to an area of refuge should not be greater than the travel distance specified in Section 1017.1 of the IBC 2015.



```
compliance_of_travel_distance(Travel_distance) :- travel_distance(Travel_distance), accessible_space(Accessible_space),
area_of_refuge(Area_of_refuge), from(Travel_distance,
Accessible_space), to(Travel_distance, Area_of_refuge),
travel_distance_2(Travel_distance_2), not greater_than(Travel_distance,
Travel_distance_2), section_1017_1(Section_1017_1), in_accordance_with
(Travel_distance_2, Section_1017_1).
```

Figure 0.1. Example Logic Clause

In the manual transformation of building code, domain experts complete the transformation based on their understanding of building code requirements. In the automated transformation, a pattern matching-based regulatory information transformation ruleset is used to complete this transformation automatically. To support the matching pattern development in the ruleset, building codes undergo feature enhancement by POS tagging and ontology matching (Figure 0.2).

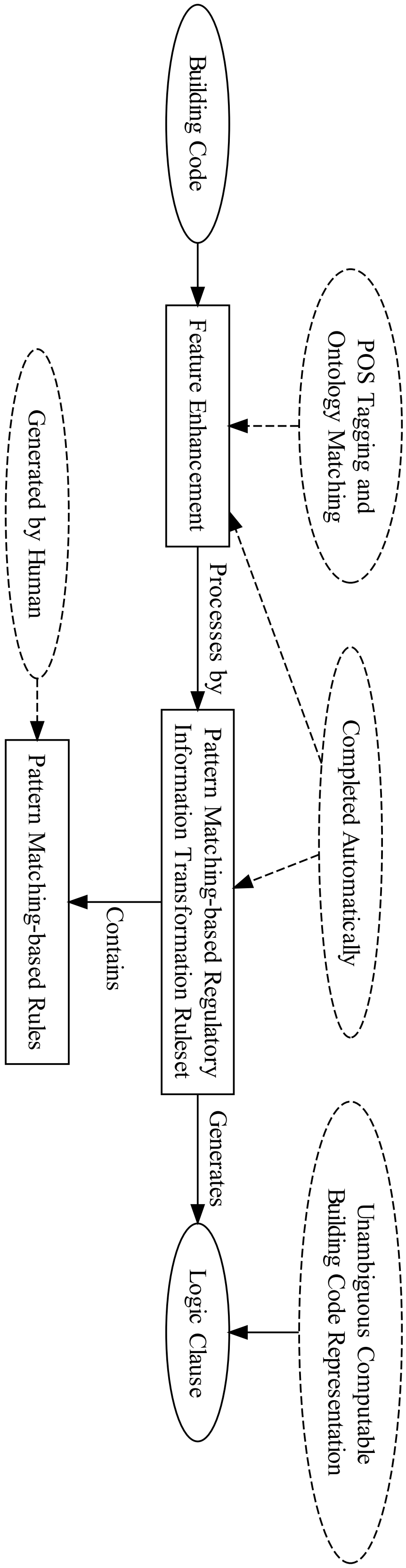

Building Code
Feature Enhancement
POS Tagging and Ontology Matching
Generated by Human
Processes by
Completed Automatically
Pattern Matching-based Regulatory Information Transformation Ruleset
Pattern Matching-based Rules
Contains
Generates
Logic Clause
Unambiguous Computable Building Code Representation

Figure 0.2. Automated Logic Clause Generation

The goal of the research in this chapter is to develop an efficient and effective method to extend an existing pattern matching-based regulatory information transformation ruleset. Although it is possible to develop a new ruleset from scratch, the authors chose to expand an existing ruleset developed by Zhang and El-Gohary (2015), based on the assumption that asymptotic full coverage of building codes could be achieved by expanding an existing ruleset. In addition, expanding an existing ruleset, instead of generating new rulesets, has the benefits of: (1) reducing rule generation workload, and (2) allowing the expanded ruleset to capture patterns absent in the training dataset, while maintaining the compatibility of the expanded ruleset with the automated building code compliance checking system. The expansion of an existing ruleset requires new rules to be added. The added rules must meet certain standards or have certain characteristics to realize the two benefit goals above. For example, the amount of effort/time spent on new rules development should be significantly less than (e.g., no greater than 20% of) the development of the original ruleset. To achieve the first goal, the number of added rules should be small. To achieve the second goal, the process of adding new rules needs to be selective. The added rules should be valid and general. A rule is valid when it correctly extracts the regulatory information it is designed to extract. A rule is considered general when it has been applied at least twice in the training dataset. The combination of multiple valid and simple pattern matching-based rules can be used to represent more complex patterns in building codes. The added rules also need to be general to capture patterns that are not in the training data. Building codes are legal documents composed by a panel of experts following strict guidelines. Therefore, the same patterns may be shared by different chapters of the building code. The generality requirement allows pattern matching-based rules to capture common patterns shared by different chapters of the same building code, or different building codes. In other words, different building codes, or at least different chapters of the same building code, should follow a set of common patterns, according to English grammar and the way building codes were compiled. The ruleset expansion method proposed in this chapter is designed to ensure the generality and validity of added rules. The transformation rule generation and validation are manually conducted in the proposed method. However, once the transformation rules are generated and validated, they can be used to fully automatically transform building code requirements into logic clauses.

4.2.1 Ruleset Expansion Method

The proposed ruleset expansion method takes an iterative approach to add new rules into an existing regulatory information transformation ruleset. The goal of the research in this chapter is to develop an efficient method to expand an existing pattern matching-based regulatory information transformation ruleset and ensure the generality and expandability of the added rules. To achieve this goal, the authors should identify missing regulatory information and generate new rules to capture it. There are two approaches to completing this task: (1) identify all missed regulatory information and generate corresponding rules in a single pass, or (2) identify one piece of missed regulatory information at a time, generate a rule to capture the missed information, review the performance of the new rule, modify the new rule, and then proceed to the next iteration of identifying missed information. Both approaches can generate a ruleset that captures all regulatory information. However, the first approach does not consider the validity and generality of the added rules and the interaction among them (i.e., multiple rules may match the same regulatory information). The second approach iteratively adds new rules and tests them before they are eventually added to the ruleset. The second approach has the potential to generate more valid and general rules than the first approach. What really distinguishes the first and second approaches is the granularity of pattern matching-based rules. The first approach aims to extract regulatory information at the chapter level or even at the whole building code level. Whereas the second approach extracts regulatory information at the sentence level. Because logic clauses are generated at the sentence level, the second approach naturally fits better. In addition, the shorter the pattern lengths are, the more flexible they become and the better scalability the whole ruleset will have. Patterns may match the whole sentence of a building code requirement or (most likely) part of a sentence. Data-driven expansion of the existing ruleset is also made possible through the second approach, whereas it would not have been possible with the first approach. Previous rule-based NLP applications (Abacha & Zweigenbaum, 2011; Bird et al., 2009; Zhang et al., 2009) with manually developed rules also supported this point. Testing rules one-by-one before they are added to the ruleset is also a more rigorous rule development process than generating all the rules and testing them together in one shot.

Because of the above-mentioned advantages, the ruleset expansion method proposed in the research in this chapter takes the second approach, which is shown in

Figure 0.3. One candidate pattern matching-based rule is generated to capture missing regulatory information one piece at a time, until all the missed regulatory information in the training dataset is captured. A version of logic clauses is first generated by the ruleset to identify missing regulatory information. If an instance of missing regulatory information is identified in this version of logic clauses, a candidate pattern matching-based rule is generated to extract it. The candidate pattern matching-based rule will be added to the ruleset if it is proven to be general and valid. The generality and validity of the candidate rule are tested by inspecting a new version of logic clauses generated by the ruleset when the candidate rule is included. A valid rule must correctly extract the information it is designed to extract and does not introduce errors when it is applied to other parts of the training text. A general rule needs to be able to be applied at least twice in the training dataset, which means it should be applied at least once outside of the sentence it is extracted from. The validity of a rule has a higher priority than its generality. If the rule introduces any errors in the training text, it will be modified until it introduces no errors. At the same time, the validity of rules will not be sacrificed to make a new rule general. It is possible that a pattern appears only once in the training text, and it is still necessary to capture an instance of regulatory information. The ruleset expansion process continues until the expanded ruleset captures all the regulatory information in the training data.

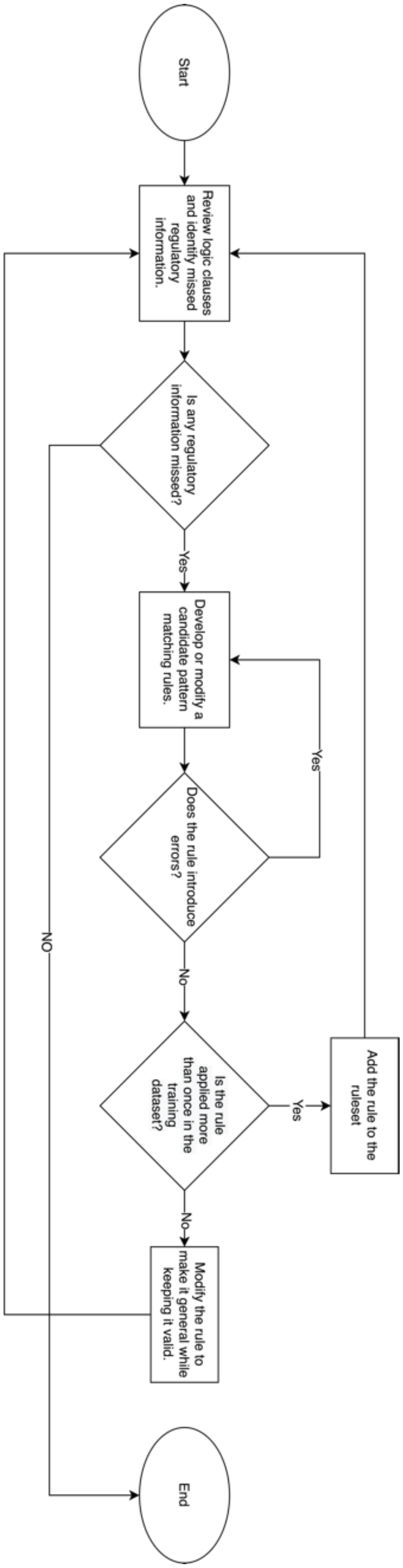

Start
Review logic clauses and identify missed regulatory information.
Is any regulatory information missed?
Yes
Develop or modify a candidate pattern matching rules.
Does the rule introduce errors?
Yes
No
Is the rule applied more than once in the training dataset?
Yes
Add the rule to the ruleset
No
Modify the rule to make it general while keeping it valid.
NO
End

Figure 0.3. Ruleset Expansion Method

4.2.2 Feature Enhancement

The input textual data (building code in plain text) are first enhanced by POS tagging and ontology matching, to generate extra features. The enhanced building codes are more informative and support more complex operations than the original building codes without extra features. Building codes are labeled with information tags in this step. Extra features can provide more information about building code expression patterns and, therefore, increase the performance of pattern matching-based rules. To understand building codes, it requires knowledge both of the English language and of the AEC domain. The feature enhancement makes the ACC system better at processing building codes by introducing such knowledge to the ACC system. The authors apply POS tagging to generate syntactic features (i.e., for knowledge of the English language) and applies ontology matching to introduce AEC domain knowledge in this step.

The ACC system uses POS tagging, which captures the grammatical roles of words in a sentence, to generate syntactic features from building code text. The same word in different POS categories can have distinct meanings. For example, when the word “run” is a verb, it means an action of moving through a space. When the word “run” is a noun, it refers to a physical object. Therefore, syntactic features together with semantic features can disambiguate words in building codes. For example, when the word “runs” is a noun in the sentence “The extensions of handrails shall be in the same direction of the flights of stairs at stairways and the ramp runs at ramps” (Section 1014.6 of IBC 2015) (International Code Council, 2015), it represents a physical object with attributes governed by the building code. However, when the word “runs” is a verb in the sentence “Where a partition containing piping runs parallel to the floor joists” (Section 2308.5.8 of IBC 2015) (International Code Council, 2015), such a possibility can be ruled out. In this chapter, the authors used the A Nearly-New Information Extraction System (ANNIE) POS tagger in the General Architecture for Text Engineering (GATE) (Cunningham, 2002) with proven performance in tagging building codes to generate accurate syntactic features. Such external POS taggers, which were trained on a larger body of text and fine-tuned by domain experts, can bring additional grammatical knowledge to the ACC system.

The ACC system also uses an ontology to introduce AEC domain knowledge for logic clause generation. In manual code compliance checking, reviewers already have domain knowledge needed to understand building codes, based on their education, training, and experience. However, in ACC systems, such knowledge needs to be explicitly provided. An ontology allows the ACC system to access domain knowledge and apply domain knowledge in rule generation. For example, with an ontology, the method can treat “International Fire Code” and “automatic sprinkler system” as integral phrases instead of multiple individual words. It also makes the method treat “inches” and “feet” as units specifying a numerical constraint, instead of regular nouns. In addition, the ontology also supports the disambiguation of vague terms.

The used ontology has two main types of items: (1) essential information, and (2) secondary information. Essential information includes: (1) subject of a particular regulatory requirement (e.g., building), (2) attribute (e.g., building height), (3) comparative relationship (less than, greater than), (4) quantity (e.g., value or range of value), (5) quantity unit (e.g., inch, feet), and (6) reference to other quantity. Secondary information includes restrictions and exceptions. Restrictions mean constraints to subjects and attributes (Dimyadi et al., 2016; Zhang & El-Gohary, 2015). For example, in the sentence “Exterior exit stairways and ramps serving as an element of a required means of egress shall be open on not less than one side, except for required structural columns, beams, handrails and guards, (International Code Council, 2015)” “serving as an element of a required means of egress” is a constraint to “Exterior exit stairways and ramps.” Exceptions are the conditions where a requirement does not apply. The ontology was created and tested in a previous study (Zhang & El-Gohary, 2015). With an expansion for this specific task, its comprehensiveness is ensured in the context of this application by enumerating all covered concepts in the corresponding code requirements. The ontology is also scalable. Similar to the ruleset itself, the ontology could also be accumulatively and continuously developed to fulfill the need for processing different building codes, until it reaches or asymptotically approaches the saturated state where any potentially related concept to building codes is included. It is editable in GATE (Cunningham, 2002) or using a plain text editor. Ontology editing tools provide support for the scalability of the ontology as the size and complexity of the ontology increases.

4.2.3 Pattern Extraction

There are two approaches to extracting regulatory information from building codes: the top-down approach, and the bottom-up approach. In the top-down approach, the information extraction algorithm constructs a global logic clause framework that matches the overall structure of a sentence and fills in the building code requirements into the framework. In the bottom-up approach, the information extraction algorithm captures building code requirements at a local level and assembles them into a logic clause. The building codes consist of a lot of long and complex sentences with diverse structures. The versatility and complexity of building code sentences means developing enough sentence-level frameworks to accommodate all sentences in building codes may require a similar amount of manual effort as directly converting building codes to logic clauses manually. It is possible that every sentence, or at least every few sentences, requires a different framework. The authors propose the use of the bottom-up approach. The complex and versatile structures of building code sentences may require many complex sentence-level frameworks, but pattern matching-based rules can assemble these structures from simple local patterns.

4.3 Experiment

4.3.1 Ruleset Expansion Experiment

The authors tested the effectiveness of the proposed ruleset expansion method by measuring its precision, recall, and F1-score in logic clause generation, which in turn tested the generality of the original ruleset. Chapter 10 of the International Building Code 2015 (IBC 2015) was selected as the training data for the experiment and Chapter 5 of the IBC 2015 was selected as the testing data. Zhang and El-Gohary developed the original ruleset based on Chapters 12 and 23 of IBC 2006 (Zhang & El-Gohary, 2015). The authors used the ruleset expansion method to generate new rules based on Chapter 10 of the IBC 2015 and tested the expanded ruleset on Chapter 5 of the IBC 2015, in comparison with the original ruleset.

In the first part of the experiment, the original ruleset generated a baseline version of logic clauses from the training data. The training data was first pre-processed by a POS tagger and an ontology to generate enhanced features. The POS tagger used in the research in this chapter is the ANNIE tagger from the GATE tool (Cunningham, 2002). The authors used the ontology developed in (Zhang & El-Gohary, 2015) with expansions on Chapters 5 and 10 of the International Building Code 2015. After that, the authors used the ruleset expansion method to expand the original ruleset.

First, the authors identified missing regulatory information and updated the original ruleset with a candidate pattern matching-based rule to capture the missing regulatory information. For example, the original ruleset did not have enough patterns to extract all the essential information for requirements that are described using negation together with a past participle verb, so a corresponding pattern and candidate rule was added. The expanded ruleset also includes all rules in the original ruleset. The authors added 64 new rules to the original ruleset, a much smaller number compared to the 306 rules already in the original ruleset. Two of the 64 new rules were developed to extract missed regulatory information in the example type mentioned above. One rule with the pattern “modal verb, negation, base form verb, [adjective, past participle verb, past tense verb]” was generated to extract the regulatory requirement of a subject. This rule can process building code requirement sentences like “A basement (candidate subject) provided with one exit shall (modal verb) not (negation) be (base form verb) located (past participle verb) more than one story below grade plane” (Section 1006.3.2.2 of IBC 2015) (International Code Council, 2015), and “The area of a Group F-2 or S-2 building (candidate subject) no more than one story in height shall (modal verb) not (negation) be (base form verb) limited (past participle verb) where the building is surrounded and adjoined by public ways or yards not less than 60 feet in width” (Section 507.3 of IBC 2015)) (International Code Council, 2015). The first subject is extracted by identifying the first subject candidate to the left of (not necessarily immediately next to) the relationship. This arrangement makes pattern matching flexible. The rule is both general and valid, because it was applied twice in the training dataset and correctly extracted the regulatory information it was designed to extract. Another pattern “candidate subject, preposition, comparative relation, value, unit” was generated to extract the quantitative regulatory requirement of a subject. This rule can process building code requirement sentences like, “The ladder or steps shall not encroach into the required dimensions of the window well (candidate subject) by (preposition) more than (comparative relation) 6 (value) inches (unit).” (Section 1030.5.2 of IBC 2015)) (International Code Council, 2015). Only the “of” relationship between “the required dimension” and “the window well” was extracted by the original ruleset. The newly added rule was not general in current training dataset (i.e., it was applied only once in the training dataset), but it was still valid, because it correctly extracted the regulatory information it was designed to extract. Although the rule was not general, it was still needed to capture an instance of regulatory information. Therefore, it was still added to the ruleset. The complete set of the 64 rules can be found in Appendix A. In the second part of the experiment, the expanded ruleset was tested on Chapter 5 of the IBC 2015 to automatically convert building codes to logic clauses. The automatically generated logical clauses were compared against a gold standard.

4.3.2 Gold Standard Generation

Chapters 10 and 5 of the IBC 2015 were transformed into logic clauses by three annotators semi-automatically to create a gold standard of information transformation and logic clause generation. All three annotators have background AEC knowledge to understand building codes, and necessary skills to transform building codes into logic clauses. The authors provided the annotators a clear annotation protocol, a brief training section before annotation, and machine-generated logic clauses for reference during their annotation. They worked independently without access to the logic clauses generated by other annotators. However, they were presented with the machine-generated logic clauses, which could help annotators align with the rule generation mechanism of pattern matching-based rules, achieve higher inter-annotator agreement, and reduce rule generation time. It also ensures the compatibility of human-generated logic clauses with the automated code compliance checking system.

Annotators were required to use the exact words that came from the building code in their generated logic clauses. For example, if the building code uses the word “exterior” for exterior walls, annotators must also use the word “exterior” in their generated logic clauses to represent exterior wall, rather than using “external wall” or other names. Therefore, the problem that annotators may use different words for the same meaning is prevented. The product of the manual transformation process was three versions of logic clauses, with each version independently and manually generated by one of the annotators. After that, annotators reviewed each other’s work and collectively generated the final gold standard. All annotators agreed that the gold standard represents the meaning of building codes accurately and approved it.

To evaluate the quality of human-generated logic clauses, the authors measured the similarity between the logic clauses generated by different annotators. Because annotators transformed the same building code, they should generate similar logic clauses. A high similarity among human-generated logic clauses of different annotators indicates a high quality of the logic clause generation. The authors chose to measure logic clause similarity by comparing characters and words at the string level in the research in this chapter. While text vectorization and cosine similarity measure the meaning-wise similarity of natural language text, because the logic clauses generated in the research in this chapter are not in natural language and the similarity measurement focuses on the existence of logic clause components rather than the meaning of the logic clauses, vectorization of text and cosine similarity were therefore not used. The authors measured the similarity among logic clauses in two different ways: (1) the Levenshtein Distance and the Jaro Winkler Similarity were used to measure the character-level similarity between the human-generated logic clauses; and (2) the Jaccard Distance was used to measure the predicate-level similarity between the human-generated logic clauses. For example, the Levenshtein Distance, the Jaccard Distance, and the Jaro Winkler Similarity between the two sample logic clauses in Table 0.1 were 14, 0.67, and 0.97, respectively. Overall, annotators reached an average Levenshtein Distance of 6.88, an average Jaccard Distance of 0.63, and an average Jaro Winkler Similarity of 0.99.

Table 0.1. Sample Logic Clauses Generated by Annotators

Building Code Sentence	Logic Clause 1	Logic Clause 2
The maximum width of a swinging door leaf shall be 48 inches (1219 mm) nominal.	compliance_width_of_swinging_door_leaf257(Swinging_door_leaf):-width(Width),swinging_door_leaf(Swinging_door_leaf),has(Swinging_door_leaf,Width),**less_than_or_equal_to**(Width,quantity(48,inches)).	compliance_width_of_swinging_door_leaf257(Swinging_door_leaf):-width(Width),swinging_door_leaf(Swinging_door_leaf),has(Swinging_door_leaf,Width),**equal_to**(Width,quantity(48,inches)).

Because text similarity is task-specific, there was no universally applicable standard for it. Instead, NLP researchers developed their own metrics according to the needs of tasks (Penumatsa et al., 2006; Rekabsaz et al., 2017). The 6.88 Levenshtein Distance seems high, but it does not consider the length of the text string. If the length of text string is considered, the 0.99 Jaro Winkler Similarity proves that human-generated logic clauses are similar at the character level. The 0.63 Jaccard Distance is relatively low. It indicates a significant number of predicates are different in human-generated logic clauses. However, the difference could be overstated because the Jaccard Distance requires two predicates to be completely identical in order to be considered the same. If two predicates are off by even one character, they are still considered different and accounted for in the Jaccard Distance. Combining the use of all three measures illustrates that human-generated logic clauses are similar in general. If two predicates are different, they usually only differ by a few characters. Measurement results using the three metrics show that the three annotators reached a reasonable alignment and the quality of the gold standard was good (Cahyono, 2019; Kloo et al., 2019).

4.4 Result

To evaluate the performance of the ruleset expansion method, the machine-generated logic clauses were compared against the human-generated gold standard. The closer the machine-generated logic clauses are to the gold standard, the better the pattern matching-based rules are, and therefore, the better performance the ruleset expansion method is deemed to have. Predicates in logic clauses can be further broken down into predicate elements (i.e., predicate names or predicate arguments). For example, the predicate “has(Stairway, Clear_width)” is formed by three elements, “has,” “Stairway,” and “Clear_width.” Therefore, the authors calculated both predicate-level performance and predicate element-level performance of the ruleset expansion method. In the predicate-level performance, the minimum unit of measurement is a predicate. In the predicate element-level performance, the minimum unit of measurement is a word or phrase. For example, if the gold standard is “considered_by(Code_change_proposals, International_fire_code_development_committee),” and the machine-generated logic clause is “considered_by(Designation_f, International_fire_code_development_committee).” The predicate-level performance treats this predicate as one incorrect predicate. The predicate element-level performance treats this predicate as three elements, two of which are correct and one is incorrect. Because of the phenomenon that predicates could be partially correct, predicate element-level accuracy could provide a more accurate evaluation regarding the performance of the ruleset expansion method and pattern matching-based rules.

The performance of the expanded pattern matching-based regulatory information transformation ruleset is summarized in Table 0.2. The performance was measured at the predicate level and the predicate element level. The experiment focuses on logic clauses about quantitative requirements because the original ruleset focused on quantitative requirements. In the research in this chapter, sentences of building code provisions and generated logic clauses have a one-to-many mapping relationship. Patterns, on the other hand, can match the whole sentence or part of a sentence. Regulatory information that spans over multiple sentences is represented by multiple logic clauses. The original ruleset filters quantitative and nonquantitative requirements automatically. Therefore, there is no completely missed logic clause. The logic clauses generated that were not in the gold standard were counted as false positives. The logic clauses generated that functioned in the same way as those in the gold standard were counted as true positives. The logic clause-level performance is reported in Table 0.3. The predicate element-level performance was higher than the predicate-level performance, which indicates some predicates were partially correct. Through error analysis, the authors recognized four main sources of errors:

1. The partially correct predicates. After reviewing machine-generated logic clauses and the gold standard, the authors found that a significant portion of predicates in machine-generated logic clauses were partially correct. For example, when the correct predicate in the gold standard is “surrounded_by(Buildings,Public_ways),” the expanded ruleset generated a partially correct predicate “surrounded_by(Chapter_9,Public_ways).” Future development about pattern matching-based rules could focus more on capturing the correct elements in predicates.
2. The flexibility of B-Prolog logic clauses and the ambiguity of natural language. The flexibility of B-Prolog logic clauses and the inherent ambiguity of natural language left a room of interpretation to the annotators. In other words, it was possible to represent the same building code requirement in different predicates. For example, one annotator translated the “minimum fire resistant rating of 1 hour” to “greater_than(Minimum_fire_resistance_rating, quantity(1, hour)).” Another annotator translated the same phrase to “equal to (Minimum_fire_resistance_rating, quantity(1, hour)).” One annotator considered that fire resistant rating of a subject should be greater than the minimum fire-resistant rating, which is 1 h. Another annotator considered that this phrase means the minimum fire-resistant rating of a subject should be 1 h. Therefore, the precision of the rule generation was affected. A viable solution to increase inter-annotator agreement may include more detailed and stricter annotation guidelines.
3. The patterns and terminologies unseen in Chapter 10. Although most regulatory information patterns in Chapter 5 were captured in Chapter 10, a small amount of regulatory information patterns were missed. The authors attributed missed building code requirements to unseen patterns or unique terminologies in Chapter 5. Chapter 5 and Chapter 10 of the IBC 2015 focus on different topics (i.e., general building height and area, and means of egress,

respectively), so some terminologies in Chapter 5 did not occur in Chapter 10. For example, the erroneous predicate “unobstructed_to(Be,Room)” was generated instead of “unobstructed_to(Room)” due to a pattern that did not occur in Chapter 10. Such error will need to be addressed by accumulatively expanding the training dataset.

4. The backward compatibility requirement. The generality and validity requirements of the ruleset expansion method ensures the quality of the generated logic clauses and shows a promising future that pattern matching rule-based regulatory information extraction can potentially capture all building code requirements with a sufficiently comprehensive set of rules. However, the compatibility requirement also forbade the removal of any existing rule, which led to some false positives. In the future, the flexibility of modifying existing rules may need to be tested.

Table 0.2. Performance of Applying Ruleset Expansion Method

	Training				Testing			
	Predicate level		Predicate element level		Predicate level		Predicate element level	
	Before[1]	After[2]	Before[1]	After[2]	Before[1]	After[2]	Before[1]	After[2]
Precision	84.35%	96.31%	87.90%	98.33%	86.17%	95.17%	90.03%	97.48%
Recall	79.00%	98.38%	81.78%	99.39%	76.84%	96.60%	81.77%	98.65%
F1-score	81.59%	97.34%	84.73%	98.86%	81.24%	95.88%	85.70%	98.06%

[1] Performance of the original ruleset, which is the ruleset before the application of the ruleset expansion method.

[2] Performance of the expanded ruleset, which is the ruleset after the application of the ruleset expansion method.

Table 0.3. Logic Clause-Level Performance

	Training		Testing	
	Before[1]	After[2]	Before[1]	After[2]
Precision	89.86%	100.00%	93.81%	100.00%
Recall	98.27%	99.68%	96.81%	97.98%
F1-score	93.87%	99.84%	95.29%	98.98%

[1] Performance of the original ruleset, which is the ruleset before the application of the ruleset expansion method.

[2] Performance of the expanded ruleset, which is the ruleset after the application of the ruleset expansion method.

4.5 Discussion

Despite all the development and advancement of automated code compliance checking systems, existing ACC systems still heavily rely on domain experts to extract building code requirements and formalize them into a computer-processable format (Zhong et al., 2012; Zhang & El-gohary, 2015), such as decision tables (Tan et al., 2010), knowledge models (Dimyadi et al., 2016), or structured rulesets (İlal et al., 2017). The research in this chapter builds upon cutting-edge semantic NLP-based information extraction and transformation approach (Zhang & EL-Gohary, 2013; Zhang & El-Gohary, 2015; Zhang, 2015) and expands previous efforts to support automated regulatory information extraction from a wide range of building codes with little marginal cost.

In the experiment of the regulatory information transformation ruleset expansion, the proposed method was tested for expanding the range of checkable building code requirements to new chapters of building codes, which are in different domains/topics of the building code from which the original ruleset was initially developed. To further evaluate the robustness of the expanded ruleset, the authors also tested it on processing construction contracts, a fundamentally different type of construction documents compared to building codes. Nine free and openly available construction contracts or construction contract templates were collected. In total, 185 sentences were extracted from these contracts. The expanded ruleset was then executed to convert the extracted sentences in construction contracts to logic clauses. The performance of the expanded ruleset is illustrated in Table 0.4. Examples of contract contents and corresponding logic clauses generated are shown in Table 0.5. The results show the robustness of the expanded ruleset is promising (although not perfect) for processing construction documents beyond the original intent of building codes.

Table 0.4. Performance on Processing Construction Contract

	Predicate level	Predicate element level
Precision	90.52%	97.20%
Recall	92.92%	98.42%
F1-score	91.70%	97.81%

Table 0.5. Examples of Contract Sentences and Corresponding Logic Clauses Generated

Contract sentence	Logic clause
Two copies of the Contract Documents shall be signed by the Owner and the Contractor. (Montrose County, 2018)	compliance_Owner3(Owner):-number_prep(Number),copies(Copies),has(Copies,Number),contract_documents(Contract_Documents),has(Contract_Documents,Copies),owner(Owner);contractor(Owner)),signed_by(Contract_Documents,Owner),equal_to(Number,quantity(2,one)).
Contractor shall maintain in a safe place at the Property one record copy of all drawings, specifications, addenda, written amendments, and the like in good order and annotated to show all changes made during construction, which will be delivered to Owner upon completion of the Work. (Legaltemplates, 2022)	compliance_Number1(Number):-maintain_in(Contractor,Safe_place),contractor(Contractor),safe_place(Safe_place),at(Safe_place,Property),property(Property),number_prep(Number),record_copy(Record_copy),has(Record_copy,Number),drawings(Drawings),has(Drawings,Record_copy),like_in(Like,Good_order),like(Like),good_order(Good_order),to_show(Good_order,Changes),changes(Changes),made_during(Changes,Construction),construction(Construction),delivered_to(Good_order,Owner),owner(Owner),upon(Owner,Completion),completion(Completion),work(Work),has(Work,Completion),equal_to(Number,quantity(1,one)),associated(Safe_place,Good_order).
If the final amount of the ALLOWANCE work is less than the ALLOWANCE line item amount listed in the Agreement, a credit will be issued to Owner after all billings related to this particular line item ALLOWANCE work have been received by Contractor. (Building Advisor, 2019)	compliance_Final_amount7(Final_amount):-final_amount(Final_amount),if(Final_amount),allowance_work(ALLOWANCE_work),has(ALLOWANCE_work,Final_amount),allowance_line_item_amount(ALLOWANCE_line_item_amount),listed_in(ALLOWANCE_line_item_amount,Agreement),agreement(Agreement),credit(Credit),owner(Owner),issued_to(Credit,Owner),after(Owner,Billings),billings(Billings),related_to(Billings,This_particular_line_item_ALLOWANCE_work),this_particular_line_item_allowance_work(This_particular_line_item_ALLOWANCE_work),received_by(This_particular_line_item_ALLOWANCE_work,Contractor),contractor(Contractor),less_than(Final_amount,quantity(1,ALLOWANCE_line_item_amount)).

4.6 Contributions to the Body of Knowledge

This chapter's research contributes to the body of knowledge in four main ways. First, it proves the feasibility of expanding the range of checkable building code requirements by expanding an existing regulatory information transformation ruleset. The authors expanded the range of checkable building code requirements of an automated code compliance checking system to cover Chapter 5 and Chapter 10 of the IBC 2015. This expansion was achieved by 64 new rules. It shows that different chapters of the IBC share similar patterns, and the number of new pattern matching-based rules needed to expand the range of checkable code requirements is small. Second, the research in this chapter was conducted to provide a new ruleset expansion method. This method ensures the quality of added pattern matching-based rules and, therefore, the quality of logic clauses generated by the pattern matching-based rules. In a previous study, three hundred and six rules were developed to cover two chapters of building code. In comparison, only sixty-four new rules were developed to cover two new chapters of building code. It shows that the marginal cost of expanding the range of checkable building code requirements is low. It provides a workable and low-cost method to expand the range of checkable code requirements of ACC systems. The cost of expanding the range of checkable building codes by expanding an existing regulatory information transformation ruleset could further decrease in the future as the number of existing rules increases, because building codes share similar patterns and the number of unseen patterns in new building codes could decrease as existing pattern matching-based rules cover more patterns in building codes. Future researchers and developers can adopt this method to expand the range of checkable code requirements of the ACC system and bring the ACC system to full deployment in the AEC industry. While the research in this chapter has a main focus on processing building codes, the testing results of transforming construction contracts show that the proposed ruleset expansion method is potentially robust in processing different types of construction documents. Third, the research in this chapter also generated a dataset of building codes in logic clauses. This dataset can facilitate other regulatory information transformation research, such as machine learning-based logic clause generation. Last but not least, the research in this chapter facilities the adoption of ACC in the AEC industry. With an expanded range of checkable code requirements, the utility of ACC is enhanced. ACC can reduce the time, cost, and human-errors in code compliance checking and encourages the AEC industry to shift towards a digital paradigm.

