AUTOMATED ENERGY COMPLIANCE CHECKING IN CONSTRUCTION

BY

PENG ZHOU

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Civil Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Doctoral Committee:

Associate Professor Nora El-Gohary, Chair
Professor Khaled A. El-Rayes
Associate Professor Corina Roxana Girju
Associate Professor Liang Y. Liu
Associate Professor Mani Golparvar-Fard

ABSTRACT

Automated energy compliance checking aims to automatically check the compliance of a building design – in a building information model (BIM) – with applicable energy requirements. A significant number of efforts in both industry and academia have been undertaken to automate the compliance checking process. Such efforts have achieved various levels of automation, expressivity, representativeness, accuracy, and efficiency. Despite the contributions of these efforts, there are two main gaps in existing automated compliance checking (ACC) efforts. First, existing methods are not fully-automated and/or not generalizable across different types of documents. They require different degrees of manual efforts to extract the requirements from the text into computer-processable representations, and match the concept representations of the extracted requirements to those of the BIM. Second, existing methods only focused on code checking. There is still a lack of efforts that address contract specification checking. To address these gaps, this thesis aims to develop a fully-automated ACC method for checking BIM-represented building designs for compliance with energy codes and contract specifications. The research included six primary research tasks: (1) conducting a comprehensive literature review; (2) developing a domain-specific semantic text classification method and algorithm for classifying energy regulatory documents (including energy codes) and contract specifications for supporting energy ACC in construction; (3) developing a semantic, natural language processing (NLP)-enabled, rule-based information extraction method and algorithm for automated extraction of energy requirements from energy codes; (4) developing a semantic, NLP-enabled, rule-based

information extraction method and algorithm for automated extraction of energy requirements from contract specifications; (5) developing a fully-automated semantic information alignment method and algorithm for aligning the concept representations of the BIMs to the concept representations of the requirements in the energy codes and contract specifications; and (6) implementing the aforementioned methods and algorithms in a fully-automated energy compliance checking prototype, called EnergyACC, and conducting a case study to identify the feasibility and challenges for fully-automated and generalized compliance checking across different types of regulatory documents – particularly energy codes versus contract specifications. Promising noncompliance detection performance was achieved for both energy code checking (95.7% recall and 85.9% precision) and contract specification checking (100% recall and 86.5% precision).

ACKNOWLEDGMENTS

First, I would like to express my profound gratitude to my advisor, Dr. Nora El-Gohary, for her patient guidance and unconditional support along the way. She has taught me a lot about research and about life, and beared with me throughout this journey. I am inspired by the high standards that she sets for herself and those around her, her attention to detail, her dedication to the profession, and her intense commitment to her work. I have thoroughly enjoyed working with her, and truly feel that I have learned so much. I also deeply appreciate the guidance and support from my committee members: Dr. Corina Roxana Girju, Dr. Khaled El-Rayes, Dr. Liang Y. Liu, and Dr. Mani Golparvar-Fard.

I would like to give special thanks to my parents for being incredibly supportive, and for reminding me not to work too hard and to take good care of myself. I am so grateful for all the sacrifices that they have made to give me a chance to pursue my dream and see the outside world. Special thanks also go to my "future" finance, Qian Yang, for staying by my side – I am sure she will always do, whether in sickness or in health, for poorer or richer, in sorrow or in joy.

I especially appreciate my best friends for being there for me during all times and for always helping me have the courage to fight for my dreams. Special thanks also go to my colleagues in the research group for their kind help in my research. It is my honor to work with such a talented group of people.

Finally, I gratefully acknowledge the financial support of the National Science Foundation.

TABLE OF CONTENTS

CHAPTER 1 – INTRODUCTION

1.1 Motivation, Gaps in Automated Compliance Checking, and Overview

Compliance checking aims to ensure the compliance of a project with applicable norms such as laws, regulations, codes, and contract requirements. Manual compliance checking is a time-consuming, costly, and error-prone task (Eastman et al. 2009; Tan et al. 2010). Automated compliance checking (ACC) has, therefore, attracted much research effort to reduce the time, cost, and errors of this task. Examples of the most recent efforts in the area of ACC in construction include: (1) using manually or semi-automatically-coded rules for building design checking (e.g., Dimyadi et al. 2014; Jiang and Leicht 2015; Preidel and Borrmann 2016; Beach et al. 2015; Solihin and Eastman 2016; Dimyadi et al. 2016a; İlal and Murat Günaydın 2017); and (2) using natural language processing (NLP) techniques for fully-automated building code checking (Zhang and El-Gohary 2013; 2014a; 2015b; 2017).