4.7 Conclusion

The research in this chapter was conducted to provide a ruleset expansion method that can expand the range of checkable code requirements of ACC systems to different chapters of the IBC, which can potentially be applied to other codes beyond the IBC and other construction documents such as contracts. The proposed method takes an iterative approach to ensure the generality and validity of the added pattern matching-based rules and the generated logic clauses. Experimental results on Chapters 5 and 10 of IBC 2015 showed the expanded ruleset generated logic clauses with 95.17% predicate-level precision, 96.60% predicate-level recall, and 95.88% predicate-level F1-score. This performance proved the effectiveness of the ruleset expansion method and the expanded ruleset. Through error analysis, the authors attributed the remaining errors to the flexibility of B-Prolog language, the ambiguity in natural language, missed building code requirement patterns, and the compatibility requirement. The authors also suggested solutions to further increase the performance of the ruleset expansion method, such as expanding the training dataset and providing stricter annotation guidelines. The research in this chapter also generated a dataset of logic clauses. This dataset has the potential to facilitate research on different regulatory information transformation approaches, such as machine learning-based logic clause generation. The research findings in this chapter can be used to build fully automated building code compliance checking systems with wider code requirements coverage than the state of the art. The demonstrated decreasing marginal cost of transformation rule development and high predicate-level performance renders the rule-based processing of building code requirements promising to bring fully automated building code compliance checking to real-world applications. Future research is needed to discover the boundary of the theoretical “superset” of common patterns used in building code transformation rules, for guiding the practical implementation of the demonstrated rule-based processing of building code requirements in real ACC systems. Furthermore, the successful demonstration of such processing in construction contracts in the research in this chapter helps open the door to rule-based processing of a variety of textual documents in the AEC industry, to support future automation and AI applications in the AEC industry in general.

4.8 Acknowledgement

The authors would like to thank the National Science Foundation (NSF). The material in this chapter is based on work supported by the NSF under Grant No. 1827733. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

5 SEMI-AUTOMATED GENERATION OF LOGIC RULES FOR TABULAR INFORMATION IN BUILDING CODES TO SUPPORT AUTOMATED CODE COMPLIANCE CHECKING

Xiaorui Xue, S.M.ASCE[1]; Jin Wu, S.M.ASCE[2]; Jiansong Zhang, Ph.D., A.M.ASCE[3]

A portion of this chapter was previously published by:

Xue, X., Wu, J., Zhang, J. (2022). Semi-automated generation of logic rules for tabular information in building codes to support automated code compliance checking. *Journal of Computing in Civil Engineering*, *36*(1), 04021033. https://doi.org/10.1061/(ASCE)CP.1943-5487.0001000

Author Contributions

The authors confirmed contribution to the paper as follows:

Study conception and design: Xiaorui Xue, Jin Wu, Jiansong Zhang.

Data collection: Xiaorui Xue, Jin Wu, Jiansong Zhang.

Analysis and interpretation of results: Xiaorui Xue, Jin Wu, Jiansong Zhang.

Draft manuscript preparation: Xiaorui Xue, Jin Wu, Jiansong Zhang.

All authors reviewed the results and approved the final version of the manuscript.

5.1 Literature Review

5.1.1 Existing Work in Automated Building Code Compliance Checking

From a historical perspective, the traditional building code compliance checking process, or building plan review, is a laborious, time-consuming, and error-prone process that demands automation (Alghamdi et al., 2017; Lee et al., 2018; Preidel & Borrmann, 2017). The automation of the code compliance checking process can significantly cut its cost, time, and manual efforts. In the manual code compliance checking process, designers need to wait a long time for building authorities to issue a building permit or ask for further modifications to the design documents, and may have to modify design documents multiple cycles. The plan review process may last a few months (City of San Clemente, 2019). On the other hand, automated code compliance checking systems can return compliance checking results in a much shorter time with a limited need for manual input. Thus, automated building code compliance checking is faster and cheaper than the traditional manual code compliance checking approach.

The automated code compliance checking systems emerged in the 1960s when Fenves introduced decision tables to check the design of steel structures (Fenves, 1966). Systems for checking different aspects of building design were then developed over the years. For example, Pauwels et al. (2011) implemented a sematic rule checking environment to check the acoustic performance of buildings. Tan et al. (2010) provided a series of decision tables to check the design of building envelopes. Getuli et al. (2017) developed a BIM-based workflow that checks against the compliance of Italian construction safety and health code. Malsane et al. (2015) suggested an Industry Foundation Class (IFC)-powered, object-oriented approach to check against fire codes in England and Wales. Bus et al. (2019) developed an ontology-based system to achieve automated compliance checking of semantic rules in French fire safety and accessibility codes. However, existing automated code compliance checking systems only check a limited set of code rules and, according to the authors' literature review, never automatically processed building code requirements in tables.

5.1.2 Table Processing

The demand to extract information from documents that are not in plain textual formats, such as tables and images that are hard for machines to process, is urgent (Correa & Zander, 2017). Most existing methods take a two-step approach to extract tabular information: (1) table detection, and (2) table sub-structure identification (i.e., cells, rows, columns) (Paliwal et al., 2019). Challenges in tabular information extraction include: (1) reliance on the context of tables to interpret tables, (2) document indexing, (3) database curation, and (4) abbreviation of phases (Shmanina et al., 2016). Table detection algorithms can construct table hierarchies in two approaches: top-down and bottom-up. In the top-down approach, the algorithm first identifies tables in documents and then slice identified tables into components. On the contrary, in the bottom-up approach, the algorithm first identifies components of tables and then assembles the components to tables (Krüpl & Herzog, 2006). Different technologies were developed to process table information for various purposes. For example, Vasileiadis et al. (2017) developed a rule-based, bottom-up tabular information extraction system for access by visually impaired people. Buitelaar et al. (2006) published an ontology-based table processing method to extract information from webpages as part of a multi-modal dialog system. Shafait and Smith (2010) used Optical Character Recognition (OCR) technology to process tables with different layouts for analyzing tables in heterogeneous documents. Qasim et al. (2019) treated the table detection problem as a graph problem and generated a Graph Neural Network to detect the structure of tables, which was successfully tested on public table detection datasets (e.g., UW3, UNLV, and ICDAR 2013). Sinha et al. (2019) used OCR to localize tables in Piping and Instrumentation Diagrams (P&IDs) and used regular expressions to enhance the accuracy of text extraction. Although most researchers treated table detection and table structure identification as two separate steps, Paliwal et al. (2019) proposed TableNet, a neural network with an encoder-decoder structure, to detect table existence and identify table structure in one unified step jointly. Although existing table processing methods reached high accuracy on their respective domains, they did not touch upon automation of processing tables in building codes, and data from such tables were still manually interpreted and processed.

5.2 Methodology

In this chapter, the authors proposed a semi-automated table processing method for tables in building codes. The proposed method takes a two-step approach to process tabular information in building codes: (1) tabular information extraction, and (2) information conversion to databases. The developed method needs to be robust over a wide range of tables, i.e., to be able to process tables in an unseen format. The tabular information extraction method needs to extract building code requirements from tables in building codes and store the extracted information in a structured format. The extraction process needs to reach a very high precision to meet the 100% recall goal of noncompliance detection in automated code compliance checking (Salama & El-Gohary, 2016). The format to store extracted building code requirements needs to support easy information access and processing to ensure the performance of the automated code compliance checking system. Integrated methods that directly convert building code tables to logic rules and store these rules in automated code compliance checking system are the most straightforward and intuitive method to process building code requirements from tables. However, state-of-the-art integrated methods lack robustness in processing tables in different layouts and the manual effort to maintain integrated methods may not be less than the effort in the manual encoding of building codes per se. The diverse layouts of tables may require customized methods for each table. Frequent updates of building codes will therefore require constant method updates. To address that, the authors proposed the separation of information extraction from rule generation to increase the robustness and reduce the maintenance need of the method.

5.2.1 Information Extraction

The proposed method takes a semi-automated approach to extract tabular information from building codes. Users need to collect tables from building codes in digital format and provide them together with some structural information of these input tables. The method then processes one table at a time. Structural information of tables helps the information extraction method identify the layout of the table. For tables with different layouts, the underlying relationships between cells are different. For example, some tables use a single cell to store an entry of building code requirement, and some tables use an entire row to store an entry of building code requirement. Layouts of tables implicitly specify how tables store building code requirements. One type of table, for example, uses a cell and its corresponding row header and column header to represent one requirement to buildings. Another type of table uses all cells in a row and their corresponding column headers to represent one requirement to buildings. The proposed method uses structural information provided by the users to automatically distinguish layouts of tables and uncover underlying relationships and information inferred by layouts.

The authors took an iterative approach to develop the sub-algorithms in the tabular information extraction algorithm, i.e., the sub-algorithms are continuously improved until they can correctly extract all tabular information from training data. The basic unit of a table is the cell. Cells can be classified into four types: (1) row header, (2) column header, (3) footnote, and (4) content (Figure 0.1). The four types of cells form the body of a table. The finished algorithm can recognize the cell type and connect the information in each cell (e.g., texts, numbers). As a result, the authors developed: (1) a header detection sub-algorithm to recognize the boundaries of cells for each type, (2) a table layout detection sub-algorithm to distinguish layouts of tables, and (3) two information transformation sub-algorithms to connect contents in the cells.

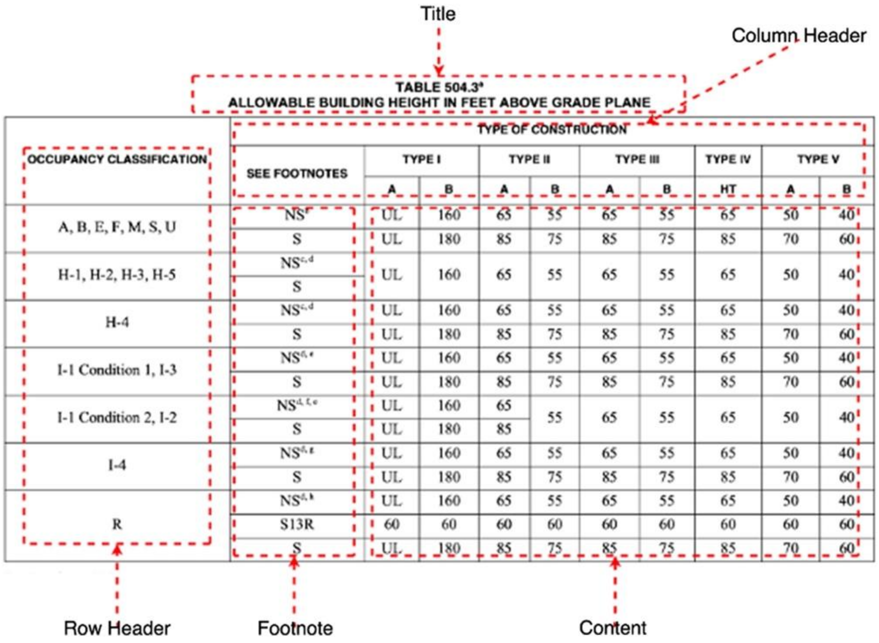


TABLE 504.3[a]
ALLOWABLE BUILDING HEIGHT IN FEET ABOVE GRADE PLANE

OCCUPANCY CLASSIFICATION	TYPE OF CONSTRUCTION									
	SEE FOOTNOTES	TYPE I		TYPE II		TYPE III		TYPE IV	TYPE V	
		A	B	A	B	A	B	HT	A	B
A, B, E, F, M, S, U	NS[b]	UL	160	65	55	65	55	65	50	40
	S	UL	180	85	75	85	75	85	70	60
H-1, H-2, H-3, H-5	NS[c, d]	UL	160	65	55	65	55	65	50	40
	S									
H-4	NS[c, d]	UL	160	65	55	65	55	65	50	40
	S	UL	180	85	75	85	75	85	70	60
I-1 Condition 1, I-3	NS[d, e]	UL	160	65	55	65	55	65	50	40
	S	UL	180	85	75	85	75	85	70	60
I-1 Condition 2, I-2	NS[d, f, e]	UL	160	65	55	65	55	65	50	40
	S	UL	180	85						
I-4	NS[d, g]	UL	160	65	55	65	55	65	50	40
	S	UL	180	85	75	85	75	85	70	60
R	NS[d, h]	UL	160	65	55	65	55	65	50	40
	S13R	60	60	60	60	60	60	60	60	60
	S	UL	180	85	75	85	75	85	70	60

Figure 0.1. Example Table with the Four Types of Cell Components and Title. (Reprinted from IBC 2015 with permission from the International Code Council.)

The header detection sub-algorithm uses the structural information of the table to detect information components. The algorithm requires three inputs from the user for locations of row headers, column headers, and footnotes, respectively. Users then provide: (1) the number of columns used for row headers *X1*, (2) the number of rows used for column headers *X2*, and (3) the number of columns used for footnotes *X3*. There may be no footnotes (i.e., zero for *X3*) or row headers (i.e., zero for *X1*). The header detection sub-algorithm can then automatically identify the locations of different contents and split the table into different information components according to inputs from the user.

After that, the layout detection sub-algorithm distinguishes the layouts of the tables based on their structural information. Tables in building codes have diverse layouts. The authors identified two master layouts based on how the information is organized in a table. Tables with row headers are considered to be in Master Layout One: a single cell is used to store an entry of building code requirement (Figure 0.2). Tables without row headers are considered to be in Master Layout Two: a row of cells is used to store an entry of building code requirement (Figure 0.3). Master layouts ensure the robustness of this algorithm and simplify the information extraction process. The layout detection sub-algorithm can classify all tables in building codes into these two master layouts depending on whether a table has a row header or not. The authors kept the algorithm simple to ensure the robustness of the entire table information processing.

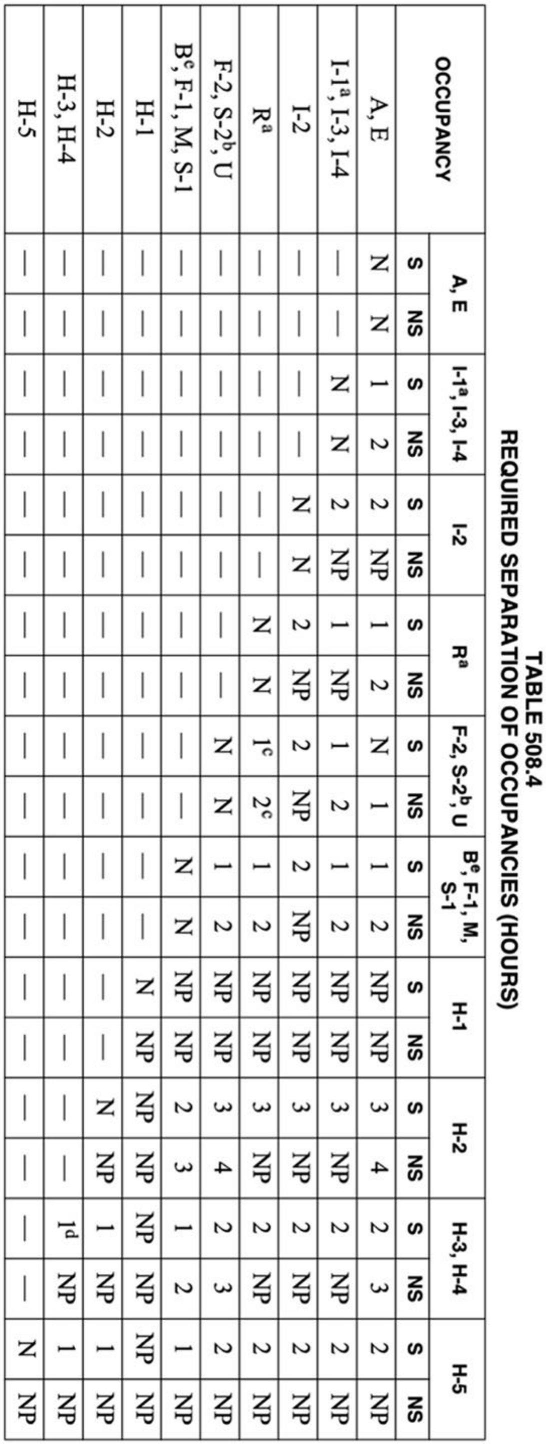


TABLE 508.4
REQUIRED SEPARATION OF OCCUPANCIES (HOURS)

OCCUPANCY	A, E		I-1[a], I-3, I-4		I-2		R[a]		F-2, S-2[b], U		B[e], F-1, M, S-1		H-1		H-2		H-3, H-4		H-5	
	S	NS	S	NS	S	NS	S	NS	S	NS	S	NS	S	NS	S	NS	S	NS	S	NS
A, E	N	N	1	2	2	NP	1	2	N	1	1	2	NP	NP	3	4	2	3	2	NP
I-1[a], I-3, I-4	—	—	N	N	2	NP	1	NP	1	2	1	2	NP	NP	3	NP	2	NP	2	NP
I-2	—	—	—	—	N	N	2	NP	2	NP	2	NP	NP	NP	3	NP	2	NP	2	NP
R[a]	—	—	—	—	—	—	N	N	1[c]	2[c]	1	2	NP	NP	3	NP	2	NP	2	NP
F-2, S-2[b] U	—	—	—	—	—	—	—	—	N	N	1	2	NP	NP	3	4	2	3	2	NP
B[e], F-1, M, S-1	—	—	—	—	—	—	—	—	—	—	N	N	NP	NP	2	3	1	2	1	NP
H-1	—	—	—	—	—	—	—	—	—	—	—	—	N	NP	NP	NP	NP	NP	NP	NP
H-2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	N	NP	1	NP	1	NP
H-3, H-4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	1[d]	NP	1	NP
H-5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	N	NP

Figure 0.2. Example Table in Master Layout 1. (Reprinted from IBC 2015 with Permission from the International Code Council.)

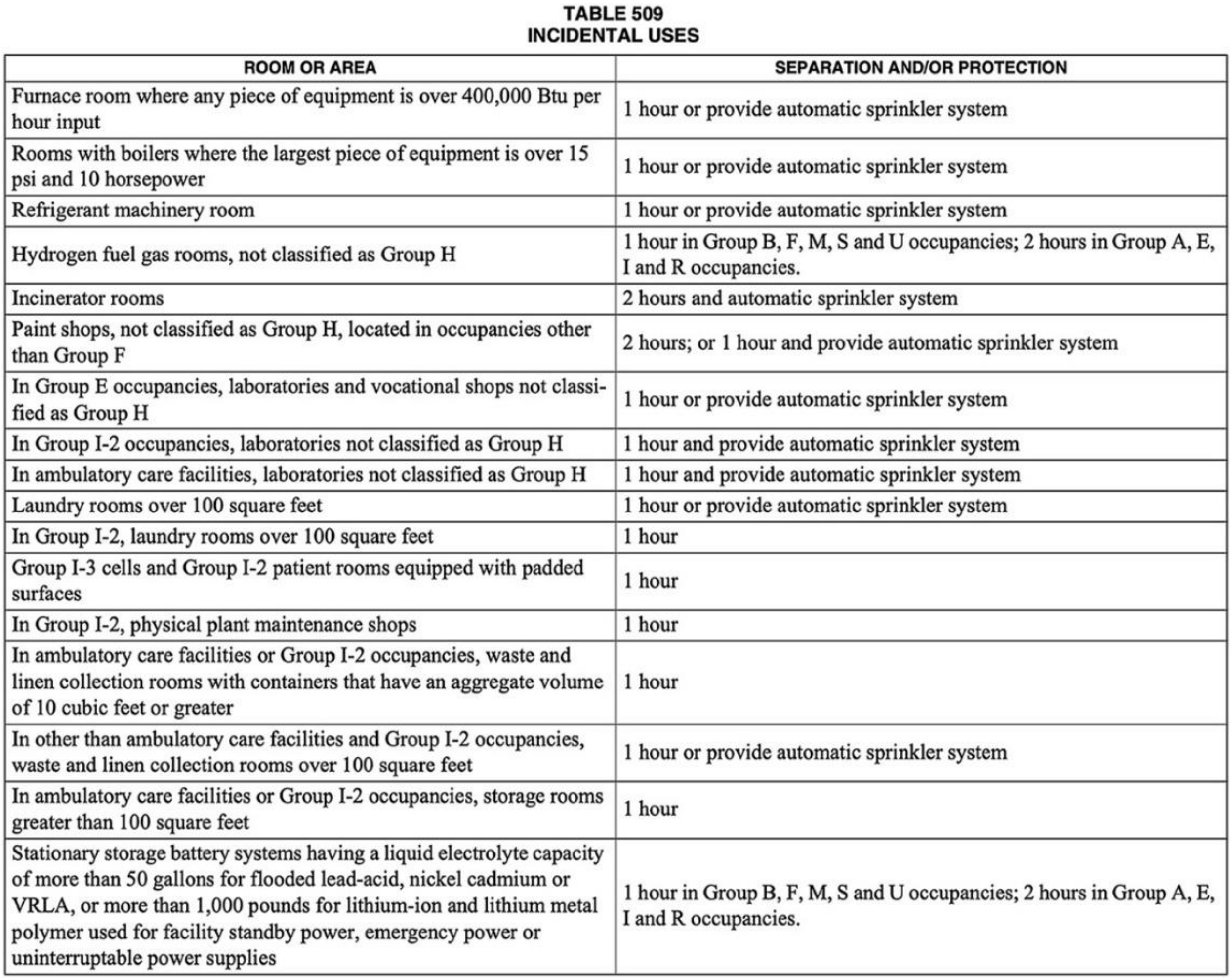


TABLE 509
INCIDENTAL USES

ROOM OR AREA	SEPARATION AND/OR PROTECTION
Furnace room where any piece of equipment is over 400,000 Btu per hour input	1 hour or provide automatic sprinkler system
Rooms with boilers where the largest piece of equipment is over 15 psi and 10 horsepower	1 hour or provide automatic sprinkler system
Refrigerant machinery room	1 hour or provide automatic sprinkler system
Hydrogen fuel gas rooms, not classified as Group H	1 hour in Group B, F, M, S and U occupancies; 2 hours in Group A, E, I and R occupancies.
Incinerator rooms	2 hours and automatic sprinkler system
Paint shops, not classified as Group H, located in occupancies other than Group F	2 hours; or 1 hour and provide automatic sprinkler system
In Group E occupancies, laboratories and vocational shops not classified as Group H	1 hour or provide automatic sprinkler system
In Group I-2 occupancies, laboratories not classified as Group H	1 hour and provide automatic sprinkler system
In ambulatory care facilities, laboratories not classified as Group H	1 hour and provide automatic sprinkler system
Laundry rooms over 100 square feet	1 hour or provide automatic sprinkler system
In Group I-2, laundry rooms over 100 square feet	1 hour
Group I-3 cells and Group I-2 patient rooms equipped with padded surfaces	1 hour
In Group I-2, physical plant maintenance shops	1 hour
In ambulatory care facilities or Group I-2 occupancies, waste and linen collection rooms with containers that have an aggregate volume of 10 cubic feet or greater	1 hour
In other than ambulatory care facilities and Group I-2 occupancies, waste and linen collection rooms over 100 square feet	1 hour or provide automatic sprinkler system
In ambulatory care facilities or Group I-2 occupancies, storage rooms greater than 100 square feet	1 hour
Stationary storage battery systems having a liquid electrolyte capacity of more than 50 gallons for flooded lead-acid, nickel cadmium or VRLA, or more than 1,000 pounds for lithium-ion and lithium metal polymer used for facility standby power, emergency power or uninterruptable power supplies	1 hour in Group B, F, M, S and U occupancies; 2 hours in Group A, E, I and R occupancies.

Figure 0.3. Example Table in Master Layout 2. (Reprinted from IBC 2015 with Permission from the International Code Council.)

The end product of this step is a database that stores information from the table. The information conversion sub-algorithm connects information in different components of a table and inserts connected information into the database. Each master layout has a customized information conversion sub-algorithm. Customized information conversion sub-algorithm ensures the correct extraction of information inferred by the layout of tables. Tables in the same master layout use the same information conversion sub-algorithm. For tables in the same master layout, variations exist, such as having or not having a column for footnotes, having or not having a different number of rows in the column header. The information transformation sub-algorithms are sufficiently robust to process such variations of tables in the same master layout.

The sub-algorithm for the Master Layout One, which is for tables that use a single cell to store an entry of building code requirement, connects the cell, its corresponding row header and column header, and its corresponding footnote (if exists) together and generates a command to insert the entry of building code requirement into the database. The sub-algorithm for the Master Layout Two, which is for tables that use an entire row to store an entry of building code requirement, connects each cell in the row with its corresponding column header and generates a command to insert the entry of building code requirement into the database. Once a command is generated, both sub-algorithms execute the command to insert building code requirements into the database.

5.3 Experiment

The header detection sub-algorithm, the layout detection sub-algorithm, and two information conversion sub-algorithms were developed based on tables (

Table 0.1) in Chapter 5 of IBC 2015 and were tested on tables in Chapter 10 (Table 0.2) of IBC 2015. Inputs of the developed algorithms were digital tables. Digital tables left less space for errors comparing to tables collected as scanned images. The authors manually inspected the extraction results by the algorithm to examine their performance.

Table 0.1. Header and Cell Count of Training Tables

Table Index	Heading	Number of Headers	Number of Contents
504.3	ALLOWABLE BUILDING HEIGHT IN FEET ABOVE GRADE PLANE	39	120
504.4	ALLOWABLE NUMBER OF STORIES ABOVE GRADE PLANE	102	455
506.2	ALLOWABLE AREA FACTOR (A_t = NS, S1, S13R, or SM, as applicable) IN SQUARE FEET	124	612
508.4	REQUIRED SEPARATION OF OCCUPANCIES (HOURS)	41	200
509	INCIDENTAL USES	2	34

Table 0.2. Header and Cell Count of Testing Tables

Table Index	Heading	Number of Headers	Number of Contents
1004.1.2	MAXIMUM FLOOR AREA ALLOWANCES PER OCCUPANT	2	54
1006.2.1	SPACES WITH ONE EXIT OR EXIT ACCESS DOORWAY	20	52
1006.3.1	MINIMUM NUMBER OF EXITS OR ACCESS TO EXITS PER STORY	2	6
1006.3.2(1)	STORIES WITH ONE EXIT OR ACCESS TO ONE EXIT FOR R-2 OCCUPANCIES	4	8
1006.3.2(2)	STORIES WITH ONE EXIT OR ACCESS TO ONE EXIT FOR OTHER OCCUPANCIES	7	18
1010.1.4.1(1)	MAXIMUM DOOR SPEED MANUAL REVOLVING DOORS	2	10
1010.1.4(2)	MAXIMUM DOOR SPEED AUTOMATIC OR POWER-OPERATED REVOLVING DOORS	2	24
1017.2	EXIT ACCESS TRAVEL DISTANCE	13	20
1020.1	CORRIDOR FIRE-RESISTANCE RATING	11	18
1020.2	MINIMUM CORRIDOR WIDTH	2	14
1029.6.2	CAPACITY FOR AISLES FOR SMOKE-PROTECTED ASSEMBLY	11	20
1029.12.2.1	SMOKE-PROTECTED ASSEMBLY AISLE ACCESSWAYS	16	32

After that, the information conversion sub-algorithm injected the extracted information into databases. The authors used the SQLite database in the implementation of information conversion sub-algorithms (Sqlite Consortium, 2000). Each table was stored in a separate database. Two information conversion sub-algorithms were developed for the two master layouts. The layout detection sub-algorithm selects which information conversion algorithm to use. The information conversion algorithm generates an SQLite insertion command based on the syntax of SQLite and the layout of the table being processed.

5.4 Result

The testing results are presented in Table 0.3. The results showed that the proposed method provided the correct results on eleven testing tables and failed in one. Correctly processed tables are the tables that are correctly converted to databases by the proposed method. The results can be verified manually using queries on the database. The failed table was Table 1006.2.1 (Figure 0.4). Therefore, the proposed method processed 91.67% of the tables in the testing dataset correctly. The reason that the proposed method failed to provide correct results in Table 1006.2.1 was that this table had four levels of column headers. No table in Chapter 5 of 2015 IBC (i.e., training data) had more than two levels of column headers. The authors then updated the developed algorithm to accommodate tables with different levels of column headers. The updated algorithm was then tested on all testing tables again. The updated algorithm provided correct results on all tables.

Table 0.3. Results of Testing

Table Index	Heading	Trained Algorithms	Updated Algorithms
1004.1.2	MAXIMUM FLOOR AREA ALLOWANCES PER OCCUPANT	Success	Success
1006.2.1	SPACES WITH ONE EXIT OR EXIT ACCESS DOORWAY	Fail	Success
1006.3.1	MINIMUM NUMBER OF EXITS OR ACCESS TO EXITS PER STORY	Success	Success
1006.3.2(1)	STORIES WITH ONE EXIT OR ACCESS TO ONE EXIT FOR R-2 OCCUPANCIES	Success	Success
1006.3.2(2)	STORIES WITH ONE EXIT OR ACCESS TO ONE EXIT FOR OTHER OCCUPANCIES	Success	Success
1010.1.4.1(1)	MAXIMUM DOOR SPEED MANUAL REVOLVING DOORS	Success	Success
1010.1.4(2)	MAXIMUM DOOR SPEED AUTOMATIC OR POWER-OPERATED REVOLVING DOORS	Success	Success
1017.2	EXIT ACCESS TRAVEL DISTANCE	Success	Success
1020.1	CORRIDOR FIRE-RESISTANCE RATING	Success	Success
1020.2	MINIMUM CORRIDOR WIDTH	Success	Success
1029.6.2	CAPACITY FOR AISLES FOR SMOKE-PROTECTED ASSEMBLY	Success	Success
1029.12.2.1	SMOKE-PROTECTED ASSEMBLY AISLE ACCESSWAYS	Success	Success

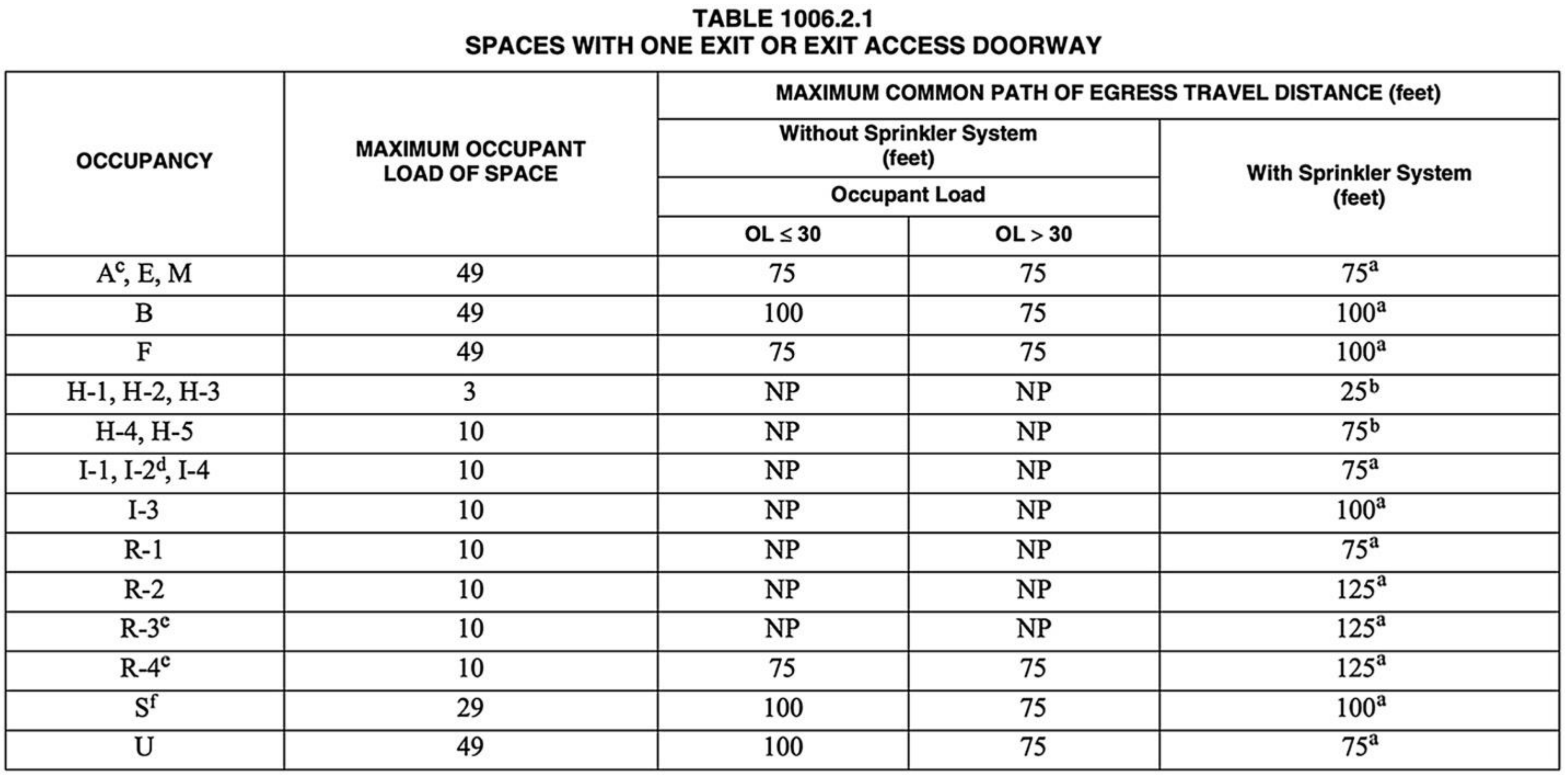


TABLE 1006.2.1
SPACES WITH ONE EXIT OR EXIT ACCESS DOORWAY

OCCUPANCY	MAXIMUM OCCUPANT LOAD OF SPACE	MAXIMUM COMMON PATH OF EGRESS TRAVEL DISTANCE (feet)		
		Without Sprinkler System (feet)		With Sprinkler System (feet)
		Occupant Load		
		OL ≤ 30	OL > 30	
A[c], E, M	49	75	75	75[a]
B	49	100	75	100[a]
F	49	75	75	100[a]
H-1, H-2, H-3	3	NP	NP	25[b]
H-4, H-5	10	NP	NP	75[b]
I-1, I-2[d], I-4	10	NP	NP	75[a]
I-3	10	NP	NP	100[a]
R-1	10	NP	NP	75[a]
R-2	10	NP	NP	125[a]
R-3[c]	10	NP	NP	125[a]
R-4[c]	10	75	75	125[a]
S[f]	29	100	75	100[a]
U	49	100	75	75[a]

Figure 0.4. Table 1006.2.1 from IBC 2015. (Reprinted with Permission from the International Code Council.)

The following experiment was further conducted to test if the information extraction sub-algorithm correctly preserved the information inferred by the layout of tables and correctly extracted building code requirements in the cells. The accuracy of the algorithm was tested by checking if the generated database returns the correct results when queried. Correct results were where the corresponding value of a building code requirement in tables can be successfully returned by the query. For example, when the database for Table 1006.2.1 is queried for the maximum occupant load of space of occupancy Type B, it should return 49. The authors queried every entry in the generated databases for every table in the testing dataset and reviewed the returned values of every query. In 100% of cases, the query returned correct results. The generated database preserves all information inferred by the layout of the tables. Another reason for the 100% accuracy is the authors used digital tables, instead of scanned tables, as inputs to the information extraction sub-algorithm. Errors in recognizing the content of scanned tables were therefore prevented. For example, the algorithm did not suffer from errors in OCR.

5.5 Discussion

In the past decade, automated code compliance checking domain developed at a fast pace. However, limited range of checkable codes hinders wide-spread application of automated code compliance checking systems. No construction industry practitioner will likely use automated code compliance checking systems that require them to manually check part of building codes. After an intense literature review, the authors found that no previous automated code compliance checking research targeted building code requirements in a tabular format, i.e., regulations associated with tables. However, almost all building codes store a large amount of building code requirements in table format. Automated code compliance checking systems that do not cover building code requirements in tables cannot achieve the goal of making automated code compliance checking systems with full coverage (Salama & El-gohary, 2016). The research in this chapter expanded the range of checkable building code requirements of automated code compliance checking systems to tables in building code and facilitates the industry adoption of automated code compliance checking systems.

The following limitations of the building codes tabular information processing method are acknowledged. First, the proposed method requires digital tables as inputs and manual conversion or third-party software to process tables from hard copy or images into digital tables.

Future versions of the proposed method should incorporate the processing of scanned tables, e.g., using OCR functions. Second, the proposed method requires manual inputs in layout detection. The proposed method cannot detect layouts of tables without such inputs from users in spite of the fact that such inputs are minimal. The authors propose to develop a fully automated layout detection algorithm for tables in building codes in the future work.

5.6 Contributions to the Body of Knowledge

The research in this chapter was conducted to provide a new method to extend the range of checkable building code requirements of automated building code compliance checking systems to cover tables in building codes. The contributions to the body of knowledge are four-fold. First, the extension of checkable building code requirements to tables proves the feasibility of checking non-textual building code requirements in a semi-automated way. Second, the research in this chapter could help incorporate more building code requirement details into fully automated code compliance checking systems in a more efficient way, comparing to the state of the art. With an enlarged range of checkable building code requirements, an automated code compliance checking system can provide more value to its users, which could lead to a wider adoption of automated building code compliance checking and synergistically facilitating the adoption of BIM. Third, the authors enhanced the robustness of automated code compliance checking systems. By storing database and generating logic rules on the go, automated code compliance checking systems will benefit from a smaller rule set which has better maintainability comparing to a larger one. Last but not least, the authors calculated that 1,542 logic rules can be generated from tables in the training and test datasets, sourced from 17 tables in two chapters of IBC 2015, which has 35 chapters in total. After interpolation, the authors estimated that the proposed method can help complete about 26,985 new rules with tabular information for IBC 2015. The proposed method can therefore significantly expand the range of checkable building code requirements of ACC systems.

5.7 Conclusion

This proposed method in this chapter incorporated tabular information in building codes into automated code compliance checking systems. The initial tabular information extraction method achieved a 91.67% success rate on tables in the testing dataset. The updated information extraction method could successfully process all tables in the testing dataset and correctly preserved information inferred by the layout of tables. The proposed method still requires minor human input, which the authors will further address in the future work.

5.8 Acknowledgement

The authors would like to thank the National Science Foundation (NSF). The material in this chapter is based on work supported by the NSF under Grant No. 1827733. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

6 DISCUSSION AND SUGGESTIONS FOR FUTURE RESEARCH

6.1 Discussion

This dissertation research is innovative in many ways when compared to the state of the art. NLP research nowadays mostly focused on creating larger and larger machine learning models to exceed state-of-the-art performance on generic datasets (Devlin et al., 2019). This dissertation research successfully addressed a domain-specific NLP task of information extraction from building codes by combining both deep learning and rule-based approaches. (1) The first research question, “How to improve the performance of POS tagging on building codes compared to the state of the art?” was addressed by performing a domain-specific POS tagging of building codes with a new POS tagging method (in Chapter Three). The proposed POS tagger combined error-driven transformational rules (which is illustrated in Chapter Two) and a neural network model. (2) Chapter Four and Chapter Five addressed the research question of “How to expand the range of checkable building code requirements that can be used in state-of-the-art automated code compliance checking systems?” Chapter Four provided a ruleset expansion method that can expand the range of checkable building code requirements while incurring a minimum amount of marginal cost. While earlier automated code compliance checking research has concentrated solely on building code requirements in textual format, this dissertation research presented a mechanism for extracting regulatory information from building code tables (in Chapter Five).

Information extraction methods from building codes in order to further expand the range of checkable building code requirements of automated code compliance checking systems is a future research direction. The findings of this dissertation research can be employed in automated code compliance checking systems to identify nonconformities in building designs. Possible customers of automated building code compliance checking systems supported by the methods and technologies presented in this dissertation research include authorities having jurisdictions that oversee plan review and permit issuance and building designers who want to double-check their designs before submitting them to the government agencies. This dissertation research’s findings can help reduce the duration of code compliance checking from weeks or months to seconds. The productivity of the AEC industry overall can benefit as the speed of compliance checking increases. Construction productivity will be improved as construction projects' overall duration (i.e., spanning its life cycle starting from planning and design) is shortened. While this dissertation research cannot currently offer fully automated code compliance checking, which involves generating code compliance checking results without any user intervention, it can still help plan reviewers speed up the code compliance checking process significantly. In many states, building construction cannot begin without first acquiring a building permit. Automated code compliance checking can shorten the time between submitting a building design and receiving a building permit and therefore can reduce the project's overall duration and lower construction costs significantly (e.g., by reducing cooperative overhead).

6.2 Suggestions for Future Research

The dissertation research points to serval directions of future research, including:

1. Further improving POS tagging accuracy on building codes. As an early step of NLP-based information extraction for automated code compliance checking systems, errors in POS tagging will cascade to future steps of code checking. In this dissertation research, the author reached an accuracy of 96.85% on the PTBC dataset, which advanced the state of the art. However, there is still space for improvement. Therefore, to ensure the best performance of code checking, POS tagging accuracy of building codes still needs to be improved in future research.
2. Identifying previously unknown patterns in POS tagging error fixing. The error-driven transformational rules fixed more than 60% errors that are made by POS taggers. However, 40% of errors still remained unfixed. The error-driven transformational rules used unigram and bigram in patterns, which were not able to fix all errors. Future research could focus on more complex patterns with more features, such as longer patterns and skip-gram patterns, which have the potential to fix the remaining errors.
3. Achieving automated cell classification in tabular information extraction. The proposed method still required some manual effort in cell classification. To achieve full automation in tabular information extraction, automated cell classification needs to be investigated in future research.

APPENDIX A: PATTERNS USED IN EXPANDED PATTERN MATCHING-BASED RULES

1. [complementary subject, candidate subject, candidate compliance checking attribute], inter clause boundary relation, [complementary subject, candidate subject, candidate compliance checking attribute], indicating “part_of” or “belongs_to” relation by the term "of", [complementary subject, candidate subject, candidate compliance checking attribute], Conjunctive Term, [complementary subject, candidate subject, candidate compliance checking attribute].
2. candidate subject, preposition, complementary subject, inter clause boundary relation, candidate subject, adjective, preposition, candidate subject.
3. [candidate subject, complementary subject, comparative relation], inter clause boundary relation, gerund or present participle verb, [candidate subject, complementary subject, comparative relation].
4. [complementary subject, candidate subject, candidate compliance checking attribute], past participle verb, comparative relation, value, unit, conjunctive term, comparative relation, value, unit.
5. complementary subject, modal verb, base form verb, value, unit, comparative relation, comparative relation, conjunctive term, value, unit, adjective, preposition, candidate subject.
6. [complementary subject, candidate subject, candidate compliance checking attribute], modal verb, negation, base form verb, comparative relation, value, unit, preposition, complementary subject.
7. [candidate subject, complementary subject, comparative relation], gerund or present participle verb, inter clause boundary relation, [candidate subject, complementary subject, comparative relation].
8. [candidate subject, complementary subject], inter clause boundary relation, [candidate subject, complementary subject], inter clause boundary relation
9. [candidate subject, complementary subject], slash “/”, [candidate subject, complementary subject].
10. candidate compliance checking attribute, indicating “part_of” or “belongs_to” relation by the term “of”, for each, candidate subject.
11. candidate subject, relation verb, inter clause boundary relation, candidate subject
12. candidate compliance checking attribute, modal verb, base form verb, negation, comparative relation, candidate compliance checking attribute.
13. value, complementary subject, preposition, for each, value, unit, indicating “part_of” or “belongs_to” relation by the term “of”, candidate compliance checking attribute.
14. [candidate subject, complementary subject, candidate compliance checking attribute], preposition, for each..
15. [complementary subject, candidate subject, candidate compliance checking attribute], [non-3rd person singular present verb, modal verb, base form verb],possessive subject restriction, value,[complementary subject, candidate subject, candidate compliance checking attribute].
16. [complementary subject, candidate subject, candidate compliance checking attribute], [non-3rd person singular present verb, base form verb, 3rd person singular present verb],

comparative relation, value, [complementary subject, candidate subject, candidate compliance checking attribute].

17. [candidate subject, complementary subject, comparative relation], inter clause boundary relation, conjunctive term, [candidate subject, complementary subject, comparative relation].
18. complementary subject, preposition, complementary subject, modal verb, negation, base form verb, candidate compliance checking attribute.
19. [complementary subject, candidate subject, candidate compliance checking attribute], indicating "part_of" or "belongs_to" relation by the term "of", complementary subject, non-3rd person singular present verb,3rd person singular present verb, negation, comparative relation, value, unit.
20. [candidate subject, complementary subject, candidate compliance checking attribute], for "with" or "with in" relation, [candidate subject, complementary subject, candidate compliance checking attribute].
21. base form verb, preposition, [candidate subject, complementary subject, candidate compliance checking attribute.
22. candidate subject, modal verb, negation, base form verb, candidate subject.
23. [candidate subject, candidate compliance checking attribute], relation verb, [complementary subject, candidate compliance checking attribute], inter clause boundary relation, complementary subject.
24. [candidate subject, complementary subject, comparative relation], indicating "part_of" or "belongs_to" relation by the term "of", for each, [candidate subject, complementary subject, comparative relation].
25. complementary subject, character, cardinal number.
26. [candidate subject, candidate compliance checking attribute], indicating "part_of" or "belongs_to" relation by the term "of", value, unit, adjective.
27. candidate compliance checking attribute, gerund or present participle verb, inter clause boundary relation, [candidate subject, complementary subject, comparative relation, candidate compliance checking attribute].
28. [complementary subject, candidate subject, candidate compliance checking attribute], preposition, [complementary subject, candidate subject, candidate compliance checking attribute], [preposition, the word "to"], [complementary subject, candidate subject, candidate compliance checking attribute].
29. complementary subject, modal verb, base form verb, preposition, candidate subject, value.
30. candidate compliance checking attribute, modal verb, negation, base form verb, the word "to", candidate subject.
31. [complementary subject, candidate subject, candidate compliance checking attribute], relation verb, value, unit, conjunctive term, value, unit.
32. complementary subject, modal verb, base form verb, negation, comparative relation, value, unit, preposition, candidate compliance checking attribute.
33. candidate compliance checking attribute, conjunctive term, past participle verb, candidate compliance checking attribute.
34. [candidate subject, complementary subject, candidate compliance checking attribute, comparative relation], 3rd person singular present verb, past participle verb.
35. [complementary subject, candidate compliance checking attribute], modal verb, base form verb, relation verb, [complementary subject, candidate subject, candidate compliance

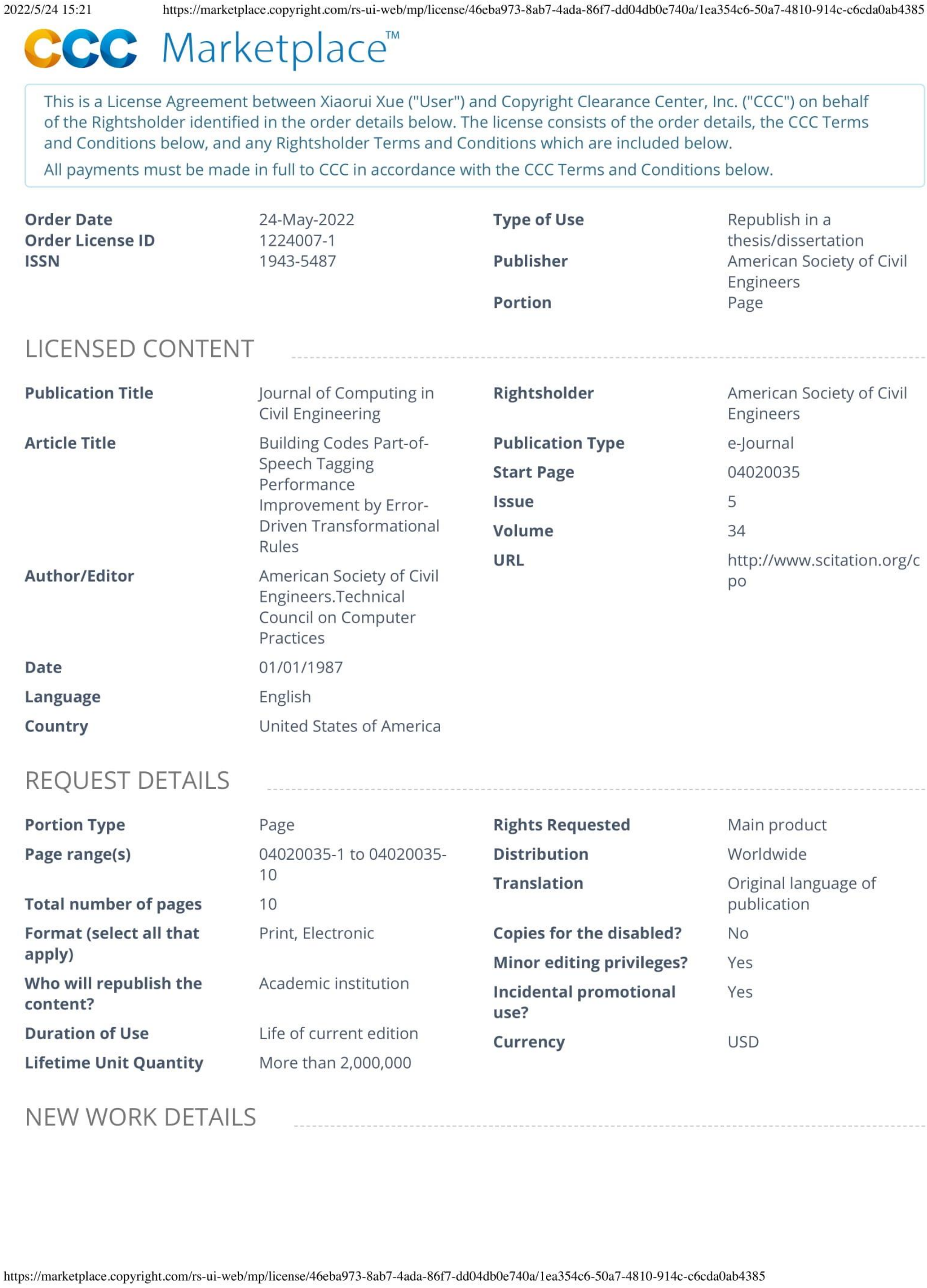
checking attribute.

36. candidate subject, modal verb, negation, base form verb, candidate compliance checking attribute.
37. preposition, value, unit, candidate subject.
38. modal verb, negation, base form verb, [adjective, past participle verb, past tense verb]
39. value, unit, preposition, candidate subject.
40. preposition, value, unit, indicating “part_of” or “belongs_to” relation by the term “of”, candidate subject.
41. relation verb, candidate compliance checking attribute, indicating “part_of” or “belongs_to” relation by the term “of”, cardinal number, conjunctive term, comparative adjective.
42. preposition, past participle verb, candidate compliance checking attribute.
43. complementary subject, the word “to”, value, complementary subject.
44. negation, comparative relation, value, slash “/”, unit, candidate compliance checking attribute.
45. candidate subject, possessive subject restriction.
46. complementary subject, candidate compliance checking attribute.
47. preposition, value, unit, candidate subject, indicating “part_of” or “belongs_to” relation by the term “of”, candidate compliance checking attribute.
48. complementary subject, modal verb, possessive subject restriction, complementary subject.
49. complementary subject, candidate subject, candidate compliance checking attribute], value, conjunctive term, comparative adjective, [complementary subject, candidate subject, candidate compliance checking attribute].
50. adjective, indicating “part_of” or “belongs_to” relation by the term “of”, [candidate subject, complementary subject, candidate compliance checking attribute].
51. preposition, value, unit, preposition, candidate compliance checking attribute.
52. candidate compliance checking attribute, indicating “part_of” or “belongs_to” relation by the term “of”, value, unit, the word “to”, value, unit.
53. [candidate subject, complementary subject, candidate compliance checking attribute, comparative relation], indicating “part_of” or “belongs_to” relation by the term “of”, gerund or present participle verb, [candidate subject, complementary subject, candidate compliance checking attribute, comparative relation].
54. comparative relation, value, [candidate subject, complementary subject].
55. [candidate subject, complementary subject, candidate compliance checking attribute, comparative relation], indicating “part_of” or “belongs_to” relation by the term “of”, comparative adjective, [candidate subject, complementary subject, candidate compliance checking attribute, comparative relation].
56. complementary subject, candidate subject, candidate compliance checking attribute], negation, comparative relation, value, [complementary subject, candidate subject, candidate compliance checking attribute].
57. candidate subject, gerund or present participle verb, candidate compliance checking attribute, indicating “part_of” or “belongs_to” relation by the term “of”, comparative relation, value.
58. negation, comparative relation, value, indicating “part_of” or “belongs_to” relation by the term “of”, candidate compliance checking attribute, indicating “part_of” or “belongs_to” relation by the term “of”, complementary subject.
59. candidate compliance checking attribute, modal verb, base form verb, past participle verb.

60. negation, comparative relation, value, indicating “part_of” or “belongs_to” relation by the term “of”, candidate compliance checking attribute.
61. candidate compliance checking attribute, 3rd person singular present verb, negation, comparative relation, value, unit.
62. candidate subject, comparative relation, value, preposition, complementary subject.
63. negation, possessive subject restriction, comparative relation, value, unit, indicating “part_of” or “belongs_to” relation by the term “of”, candidate compliance checking attribute.
64. candidate subject, preposition, comparative relation, value, unit.

Note: brackets means words in multiple patterns can be fit into the slot of the pattern.

APPENDIX B: PERMISSION FROM PUBLISHER



2022/5/24 15:21 https://marketplace.copyright.com/rs-ui-web/mp/license/46eba973-8ab7-4ada-86f7-dd04db0e740a/1ea354c6-50a7-4810-914c-c6cda0ab4385

CCC Marketplace™

This is a License Agreement between Xiaorui Xue ("User") and Copyright Clearance Center, Inc. ("CCC") on behalf of the Rightsholder identified in the order details below. The license consists of the order details, the CCC Terms and Conditions below, and any Rightsholder Terms and Conditions which are included below.

All payments must be made in full to CCC in accordance with the CCC Terms and Conditions below.

Order Date	24-May-2022	**Type of Use**	Republish in a thesis/dissertation
Order License ID	1224007-1	**Publisher**	American Society of Civil Engineers
ISSN	1943-5487	**Portion**	Page

LICENSED CONTENT

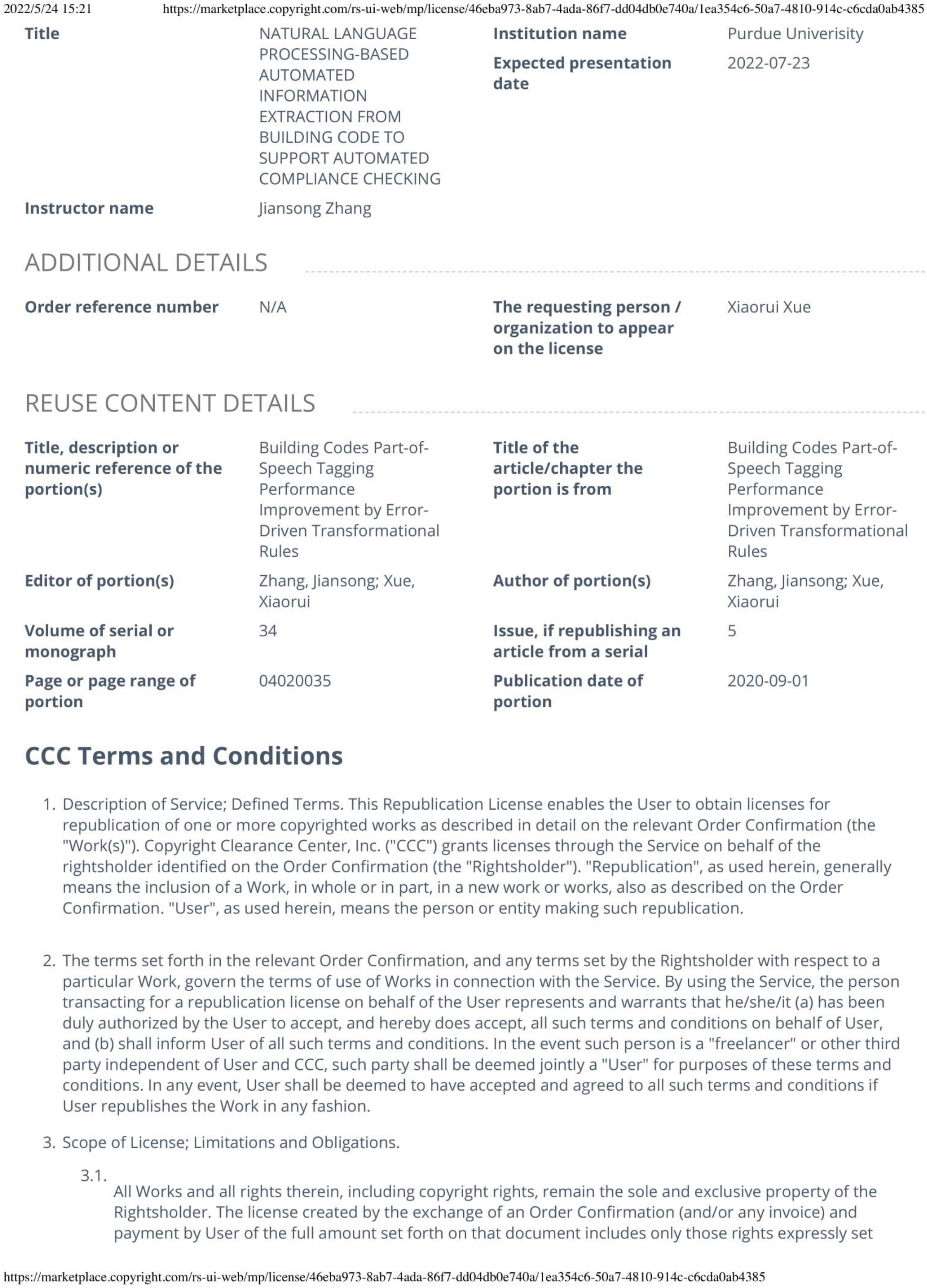
Publication Title	Journal of Computing in Civil Engineering	**Rightsholder**	American Society of Civil Engineers
Article Title	Building Codes Part-of-Speech Tagging Performance Improvement by Error-Driven Transformational Rules	**Publication Type**	e-Journal
		Start Page	04020035
		Issue	5
		Volume	34
		URL	http://www.scitation.org/cpo
Author/Editor	American Society of Civil Engineers.Technical Council on Computer Practices		
Date	01/01/1987		
Language	English		
Country	United States of America		

REQUEST DETAILS

Portion Type	Page	**Rights Requested**	Main product
Page range(s)	04020035-1 to 04020035-10	**Distribution**	Worldwide
		Translation	Original language of publication
Total number of pages	10		
Format (select all that apply)	Print, Electronic	**Copies for the disabled?**	No
		Minor editing privileges?	Yes
Who will republish the content?	Academic institution	**Incidental promotional use?**	Yes
Duration of Use	Life of current edition	**Currency**	USD
Lifetime Unit Quantity	More than 2,000,000		

NEW WORK DETAILS

https://marketplace.copyright.com/rs-ui-web/mp/license/46eba973-8ab7-4ada-86f7-dd04db0e740a/1ea354c6-50a7-4810-914c-c6cda0ab4385



Title	NATURAL LANGUAGE PROCESSING-BASED AUTOMATED INFORMATION EXTRACTION FROM BUILDING CODE TO SUPPORT AUTOMATED COMPLIANCE CHECKING	**Institution name** **Expected presentation date**	Purdue Univerisity 2022-07-23
Instructor name	Jiansong Zhang		

ADDITIONAL DETAILS

Order reference number	N/A	**The requesting person / organization to appear on the license**	Xiaorui Xue

REUSE CONTENT DETAILS

Title, description or numeric reference of the portion(s)	Building Codes Part-of-Speech Tagging Performance Improvement by Error-Driven Transformational Rules	**Title of the article/chapter the portion is from**	Building Codes Part-of-Speech Tagging Performance Improvement by Error-Driven Transformational Rules
Editor of portion(s)	Zhang, Jiansong; Xue, Xiaorui	**Author of portion(s)**	Zhang, Jiansong; Xue, Xiaorui
Volume of serial or monograph	34	**Issue, if republishing an article from a serial**	5
Page or page range of portion	04020035	**Publication date of portion**	2020-09-01

CCC Terms and Conditions

1. Description of Service; Defined Terms. This Republication License enables the User to obtain licenses for republication of one or more copyrighted works as described in detail on the relevant Order Confirmation (the "Work(s)"). Copyright Clearance Center, Inc. ("CCC") grants licenses through the Service on behalf of the rightsholder identified on the Order Confirmation (the "Rightsholder"). "Republication", as used herein, generally means the inclusion of a Work, in whole or in part, in a new work or works, also as described on the Order Confirmation. "User", as used herein, means the person or entity making such republication.

2. The terms set forth in the relevant Order Confirmation, and any terms set by the Rightsholder with respect to a particular Work, govern the terms of use of Works in connection with the Service. By using the Service, the person transacting for a republication license on behalf of the User represents and warrants that he/she/it (a) has been duly authorized by the User to accept, and hereby does accept, all such terms and conditions on behalf of User, and (b) shall inform User of all such terms and conditions. In the event such person is a "freelancer" or other third party independent of User and CCC, such party shall be deemed jointly a "User" for purposes of these terms and conditions. In any event, User shall be deemed to have accepted and agreed to all such terms and conditions if User republishes the Work in any fashion.

3. Scope of License; Limitations and Obligations.

   3.1.
   All Works and all rights therein, including copyright rights, remain the sole and exclusive property of the Rightsholder. The license created by the exchange of an Order Confirmation (and/or any invoice) and payment by User of the full amount set forth on that document includes only those rights expressly set

forth in the Order Confirmation and in these terms and conditions, and conveys no other rights in the Work(s) to User. All rights not expressly granted are hereby reserved.

3.2. General Payment Terms: You may pay by credit card or through an account with us payable at the end of the month. If you and we agree that you may establish a standing account with CCC, then the following terms apply: Remit Payment to: Copyright Clearance Center, 29118 Network Place, Chicago, IL 60673-1291. Payments Due: Invoices are payable upon their delivery to you (or upon our notice to you that they are available to you for downloading). After 30 days, outstanding amounts will be subject to a service charge of 1-1/2% per month or, if less, the maximum rate allowed by applicable law. Unless otherwise specifically set forth in the Order Confirmation or in a separate written agreement signed by CCC, invoices are due and payable on "net 30" terms. While User may exercise the rights licensed immediately upon issuance of the Order Confirmation, the license is automatically revoked and is null and void, as if it had never been issued, if complete payment for the license is not received on a timely basis either from User directly or through a payment agent, such as a credit card company.

3.3. Unless otherwise provided in the Order Confirmation, any grant of rights to User (i) is "one-time" (including the editions and product family specified in the license), (ii) is non-exclusive and non-transferable and (iii) is subject to any and all limitations and restrictions (such as, but not limited to, limitations on duration of use or circulation) included in the Order Confirmation or invoice and/or in these terms and conditions. Upon completion of the licensed use, User shall either secure a new permission for further use of the Work(s) or immediately cease any new use of the Work(s) and shall render inaccessible (such as by deleting or by removing or severing links or other locators) any further copies of the Work (except for copies printed on paper in accordance with this license and still in User's stock at the end of such period).

3.4. In the event that the material for which a republication license is sought includes third party materials (such as photographs, illustrations, graphs, inserts and similar materials) which are identified in such material as having been used by permission, User is responsible for identifying, and seeking separate licenses (under this Service or otherwise) for, any of such third party materials; without a separate license, such third party materials may not be used.

3.5. Use of proper copyright notice for a Work is required as a condition of any license granted under the Service. Unless otherwise provided in the Order Confirmation, a proper copyright notice will read substantially as follows: "Republished with permission of [Rightsholder's name], from [Work's title, author, volume, edition number and year of copyright]; permission conveyed through Copyright Clearance Center, Inc. " Such notice must be provided in a reasonably legible font size and must be placed either immediately adjacent to the Work as used (for example, as part of a by-line or footnote but not as a separate electronic link) or in the place where substantially all other credits or notices for the new work containing the republished Work are located. Failure to include the required notice results in loss to the Rightsholder and CCC, and the User shall be liable to pay liquidated damages for each such failure equal to twice the use fee specified in the Order Confirmation, in addition to the use fee itself and any other fees and charges specified.

3.6. User may only make alterations to the Work if and as expressly set forth in the Order Confirmation. No Work may be used in any way that is defamatory, violates the rights of third parties (including such third parties' rights of copyright, privacy, publicity, or other tangible or intangible property), or is otherwise illegal, sexually explicit or obscene. In addition, User may not conjoin a Work with any other material that may result in damage to the reputation of the Rightsholder. User agrees to inform CCC if it becomes aware of any infringement of any rights in a Work and to cooperate with any reasonable request of CCC or the Rightsholder in connection therewith.

4. Indemnity. User hereby indemnifies and agrees to defend the Rightsholder and CCC, and their respective employees and directors, against all claims, liability, damages, costs and expenses, including legal fees and expenses, arising out of any use of a Work beyond the scope of the rights granted herein, or any use of a Work which has been altered in any unauthorized way by User, including claims of defamation or infringement of rights of copyright, publicity, privacy or other tangible or intangible property.

5. Limitation of Liability. UNDER NO CIRCUMSTANCES WILL CCC OR THE RIGHTSHOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL OR INCIDENTAL DAMAGES (INCLUDING WITHOUT LIMITATION DAMAGES FOR LOSS OF BUSINESS PROFITS OR INFORMATION, OR FOR BUSINESS INTERRUPTION) ARISING OUT OF THE USE OR INABILITY TO USE A WORK, EVEN IF ONE OF THEM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In any event, the total liability of the Rightsholder and CCC (including their respective employees and directors) shall not exceed the total amount actually paid by User for this license. User assumes full liability for the actions and omissions of its principals, employees, agents, affiliates, successors and assigns.

6. Limited Warranties. THE WORK(S) AND RIGHT(S) ARE PROVIDED "AS IS". CCC HAS THE RIGHT TO GRANT TO USER THE RIGHTS GRANTED IN THE ORDER CONFIRMATION DOCUMENT. CCC AND THE RIGHTSHOLDER DISCLAIM ALL OTHER WARRANTIES RELATING TO THE WORK(S) AND RIGHT(S), EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. ADDITIONAL RIGHTS MAY BE REQUIRED TO USE ILLUSTRATIONS, GRAPHS, PHOTOGRAPHS, ABSTRACTS, INSERTS OR OTHER PORTIONS OF THE WORK (AS OPPOSED TO THE ENTIRE WORK) IN A MANNER CONTEMPLATED BY USER; USER UNDERSTANDS AND AGREES THAT NEITHER CCC NOR THE RIGHTSHOLDER MAY HAVE SUCH ADDITIONAL RIGHTS TO GRANT.

7. Effect of Breach. Any failure by User to pay any amount when due, or any use by User of a Work beyond the scope of the license set forth in the Order Confirmation and/or these terms and conditions, shall be a material breach of the license created by the Order Confirmation and these terms and conditions. Any breach not cured within 30 days of written notice thereof shall result in immediate termination of such license without further notice. Any unauthorized (but licensable) use of a Work that is terminated immediately upon notice thereof may be liquidated by payment of the Rightsholder's ordinary license price therefor; any unauthorized (and unlicensable) use that is not terminated immediately for any reason (including, for example, because materials containing the Work cannot reasonably be recalled) will be subject to all remedies available at law or in equity, but in no event to a payment of less than three times the Rightsholder's ordinary license price for the most closely analogous licensable use plus Rightsholder's and/or CCC's costs and expenses incurred in collecting such payment.

8. Miscellaneous.

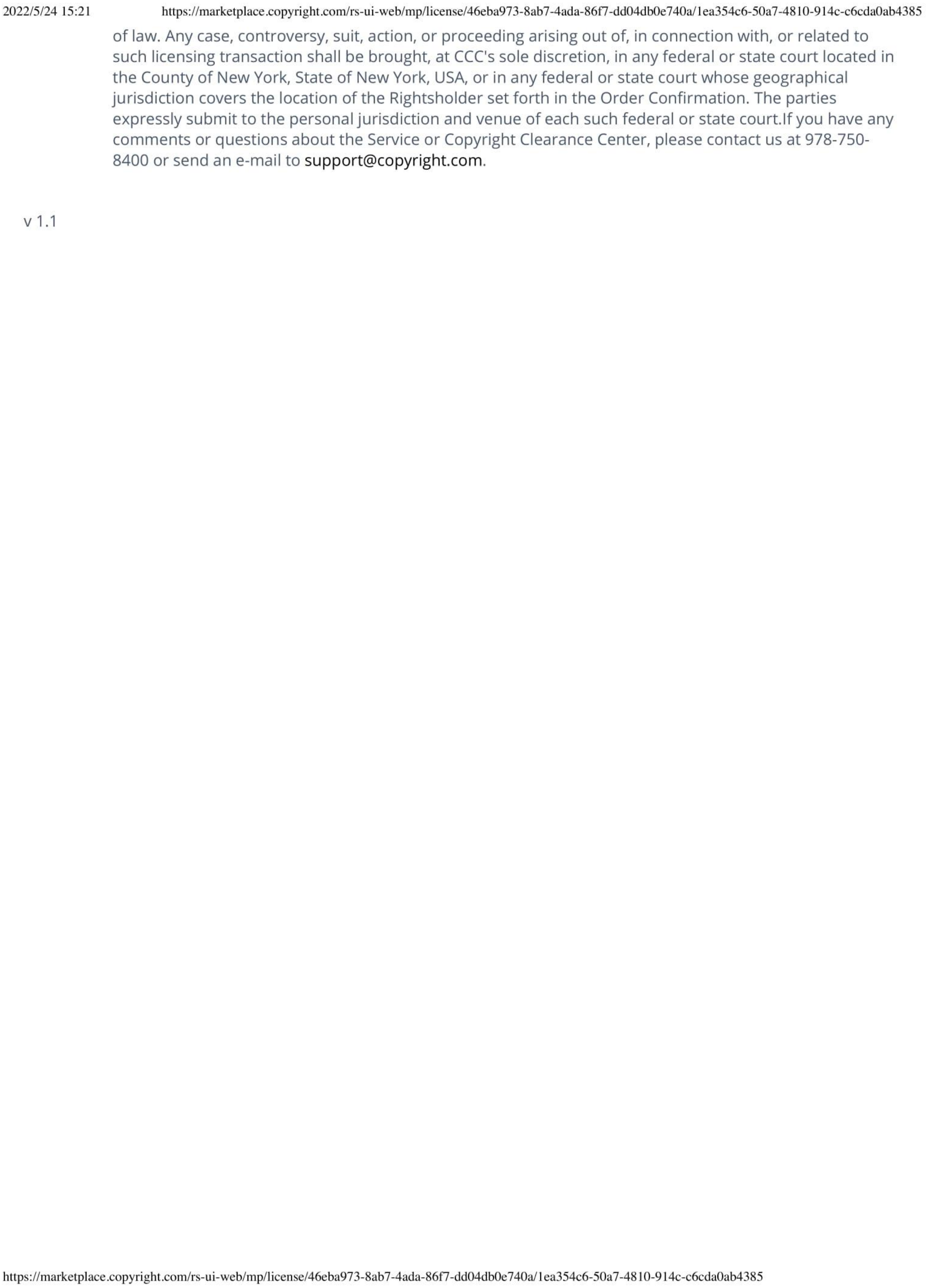
    8.1. User acknowledges that CCC may, from time to time, make changes or additions to the Service or to these terms and conditions, and CCC reserves the right to send notice to the User by electronic mail or otherwise for the purposes of notifying User of such changes or additions; provided that any such changes or additions shall not apply to permissions already secured and paid for.

    8.2. Use of User-related information collected through the Service is governed by CCC's privacy policy, available online here:https://marketplace.copyright.com/rs-ui-web/mp/privacy-policy

    8.3. The licensing transaction described in the Order Confirmation is personal to User. Therefore, User may not assign or transfer to any other person (whether a natural person or an organization of any kind) the license created by the Order Confirmation and these terms and conditions or any rights granted hereunder; provided, however, that User may assign such license in its entirety on written notice to CCC in the event of a transfer of all or substantially all of User's rights in the new material which includes the Work(s) licensed under this Service.

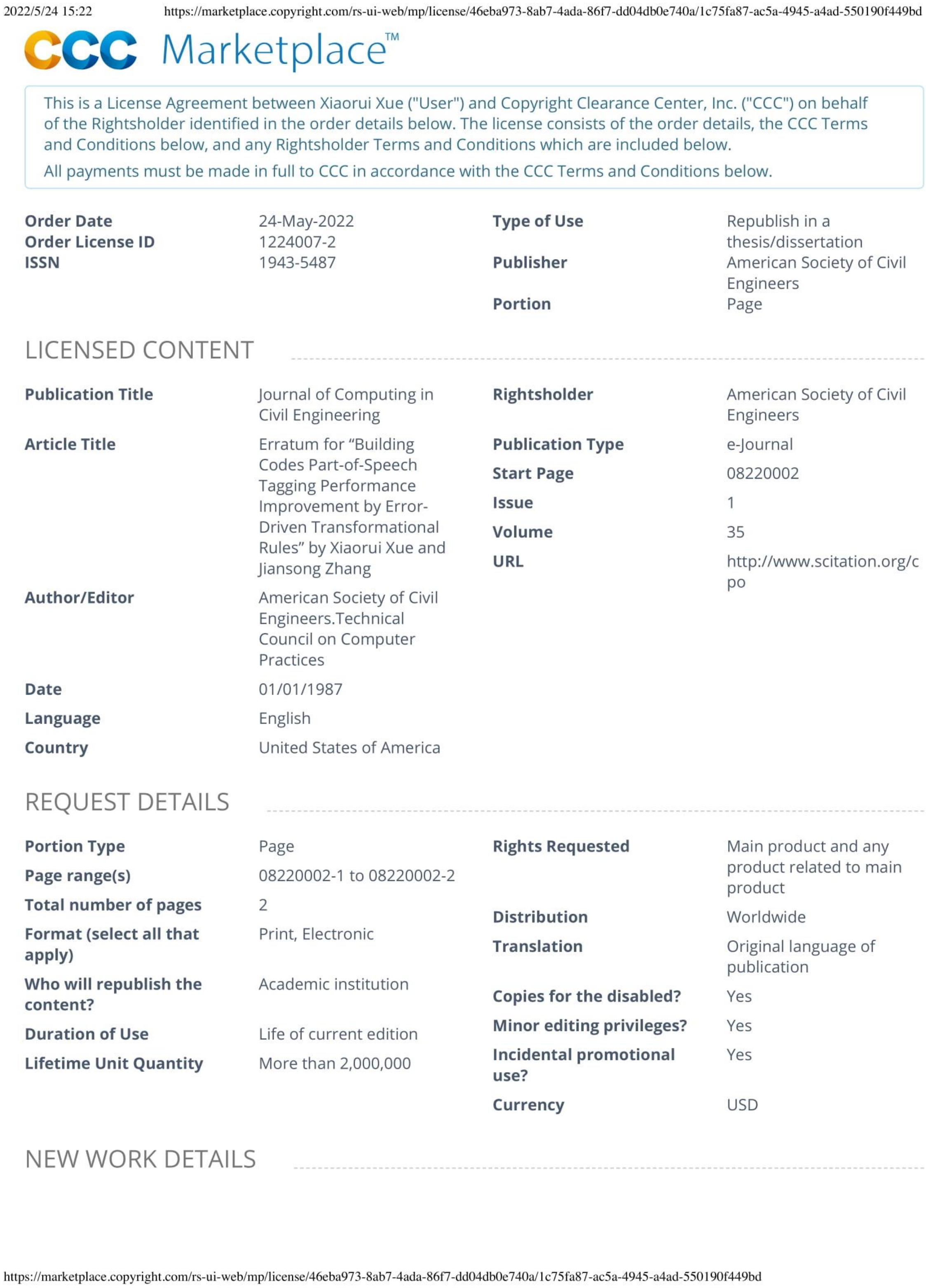
    8.4. No amendment or waiver of any terms is binding unless set forth in writing and signed by the parties. The Rightsholder and CCC hereby object to any terms contained in any writing prepared by the User or its principals, employees, agents or affiliates and purporting to govern or otherwise relate to the licensing transaction described in the Order Confirmation, which terms are in any way inconsistent with any terms set forth in the Order Confirmation and/or in these terms and conditions or CCC's standard operating procedures, whether such writing is prepared prior to, simultaneously with or subsequent to the Order Confirmation, and whether such writing appears on a copy of the Order Confirmation or in a separate instrument.

    8.5. The licensing transaction described in the Order Confirmation document shall be governed by and construed under the law of the State of New York, USA, without regard to the principles thereof of conflicts



of law. Any case, controversy, suit, action, or proceeding arising out of, in connection with, or related to such licensing transaction shall be brought, at CCC's sole discretion, in any federal or state court located in the County of New York, State of New York, USA, or in any federal or state court whose geographical jurisdiction covers the location of the Rightsholder set forth in the Order Confirmation. The parties expressly submit to the personal jurisdiction and venue of each such federal or state court.If you have any comments or questions about the Service or Copyright Clearance Center, please contact us at 978-750-8400 or send an e-mail to support@copyright.com.

v 1.1



CCC Marketplace™

This is a License Agreement between Xiaorui Xue ("User") and Copyright Clearance Center, Inc. ("CCC") on behalf of the Rightsholder identified in the order details below. The license consists of the order details, the CCC Terms and Conditions below, and any Rightsholder Terms and Conditions which are included below.

All payments must be made in full to CCC in accordance with the CCC Terms and Conditions below.

Order Date	24-May-2022	**Type of Use**	Republish in a thesis/dissertation
Order License ID	1224007-2	**Publisher**	American Society of Civil Engineers
ISSN	1943-5487	**Portion**	Page

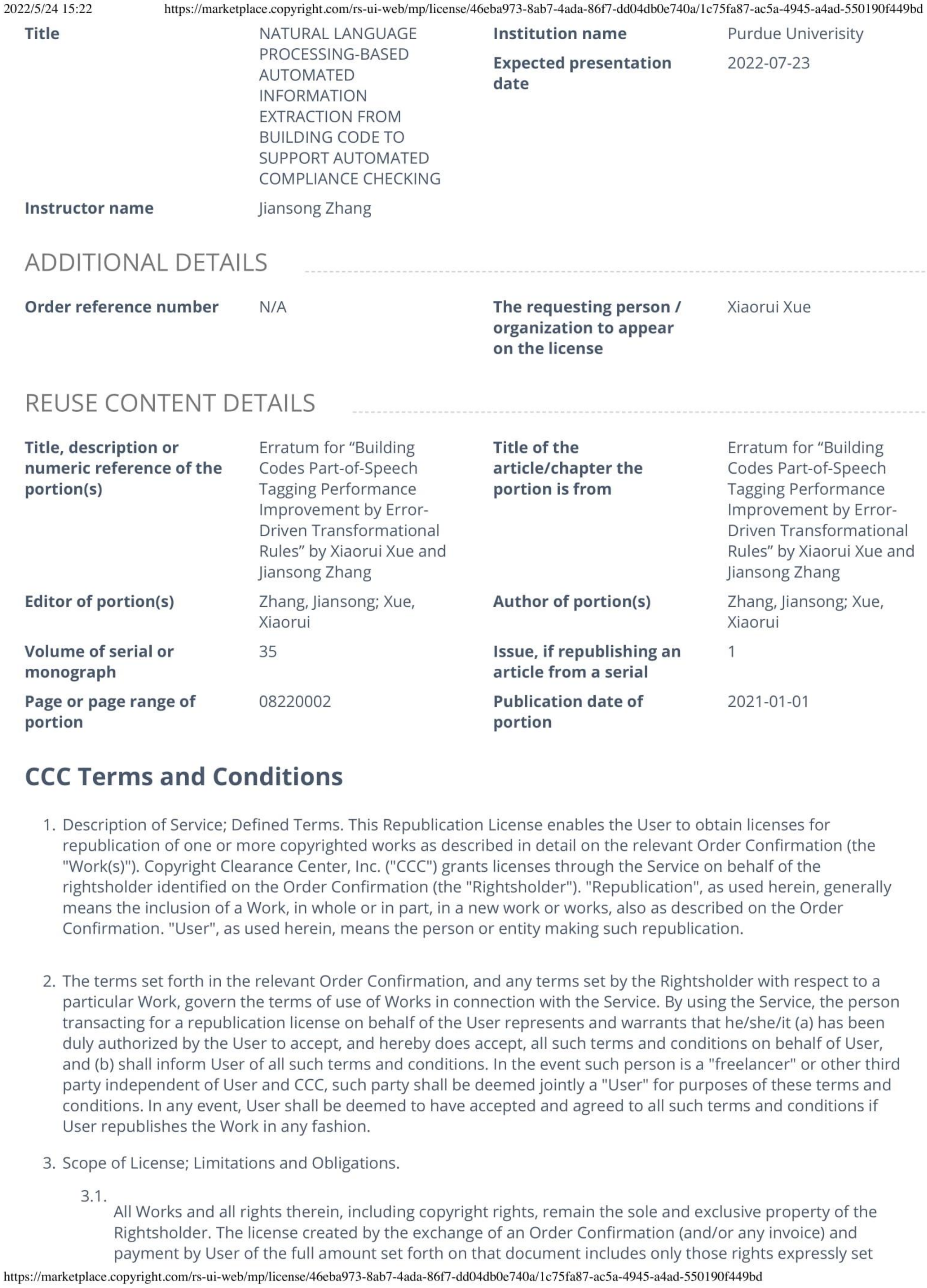
LICENSED CONTENT

Publication Title	Journal of Computing in Civil Engineering	**Rightsholder**	American Society of Civil Engineers
Article Title	Erratum for "Building Codes Part-of-Speech Tagging Performance Improvement by Error-Driven Transformational Rules" by Xiaorui Xue and Jiansong Zhang	**Publication Type**	e-Journal
		Start Page	08220002
		Issue	1
		Volume	35
		URL	http://www.scitation.org/cpo
Author/Editor	American Society of Civil Engineers.Technical Council on Computer Practices		
Date	01/01/1987		
Language	English		
Country	United States of America		

REQUEST DETAILS

Portion Type	Page	**Rights Requested**	Main product and any product related to main product
Page range(s)	08220002-1 to 08220002-2		
Total number of pages	2	**Distribution**	Worldwide
Format (select all that apply)	Print, Electronic	**Translation**	Original language of publication
Who will republish the content?	Academic institution	**Copies for the disabled?**	Yes
Duration of Use	Life of current edition	**Minor editing privileges?**	Yes
Lifetime Unit Quantity	More than 2,000,000	**Incidental promotional use?**	Yes
		Currency	USD

NEW WORK DETAILS



Title	NATURAL LANGUAGE PROCESSING-BASED AUTOMATED INFORMATION EXTRACTION FROM BUILDING CODE TO SUPPORT AUTOMATED COMPLIANCE CHECKING	**Institution name** **Expected presentation date**	Purdue Univerisity 2022-07-23
Instructor name	Jiansong Zhang		

ADDITIONAL DETAILS

Order reference number	N/A	**The requesting person / organization to appear on the license**	Xiaorui Xue

REUSE CONTENT DETAILS

Title, description or numeric reference of the portion(s)	Erratum for “Building Codes Part-of-Speech Tagging Performance Improvement by Error-Driven Transformational Rules” by Xiaorui Xue and Jiansong Zhang	**Title of the article/chapter the portion is from**	Erratum for “Building Codes Part-of-Speech Tagging Performance Improvement by Error-Driven Transformational Rules” by Xiaorui Xue and Jiansong Zhang
Editor of portion(s)	Zhang, Jiansong; Xue, Xiaorui	**Author of portion(s)**	Zhang, Jiansong; Xue, Xiaorui
Volume of serial or monograph	35	**Issue, if republishing an article from a serial**	1
Page or page range of portion	08220002	**Publication date of portion**	2021-01-01

CCC Terms and Conditions

1. Description of Service; Defined Terms. This Republication License enables the User to obtain licenses for republication of one or more copyrighted works as described in detail on the relevant Order Confirmation (the "Work(s)"). Copyright Clearance Center, Inc. ("CCC") grants licenses through the Service on behalf of the rightsholder identified on the Order Confirmation (the "Rightsholder"). "Republication", as used herein, generally means the inclusion of a Work, in whole or in part, in a new work or works, also as described on the Order Confirmation. "User", as used herein, means the person or entity making such republication.

2. The terms set forth in the relevant Order Confirmation, and any terms set by the Rightsholder with respect to a particular Work, govern the terms of use of Works in connection with the Service. By using the Service, the person transacting for a republication license on behalf of the User represents and warrants that he/she/it (a) has been duly authorized by the User to accept, and hereby does accept, all such terms and conditions on behalf of User, and (b) shall inform User of all such terms and conditions. In the event such person is a "freelancer" or other third party independent of User and CCC, such party shall be deemed jointly a "User" for purposes of these terms and conditions. In any event, User shall be deemed to have accepted and agreed to all such terms and conditions if User republishes the Work in any fashion.

3. Scope of License; Limitations and Obligations.

   3.1.
   All Works and all rights therein, including copyright rights, remain the sole and exclusive property of the Rightsholder. The license created by the exchange of an Order Confirmation (and/or any invoice) and payment by User of the full amount set forth on that document includes only those rights expressly set

forth in the Order Confirmation and in these terms and conditions, and conveys no other rights in the Work(s) to User. All rights not expressly granted are hereby reserved.

3.2. General Payment Terms: You may pay by credit card or through an account with us payable at the end of the month. If you and we agree that you may establish a standing account with CCC, then the following terms apply: Remit Payment to: Copyright Clearance Center, 29118 Network Place, Chicago, IL 60673-1291. Payments Due: Invoices are payable upon their delivery to you (or upon our notice to you that they are available to you for downloading). After 30 days, outstanding amounts will be subject to a service charge of 1-1/2% per month or, if less, the maximum rate allowed by applicable law. Unless otherwise specifically set forth in the Order Confirmation or in a separate written agreement signed by CCC, invoices are due and payable on "net 30" terms. While User may exercise the rights licensed immediately upon issuance of the Order Confirmation, the license is automatically revoked and is null and void, as if it had never been issued, if complete payment for the license is not received on a timely basis either from User directly or through a payment agent, such as a credit card company.

3.3. Unless otherwise provided in the Order Confirmation, any grant of rights to User (i) is "one-time" (including the editions and product family specified in the license), (ii) is non-exclusive and non-transferable and (iii) is subject to any and all limitations and restrictions (such as, but not limited to, limitations on duration of use or circulation) included in the Order Confirmation or invoice and/or in these terms and conditions. Upon completion of the licensed use, User shall either secure a new permission for further use of the Work(s) or immediately cease any new use of the Work(s) and shall render inaccessible (such as by deleting or by removing or severing links or other locators) any further copies of the Work (except for copies printed on paper in accordance with this license and still in User's stock at the end of such period).

3.4. In the event that the material for which a republication license is sought includes third party materials (such as photographs, illustrations, graphs, inserts and similar materials) which are identified in such material as having been used by permission, User is responsible for identifying, and seeking separate licenses (under this Service or otherwise) for, any of such third party materials; without a separate license, such third party materials may not be used.

3.5. Use of proper copyright notice for a Work is required as a condition of any license granted under the Service. Unless otherwise provided in the Order Confirmation, a proper copyright notice will read substantially as follows: "Republished with permission of [Rightsholder's name], from [Work's title, author, volume, edition number and year of copyright]; permission conveyed through Copyright Clearance Center, Inc. " Such notice must be provided in a reasonably legible font size and must be placed either immediately adjacent to the Work as used (for example, as part of a by-line or footnote but not as a separate electronic link) or in the place where substantially all other credits or notices for the new work containing the republished Work are located. Failure to include the required notice results in loss to the Rightsholder and CCC, and the User shall be liable to pay liquidated damages for each such failure equal to twice the use fee specified in the Order Confirmation, in addition to the use fee itself and any other fees and charges specified.

3.6. User may only make alterations to the Work if and as expressly set forth in the Order Confirmation. No Work may be used in any way that is defamatory, violates the rights of third parties (including such third parties' rights of copyright, privacy, publicity, or other tangible or intangible property), or is otherwise illegal, sexually explicit or obscene. In addition, User may not conjoin a Work with any other material that may result in damage to the reputation of the Rightsholder. User agrees to inform CCC if it becomes aware of any infringement of any rights in a Work and to cooperate with any reasonable request of CCC or the Rightsholder in connection therewith.

4. Indemnity. User hereby indemnifies and agrees to defend the Rightsholder and CCC, and their respective employees and directors, against all claims, liability, damages, costs and expenses, including legal fees and expenses, arising out of any use of a Work beyond the scope of the rights granted herein, or any use of a Work which has been altered in any unauthorized way by User, including claims of defamation or infringement of rights of copyright, publicity, privacy or other tangible or intangible property.

5. Limitation of Liability. UNDER NO CIRCUMSTANCES WILL CCC OR THE RIGHTSHOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL OR INCIDENTAL DAMAGES (INCLUDING WITHOUT LIMITATION DAMAGES FOR LOSS OF BUSINESS PROFITS OR INFORMATION, OR FOR BUSINESS INTERRUPTION) ARISING OUT OF THE USE OR INABILITY TO USE A WORK, EVEN IF ONE OF THEM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In any event, the total liability of the Rightsholder and CCC (including their respective employees and directors) shall not exceed the total amount actually paid by User for this license. User assumes full liability for the actions and omissions of its principals, employees, agents, affiliates, successors and assigns.

6. Limited Warranties. THE WORK(S) AND RIGHT(S) ARE PROVIDED "AS IS". CCC HAS THE RIGHT TO GRANT TO USER THE RIGHTS GRANTED IN THE ORDER CONFIRMATION DOCUMENT. CCC AND THE RIGHTSHOLDER DISCLAIM ALL OTHER WARRANTIES RELATING TO THE WORK(S) AND RIGHT(S), EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. ADDITIONAL RIGHTS MAY BE REQUIRED TO USE ILLUSTRATIONS, GRAPHS, PHOTOGRAPHS, ABSTRACTS, INSERTS OR OTHER PORTIONS OF THE WORK (AS OPPOSED TO THE ENTIRE WORK) IN A MANNER CONTEMPLATED BY USER; USER UNDERSTANDS AND AGREES THAT NEITHER CCC NOR THE RIGHTSHOLDER MAY HAVE SUCH ADDITIONAL RIGHTS TO GRANT.

7. Effect of Breach. Any failure by User to pay any amount when due, or any use by User of a Work beyond the scope of the license set forth in the Order Confirmation and/or these terms and conditions, shall be a material breach of the license created by the Order Confirmation and these terms and conditions. Any breach not cured within 30 days of written notice thereof shall result in immediate termination of such license without further notice. Any unauthorized (but licensable) use of a Work that is terminated immediately upon notice thereof may be liquidated by payment of the Rightsholder's ordinary license price therefor; any unauthorized (and unlicensable) use that is not terminated immediately for any reason (including, for example, because materials containing the Work cannot reasonably be recalled) will be subject to all remedies available at law or in equity, but in no event to a payment of less than three times the Rightsholder's ordinary license price for the most closely analogous licensable use plus Rightsholder's and/or CCC's costs and expenses incurred in collecting such payment.

8. Miscellaneous.

    8.1. User acknowledges that CCC may, from time to time, make changes or additions to the Service or to these terms and conditions, and CCC reserves the right to send notice to the User by electronic mail or otherwise for the purposes of notifying User of such changes or additions; provided that any such changes or additions shall not apply to permissions already secured and paid for.

    8.2. Use of User-related information collected through the Service is governed by CCC's privacy policy, available online here:https://marketplace.copyright.com/rs-ui-web/mp/privacy-policy

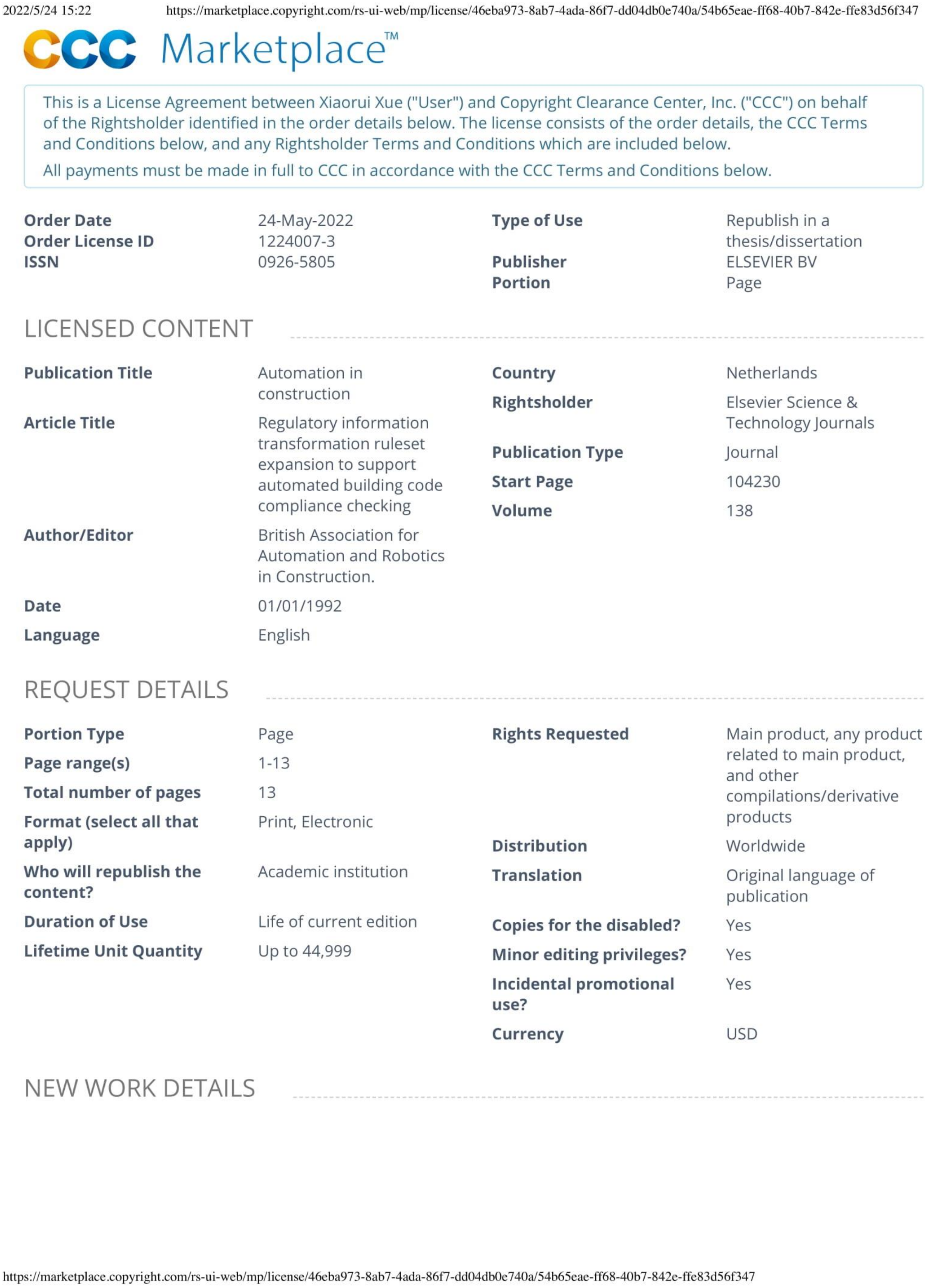
    8.3. The licensing transaction described in the Order Confirmation is personal to User. Therefore, User may not assign or transfer to any other person (whether a natural person or an organization of any kind) the license created by the Order Confirmation and these terms and conditions or any rights granted hereunder; provided, however, that User may assign such license in its entirety on written notice to CCC in the event of a transfer of all or substantially all of User's rights in the new material which includes the Work(s) licensed under this Service.

    8.4. No amendment or waiver of any terms is binding unless set forth in writing and signed by the parties. The Rightsholder and CCC hereby object to any terms contained in any writing prepared by the User or its principals, employees, agents or affiliates and purporting to govern or otherwise relate to the licensing transaction described in the Order Confirmation, which terms are in any way inconsistent with any terms set forth in the Order Confirmation and/or in these terms and conditions or CCC's standard operating procedures, whether such writing is prepared prior to, simultaneously with or subsequent to the Order Confirmation, and whether such writing appears on a copy of the Order Confirmation or in a separate instrument.

    8.5.
    The licensing transaction described in the Order Confirmation document shall be governed by and construed under the law of the State of New York, USA, without regard to the principles thereof of conflicts

of law. Any case, controversy, suit, action, or proceeding arising out of, in connection with, or related to such licensing transaction shall be brought, at CCC's sole discretion, in any federal or state court located in the County of New York, State of New York, USA, or in any federal or state court whose geographical jurisdiction covers the location of the Rightsholder set forth in the Order Confirmation. The parties expressly submit to the personal jurisdiction and venue of each such federal or state court.If you have any comments or questions about the Service or Copyright Clearance Center, please contact us at 978-750-8400 or send an e-mail to support@copyright.com.

v 1.1



2022/5/24 15:22 https://marketplace.copyright.com/rs-ui-web/mp/license/46eba973-8ab7-4ada-86f7-dd04db0e740a/54b65eae-ff68-40b7-842e-ffe83d56f347

CCC Marketplace™

This is a License Agreement between Xiaorui Xue ("User") and Copyright Clearance Center, Inc. ("CCC") on behalf of the Rightsholder identified in the order details below. The license consists of the order details, the CCC Terms and Conditions below, and any Rightsholder Terms and Conditions which are included below.

All payments must be made in full to CCC in accordance with the CCC Terms and Conditions below.

Order Date	24-May-2022	**Type of Use**	Republish in a thesis/dissertation
Order License ID	1224007-3		
ISSN	0926-5805	**Publisher**	ELSEVIER BV
		Portion	Page

LICENSED CONTENT

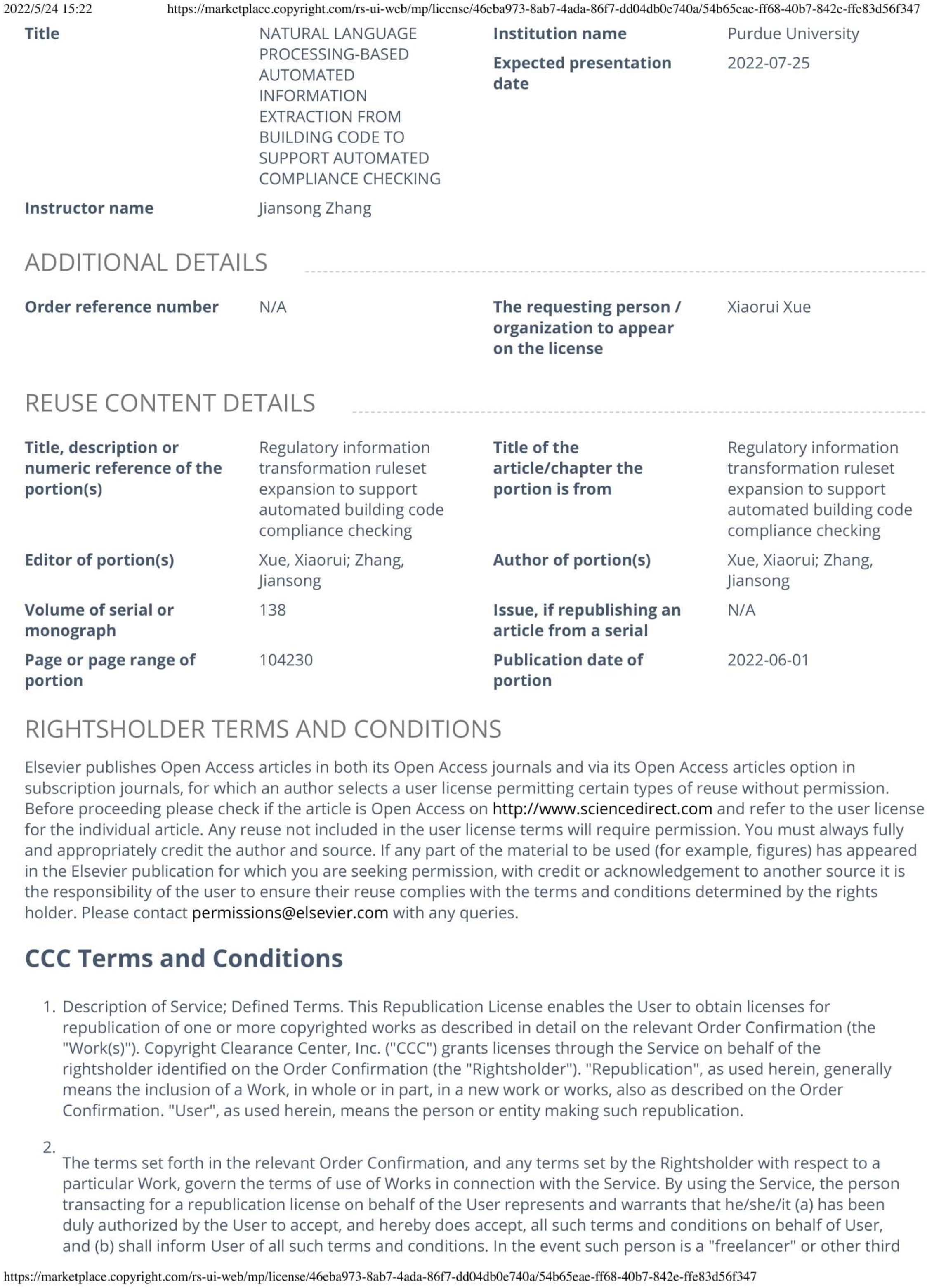
Publication Title	Automation in construction	**Country**	Netherlands
		Rightsholder	Elsevier Science & Technology Journals
Article Title	Regulatory information transformation ruleset expansion to support automated building code compliance checking	**Publication Type**	Journal
		Start Page	104230
		Volume	138
Author/Editor	British Association for Automation and Robotics in Construction.		
Date	01/01/1992		
Language	English		

REQUEST DETAILS

Portion Type	Page	**Rights Requested**	Main product, any product related to main product, and other compilations/derivative products
Page range(s)	1-13		
Total number of pages	13		
Format (select all that apply)	Print, Electronic	**Distribution**	Worldwide
Who will republish the content?	Academic institution	**Translation**	Original language of publication
Duration of Use	Life of current edition	**Copies for the disabled?**	Yes
Lifetime Unit Quantity	Up to 44,999	**Minor editing privileges?**	Yes
		Incidental promotional use?	Yes
		Currency	USD

NEW WORK DETAILS

https://marketplace.copyright.com/rs-ui-web/mp/license/46eba973-8ab7-4ada-86f7-dd04db0e740a/54b65eae-ff68-40b7-842e-ffe83d56f347



Title	NATURAL LANGUAGE PROCESSING-BASED AUTOMATED INFORMATION EXTRACTION FROM BUILDING CODE TO SUPPORT AUTOMATED COMPLIANCE CHECKING	**Institution name** **Expected presentation date**	Purdue University 2022-07-25
Instructor name	Jiansong Zhang		

ADDITIONAL DETAILS

Order reference number	N/A	**The requesting person / organization to appear on the license**	Xiaorui Xue

REUSE CONTENT DETAILS

Title, description or numeric reference of the portion(s)	Regulatory information transformation ruleset expansion to support automated building code compliance checking	**Title of the article/chapter the portion is from**	Regulatory information transformation ruleset expansion to support automated building code compliance checking
Editor of portion(s)	Xue, Xiaorui; Zhang, Jiansong	**Author of portion(s)**	Xue, Xiaorui; Zhang, Jiansong
Volume of serial or monograph	138	**Issue, if republishing an article from a serial**	N/A
Page or page range of portion	104230	**Publication date of portion**	2022-06-01

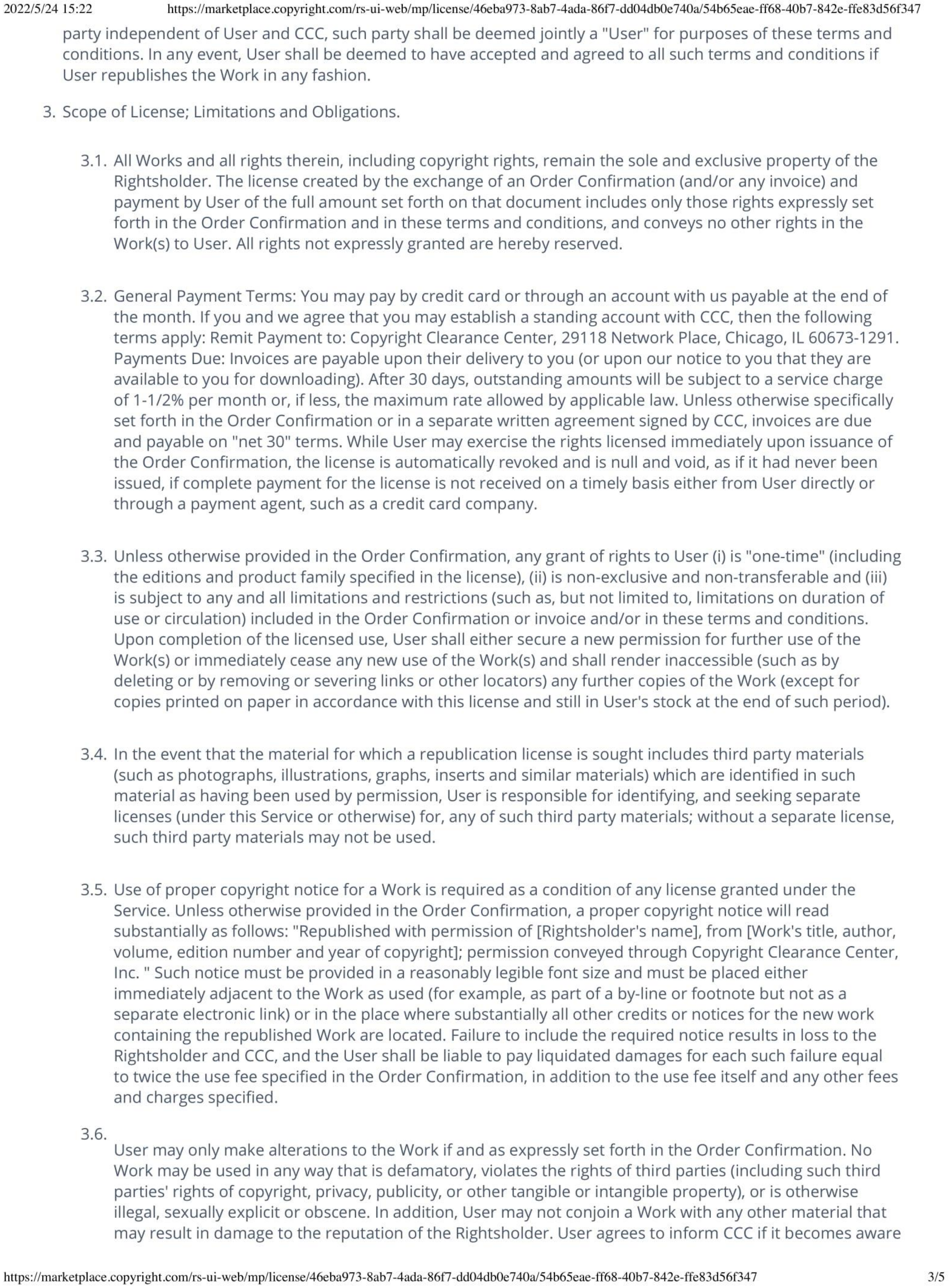
RIGHTSHOLDER TERMS AND CONDITIONS

Elsevier publishes Open Access articles in both its Open Access journals and via its Open Access articles option in subscription journals, for which an author selects a user license permitting certain types of reuse without permission. Before proceeding please check if the article is Open Access on http://www.sciencedirect.com and refer to the user license for the individual article. Any reuse not included in the user license terms will require permission. You must always fully and appropriately credit the author and source. If any part of the material to be used (for example, figures) has appeared in the Elsevier publication for which you are seeking permission, with credit or acknowledgement to another source it is the responsibility of the user to ensure their reuse complies with the terms and conditions determined by the rights holder. Please contact permissions@elsevier.com with any queries.

CCC Terms and Conditions

1. Description of Service; Defined Terms. This Republication License enables the User to obtain licenses for republication of one or more copyrighted works as described in detail on the relevant Order Confirmation (the "Work(s)"). Copyright Clearance Center, Inc. ("CCC") grants licenses through the Service on behalf of the rightsholder identified on the Order Confirmation (the "Rightsholder"). "Republication", as used herein, generally means the inclusion of a Work, in whole or in part, in a new work or works, also as described on the Order Confirmation. "User", as used herein, means the person or entity making such republication.

2. The terms set forth in the relevant Order Confirmation, and any terms set by the Rightsholder with respect to a particular Work, govern the terms of use of Works in connection with the Service. By using the Service, the person transacting for a republication license on behalf of the User represents and warrants that he/she/it (a) has been duly authorized by the User to accept, and hereby does accept, all such terms and conditions on behalf of User, and (b) shall inform User of all such terms and conditions. In the event such person is a "freelancer" or other third



party independent of User and CCC, such party shall be deemed jointly a "User" for purposes of these terms and conditions. In any event, User shall be deemed to have accepted and agreed to all such terms and conditions if User republishes the Work in any fashion.

3. Scope of License; Limitations and Obligations.

   3.1. All Works and all rights therein, including copyright rights, remain the sole and exclusive property of the Rightsholder. The license created by the exchange of an Order Confirmation (and/or any invoice) and payment by User of the full amount set forth on that document includes only those rights expressly set forth in the Order Confirmation and in these terms and conditions, and conveys no other rights in the Work(s) to User. All rights not expressly granted are hereby reserved.

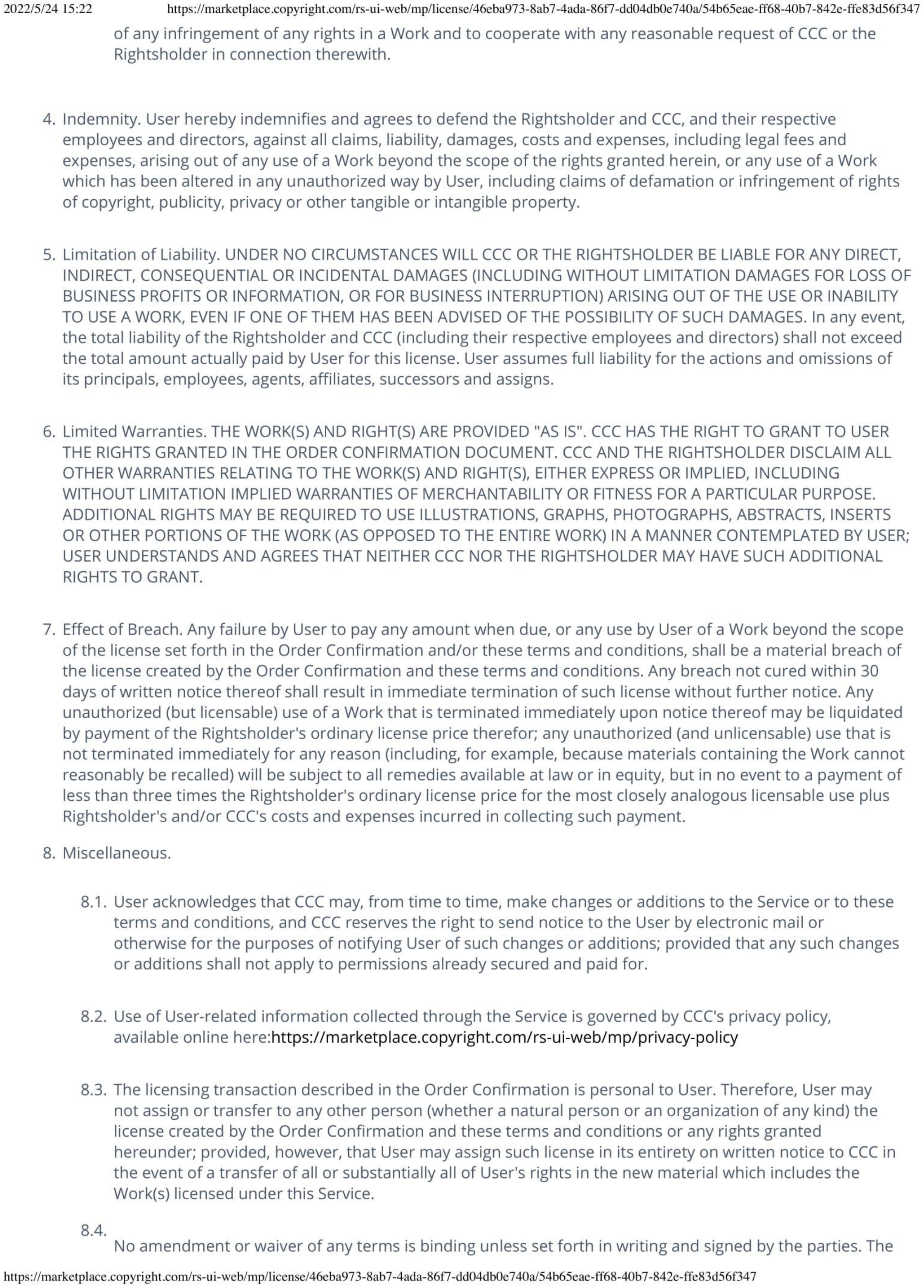
   3.2. General Payment Terms: You may pay by credit card or through an account with us payable at the end of the month. If you and we agree that you may establish a standing account with CCC, then the following terms apply: Remit Payment to: Copyright Clearance Center, 29118 Network Place, Chicago, IL 60673-1291. Payments Due: Invoices are payable upon their delivery to you (or upon our notice to you that they are available to you for downloading). After 30 days, outstanding amounts will be subject to a service charge of 1-1/2% per month or, if less, the maximum rate allowed by applicable law. Unless otherwise specifically set forth in the Order Confirmation or in a separate written agreement signed by CCC, invoices are due and payable on "net 30" terms. While User may exercise the rights licensed immediately upon issuance of the Order Confirmation, the license is automatically revoked and is null and void, as if it had never been issued, if complete payment for the license is not received on a timely basis either from User directly or through a payment agent, such as a credit card company.

   3.3. Unless otherwise provided in the Order Confirmation, any grant of rights to User (i) is "one-time" (including the editions and product family specified in the license), (ii) is non-exclusive and non-transferable and (iii) is subject to any and all limitations and restrictions (such as, but not limited to, limitations on duration of use or circulation) included in the Order Confirmation or invoice and/or in these terms and conditions. Upon completion of the licensed use, User shall either secure a new permission for further use of the Work(s) or immediately cease any new use of the Work(s) and shall render inaccessible (such as by deleting or by removing or severing links or other locators) any further copies of the Work (except for copies printed on paper in accordance with this license and still in User's stock at the end of such period).

   3.4. In the event that the material for which a republication license is sought includes third party materials (such as photographs, illustrations, graphs, inserts and similar materials) which are identified in such material as having been used by permission, User is responsible for identifying, and seeking separate licenses (under this Service or otherwise) for, any of such third party materials; without a separate license, such third party materials may not be used.

   3.5. Use of proper copyright notice for a Work is required as a condition of any license granted under the Service. Unless otherwise provided in the Order Confirmation, a proper copyright notice will read substantially as follows: "Republished with permission of [Rightsholder's name], from [Work's title, author, volume, edition number and year of copyright]; permission conveyed through Copyright Clearance Center, Inc. " Such notice must be provided in a reasonably legible font size and must be placed either immediately adjacent to the Work as used (for example, as part of a by-line or footnote but not as a separate electronic link) or in the place where substantially all other credits or notices for the new work containing the republished Work are located. Failure to include the required notice results in loss to the Rightsholder and CCC, and the User shall be liable to pay liquidated damages for each such failure equal to twice the use fee specified in the Order Confirmation, in addition to the use fee itself and any other fees and charges specified.

   3.6.
   User may only make alterations to the Work if and as expressly set forth in the Order Confirmation. No Work may be used in any way that is defamatory, violates the rights of third parties (including such third parties' rights of copyright, privacy, publicity, or other tangible or intangible property), or is otherwise illegal, sexually explicit or obscene. In addition, User may not conjoin a Work with any other material that may result in damage to the reputation of the Rightsholder. User agrees to inform CCC if it becomes aware



of any infringement of any rights in a Work and to cooperate with any reasonable request of CCC or the Rightsholder in connection therewith.

4. Indemnity. User hereby indemnifies and agrees to defend the Rightsholder and CCC, and their respective employees and directors, against all claims, liability, damages, costs and expenses, including legal fees and expenses, arising out of any use of a Work beyond the scope of the rights granted herein, or any use of a Work which has been altered in any unauthorized way by User, including claims of defamation or infringement of rights of copyright, publicity, privacy or other tangible or intangible property.

5. Limitation of Liability. UNDER NO CIRCUMSTANCES WILL CCC OR THE RIGHTSHOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL OR INCIDENTAL DAMAGES (INCLUDING WITHOUT LIMITATION DAMAGES FOR LOSS OF BUSINESS PROFITS OR INFORMATION, OR FOR BUSINESS INTERRUPTION) ARISING OUT OF THE USE OR INABILITY TO USE A WORK, EVEN IF ONE OF THEM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In any event, the total liability of the Rightsholder and CCC (including their respective employees and directors) shall not exceed the total amount actually paid by User for this license. User assumes full liability for the actions and omissions of its principals, employees, agents, affiliates, successors and assigns.

6. Limited Warranties. THE WORK(S) AND RIGHT(S) ARE PROVIDED "AS IS". CCC HAS THE RIGHT TO GRANT TO USER THE RIGHTS GRANTED IN THE ORDER CONFIRMATION DOCUMENT. CCC AND THE RIGHTSHOLDER DISCLAIM ALL OTHER WARRANTIES RELATING TO THE WORK(S) AND RIGHT(S), EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. ADDITIONAL RIGHTS MAY BE REQUIRED TO USE ILLUSTRATIONS, GRAPHS, PHOTOGRAPHS, ABSTRACTS, INSERTS OR OTHER PORTIONS OF THE WORK (AS OPPOSED TO THE ENTIRE WORK) IN A MANNER CONTEMPLATED BY USER; USER UNDERSTANDS AND AGREES THAT NEITHER CCC NOR THE RIGHTSHOLDER MAY HAVE SUCH ADDITIONAL RIGHTS TO GRANT.

7. Effect of Breach. Any failure by User to pay any amount when due, or any use by User of a Work beyond the scope of the license set forth in the Order Confirmation and/or these terms and conditions, shall be a material breach of the license created by the Order Confirmation and these terms and conditions. Any breach not cured within 30 days of written notice thereof shall result in immediate termination of such license without further notice. Any unauthorized (but licensable) use of a Work that is terminated immediately upon notice thereof may be liquidated by payment of the Rightsholder's ordinary license price therefor; any unauthorized (and unlicensable) use that is not terminated immediately for any reason (including, for example, because materials containing the Work cannot reasonably be recalled) will be subject to all remedies available at law or in equity, but in no event to a payment of less than three times the Rightsholder's ordinary license price for the most closely analogous licensable use plus Rightsholder's and/or CCC's costs and expenses incurred in collecting such payment.

8. Miscellaneous.

   8.1. User acknowledges that CCC may, from time to time, make changes or additions to the Service or to these terms and conditions, and CCC reserves the right to send notice to the User by electronic mail or otherwise for the purposes of notifying User of such changes or additions; provided that any such changes or additions shall not apply to permissions already secured and paid for.

   8.2. Use of User-related information collected through the Service is governed by CCC's privacy policy, available online here:https://marketplace.copyright.com/rs-ui-web/mp/privacy-policy

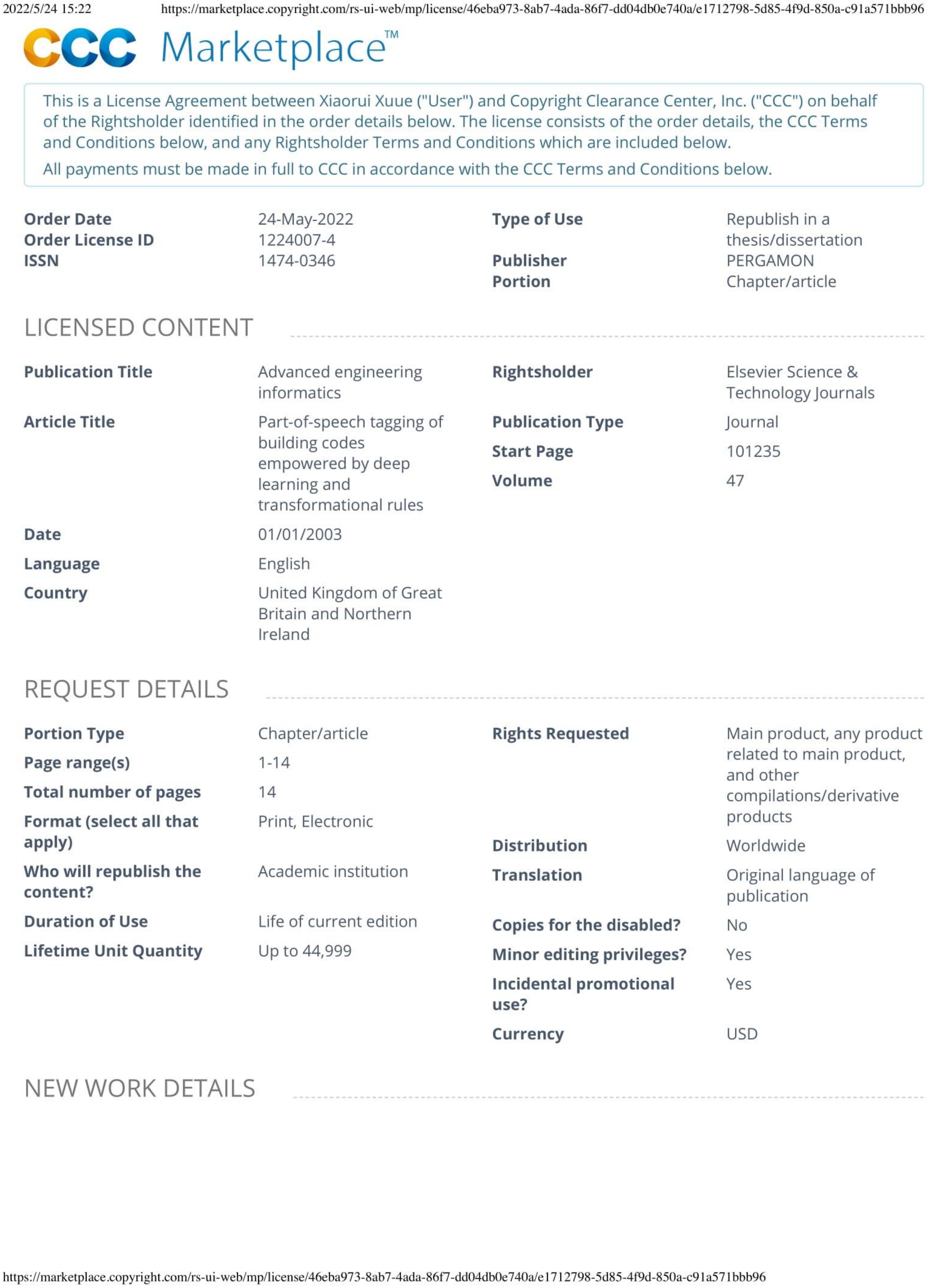
   8.3. The licensing transaction described in the Order Confirmation is personal to User. Therefore, User may not assign or transfer to any other person (whether a natural person or an organization of any kind) the license created by the Order Confirmation and these terms and conditions or any rights granted hereunder; provided, however, that User may assign such license in its entirety on written notice to CCC in the event of a transfer of all or substantially all of User's rights in the new material which includes the Work(s) licensed under this Service.

   8.4. No amendment or waiver of any terms is binding unless set forth in writing and signed by the parties. The

Rightsholder and CCC hereby object to any terms contained in any writing prepared by the User or its principals, employees, agents or affiliates and purporting to govern or otherwise relate to the licensing transaction described in the Order Confirmation, which terms are in any way inconsistent with any terms set forth in the Order Confirmation and/or in these terms and conditions or CCC's standard operating procedures, whether such writing is prepared prior to, simultaneously with or subsequent to the Order Confirmation, and whether such writing appears on a copy of the Order Confirmation or in a separate instrument.

8.5. The licensing transaction described in the Order Confirmation document shall be governed by and construed under the law of the State of New York, USA, without regard to the principles thereof of conflicts of law. Any case, controversy, suit, action, or proceeding arising out of, in connection with, or related to such licensing transaction shall be brought, at CCC's sole discretion, in any federal or state court located in the County of New York, State of New York, USA, or in any federal or state court whose geographical jurisdiction covers the location of the Rightsholder set forth in the Order Confirmation. The parties expressly submit to the personal jurisdiction and venue of each such federal or state court.If you have any comments or questions about the Service or Copyright Clearance Center, please contact us at 978-750-8400 or send an e-mail to support@copyright.com.

v 1.1



CCC Marketplace™

This is a License Agreement between Xiaorui Xuue ("User") and Copyright Clearance Center, Inc. ("CCC") on behalf of the Rightsholder identified in the order details below. The license consists of the order details, the CCC Terms and Conditions below, and any Rightsholder Terms and Conditions which are included below.

All payments must be made in full to CCC in accordance with the CCC Terms and Conditions below.

Order Date	24-May-2022	**Type of Use**	Republish in a thesis/dissertation
Order License ID	1224007-4		
ISSN	1474-0346	**Publisher**	PERGAMON
		Portion	Chapter/article

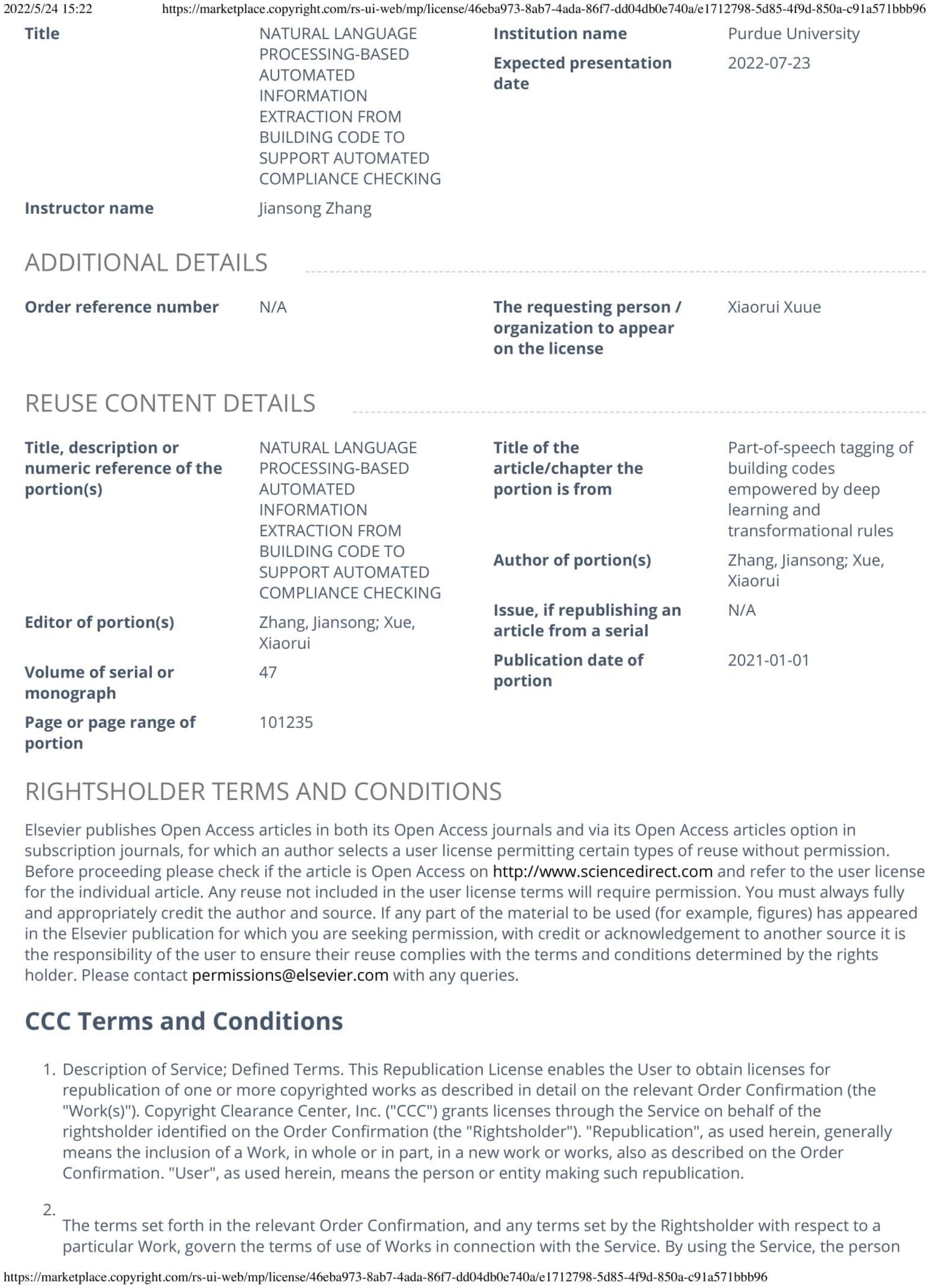
LICENSED CONTENT

Publication Title	Advanced engineering informatics	**Rightsholder**	Elsevier Science & Technology Journals
Article Title	Part-of-speech tagging of building codes empowered by deep learning and transformational rules	**Publication Type**	Journal
		Start Page	101235
		Volume	47
Date	01/01/2003		
Language	English		
Country	United Kingdom of Great Britain and Northern Ireland		

REQUEST DETAILS

Portion Type	Chapter/article	**Rights Requested**	Main product, any product related to main product, and other compilations/derivative products
Page range(s)	1-14		
Total number of pages	14		
Format (select all that apply)	Print, Electronic	**Distribution**	Worldwide
Who will republish the content?	Academic institution	**Translation**	Original language of publication
Duration of Use	Life of current edition	**Copies for the disabled?**	No
Lifetime Unit Quantity	Up to 44,999	**Minor editing privileges?**	Yes
		Incidental promotional use?	Yes
		Currency	USD

NEW WORK DETAILS



Title	NATURAL LANGUAGE PROCESSING-BASED AUTOMATED INFORMATION EXTRACTION FROM BUILDING CODE TO SUPPORT AUTOMATED COMPLIANCE CHECKING	**Institution name**	Purdue University
		Expected presentation date	2022-07-23
Instructor name	Jiansong Zhang		

ADDITIONAL DETAILS

Order reference number	N/A	**The requesting person / organization to appear on the license**	Xiaorui Xuue

REUSE CONTENT DETAILS

Title, description or numeric reference of the portion(s)	NATURAL LANGUAGE PROCESSING-BASED AUTOMATED INFORMATION EXTRACTION FROM BUILDING CODE TO SUPPORT AUTOMATED COMPLIANCE CHECKING	**Title of the article/chapter the portion is from**	Part-of-speech tagging of building codes empowered by deep learning and transformational rules
		Author of portion(s)	Zhang, Jiansong; Xue, Xiaorui
Editor of portion(s)	Zhang, Jiansong; Xue, Xiaorui	**Issue, if republishing an article from a serial**	N/A
Volume of serial or monograph	47	**Publication date of portion**	2021-01-01
Page or page range of portion	101235		

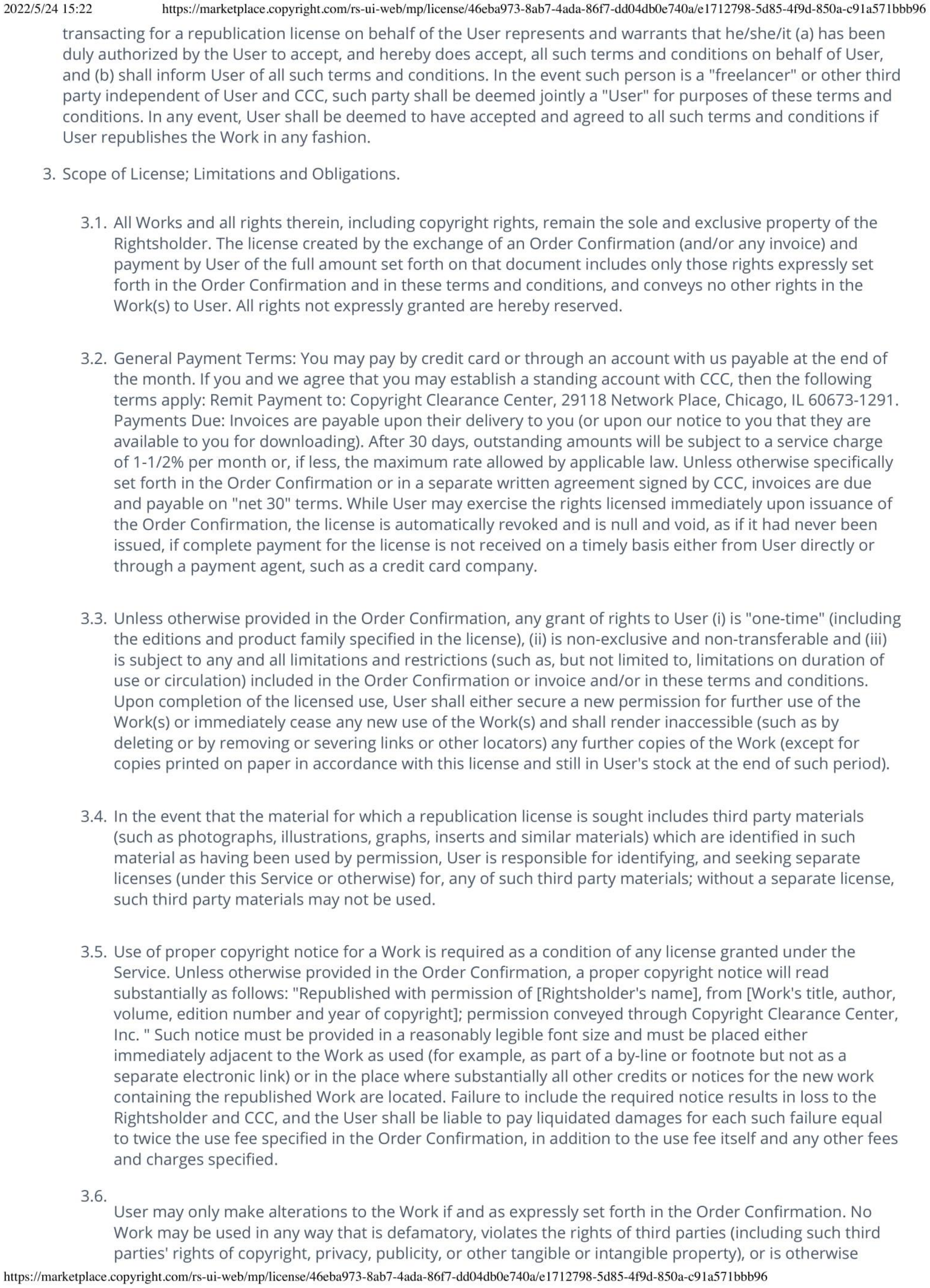
RIGHTSHOLDER TERMS AND CONDITIONS

Elsevier publishes Open Access articles in both its Open Access journals and via its Open Access articles option in subscription journals, for which an author selects a user license permitting certain types of reuse without permission. Before proceeding please check if the article is Open Access on http://www.sciencedirect.com and refer to the user license for the individual article. Any reuse not included in the user license terms will require permission. You must always fully and appropriately credit the author and source. If any part of the material to be used (for example, figures) has appeared in the Elsevier publication for which you are seeking permission, with credit or acknowledgement to another source it is the responsibility of the user to ensure their reuse complies with the terms and conditions determined by the rights holder. Please contact permissions@elsevier.com with any queries.

CCC Terms and Conditions

1. Description of Service; Defined Terms. This Republication License enables the User to obtain licenses for republication of one or more copyrighted works as described in detail on the relevant Order Confirmation (the "Work(s)"). Copyright Clearance Center, Inc. ("CCC") grants licenses through the Service on behalf of the rightsholder identified on the Order Confirmation (the "Rightsholder"). "Republication", as used herein, generally means the inclusion of a Work, in whole or in part, in a new work or works, also as described on the Order Confirmation. "User", as used herein, means the person or entity making such republication.

2. The terms set forth in the relevant Order Confirmation, and any terms set by the Rightsholder with respect to a particular Work, govern the terms of use of Works in connection with the Service. By using the Service, the person



transacting for a republication license on behalf of the User represents and warrants that he/she/it (a) has been duly authorized by the User to accept, and hereby does accept, all such terms and conditions on behalf of User, and (b) shall inform User of all such terms and conditions. In the event such person is a "freelancer" or other third party independent of User and CCC, such party shall be deemed jointly a "User" for purposes of these terms and conditions. In any event, User shall be deemed to have accepted and agreed to all such terms and conditions if User republishes the Work in any fashion.

3. Scope of License; Limitations and Obligations.

3.1. All Works and all rights therein, including copyright rights, remain the sole and exclusive property of the Rightsholder. The license created by the exchange of an Order Confirmation (and/or any invoice) and payment by User of the full amount set forth on that document includes only those rights expressly set forth in the Order Confirmation and in these terms and conditions, and conveys no other rights in the Work(s) to User. All rights not expressly granted are hereby reserved.

3.2. General Payment Terms: You may pay by credit card or through an account with us payable at the end of the month. If you and we agree that you may establish a standing account with CCC, then the following terms apply: Remit Payment to: Copyright Clearance Center, 29118 Network Place, Chicago, IL 60673-1291. Payments Due: Invoices are payable upon their delivery to you (or upon our notice to you that they are available to you for downloading). After 30 days, outstanding amounts will be subject to a service charge of 1-1/2% per month or, if less, the maximum rate allowed by applicable law. Unless otherwise specifically set forth in the Order Confirmation or in a separate written agreement signed by CCC, invoices are due and payable on "net 30" terms. While User may exercise the rights licensed immediately upon issuance of the Order Confirmation, the license is automatically revoked and is null and void, as if it had never been issued, if complete payment for the license is not received on a timely basis either from User directly or through a payment agent, such as a credit card company.

3.3. Unless otherwise provided in the Order Confirmation, any grant of rights to User (i) is "one-time" (including the editions and product family specified in the license), (ii) is non-exclusive and non-transferable and (iii) is subject to any and all limitations and restrictions (such as, but not limited to, limitations on duration of use or circulation) included in the Order Confirmation or invoice and/or in these terms and conditions. Upon completion of the licensed use, User shall either secure a new permission for further use of the Work(s) or immediately cease any new use of the Work(s) and shall render inaccessible (such as by deleting or by removing or severing links or other locators) any further copies of the Work (except for copies printed on paper in accordance with this license and still in User's stock at the end of such period).

3.4. In the event that the material for which a republication license is sought includes third party materials (such as photographs, illustrations, graphs, inserts and similar materials) which are identified in such material as having been used by permission, User is responsible for identifying, and seeking separate licenses (under this Service or otherwise) for, any of such third party materials; without a separate license, such third party materials may not be used.

3.5. Use of proper copyright notice for a Work is required as a condition of any license granted under the Service. Unless otherwise provided in the Order Confirmation, a proper copyright notice will read substantially as follows: "Republished with permission of [Rightsholder's name], from [Work's title, author, volume, edition number and year of copyright]; permission conveyed through Copyright Clearance Center, Inc. " Such notice must be provided in a reasonably legible font size and must be placed either immediately adjacent to the Work as used (for example, as part of a by-line or footnote but not as a separate electronic link) or in the place where substantially all other credits or notices for the new work containing the republished Work are located. Failure to include the required notice results in loss to the Rightsholder and CCC, and the User shall be liable to pay liquidated damages for each such failure equal to twice the use fee specified in the Order Confirmation, in addition to the use fee itself and any other fees and charges specified.

3.6.
User may only make alterations to the Work if and as expressly set forth in the Order Confirmation. No Work may be used in any way that is defamatory, violates the rights of third parties (including such third parties' rights of copyright, privacy, publicity, or other tangible or intangible property), or is otherwise

illegal, sexually explicit or obscene. In addition, User may not conjoin a Work with any other material that may result in damage to the reputation of the Rightsholder. User agrees to inform CCC if it becomes aware of any infringement of any rights in a Work and to cooperate with any reasonable request of CCC or the Rightsholder in connection therewith.

4. Indemnity. User hereby indemnifies and agrees to defend the Rightsholder and CCC, and their respective employees and directors, against all claims, liability, damages, costs and expenses, including legal fees and expenses, arising out of any use of a Work beyond the scope of the rights granted herein, or any use of a Work which has been altered in any unauthorized way by User, including claims of defamation or infringement of rights of copyright, publicity, privacy or other tangible or intangible property.

5. Limitation of Liability. UNDER NO CIRCUMSTANCES WILL CCC OR THE RIGHTSHOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL OR INCIDENTAL DAMAGES (INCLUDING WITHOUT LIMITATION DAMAGES FOR LOSS OF BUSINESS PROFITS OR INFORMATION, OR FOR BUSINESS INTERRUPTION) ARISING OUT OF THE USE OR INABILITY TO USE A WORK, EVEN IF ONE OF THEM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In any event, the total liability of the Rightsholder and CCC (including their respective employees and directors) shall not exceed the total amount actually paid by User for this license. User assumes full liability for the actions and omissions of its principals, employees, agents, affiliates, successors and assigns.

6. Limited Warranties. THE WORK(S) AND RIGHT(S) ARE PROVIDED "AS IS". CCC HAS THE RIGHT TO GRANT TO USER THE RIGHTS GRANTED IN THE ORDER CONFIRMATION DOCUMENT. CCC AND THE RIGHTSHOLDER DISCLAIM ALL OTHER WARRANTIES RELATING TO THE WORK(S) AND RIGHT(S), EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. ADDITIONAL RIGHTS MAY BE REQUIRED TO USE ILLUSTRATIONS, GRAPHS, PHOTOGRAPHS, ABSTRACTS, INSERTS OR OTHER PORTIONS OF THE WORK (AS OPPOSED TO THE ENTIRE WORK) IN A MANNER CONTEMPLATED BY USER; USER UNDERSTANDS AND AGREES THAT NEITHER CCC NOR THE RIGHTSHOLDER MAY HAVE SUCH ADDITIONAL RIGHTS TO GRANT.

7. Effect of Breach. Any failure by User to pay any amount when due, or any use by User of a Work beyond the scope of the license set forth in the Order Confirmation and/or these terms and conditions, shall be a material breach of the license created by the Order Confirmation and these terms and conditions. Any breach not cured within 30 days of written notice thereof shall result in immediate termination of such license without further notice. Any unauthorized (but licensable) use of a Work that is terminated immediately upon notice thereof may be liquidated by payment of the Rightsholder's ordinary license price therefor; any unauthorized (and unlicensable) use that is not terminated immediately for any reason (including, for example, because materials containing the Work cannot reasonably be recalled) will be subject to all remedies available at law or in equity, but in no event to a payment of less than three times the Rightsholder's ordinary license price for the most closely analogous licensable use plus Rightsholder's and/or CCC's costs and expenses incurred in collecting such payment.

8. Miscellaneous.

   8.1. User acknowledges that CCC may, from time to time, make changes or additions to the Service or to these terms and conditions, and CCC reserves the right to send notice to the User by electronic mail or otherwise for the purposes of notifying User of such changes or additions; provided that any such changes or additions shall not apply to permissions already secured and paid for.

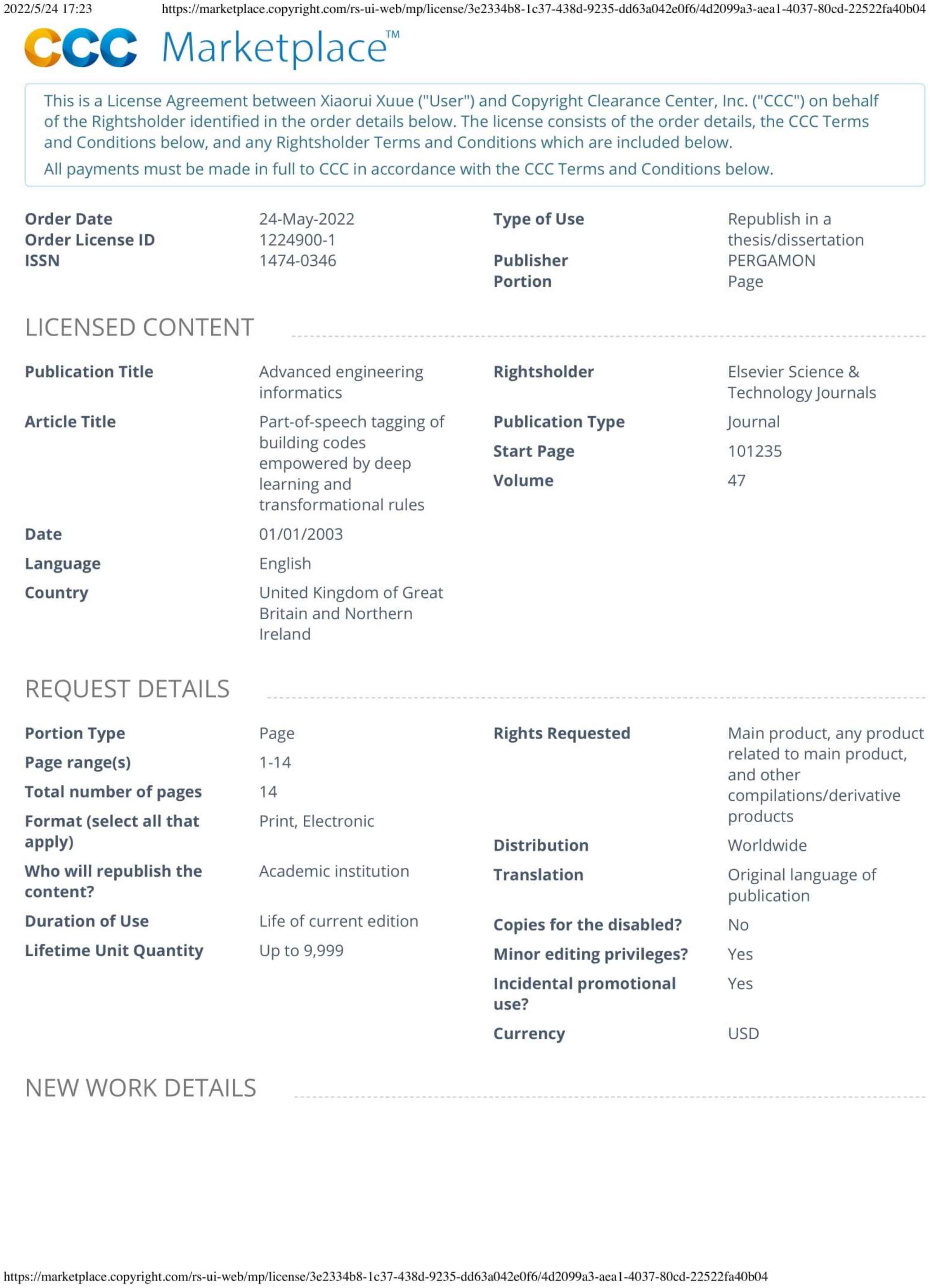
   8.2. Use of User-related information collected through the Service is governed by CCC's privacy policy, available online here:https://marketplace.copyright.com/rs-ui-web/mp/privacy-policy

   8.3. The licensing transaction described in the Order Confirmation is personal to User. Therefore, User may not assign or transfer to any other person (whether a natural person or an organization of any kind) the license created by the Order Confirmation and these terms and conditions or any rights granted hereunder; provided, however, that User may assign such license in its entirety on written notice to CCC in the event of a transfer of all or substantially all of User's rights in the new material which includes the Work(s) licensed under this Service.

8.4. No amendment or waiver of any terms is binding unless set forth in writing and signed by the parties. The Rightsholder and CCC hereby object to any terms contained in any writing prepared by the User or its principals, employees, agents or affiliates and purporting to govern or otherwise relate to the licensing transaction described in the Order Confirmation, which terms are in any way inconsistent with any terms set forth in the Order Confirmation and/or in these terms and conditions or CCC's standard operating procedures, whether such writing is prepared prior to, simultaneously with or subsequent to the Order Confirmation, and whether such writing appears on a copy of the Order Confirmation or in a separate instrument.

8.5. The licensing transaction described in the Order Confirmation document shall be governed by and construed under the law of the State of New York, USA, without regard to the principles thereof of conflicts of law. Any case, controversy, suit, action, or proceeding arising out of, in connection with, or related to such licensing transaction shall be brought, at CCC's sole discretion, in any federal or state court located in the County of New York, State of New York, USA, or in any federal or state court whose geographical jurisdiction covers the location of the Rightsholder set forth in the Order Confirmation. The parties expressly submit to the personal jurisdiction and venue of each such federal or state court.If you have any comments or questions about the Service or Copyright Clearance Center, please contact us at 978-750-8400 or send an e-mail to support@copyright.com.

v 1.1



2022/5/24 17:23 https://marketplace.copyright.com/rs-ui-web/mp/license/3e2334b8-1c37-438d-9235-dd63a042e0f6/4d2099a3-aea1-4037-80cd-22522fa40b04

CCC Marketplace™

This is a License Agreement between Xiaorui Xuue ("User") and Copyright Clearance Center, Inc. ("CCC") on behalf of the Rightsholder identified in the order details below. The license consists of the order details, the CCC Terms and Conditions below, and any Rightsholder Terms and Conditions which are included below.

All payments must be made in full to CCC in accordance with the CCC Terms and Conditions below.

Order Date	24-May-2022	**Type of Use**	Republish in a thesis/dissertation
Order License ID	1224900-1		
ISSN	1474-0346	**Publisher**	PERGAMON
		Portion	Page

LICENSED CONTENT

Publication Title	Advanced engineering informatics	**Rightsholder**	Elsevier Science & Technology Journals
Article Title	Part-of-speech tagging of building codes empowered by deep learning and transformational rules	**Publication Type**	Journal
		Start Page	101235
		Volume	47
Date	01/01/2003		
Language	English		
Country	United Kingdom of Great Britain and Northern Ireland		

REQUEST DETAILS

Portion Type	Page	**Rights Requested**	Main product, any product related to main product, and other compilations/derivative products
Page range(s)	1-14		
Total number of pages	14		
Format (select all that apply)	Print, Electronic		
		Distribution	Worldwide
Who will republish the content?	Academic institution	**Translation**	Original language of publication
Duration of Use	Life of current edition	**Copies for the disabled?**	No
Lifetime Unit Quantity	Up to 9,999	**Minor editing privileges?**	Yes
		Incidental promotional use?	Yes
		Currency	USD

NEW WORK DETAILS

https://marketplace.copyright.com/rs-ui-web/mp/license/3e2334b8-1c37-438d-9235-dd63a042e0f6/4d2099a3-aea1-4037-80cd-22522fa40b04

Title	NATURAL LANGUAGE PROCESSING-BASED AUTOMATED INFORMATION EXTRACTION FROM BUILDING CODE TO SUPPORT AUTOMATED COMPLIANCE CHECKING	**Institution name**	Purdue University
		Expected presentation date	2022-07-25
Instructor name	Jiansong Zhang		

ADDITIONAL DETAILS

Order reference number	N/A	**The requesting person / organization to appear on the license**	Xiaorui Xuue

REUSE CONTENT DETAILS

Title, description or numeric reference of the portion(s)	Part-of-speech tagging of building codes empowered by deep learning and transformational rules	**Title of the article/chapter the portion is from**	Part-of-speech tagging of building codes empowered by deep learning and transformational rules
Editor of portion(s)	Zhang, Jiansong; Xue, Xiaorui	**Author of portion(s)**	Zhang, Jiansong; Xue, Xiaorui
Volume of serial or monograph	47	**Issue, if republishing an article from a serial**	N/A
Page or page range of portion	101235	**Publication date of portion**	2021-01-01

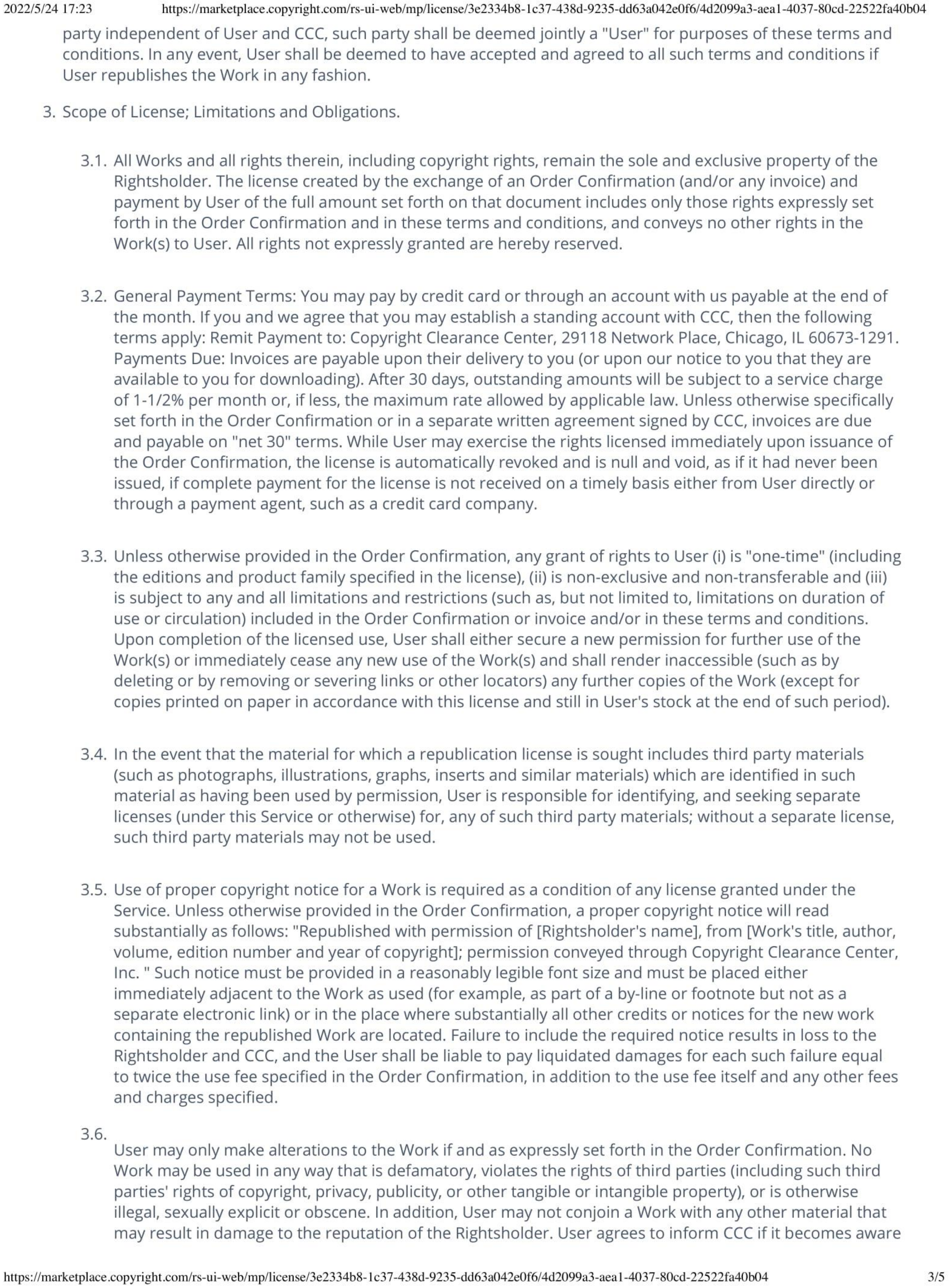
RIGHTSHOLDER TERMS AND CONDITIONS

Elsevier publishes Open Access articles in both its Open Access journals and via its Open Access articles option in subscription journals, for which an author selects a user license permitting certain types of reuse without permission. Before proceeding please check if the article is Open Access on http://www.sciencedirect.com and refer to the user license for the individual article. Any reuse not included in the user license terms will require permission. You must always fully and appropriately credit the author and source. If any part of the material to be used (for example, figures) has appeared in the Elsevier publication for which you are seeking permission, with credit or acknowledgement to another source it is the responsibility of the user to ensure their reuse complies with the terms and conditions determined by the rights holder. Please contact permissions@elsevier.com with any queries.

CCC Terms and Conditions

1. Description of Service; Defined Terms. This Republication License enables the User to obtain licenses for republication of one or more copyrighted works as described in detail on the relevant Order Confirmation (the "Work(s)"). Copyright Clearance Center, Inc. ("CCC") grants licenses through the Service on behalf of the rightsholder identified on the Order Confirmation (the "Rightsholder"). "Republication", as used herein, generally means the inclusion of a Work, in whole or in part, in a new work or works, also as described on the Order Confirmation. "User", as used herein, means the person or entity making such republication.

2. The terms set forth in the relevant Order Confirmation, and any terms set by the Rightsholder with respect to a particular Work, govern the terms of use of Works in connection with the Service. By using the Service, the person transacting for a republication license on behalf of the User represents and warrants that he/she/it (a) has been duly authorized by the User to accept, and hereby does accept, all such terms and conditions on behalf of User, and (b) shall inform User of all such terms and conditions. In the event such person is a "freelancer" or other third



party independent of User and CCC, such party shall be deemed jointly a "User" for purposes of these terms and conditions. In any event, User shall be deemed to have accepted and agreed to all such terms and conditions if User republishes the Work in any fashion.

3. Scope of License; Limitations and Obligations.

3.1. All Works and all rights therein, including copyright rights, remain the sole and exclusive property of the Rightsholder. The license created by the exchange of an Order Confirmation (and/or any invoice) and payment by User of the full amount set forth on that document includes only those rights expressly set forth in the Order Confirmation and in these terms and conditions, and conveys no other rights in the Work(s) to User. All rights not expressly granted are hereby reserved.

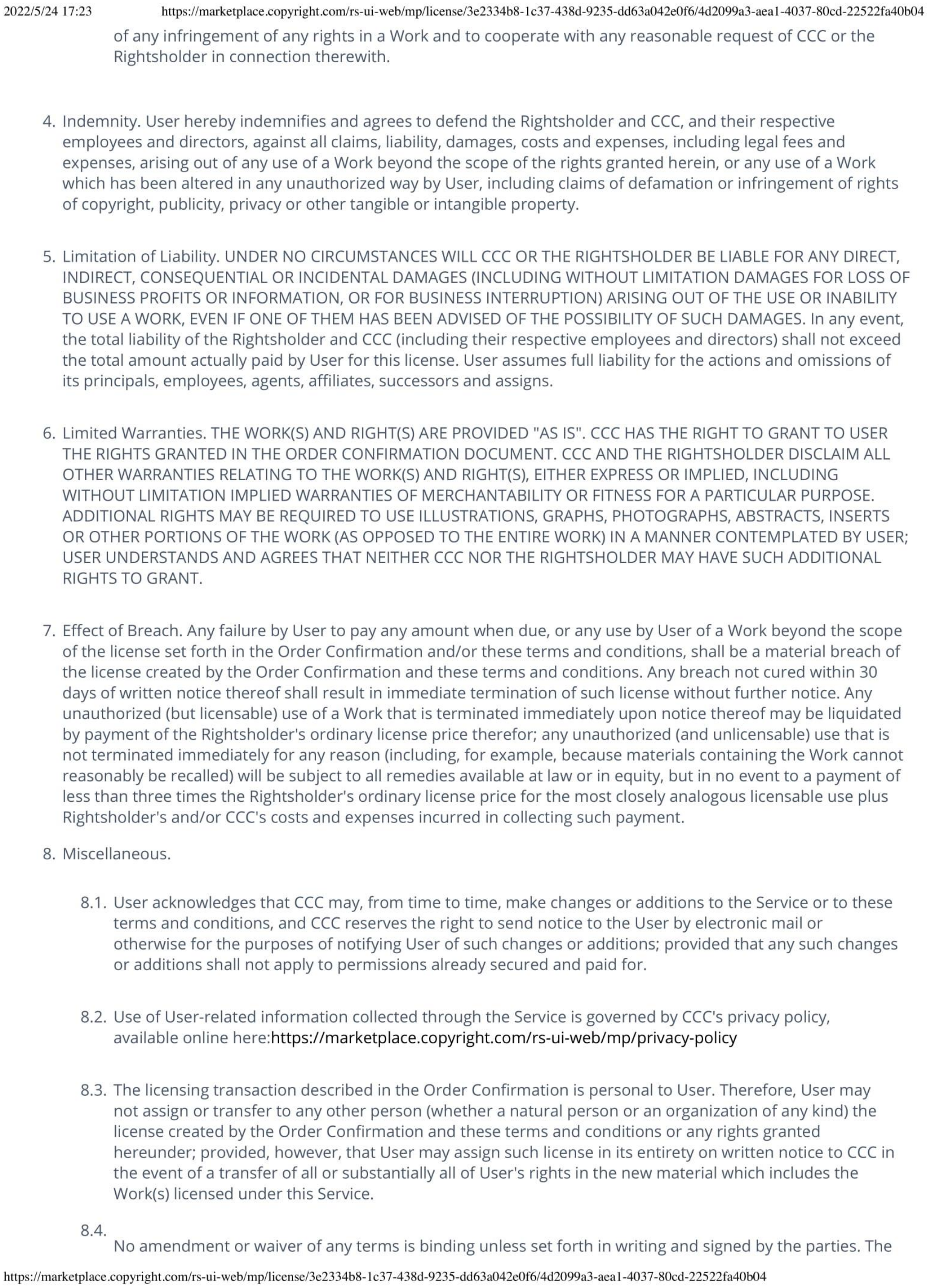
3.2. General Payment Terms: You may pay by credit card or through an account with us payable at the end of the month. If you and we agree that you may establish a standing account with CCC, then the following terms apply: Remit Payment to: Copyright Clearance Center, 29118 Network Place, Chicago, IL 60673-1291. Payments Due: Invoices are payable upon their delivery to you (or upon our notice to you that they are available to you for downloading). After 30 days, outstanding amounts will be subject to a service charge of 1-1/2% per month or, if less, the maximum rate allowed by applicable law. Unless otherwise specifically set forth in the Order Confirmation or in a separate written agreement signed by CCC, invoices are due and payable on "net 30" terms. While User may exercise the rights licensed immediately upon issuance of the Order Confirmation, the license is automatically revoked and is null and void, as if it had never been issued, if complete payment for the license is not received on a timely basis either from User directly or through a payment agent, such as a credit card company.

3.3. Unless otherwise provided in the Order Confirmation, any grant of rights to User (i) is "one-time" (including the editions and product family specified in the license), (ii) is non-exclusive and non-transferable and (iii) is subject to any and all limitations and restrictions (such as, but not limited to, limitations on duration of use or circulation) included in the Order Confirmation or invoice and/or in these terms and conditions. Upon completion of the licensed use, User shall either secure a new permission for further use of the Work(s) or immediately cease any new use of the Work(s) and shall render inaccessible (such as by deleting or by removing or severing links or other locators) any further copies of the Work (except for copies printed on paper in accordance with this license and still in User's stock at the end of such period).

3.4. In the event that the material for which a republication license is sought includes third party materials (such as photographs, illustrations, graphs, inserts and similar materials) which are identified in such material as having been used by permission, User is responsible for identifying, and seeking separate licenses (under this Service or otherwise) for, any of such third party materials; without a separate license, such third party materials may not be used.

3.5. Use of proper copyright notice for a Work is required as a condition of any license granted under the Service. Unless otherwise provided in the Order Confirmation, a proper copyright notice will read substantially as follows: "Republished with permission of [Rightsholder's name], from [Work's title, author, volume, edition number and year of copyright]; permission conveyed through Copyright Clearance Center, Inc. " Such notice must be provided in a reasonably legible font size and must be placed either immediately adjacent to the Work as used (for example, as part of a by-line or footnote but not as a separate electronic link) or in the place where substantially all other credits or notices for the new work containing the republished Work are located. Failure to include the required notice results in loss to the Rightsholder and CCC, and the User shall be liable to pay liquidated damages for each such failure equal to twice the use fee specified in the Order Confirmation, in addition to the use fee itself and any other fees and charges specified.

3.6.
User may only make alterations to the Work if and as expressly set forth in the Order Confirmation. No Work may be used in any way that is defamatory, violates the rights of third parties (including such third parties' rights of copyright, privacy, publicity, or other tangible or intangible property), or is otherwise illegal, sexually explicit or obscene. In addition, User may not conjoin a Work with any other material that may result in damage to the reputation of the Rightsholder. User agrees to inform CCC if it becomes aware



of any infringement of any rights in a Work and to cooperate with any reasonable request of CCC or the Rightsholder in connection therewith.

4. Indemnity. User hereby indemnifies and agrees to defend the Rightsholder and CCC, and their respective employees and directors, against all claims, liability, damages, costs and expenses, including legal fees and expenses, arising out of any use of a Work beyond the scope of the rights granted herein, or any use of a Work which has been altered in any unauthorized way by User, including claims of defamation or infringement of rights of copyright, publicity, privacy or other tangible or intangible property.

5. Limitation of Liability. UNDER NO CIRCUMSTANCES WILL CCC OR THE RIGHTSHOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL OR INCIDENTAL DAMAGES (INCLUDING WITHOUT LIMITATION DAMAGES FOR LOSS OF BUSINESS PROFITS OR INFORMATION, OR FOR BUSINESS INTERRUPTION) ARISING OUT OF THE USE OR INABILITY TO USE A WORK, EVEN IF ONE OF THEM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In any event, the total liability of the Rightsholder and CCC (including their respective employees and directors) shall not exceed the total amount actually paid by User for this license. User assumes full liability for the actions and omissions of its principals, employees, agents, affiliates, successors and assigns.

6. Limited Warranties. THE WORK(S) AND RIGHT(S) ARE PROVIDED "AS IS". CCC HAS THE RIGHT TO GRANT TO USER THE RIGHTS GRANTED IN THE ORDER CONFIRMATION DOCUMENT. CCC AND THE RIGHTSHOLDER DISCLAIM ALL OTHER WARRANTIES RELATING TO THE WORK(S) AND RIGHT(S), EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. ADDITIONAL RIGHTS MAY BE REQUIRED TO USE ILLUSTRATIONS, GRAPHS, PHOTOGRAPHS, ABSTRACTS, INSERTS OR OTHER PORTIONS OF THE WORK (AS OPPOSED TO THE ENTIRE WORK) IN A MANNER CONTEMPLATED BY USER; USER UNDERSTANDS AND AGREES THAT NEITHER CCC NOR THE RIGHTSHOLDER MAY HAVE SUCH ADDITIONAL RIGHTS TO GRANT.

7. Effect of Breach. Any failure by User to pay any amount when due, or any use by User of a Work beyond the scope of the license set forth in the Order Confirmation and/or these terms and conditions, shall be a material breach of the license created by the Order Confirmation and these terms and conditions. Any breach not cured within 30 days of written notice thereof shall result in immediate termination of such license without further notice. Any unauthorized (but licensable) use of a Work that is terminated immediately upon notice thereof may be liquidated by payment of the Rightsholder's ordinary license price therefor; any unauthorized (and unlicensable) use that is not terminated immediately for any reason (including, for example, because materials containing the Work cannot reasonably be recalled) will be subject to all remedies available at law or in equity, but in no event to a payment of less than three times the Rightsholder's ordinary license price for the most closely analogous licensable use plus Rightsholder's and/or CCC's costs and expenses incurred in collecting such payment.

8. Miscellaneous.

   8.1. User acknowledges that CCC may, from time to time, make changes or additions to the Service or to these terms and conditions, and CCC reserves the right to send notice to the User by electronic mail or otherwise for the purposes of notifying User of such changes or additions; provided that any such changes or additions shall not apply to permissions already secured and paid for.

   8.2. Use of User-related information collected through the Service is governed by CCC's privacy policy, available online here:https://marketplace.copyright.com/rs-ui-web/mp/privacy-policy

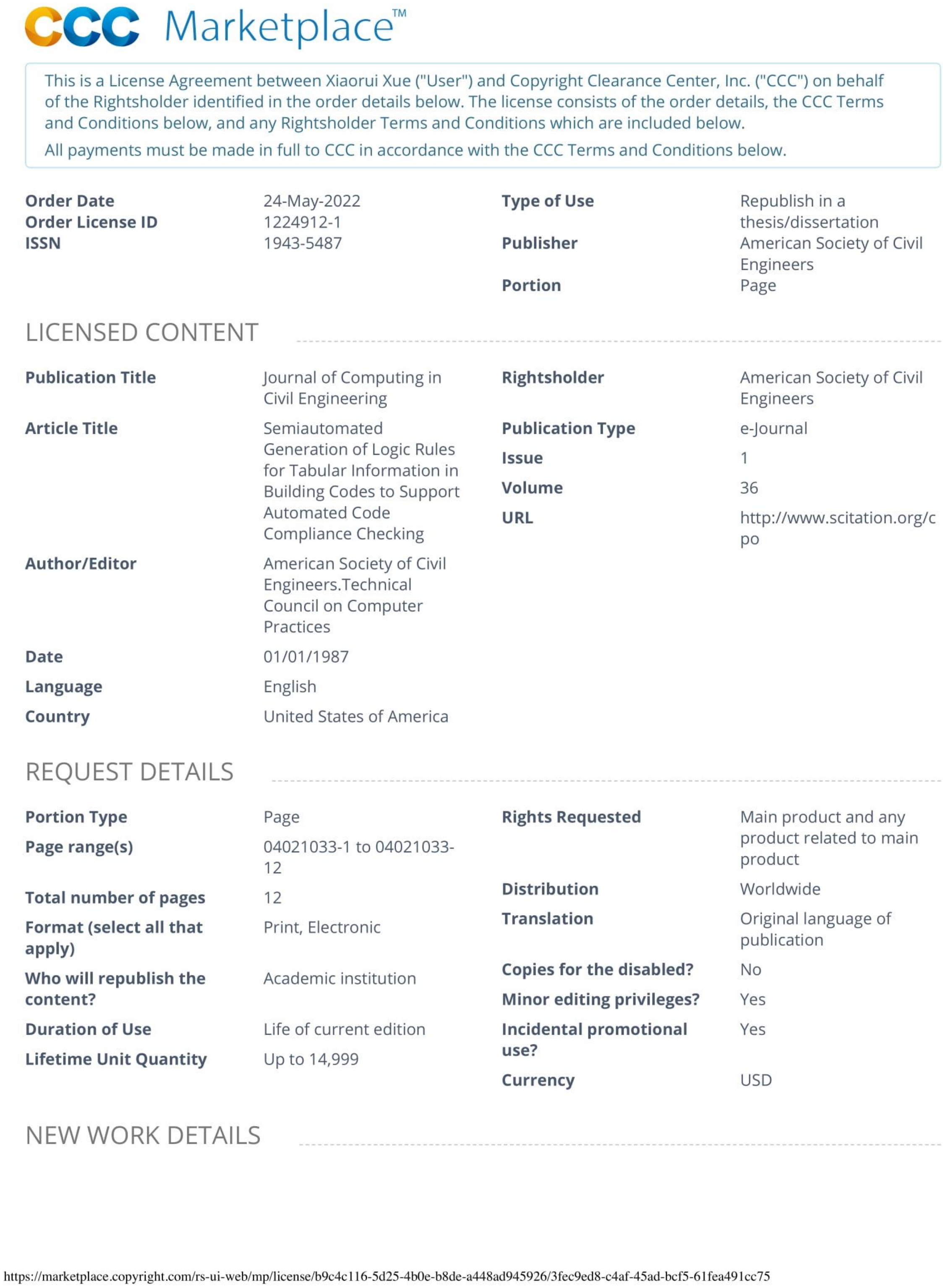
   8.3. The licensing transaction described in the Order Confirmation is personal to User. Therefore, User may not assign or transfer to any other person (whether a natural person or an organization of any kind) the license created by the Order Confirmation and these terms and conditions or any rights granted hereunder; provided, however, that User may assign such license in its entirety on written notice to CCC in the event of a transfer of all or substantially all of User's rights in the new material which includes the Work(s) licensed under this Service.

   8.4. No amendment or waiver of any terms is binding unless set forth in writing and signed by the parties. The

Rightsholder and CCC hereby object to any terms contained in any writing prepared by the User or its principals, employees, agents or affiliates and purporting to govern or otherwise relate to the licensing transaction described in the Order Confirmation, which terms are in any way inconsistent with any terms set forth in the Order Confirmation and/or in these terms and conditions or CCC's standard operating procedures, whether such writing is prepared prior to, simultaneously with or subsequent to the Order Confirmation, and whether such writing appears on a copy of the Order Confirmation or in a separate instrument.

8.5. The licensing transaction described in the Order Confirmation document shall be governed by and construed under the law of the State of New York, USA, without regard to the principles thereof of conflicts of law. Any case, controversy, suit, action, or proceeding arising out of, in connection with, or related to such licensing transaction shall be brought, at CCC's sole discretion, in any federal or state court located in the County of New York, State of New York, USA, or in any federal or state court whose geographical jurisdiction covers the location of the Rightsholder set forth in the Order Confirmation. The parties expressly submit to the personal jurisdiction and venue of each such federal or state court.If you have any comments or questions about the Service or Copyright Clearance Center, please contact us at 978-750-8400 or send an e-mail to support@copyright.com.

v 1.1



CCC Marketplace™

This is a License Agreement between Xiaorui Xue ("User") and Copyright Clearance Center, Inc. ("CCC") on behalf of the Rightsholder identified in the order details below. The license consists of the order details, the CCC Terms and Conditions below, and any Rightsholder Terms and Conditions which are included below.

All payments must be made in full to CCC in accordance with the CCC Terms and Conditions below.

Order Date	24-May-2022	**Type of Use**	Republish in a thesis/dissertation
Order License ID	1224912-1	**Publisher**	American Society of Civil Engineers
ISSN	1943-5487	**Portion**	Page

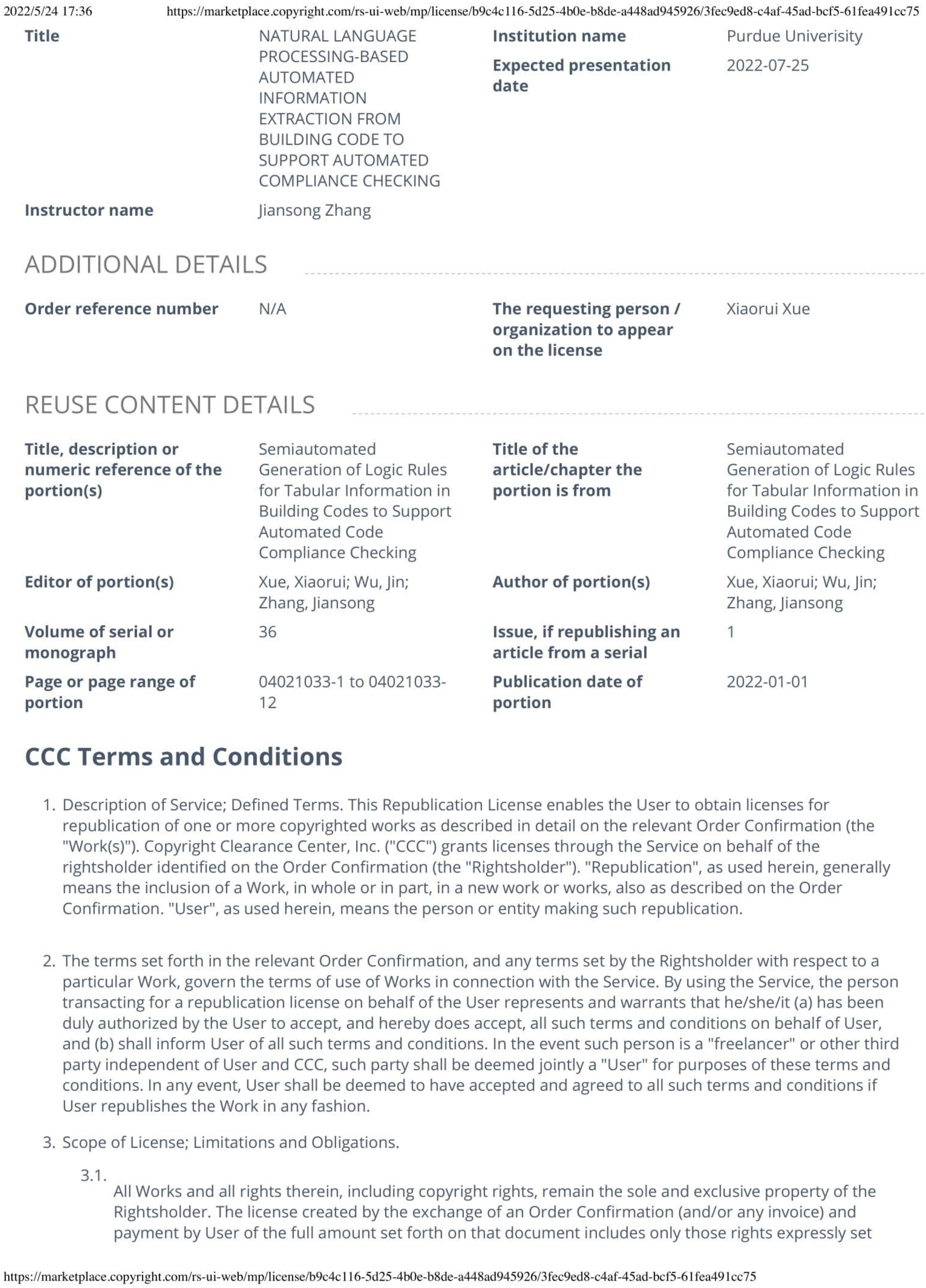
LICENSED CONTENT

Publication Title	Journal of Computing in Civil Engineering	**Rightsholder**	American Society of Civil Engineers
Article Title	Semiautomated Generation of Logic Rules for Tabular Information in Building Codes to Support Automated Code Compliance Checking	**Publication Type**	e-Journal
		Issue	1
		Volume	36
		URL	http://www.scitation.org/cpo
Author/Editor	American Society of Civil Engineers.Technical Council on Computer Practices		
Date	01/01/1987		
Language	English		
Country	United States of America		

REQUEST DETAILS

Portion Type	Page	**Rights Requested**	Main product and any product related to main product
Page range(s)	04021033-1 to 04021033-12		
Total number of pages	12	**Distribution**	Worldwide
		Translation	Original language of publication
Format (select all that apply)	Print, Electronic		
Who will republish the content?	Academic institution	**Copies for the disabled?**	No
		Minor editing privileges?	Yes
Duration of Use	Life of current edition	**Incidental promotional use?**	Yes
Lifetime Unit Quantity	Up to 14,999		
		Currency	USD

NEW WORK DETAILS



Title	NATURAL LANGUAGE PROCESSING-BASED AUTOMATED INFORMATION EXTRACTION FROM BUILDING CODE TO SUPPORT AUTOMATED COMPLIANCE CHECKING	**Institution name** **Expected presentation date**	Purdue Univeristy 2022-07-25
Instructor name	Jiansong Zhang		

ADDITIONAL DETAILS

Order reference number	N/A	**The requesting person / organization to appear on the license**	Xiaorui Xue

REUSE CONTENT DETAILS

Title, description or numeric reference of the portion(s)	Semiautomated Generation of Logic Rules for Tabular Information in Building Codes to Support Automated Code Compliance Checking	**Title of the article/chapter the portion is from**	Semiautomated Generation of Logic Rules for Tabular Information in Building Codes to Support Automated Code Compliance Checking
Editor of portion(s)	Xue, Xiaorui; Wu, Jin; Zhang, Jiansong	**Author of portion(s)**	Xue, Xiaorui; Wu, Jin; Zhang, Jiansong
Volume of serial or monograph	36	**Issue, if republishing an article from a serial**	1
Page or page range of portion	04021033-1 to 04021033-12	**Publication date of portion**	2022-01-01

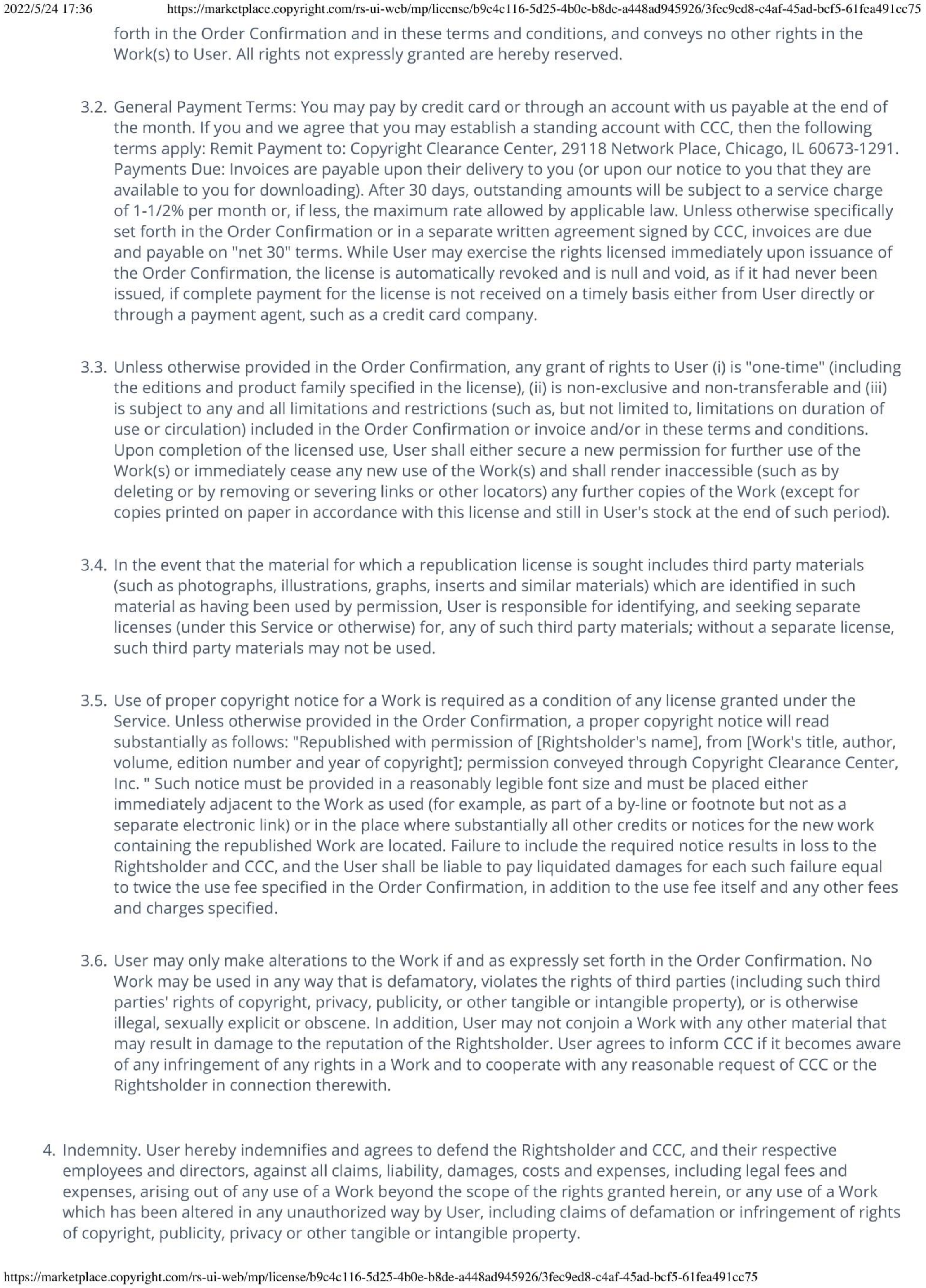
CCC Terms and Conditions

1. Description of Service; Defined Terms. This Republication License enables the User to obtain licenses for republication of one or more copyrighted works as described in detail on the relevant Order Confirmation (the "Work(s)"). Copyright Clearance Center, Inc. ("CCC") grants licenses through the Service on behalf of the rightsholder identified on the Order Confirmation (the "Rightsholder"). "Republication", as used herein, generally means the inclusion of a Work, in whole or in part, in a new work or works, also as described on the Order Confirmation. "User", as used herein, means the person or entity making such republication.

2. The terms set forth in the relevant Order Confirmation, and any terms set by the Rightsholder with respect to a particular Work, govern the terms of use of Works in connection with the Service. By using the Service, the person transacting for a republication license on behalf of the User represents and warrants that he/she/it (a) has been duly authorized by the User to accept, and hereby does accept, all such terms and conditions on behalf of User, and (b) shall inform User of all such terms and conditions. In the event such person is a "freelancer" or other third party independent of User and CCC, such party shall be deemed jointly a "User" for purposes of these terms and conditions. In any event, User shall be deemed to have accepted and agreed to all such terms and conditions if User republishes the Work in any fashion.

3. Scope of License; Limitations and Obligations.

   3.1. All Works and all rights therein, including copyright rights, remain the sole and exclusive property of the Rightsholder. The license created by the exchange of an Order Confirmation (and/or any invoice) and payment by User of the full amount set forth on that document includes only those rights expressly set



forth in the Order Confirmation and in these terms and conditions, and conveys no other rights in the Work(s) to User. All rights not expressly granted are hereby reserved.

3.2. General Payment Terms: You may pay by credit card or through an account with us payable at the end of the month. If you and we agree that you may establish a standing account with CCC, then the following terms apply: Remit Payment to: Copyright Clearance Center, 29118 Network Place, Chicago, IL 60673-1291. Payments Due: Invoices are payable upon their delivery to you (or upon our notice to you that they are available to you for downloading). After 30 days, outstanding amounts will be subject to a service charge of 1-1/2% per month or, if less, the maximum rate allowed by applicable law. Unless otherwise specifically set forth in the Order Confirmation or in a separate written agreement signed by CCC, invoices are due and payable on "net 30" terms. While User may exercise the rights licensed immediately upon issuance of the Order Confirmation, the license is automatically revoked and is null and void, as if it had never been issued, if complete payment for the license is not received on a timely basis either from User directly or through a payment agent, such as a credit card company.

3.3. Unless otherwise provided in the Order Confirmation, any grant of rights to User (i) is "one-time" (including the editions and product family specified in the license), (ii) is non-exclusive and non-transferable and (iii) is subject to any and all limitations and restrictions (such as, but not limited to, limitations on duration of use or circulation) included in the Order Confirmation or invoice and/or in these terms and conditions. Upon completion of the licensed use, User shall either secure a new permission for further use of the Work(s) or immediately cease any new use of the Work(s) and shall render inaccessible (such as by deleting or by removing or severing links or other locators) any further copies of the Work (except for copies printed on paper in accordance with this license and still in User's stock at the end of such period).

3.4. In the event that the material for which a republication license is sought includes third party materials (such as photographs, illustrations, graphs, inserts and similar materials) which are identified in such material as having been used by permission, User is responsible for identifying, and seeking separate licenses (under this Service or otherwise) for, any of such third party materials; without a separate license, such third party materials may not be used.

3.5. Use of proper copyright notice for a Work is required as a condition of any license granted under the Service. Unless otherwise provided in the Order Confirmation, a proper copyright notice will read substantially as follows: "Republished with permission of [Rightsholder's name], from [Work's title, author, volume, edition number and year of copyright]; permission conveyed through Copyright Clearance Center, Inc. " Such notice must be provided in a reasonably legible font size and must be placed either immediately adjacent to the Work as used (for example, as part of a by-line or footnote but not as a separate electronic link) or in the place where substantially all other credits or notices for the new work containing the republished Work are located. Failure to include the required notice results in loss to the Rightsholder and CCC, and the User shall be liable to pay liquidated damages for each such failure equal to twice the use fee specified in the Order Confirmation, in addition to the use fee itself and any other fees and charges specified.

3.6. User may only make alterations to the Work if and as expressly set forth in the Order Confirmation. No Work may be used in any way that is defamatory, violates the rights of third parties (including such third parties' rights of copyright, privacy, publicity, or other tangible or intangible property), or is otherwise illegal, sexually explicit or obscene. In addition, User may not conjoin a Work with any other material that may result in damage to the reputation of the Rightsholder. User agrees to inform CCC if it becomes aware of any infringement of any rights in a Work and to cooperate with any reasonable request of CCC or the Rightsholder in connection therewith.

4. Indemnity. User hereby indemnifies and agrees to defend the Rightsholder and CCC, and their respective employees and directors, against all claims, liability, damages, costs and expenses, including legal fees and expenses, arising out of any use of a Work beyond the scope of the rights granted herein, or any use of a Work which has been altered in any unauthorized way by User, including claims of defamation or infringement of rights of copyright, publicity, privacy or other tangible or intangible property.

5. Limitation of Liability. UNDER NO CIRCUMSTANCES WILL CCC OR THE RIGHTSHOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL OR INCIDENTAL DAMAGES (INCLUDING WITHOUT LIMITATION DAMAGES FOR LOSS OF BUSINESS PROFITS OR INFORMATION, OR FOR BUSINESS INTERRUPTION) ARISING OUT OF THE USE OR INABILITY TO USE A WORK, EVEN IF ONE OF THEM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In any event, the total liability of the Rightsholder and CCC (including their respective employees and directors) shall not exceed the total amount actually paid by User for this license. User assumes full liability for the actions and omissions of its principals, employees, agents, affiliates, successors and assigns.

6. Limited Warranties. THE WORK(S) AND RIGHT(S) ARE PROVIDED "AS IS". CCC HAS THE RIGHT TO GRANT TO USER THE RIGHTS GRANTED IN THE ORDER CONFIRMATION DOCUMENT. CCC AND THE RIGHTSHOLDER DISCLAIM ALL OTHER WARRANTIES RELATING TO THE WORK(S) AND RIGHT(S), EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. ADDITIONAL RIGHTS MAY BE REQUIRED TO USE ILLUSTRATIONS, GRAPHS, PHOTOGRAPHS, ABSTRACTS, INSERTS OR OTHER PORTIONS OF THE WORK (AS OPPOSED TO THE ENTIRE WORK) IN A MANNER CONTEMPLATED BY USER; USER UNDERSTANDS AND AGREES THAT NEITHER CCC NOR THE RIGHTSHOLDER MAY HAVE SUCH ADDITIONAL RIGHTS TO GRANT.

7. Effect of Breach. Any failure by User to pay any amount when due, or any use by User of a Work beyond the scope of the license set forth in the Order Confirmation and/or these terms and conditions, shall be a material breach of the license created by the Order Confirmation and these terms and conditions. Any breach not cured within 30 days of written notice thereof shall result in immediate termination of such license without further notice. Any unauthorized (but licensable) use of a Work that is terminated immediately upon notice thereof may be liquidated by payment of the Rightsholder's ordinary license price therefor; any unauthorized (and unlicensable) use that is not terminated immediately for any reason (including, for example, because materials containing the Work cannot reasonably be recalled) will be subject to all remedies available at law or in equity, but in no event to a payment of less than three times the Rightsholder's ordinary license price for the most closely analogous licensable use plus Rightsholder's and/or CCC's costs and expenses incurred in collecting such payment.

8. Miscellaneous.

    8.1. User acknowledges that CCC may, from time to time, make changes or additions to the Service or to these terms and conditions, and CCC reserves the right to send notice to the User by electronic mail or otherwise for the purposes of notifying User of such changes or additions; provided that any such changes or additions shall not apply to permissions already secured and paid for.

    8.2. Use of User-related information collected through the Service is governed by CCC's privacy policy, available online here:https://marketplace.copyright.com/rs-ui-web/mp/privacy-policy

    8.3. The licensing transaction described in the Order Confirmation is personal to User. Therefore, User may not assign or transfer to any other person (whether a natural person or an organization of any kind) the license created by the Order Confirmation and these terms and conditions or any rights granted hereunder; provided, however, that User may assign such license in its entirety on written notice to CCC in the event of a transfer of all or substantially all of User's rights in the new material which includes the Work(s) licensed under this Service.

    8.4. No amendment or waiver of any terms is binding unless set forth in writing and signed by the parties. The Rightsholder and CCC hereby object to any terms contained in any writing prepared by the User or its principals, employees, agents or affiliates and purporting to govern or otherwise relate to the licensing transaction described in the Order Confirmation, which terms are in any way inconsistent with any terms set forth in the Order Confirmation and/or in these terms and conditions or CCC's standard operating procedures, whether such writing is prepared prior to, simultaneously with or subsequent to the Order Confirmation, and whether such writing appears on a copy of the Order Confirmation or in a separate instrument.

    8.5. The licensing transaction described in the Order Confirmation document shall be governed by and construed under the law of the State of New York, USA, without regard to the principles thereof of conflicts of law. Any case, controversy, suit, action, or proceeding arising out of, in connection with, or related to such licensing transaction shall be brought, at CCC's sole discretion, in any federal or state court located in the County of New York, State of New York, USA, or in any federal or state court whose geographical jurisdiction covers the location of the Rightsholder set forth in the Order Confirmation. The parties expressly submit to the personal jurisdiction and venue of each such federal or state court.If you have any comments or questions about the Service or Copyright Clearance Center, please contact us at 978-750-8400 or send an e-mail to support@copyright.com.

v 1.1

REFERENCE

Abacha, A., & Zweigenbaum, P. (2011). Automatic extraction of semantic relations between medical entities: A rule based approach. *Journal of Biomedical Semantics, 2*(5), 1-11. https://doi.org/S4.10.1186/2041-1480-2-S5-S4

Abzianidze, L., & Bos, J. (2017). Towards universal semantic tagging. *Proceedings of the IWCS 2017-12th International Conference on Computational Semantics- Short papers,* Association for Computational Linguistics, 1-9. https://doi.org/10.48550/arXiv.1709.10381

Agarwal, A., Yadav, A., & Vishwakarma, D. K. (2019). Multimodal sentiment analysis via RNN variants. *Proceedings of the 2019 IEEE International Conference on Big Data, Cloud Computing, Data Science & Engineering (BCD)*, Institute of Electrical and Electronics Engineers,19-23. https://doi.org/10.1109/BCD.2019.8885108

Akanbi, T., & Zhang, J. (2022). Framework for developing IFC-based 3D documentation from 2D bridge drawings. *Journal of Computing in Civil Engineering, 36*(1), 04021031. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000986

Akanbi, T., Zhang, J., & Lee, Y. C. (2020). Data-driven reverse engineering algorithm development method for developing interoperable quantity takeoff algorithms using IFC-based BIM. *Journal of Computing in Civil Engineering, 34*(5). https://doi.org/10.1061/(ASCE)CP.1943-5487.0000909

Alghamdi, A., Sulaiman, M., Alghamdi, A., Alhosan, M., Mastali, M., & Zhang, J. (2017). Building accessibility code compliance verification using game simulations in virtual reality. *Proceedings of Computing in Civil Engineering 2017*, American Society of Civil Engineers, 262-270. https://doi.org/10.1061/9780784480830.033

Ambartsoumian, A., & Popowich, F. (2018). Self-attention: A better building block for sentiment analysis neural network classifiers. *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Association for Computational Linguistics, 130-139. https://doi.org/10.18653/v1/W18-6219

Amor, R., & Dimyadi, J. (2021). The promise of automated compliance checking. *Developments in the Built Environment*, *5*(March 2021), 100039. https://doi.org/10.1016/j.dibe.2020.100039

Amos, L., Anderson, D., Brody, S., Ripple, A., & Humphreys, B. L. (2020). Umls users and uses: A current overview. *Journal of the American Medical Informatics Association*, *27*(10), 1606-1611. https://doi.org/10.1093/jamia/ocaa084

Autodesk Company. (2022, March 23). What is BIM?. Autodesk Company. https://www.autodesk.com/industry/aec/bim

Awad, M., & Khanna, R. (2015). *Efficient learning machines: Theories, concepts, and applications for engineers and system designers,* Springer Nature. https://doi.org/10.1007/978-1-4302-5990-9

Azhar, S. (2011). Building information modeling (BIM): trends, benefits, risks, and challenges for the AEC industry. *Leadership and Management in Engineering*, *11*(3), 241-252. https://doi.org/10.1061/(ASCE)LM.1943-5630.0000127

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate, *Proceedings of the 3rd International Conference on Learning Representationsal,* OpenReview, 1-15. https://doi.org/10.48550/arXiv.1409.0473

Baktha, K., & Tripathy, B. (2017). Investigation of Recurrent Neural Networks in the field of sentiment analysis. *Proceedings of the 2017 International Conference on Communication and*

Signal Processing (ICCSP), Institute of Electrical and Electronics Engineers, 2047-2050. https://doi.org/10.1109/ICCSP.2017.8286763

Barone, A. V. M., Helcl, J., Sennrich, R., Haddow, B., & Birch, A. (2017). Deep architectures for neural machine translation. *Proceedings of the Second Conference on Machine Translation,* Association for Computational Linguistics, 99-107. https://doi.org/10.18653/v1/W17-4710

Barzilay, R., & Elhadad, M. (1999). Using lexical chains for text summarization. *Proceedings of the Advances in automatic text summarization*, Association for Computational Linguistics, 111-121. https://aclanthology.org/W97-0703

Bell, H., Bjorkhaug, L., & Hjelseth, E. (May 25th, 2022). *Standardized computable rules*. Standard.no, http://www.standard.no/no/Fagomrader/Bygg-og-anlegg/Digital-byggeprosess/Digitale-regelsjekkere

Bhutani, N., Suhara, Y., Tan, W.C., Halevy, A., & Jagadish, H. (2019). Open information extraction from question-answer pairs. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers),* Association for Computational Linguistics, 2294-2350. https://doi.org/10.18653/v1/N19-1239

Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with python*. O'Reilly Media, Inc. https://doi.org/10.5555/1717171

Brants, T. (2000). TNT: A statistical Partpart-of-Speech speech tagger. *Proceedings of the sixth conference on Applied Natural Language Processing*, Association for Computational Linguistics, 224-231. https://doi.org/10.3115/974147.974178

Brickley, D., Guha, R. V., & Layman, A. (1999). Resource Description Framework (RDF) schema specification. *Encyclopedia of Database Systems.* Springer. https://doi.org/10.1007/978-0-387-39940-9_1319

Brill, E. (1992). A simple rule-based Partpart-of-Speech speech tagger. *Proceedings of the third conference on Applied Natural Language Processing*, Association for Computational Linguistics,152-155. https://doi.org/10.3115/974499.974526

Brissi, S. G., Chong, O. W., Debs, L., & Zhang, J. (2021). A review on the interactions of robotic systems and lean principles in offsite construction. *Journal of Engineering, Construction and Architectural Management*. https://doi.org/10.1108/ECAM-10-2020-0809

Building Advisor. (2015, June 6). *Model Construction Contract*. Building advisor. https://buildingadvisor.com/project-management/contracts/model-construction-contract_1

BuildingSMART International. (2019, Juauary 13). *Industry Foundation Classes (IFC).* BuildingSMART International. https://www.buildingsmart.org/standards/bsi-standards/industry-foundation-classes

Buitelaar, P., Cimiano, P., Racioppa, S., & Siegel, M. (2006). Ontology-based information extraction with SOBA. *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, European Language Resources Association (ELRA), 2321-2324. http://www.lrec-conf.org/proceedings/lrec2006/pdf/93.pdf

Bureau of Economic Analysis. (2022, June 29). *BEA Industry Facts-Construction*. Bureau of Economic Analysis. https://apps.bea.gov/industry/factsheet/factsheet.html#23

Bureau of Labor Statistics. (2018, January 1). *Measuring productivity growth in construction*. Bureau of Labor Statistics. https://www.bls.gov/opub/mlr/2018/article/measuring-productivity-growth-in-construction.htm

Bus, N., Roxin, A., Picinbono, G., & Fahad, M. (2019). Towards French smart building code: Compliance checking based on semantic rules, *Proceedings of the 6th Linked Data in Architecture and Construction Workshop*, 6-15. https://doi.org/10.48550/arXiv.1910.00334

Butte College (2009, Septempter 11). *The Eight Parts parts of Speechspeech*. Butte College, http://www.butte.edu/departments/cas/tipsheets/grammar/parts_of_speech.html

Cahyono, S. (2019). Comparison of document similarity measurements in scientific writing using Jaro-Winkler distance method and paragraph vector method. *Proceedings of the IOP Conference Series: Materials Science and Engineering*, IOP Publishing, 1-9. https://doi.org/052016.10.1088/1757-899X/662/5/052016

Calijorne Soares, M. A., & Parreiras, F. S. (2020). A literature review on question answering techniques, paradigms and systems. *Journal of King Saud University - Computer and Information Sciences, 32*(6), 635-646. https://doi.org/10.1016/j.jksuci.2018.08.005

Cao, L., Li, Y., Zhang, J., Jiang, Y., Han, Y., & Wei, J. (2020). Electrical load prediction of healthcare buildings through single and ensemble learning. *Journal of Energy Reports*, 6, 2751-2767. https://doi.org/10.1016/j.egyr.2020.10.005

Chan, W., Jaitly, N., Le, Q., & Vinyals, O. (2016). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Institute of Electrical and Electronics Engineers, 4960-4964. https://doi.org/10.1109/ICASSP.2016.7472621

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 1724–1734. https://doi.org/10.3115/v1/D14-1179

Choi, J., & Kim, I. (2017). A methodology of building code checking system for building permission based on openBIM. *Proceedings of the International Symposium on Automation and Robotics in Construction*, Vilnius Gediminas Technical University, Department of Construction Economics, 1-6. https://doi.org/10.22260/ISARC2017/0131

Chollet, F. (2017). *Deep learning with python*. Manning Publications. https://doi.org/10.5555/3203489

Chowdhury, G. G. (2003). Natural Language Processing. *Annual Review of Information Science And Technology*, *37*(1), 51-89. https://doi.org/10.1002/aris.1440370103

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of Gated Recurrent Neural Networks on sequence modeling, *Proceedings of the NIPS 2014 Deep Learning and Representation Learning Workshop,* 1-9. https://doi.org/10.48550/arXiv.1412.3555

Ciregan, D., Meier, U., & Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, Institute of Electrical and Electronics Engineers, 3642-3649. *https://doi.org/*10.1109/CVPR.2012.6248110

City of San Clemente (2019, March 5). *Ordinance No.1668*. City of San Clemente, https://www.san-clemente.org/Home/ShowDocument?id=50617

Coden, A. R., Pakhomov, S. V., Ando, R. K., Duffy, P. H., & Chute, C. G. (2005). Domain-specific language models and lexicons for tagging. *Journal of Biomedical Informatics, 38*(6), 422-430. https://doi.org/10.1016/j.jbi.2005.02.009

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural Language Processing (almost) from scratch. *Journal of Machine Learning Research*, *12*(Aug), 2493-2537. https://doi.org/10.5555/1953048.2078186

Correa, A., & Zander, P. (2017). Unleashing tabular content to open data: A survey on PDF table extraction methods and tools. *Proceedings of the 18th Annual International Conference on Digital Government Research*, Association for Computational Linguistics, 54-63. https://doi.org/10.1145/3085228.3085278

Cowie, J., & Lehnert, W. (1996). Information Extraction. *Commun*, *39*(1), 80–91. https://doi.org/10.1145/234173.234209

Crowston, K., Liu, X., & Allen, E. E. (2010). Machine learning and rule-based automated coding of qualitative data. *Proceedings of the American Society for Information Science and Technology*, Association for Information Science and Technology, 1-2. https://doi.org/10.1002/meet.1450470132

Cunningham, H. (2002). GATE, a general architecture for text engineering. *Computers and the Humanities*, *36*(2), 223-254. https://doi.org/10.1023/A:1014348124664

Delis, E. A., & Delis, A. (1995). Automatic fire-code checking using expert-system technology. *Journal of Computing in Civil Engineering*, *9*(2), 141-156. https://doi.org/10.1061/(ASCE)0887-3801(1995)9:2(141)

Dell'Orletta, F. (2009). Ensemble system for Partpart-of-Speech speech tagging. *Evaluation of NLP and Speech Tools for Italian*, *9*(2009) , 1-8. https://corpusitaliano.it/static/documents/POSILC.pdf

Devlin, J., Chang, M.W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/v1/N19-1423

Dimyadi, J., & Amor, R. (2013). Automated building code compliance checking–where is it at? *Proceedings of the 19th International CIB World Building Congress*, Queensland University of Technology, 1-14. https://doi.org/10.13140/2.1.4920.4161

Dimyadi, J., Clifton, C., Spearpoint, M., & Amor, R. (2016). Computerizing regulatory knowledge for building engineering design. *Journal of Computing in Civil Engineering*, *30*(5), C4016001. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000572

Ding, L., Drogemuller, R., Rosenman, M., Marchant, D., & Gero, J. (2006). Automating code checking for building designs-DesignCheck. *Clients Driving Innovation: Moving Ideas into Practice*, Cooperative Research Centre (CRC) for Construction Innovation, 1-16. https://ro.uow.edu.au/engpapers/4842/

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

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179-211. https://doi.org/10.1016/0364-0213(90)90002-E

Explosion AI (2022, June 17). Spacy-industrial-strength Natural Language Processing in python. Explosion AI, https://spacy.io/

Fenves, S. J. (1966). Tabular decision logic for structural design. *Journal of the Structural Division*, *92*(6), 473-490. https://doi.org/ 0.1061/JSDEAG.0001567

Findwell (2020 March 17). Building Code. Findwell. https://www.findwell.com/real-estate-dictionary/definition/building-code

Firat, O., Cho, K., & Bengio, Y. (2016). Multi-way, multilingual neural machine translation with a shared attention mechanism. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, 866–875. https://doi.org/10.18653/v1/N16-1101

Francis, W. N., & Kucera, H. (1979, July 1). *Brown corpus manual*. Brown University. http://korpus.uib.no/icame/manuals/BROWN/INDEX.HTM

Gali, K., Surana, H., Vaidya, A., Shishtla, P. M., & Sharma, D. M. (2008) Aggregating machine learning and rule based heuristics for Named Entity Recognition. *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*. Association for Computational Linguistics, 25-32. https://www.aclweb.org/anthology/I08-5005

Garrett, J. H., & Fenves, S. J. (1987). A knowledge-based standards processor for structural component design. *Engineering with computers*, *2*(4), 219-238. https://doi.org/0.1007/BF01276414

Gero, J. S. (1990). Design prototypes: A knowledge representation schema for design. *AI magazine*, *11*(4), 26-26. https://doi.org/10.1609/aimag.v11i4.854

Getuli, V., Ventura, S. M., Capone, P., & Ciribini, A. L. (2017). BIM-based code checking for construction health and safety. *Procedia Engineering*, *196*(2017), 454-461. https://doi.org/10.1016/j.proeng.2017.07.224

Giménez, J., & Marquez, L. (2004). Fast and accurate Partpart-of-Speech speech tagging: The SVM approach revisited. *Recent Advances in Natural Language Processing III*, John Benjamins Publishing Company, 153-162. https://doi.org/10.1075/cilt.260.17gim

Glorot, X., Bordes, A., & Bengio, Y. Deep sparse Rectifier Neural Networks. *Proceedings of Fourteenth International Conference on Artificial Intelligence and Statistics*, MIT Press, 315-323. https://proceedings.mlr.press/v15/glorot11a/glorot11a.pdf

Gomaa, W. H., & Fahmy, A. A. (2013). A survey of text similarity approaches. *International Journal of Computer Applications*, *68*(13), 13-18. https://doi.org/10.5120/11638-7118

Greene, B. B., & Rubin, G. M. (1971). Automatic grammatical tagging of English. Department of Linguistics, Brown University. https://books.google.com/books?id=VznTygAACAAJ

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, *5*(2), 199-221. https://doi.org/10.1006/knac.1993.1008

Guo, X., Tian, C., Chen, Y., & Zhang, J. (2022). Case study of building information modeling implementation in infrastructure projects. *Journal of Transportation Research Record*, 2676(2), 663-679. https://doi.org/10.1177%2F03611981211045060

Hassan, F. U., & Le, T. (2020). Automated requirements identification from construction contract documents using Natural Language Processing. *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, *12*(2), 04520009. https://doi.org/10.1061/(ASCE)LA.1943-4170.0000379

He, B., Zhou, D., Xiao, J., Liu, Q., Yuan, N. J., & Xu, T. (2020). BERT-MK: Integrating graph contextualized knowledge into pre-trained language models. *Findings of the Association for Computational Linguistics: EMNLP 2020*, Association for Computational Linguistics, 2281-2290. https://doi.org/10.18653/v1/2020.findings-emnlp.207

Hendler, J., & McGuinness, D. L. (2000). The DARPA agent markup language. *IEEE Intelligent Systems,* 15*(6)* , 67-73. https://doi.org/10.1109/5254.895864

Hepp, M. (2008). Goodrelations: An ontology for describing products and services offers on the web. *Proceedings of the International Conference on Knowledge Engineering and Knowledge Management*, Springer, 329-346. https://doi.org/10.1007/978-3-540-87696-0_29

Hirschberg, J., & Christopher, M. (2015). Advances in Natural Language Processing. *Science, 349*(6245), 261-266. https://doi.org/10.1126/science.aaa8685

Hochreiter, S. (1998). The vanishing gradient problem during learning Recurrent Neural Nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6*(02), 107-116. https://doi.org/10.1142/S0218488598000094

Hu, D. (2019). An introductory survey on attention mechanisms in NLP problems. *SAI Intelligent Systems Conference*, Springer, 432-448. https://doi.org/10.1007/978-3-030-29513-4_31

İlal, S. M., & Günaydın, H. M. (2017). Computer representation of building codes for automated compliance checking. *Automation in Construction, 82*(October 2017), 43-58. https://doi.org/10.1016/j.autcon.2017.06.018

International Code Council (2015). *International Building Code*, International Code Council, https://codes.iccsafe.org/content/IBC2015

Jiang, J. (2012). Information extraction from text. *Mining text data*, Springer US, 11-41. https://doi.org/10.1007/978-1-4614-3223-4_2

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., & Mikolov, T. (2017). Fasttext. Zip: Compressing text classification models. *https://doi.org/10.48550/arXiv.1612.03651*

Karita, S., Chen, N., Hayashi, T., Hori, T., Inaguma, H., Jiang, Z., Someki, M., Soplin, N. E. Y., Yamamoto, R., & Wang, X. (2019). A comparative study on transformer vs RNN in speech applications. *Proceedings of the 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Institute of Electrical and Electronics Engineers, 449-456. https://doi.org/10.1109/ASRU46091.2019.9003750

Kloo, I., Dabkowski, M. F., & Huddleston, S. H. (2019). Improving record linkage for counter-threat finance intelligence with dynamic jaro-winkler thresholds. *Proceedings of the 2019 Winter Simulation Conference (WSC)*, Institute of Electrical and Electronics Engineers, 2467-2478. https://doi.org/10.1109/WSC40007.2019.9004945

Koehn, P. (2009). Statistical machine translation, *ACM Computing Surveys, 40*(3), 1-49. https://doi.org/10.1145/1380584.1380586

Kosub, S. (2019). A note on the triangle inequality for the Jaccard distance. *Pattern Recognition Letters, 120*(1 April 2019), 36-38. https://doi.org/10.1016/j.patrec.2018.12.007

Kottmann, J., Margulies, B., Ingersoll, G., Drost, I., Kosin, J., Baldridge, J., Goetz, T., Morton, T., Silva, W., & Autayeu, A. (2004, April 22). Apache OpenNLP, Apache Foundation, https://opennlp.apache.org

Krüpl, B., & Herzog, M. (2006). Visually guided bottom-up table detection and segmentation in web documents. *Proceedings of the 15th International Conference on World Wide Web,* Association for Computing Machinery, 933-934. https://doi.org/10.1145/1135777.1135951

Kwayu, K. M., Kwigizile, V., Zhang, J., & Oh, J. S. (2020). Semantic n-gram feature analysis and machine learning–based classification of drivers' hazardous actions at signal-controlled intersections. *Journal of Computing in Civil Engineering, 34*(4), 04020015. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000895

Lacny, C., & Zhang, J. (2022, March). Computer vision-based geometry mapping and matching of building elements for construction robotic applications. *Proceedings of the Construction*

Research Congress 2020, American Society of Civil Engineers, 541-549. https://doi.org/10.1061/9780784483961.057

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

Lee, J., Ham, Y., Yi, J.S., & Son, J. (2020). Effective risk positioning through automated identification of missing contract conditions from the contractor’s perspective based on fidic contract cases. *Journal of Management in Engineering*, *36*(3), 05020003. https://doi.org/10.1061/(ASCE)ME.1943-5479.0000757

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, *36*(4), 1234-1240. https://doi.org/10.1093/bioinformatics/btz682

Lee, Y.C., Ghannad, P., Dimyadi, J., Lee, J.K., Solihin, W., & Zhang, J. (2020). A comparative analysis of five rule-based model checking platforms. *Proceedings of the Construction Research Congress 2020*, American Society of Civil Engineers, 1127-1136. https://doi.org/10.1061/9780784482865.119

Lee, Y.C., Ghannad, P., Shang, N., Eastman, C., & Barrett, S. (2018). Graphical scripting approach integrated with speech recognition for BIM-based rule checking. *Proceedings of the Construction Research CSongress 2018*, American Society of Civil Engineers, 262-272. https://doi.org/10.1061/9780784481264.026

Legaltemplates (2020, November 23). Free construction contract template. Legaltemplates, https://legaltemplates.net/form/lt/construction-contract-agreement

Levenshtein, V. I. (1965). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, *10*(8), 707-710. https://ui.adsabs.harvard.edu/abs/1966SPhD...10..707L/abstract

Li, H., & Zhang, J. (2022). IFC-based information extraction and analysis of HVAC objects to support building energy modeling. *Proceedings of the 39th International Symposium on Automation and Robotics in Construction (ISARC 2022),* I.A.A.R.C., iaarc.org., 159-166.

Li, H., & Zhang, J. Interoperability between BIM and BEM using IFC. (2020). *Proceedings of the Construction Research Congress 2020*, American Society of Civil Engineers, 630-637. https://doi.org/10.1061/9780784483893.078

Li, H., Zhang, J., Xue, X., Debs, L., Chang, S., Qu, M., Sparking, A., & Goldwasser, D. (2022). Issues in bi-directional interoperability between BIM and BEM. *Proceedings of the Construction Research Congress*, American Society of Civil Engineers, 1355-1364. https://doi.org/10.1061/9780784483961.142

Li, S., Cai, H., & Vineet, R. K. (2016). Integrating Natural Language Processing and spatial reasoning for utility compliance checking. *Journal of Construction Engineering and Management, 142*(12),04016074. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001199

Li, S., Graça, J. V., & Taskar, B. (2012) Wiki-ly supervised Partpart-of-Speech speech tagging. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Association for Computational Linguistics, 1389-1398. https://doi.org/10.5555/2390948.2391106

Li, Y., Cao, L., Zhang, J., Jiang, Y., & Han, Y. (2021). Development of an energy-oriented layout planning framework for healthcare facilities. *Proceedings of the 2021 ASCE International Conference on Computing in Civil Engineering*, American Society of Civil Engineers, 1016-1023. https://doi.org/10.1061/9780784483893.125

Liao, S.H. (2005). Expert system methodologies and applications-a decade review from 1995 to 2004. *Expert Systems with Applications*, *28*(1), 93-103. https://doi.org/10.1016/j.eswa.2004.08.003

Lin, J.R., Hu, Z., & Zhang, J. (2013). BIM oriented intelligent data mining and representation. *Proceedings of the 30th CIB W78 International Conference on Applications of IT in the AEC Industry*, Queensland University of Technology, 280-289. https://itc.scix.net/pdfs/w78-2013-paper-112.pdf

Loper, E., & Bird, S. (2004). NLTK: The natural language toolkit. *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, Association for Computational Linguistics, 1-4. https://aclanthology.org/P04-3031

Lopez, L., Elam, S., & Reed, K. (1989). Software concept for checking engineering designs for conformance with codes and standards. *Engineering with Computers, 5*(2), 63-78. https://doi.org/10.1007/BF01199070

Lu, J., Yang, J., Batra, D., & Parikh, D. (2016). Hierarchical question-image co-attention for visual question answering. *Advances In Neural Information Processing Systems*, Curran Associates Inc., 289-297. https://doi.org/10.5555/3157096.3157129

Luong, M.T., Pham, H., & Manning, C. D. (2015). Effective approaches to Attention-based neural machine translation. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 1412–1421. https://doi.org/10.18653/v1/D15-1166

Malsane, S., Matthews, J., Lockley, S., Love, P. E. , & Greenwood, D. (2015). Development of an object model for automated compliance checking. *Automation in Construction, 49*(January 2015), 51-58. https://doi.org/ 10.1016/j.autcon.2014.10.004

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing toolkit. *Proceedings of the 52nd annual meeting of the Association for Computational Linguistics: system demonstrations,* Association for Computational Linguistics, 55-60. https://doi.org/10.3115/v1/P14-5010

Marcus, M., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics, 19*(2), 313-330. https://doi.org/10.5555/972470.972475

Marques, N. C., & Lopes, G. P. (2001). Tagging with small training corpora, *Advances in Intelligent Data Analysis*. Springer, 63-72. https://doi.org/10.1007/3-540-44816-0_7

Mcguinness, D. L., Fikes, R., Hendler, J., & Stein, L. A. (2002). Daml+oil: An ontology language for the semantic web. *IEEE Intelligent Systems*, *17*(5), 72-80. https://doi.org/10.1109/MIS.2002.1039835

McKeown, K. (1985). *Text generation,* Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511620751

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2,* Curran Associates Inc., 3111–3119. https://doi.org/10.5555/2999792.2999959

Mitchell, T. M. (1997). *Machine Learning*, McGraw-Hill, Inc,. https://doi.org/10.5555/541177

Montrose County. (2018, April 24). Construction Contract. Montrose County, https://www.montrosecounty.net/DocumentCenter/View/823/Sample-Construction-Contract

Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). Natural Language Processing: An introduction. *Journal of the American Medical Informatics Association*, *18*(5), 544-551. https://doi.org/ 10.1136/amiajnl-2011-000464

National Building Information Model Standard Project Committee. (2022, May 27). Frequently asked questions about the national BIM standard-United States. National Building Information Model Standard Project Committee. https://www.nationalbimstandard.org/faqs#faq1

Nenkova, A., & McKeown, K. (2012). A survey of text summarization techniques. *Mining Text Data*, Springer US, 43-76. https://doi.org/10.1007/978-1-4614-3223-4_3

Nguyen, T.H., & Kim, J.L (2011). Building code compliance checking using BIM technology. *Proceedings of the 2011 Winter Simulation Conference (WSC)*, Institute of Electrical and Electronics Engineers, 3395-3400. https://doi.org/10.1109/WSC.2011.6148035

Noy, N. F., & McGuinness, D. L. (2012, June 24). Ontology development 101: A guide to creating your first ontology. Stanford Knowledge Systems Laboratory, http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html

Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *Proceedings of the INCCST 2020: 2nd International Conference on Computational Sciences and Technologies*, Mehran University of Engineering and Technology, 124-133. https://doi.org/10.48550/arXiv.1811.03378

Paliwal, S. S., Vishwanath, D., Rahul, R., Sharma, M., & Vig, L. (2019). Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images. *Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR)*, Institute of Electrical and Electronics Engineers, 128-133. https://doi.org/0.1109/ICDAR.2019.00029

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, *22*(10), 1345-1359. https://doi.org/10.1109/TKDE.2009.191

Pauwels, P., Van Deursen, D., Verstraeten, R., De Roo, J., De Meyer, R., Van de Walle, R., & Van Campenhout, J. (2011). A semantic rule checking environment for building performance checking. *Automation in Construction*, *20*(5), 506-518. https://doi.org/10.1016/j.autcon.2010.11.017

Pennington, J., Socher, R., & Manning, C. D. (2001). Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP),* Association for Computational Linguistics, 1532-1543. https://doi.org/10.3115/v1/D14-1162

Penumatsa, P., Ventura, M., Graesser, A. C., Louwerse, M., Hu, X., Cai, Z., & Franceschetti, D. R. (2006). The right threshold value: What is the right threshold of cosine measure when using latent semantic analysis for evaluating student answers? *International Journal on Artificial Intelligence Tools*, *15*(05), 767-777. https://doi.org/10.1142/S021821300600293X

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, Association for Computational Linguistics, 2227–2237. https://doi.org/10.18653/v1/N18-1202

Petrov, S., Das, D., & McDonald, R. (2012). A universal Partpart-of-Speech speech tagset. *Proceedings of the Eighth International Conference on Language Resources and Evaluation*

(LREC'12). European Language Resources Association (ELRA), 2089–2096. https://aclanthology.org/L12-1115/

Plank, B., Søgaard, A., & Goldberg, Y. (2016). Multilingual Partpart-of-Speech speech tagging with bidirectional long short-term memory models and auxiliary loss. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers),* Association for Computational Linguistics, 412–418. https://doi.org/10.18653/v1/P16-2067

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, & Y., Schwarz, P. (2011). The Kaldi speech recognition toolkit. *Proceedings of the IEEE 2011 Workshop on Automatic Speech Recognition and Understanding,* Association for Computational Linguistics, 1-4. https://doi.org/10.1109/ASRU17718.2011

Preidel, C., & Borrmann, A. (2017). Refinement of the visual code checking language for an automated checking of building information models regarding applicable regulations. *Computing in Civil Engineering 2017,* American Society of Civil Engineers, 157-165. https://doi.org/10.1061/9780784480823.020

Qasim, S. R., Mahmood, H., & Shafait, F. (2019). Rethinking table recognition using graph neural networks. *Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR),* Institute of Electrical and Electronics Engineers,142-147. https://doi.org/10.1109/ICDAR.2019.00031

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019, February 14). Language models are unsupervised multitask learners. *OpenAI blog,* https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf

Raghavan, P., Schütze, H., & Manning, C. D. (2010). Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze: Introduction to information retrieval. *Information Retrieval, 39*(2), 192-195. https://doi.org/10.1007/s10791-009-9115-y

Rao, X., & Ke, Z. (2018). Hierarchical RNN for information extraction from lawsuit documents. *Proceedings of the International MultiConference of Engineers and Computer Scientists 2018,* International Association of Engineers, 1-5. https://doi.org/10.48550/arXiv.1804.09321

Rekabsaz, N., Lupu, M., & Hanbury, A. (2016). Exploration of a threshold for similarity based on uncertainty in word embedding. *Proceedings of the European Conference on Information Retrieval,* Springer, 396-409. https://doi.org/10.1007/978-3-319-56608-5_31

Ren, R., & Zhang, J. (2020). Comparison of BIM interoperability applications at different structural analysis stages. *Proceedings of the* Construction Research Congress 2020, American Society of Civil Engineers, 537-545. https://doi.org/10.1061/9780784482865.057

Ren, R., & Zhang, J. (2021). Semantic rule-based construction procedural information extraction to guide jobsite sensing and monitoring. *Journal of Computing in Civil Engineering, 35*(6), 04021026. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000971

Rocktäschel, T., Grefenstette, E., Hermann, K. M., Kočiský, T., & Blunsom, P. (2015). Reasoning about entailment with neural attention. *Proceedings of the 4th International Conference on Learning Representations,* 1-9. https://doi.org/10.48550/arXiv.1509.06664

Sabouni, A., & Al-Mourad, O. (1997). Quantitative knowledge based approach for preliminary design of tall buildings. *Artificial intelligence in Engineering*, *11*(2), 143-154. https://doi.org/10.1016/S0954-1810(96)00023-4

Sak, H., Senior, A. W., & Beaufays, F. (2014). Long Short-Term memory Recurrent Neural Network architectures for large scale acoustic modeling. *https://doi.org/10.48550/arXiv.1402.1128*

Salama, D. M., & El-Gohary, N. M. (2016). Semantic text classification for supporting automated compliance checking in construction. *Journal of Computing in Civil Engineering*, *30*(1), 04014106. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000301

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, *3*(3), 210-229. https://doi.org/10.1147/rd.33.0210

Schmid, H. (1994). Part-of-Speech speech tagging with neural networks. *Proceedings of the 15th conference on Computational linguistics-Volume 1*, Association for Computational Linguistics, 172-176. https://doi.org/10.3115/991886.991915

Schmid, H. (2019). Deep learning-based morphological taggers and lemmatizers for annotating historical texts. *Proceedings of the 3rd International Conference on Digital Access to Textual Cultural Heritage*, Association for Computational Linguistics, 133-137. https://doi.org/10.1145/3322905

Schmid, H., Baroni, M., Zanchetta, E., & Stein, A. (2014). The enriched tree tagger system. *Proceedings of the Evaluation of NLP and Speech Tools for Italian 2007*, Springer. http://hdl.handle.net/11572/70059

Shafait, F., & Smith, R. (2010). Table detection in heterogeneous documents. *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, Association for Computational Linguistics, 65-72. https://doi.org/10.1145/1815330.1815339

Shao, Y., Hardmeier, C., Tiedemann, J.,& Nivre, J. (2017). Character-based joint segmentation and POS tagging for Chinese using bidirectional RNN-CRF. *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, 173–183. https://aclanthology.org/I17-1018

Shmanina, T., Zukerman, I., Cheam, A. L., Bochynek, T., & Cavedon, L. (2016). A corpus of tables in full-text biomedical research publications. *Proceedings of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM2016)*, Association for Computational Linguistics, 70-79. https://aclanthology.org/W16-5108

Sinha, A., Bayer, J., & Bukhari, S. S. (2019) Table localization and field value extraction in piping and instrumentation diagram images. *Proceedings of the 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, Institute of Electrical and Electronics Engineers, 26-31. https://doi.org/10.1109/ICDARW.2019.00010

Song, J., Kim, J., & Lee, J.K. (2018). NLP and deep learning-based analysis of building regulations to support automated rule checking system. *Proceedings of the International Symposium on Automation and Robotics in Construction,* The International Association for Automation and Robotics in Construction, 1-7. https://doi.org/10.22260/ISARC2018/0080

Sqlite Consortium. (2000, August 17). Sqlite home page. Sqlite Consortium, https://www.sqlite.org/index.htm

Staudemeyer, R. C., & Morris, E. R. (2019). Understanding LSTM-a tutorial into long short-term memory Recurrent Neural Networks. https://doi.org/10.48550/arXiv.1909.09586

Su, Z., Ahn, B., Eom, K., Kang, M., Kim, J., & Kim, M. (2008). Plagiarism detection using the levenshtein distance and smith-waterman algorithm. *Proceedings of the 2008 3rd International Conference on Innovative Computing Information and Control*, American Society of Civil Engineers, 569-569. https://doi.org/10.1109/ICICIC.2008.422

Tai, W., Kung, H., Dong, X. L., Comiter, M., & Kuo, C.F. (2020). ExBERT: Extending pre-trained models with domain-specific vocabulary under constrained training resources. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, American Society of Civil Engineers, 1433-1439. https://doi.org/10.18653/v1/2020.findings-emnlp.129.

Tan, X., Hammad, A., & Fazio, P. (2010). Automated code compliance checking for building envelope design. *Journal of Computing in Civil Engineering*, *24*(2), 203-211, https://doi.org/10.1061/(ASCE)0887-3801(2010)24:2(203)

Tang, D., Qin, B., & Liu, T. (2015).Document modeling with Gated Recurrent Neural Network for sentiment classification. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 1422-1432. https://doi.org/10.18653/v1/D15-1167

Tixier, A. J.P., Hallowell, M. R., Rajagopalan, B., & Bowman, D. (2016). Automated content analysis for construction safety: A Natural Language Processing system to extract precursors and outcomes from unstructured injury reports. *Automation in Construction*, *62*(February 2016) , 45-56 .https://doi.org/10.1016/j.autcon.2015.11.001

UK BIM Task Group (2019, October, 2). BIM level 2 frequently asked questions. UK BIM Task Group. https://bim-level2.org/en/faqs/

Vasileiadis, M., Kaklanis, N., Votis, K., & Tzovaras, D. (2004). Extraction of tabular data from document images. *Proceedings of the 14th Web for All Conference on The Future of Accessible Work*, Association for Computational Linguistics, 1-2. https://doi.org/10.1145/3058555.3058581

Vaswani, A., Bengio, S., Brevdo, E., Chollet, F., Gomez, A. N., Gouws, S., Jones, L., Kaiser, Ł., Kalchbrenner, N., & Parmar, N. (2018). Tensor2tensor for neural machine translation. *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas*, Association for Machine Translation in the Americas, 193–199. https://aclanthology.org/W18-1819

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2007). Attention is all you need. *Advances in Neural Information Processing Systems*, Curran Associates Inc, 5998-6008. https://doi.org/10.5555/3295222.3295349

Wang, J., Chen, Y., Hao, S., Peng, X., & Hu, L. (2019). Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, *119*(1 March 2019), 3-11. https://doi.org/10.1016/j.patrec.2018.02.010

Wang, J., Mu, L., Zhang, J., Zhou, X., & Li, J. (2020). On intelligent fire drawings review based on building information modeling and knowledge graph. *Proceedings of the Construction Research Congress 2020*, American Society of Civil Engineers, 812-820. https://doi.org/10.1061/9780784482865.086

Winkler, W. E. (1990, Juanary 1). String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. Institute of Education Sciences, https://eric.ed.gov/?id=ED325505

Wong Chong, O., & Zhang, J. (2021). Logic representation and reasoning for automated BIM analysis to support automation in offsite construction. *Automation in Construction, 129*(September 2021), 103756. https://doi.org/10.1016/j.autcon.2021.103756

Wong Chong, O., Baker, C., Afsari, K., Zhang, J. & Roach, M. (2020). Integration of BIM processes in architectural design, structural analysis, and detailing: current status and limitations. *Proceedings of the Construction Research Congress 2020*, American Society of Civil Engineers, 1203-1212. https://doi.org/10.1016/j.autcon.2021.103756

Wong Chong, O., Zhang, J., Voyles, R.M., & Min, B. (2022). A BIM-based approach to simulate construction robotics in the assembly process of wood frames to support offsite construction automation. *Automation in Construction, 137*(May 2022), 104194. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001853

Wu, J., Akanbi, T., & Zhang, J. (2022). Constructing invariant signatures for AEC objects to support BIM-based analysis automation through object classification. *Journal of Computing in Civil Engineering, 36*(4), 04022008. https://doi.org/10.1061/(ASCE)CP.1943-5487.0001012

Wu, J., Sadraddin, H.L., Ren, R., Zhang, J., & Shao, X. (2021). Invariant signatures of architecture, engineering, and construction objects to support BIM interoperability between architectural design and structural analysis. *Journal of Construction Engineering and Management, 147*(1), 04020148. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001943

Xu, X., & Cai, H. (2020). Semantic approach to compliance checking of underground utilities. *Automation in Construction*, *109*(January 2020), 103006. https://doi.org/10.1016/j.autcon.2019.103006

Xue, X., & Zhang, J. (2019). Part-of-Speech speech tagged building codes (PTBC). https://doi.org/10.4231/Y0ZQ-4946

Xue, X., & Zhang, J. (2020). Building codes Partp-of-Speech speech tagging performance improvement by error-driven transformational rules. *Journal of Computing in Civil Engineering*, *34*(5), 04020035. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000917

Xue, X., & Zhang, J. (2020). Evaluation of seven Partpart-of-Speech speech taggers in tagging building codes: Identifying the best performing tagger and common sources of errors. *Proceedings of the Construction Research Congress 2020*, American Society of Civil Engineers, 1-9. https://doi.org/10.1061/9780784482865.053

Xue, X., & Zhang, J. (2021). Erratum for “Building codes Partpart-of-Speech speech tagging performance improvement by error-driven transformational rules” by Xiaorui Xue and Jiansong Zhang. *Journal of Computing in Civil Engineering*, *35*(1), 08220002. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000950

Xue, X., & Zhang, J. (2021). Logic clause representation of building codes dataset. https://doi.org/10.4231/XPJS-0G48

Xue, X., & Zhang, J. (2021). Part-of-Speech speech tagging of building codes empowered by deep learning and transformational rules. *Advanced Engineering Informatics, 47*(January 2021), 101235. https://doi.org/10.1016/j.aei.2020.101235

Xue, X., & Zhang, J. (2022). Regulatory information transformation ruleset expansion to support automated building code compliance checking. *Automation in Construction*, *138*(June 2022), 104230. https://doi.org/10.1016/j.autcon.2022.104230

Xue, X., Hou, Y., & Zhang, J. (2022). Automated construction contract summarization using natural language processing and deep learning. *Proceedings of the 39th International Symposium on Automation and Robotics in Construction (ISARC 2022)*, I.A.A.R.C., iaarc.org., 459-466.

Xue, X., Wu, J., & Zhang, J. (2022). Semi-automated generation of logic rules for tabular information in building codes to support automated code compliance checking. *Journal of Computing in Civil Engineering, 36*(1), 04021033. https://doi.org/10.1061/(ASCE)CP.1943-5487.0001000

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based Natural Language Processing. *Proceedings of the IEEE Computational intelligenCe magazine*, *13*(3), 55-75. https://doi.org/10.1109/MCI.2018.2840738

Yu, X., Faleńska, A., & Vu, N. T. (2017). A general-purpose tagger with Convolutional Neural Networks. *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, Association for Computational Linguistics, 24–129. https://doi.org/10.18653/v1/W17-4118

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into Deep Learning. https://doi.org/10.48550/arXiv.2106.11342

Zhang, C., Zhang, X., Jiang, W., Shen, Q., & Zhang, S. (2009). Rule-based extraction of spatial relations in natural language text. *Proceedings of the 2009 International Conference on Computational Intelligence and Software Engineering*, Institute of Electrical and Electronics Engineers, 1-4. https://doi.org/10.1109/CISE.2009.5363900

Zhang, J. (2015). Automated code compliance checking in the construction domain using semantic Natural Language Processing and logic-based reasoning. [Doctoral dissertation, University of Illinois at Urbana-Champaign]. IDEALS. http://hdl.handle.net/2142/89207

Zhang, J., & El-Gohary, N. (2013). Handling sentence complexity in information extraction for automated compliance checking in construction. *Proceedings of the CIB W78 2013*, Conseil International du Bâtiment (CIB), 770-780. https://eres.scix.net/pdfs/w78-2013-paper-89.pdf

Zhang, J., & El-Gohary, N. M. (2015). Automated extraction of information from Building Information Models into a semantic logic-based representation. *Proceedings of the 2015 International Workshop on Computing in Civil Engineering*, American Society of Civil Engineers, 173-180. https://doi.org/10.1061/9780784479247.022

Zhang, J., & El-Gohary, N. M. (2015). Automated information transformation for automated regulatory compliance checking in construction. *Journal of Computing in Civil Engineering*, *29*(4), B4015001. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000427

Zhang, J., & El-Gohary, N. M. (2016). Extending Building Information Models semiautomatically using semantic Natural Language Processing techniques. *Journal of Computing in Civil Engineering*, *30*(6), C4016004. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000536

Zhang, J., & El-Gohary, N. M. (2016). Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking. *Journal of Computing in Civil Engineering*, *30*(2), 04015014. https://doi.org/04015014.10.1061/(ASCE)CP.1943-5487.0000346

Zhang, J., & El-Gohary, N. M. (2017). Integrating semantic NLP and logic reasoning into a unified system for fully-automated code checking. *Automation in Construction*, *73*(January 2017) , 45-57. https://doi.org/10.1016/j.autcon.2016.08.027

Zhang, J., & El-Gohary, N. M. (2017). Semantic-based logic representation and reasoning for automated regulatory compliance checking. *Journal of Computing in Civil Engineering*, *31*(1), 04016037. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000583

Zhang, J., & Laddipeerla, S. (2018). A feasibility study of IFC-based BIM 4D simulation using commercial systems to support construction planning in the U.S. *Proceedings of the 54th Construction International Conference Annual International Conference*, Construction International Conference, 441-448. http://ascpro0.ascweb.org/archives/cd/2018/paper/CPRT119002018.pdf

Zhang, J., Kwigizile, V., & Oh, S. (2016). Automated hazardous action category classification using natural language processing and machine learning techniques. *Proceedings of the 16th COTA*

International Conference of Transportation Professionals, 1579-1590. https://doi.org/10.1061/9780784479896.144

Zhang, R., & El-Gohary, N. M. (2020). A machine-learning approach for semantic matching of building codes and Building Information Models (BIMs) for supporting automated code checking. *Proceedings of the International Congress and Exhibition Sustainable Civil Infrastructures,* Springer. 64-73. https://doi.org/10.1007/978-3-030-34216-6_5

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, MIT Press, 649–657. https://doi.org/10.5555/2969239.2969312

Zhong, B. T., Ding, L. Y., Luo, H. B., Zhou, Y., Hu, Y. Z., & Hu, H. M. (2012). Ontology-based semantic modeling of regulation constraint for automated construction quality compliance checking. *Automation in Construction, 28*(December 2012), 58-70. https://doi.org/10.1016/j.autcon.2012.06.006

Zhou, N.F. (2014, Feburary 23). B-prolog user's manual. Afany Software. http://www.picat-lang.org/bprolog/download/manual.pdf

Zhou, P., & El-Gohary, N. (2018). Automated matching of design information in BIM to regulatory information in energy codes. *Proceedings of the Construction Research Congress 2018*, American Society of Civil Engineers, 75-85. https://doi.org/10.1061/9780784481264.008

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *Proceedings of the IEEE international conference on computer vision*, Institute of Electrical and Electronics Engineers, 19-27. https://doi.org/10.1109/ICCV.2015.11