Despite the importance of these efforts, three main gaps in existing ACC systems and methods are identified. First, existing ACC systems and methods are not entirely automated. They require different degrees of manual effort to extract requirements from text into computer-processable representations, and match the concept representations of the extracted requirements to those of the BIM. For example, (1) Pauwels et al. (2011) manually encoded building acoustic performance requirements as SPARQL Protocol and RDF Query Language (SPARQL) queries and manually developed a mapping ruleset to match the SPARQL-represented requirements to the RDF-

represented design information, and (2) Solihin and Eastman (2016) manually modeled the general building design requirements into conceptual graphs, in which BIM terminologies were used to encode the requirements for matching to BIMs. To the best of the author's knowledge, the only effort that achieved nearly-full ACC is that by Zhang and El-Gohary (2013; 2014a; 2015a; 2017). Zhang and El-Gohary (2013; 2014a; 2015a; 2017) developed a novel approach for automatically extracting both regulatory information (in regulatory documents) and design information [in building information models (BIMs)] and representing the extracted information into a semantic, logic-based representation for automated reasoning. Their approach is, however, not entirely fully-automated because their requirement-BIM matching component is semi-automated, although requiring minimal manual effort. Fully-automated requirement-BIM matching is essential to eliminate/reduce the time-consuming and costly manual effort, and develop approaches that are more scalable across different BIM instances/concepts, different types of regulations, and changes/updates to the BIM or the regulations. Automating the process of compliance checking with building energy codes is especially important because manual requirement extraction and requirement-BIM matching would be especially labor-intensive and time-consuming for energy codes due to two main reasons: (1) energy codes change frequently: energy codes are being periodically updated to enhance the requirement stringency. For example, the international energy conservation code (IECC) undergoes a repeated three-year update cycle to enhance the specification of building energy efficiency requirements. From 2015 IECC to 2018 IECC, 104 updates in provisions have been made (including 58 additions, 2 deletions, 44 revisions) (Nevada

Governor's Office of Energy 2018); and (2) energy codes vary from location to location: energy codes vary from state to state and city to city to address the specific needs of the location such as more/less stringent energy conservation requirements for particular building elements in that location. For example, by April 2018, 48 states have developed (or are developing) their state-specific energy conservation codes based on the IECC (ICC 2018; U.S. DOE 2018a) to meet their specific needs. Even different counties/cities in the same state have their own specific energy conservation codes [e.g., Clark county in Southern Nevada adopted a localized version of 2012 IECC with 34 updates (Southern Nevada Building Officials 2013)].

Second, there is a lack of ACC systems and methods for checking the compliance of building designs with building energy codes. For example, existing ACC systems and methods focused on checking building fire safety (Dimyadi et al. 2016a; Preidel and Borrmann 2016; Malsane et al. 2015; Dimyadi et al. 2014; Fiatech 2014), building egress and accessibility (Lee et al. 2015; Fiatech 2014; Corke 2013), building sustainability (Beach et al. 2015; Kasim et al. 2013; Beach et al. 2013), and building structural integrity (Nawari 2012; Avolve Software Corporation 2011) – with only a few, limited-scope, limited-capability efforts in energy compliance checking [e.g., COMcheck and REScheck (U.S. DOE 2018b), CBECC-Com and CBECC-Res (California Energy Commission 2016)]. Automated energy compliance checking is essential: buildings consume around 41% (about 40 quadrillion British thermal units) of the total energy consumption in the United States (U.S. EIA 2015); energy compliance is critical to attain energy savings; and ACC

would help reduce the time, cost, and errors associated with energy compliance checking. Compared to automated building code checking (i.e., Zhang and El-Gohary 2017), automatically extracting requirements from energy codes is more challenging because of the text complexities of the energy codes: (1) longer provisions: provisions in energy codes are longer, which indicates that requirements are more likely to be complex and noisy; (2) more requirement exceptions: a requirement in energy codes may contain one or multiple exceptions for waiving the compliance with the requirement if one or all of a set of exception conditions are met; and (3) hierarchically-complex sentence structures: text in energy codes has more complex sentence structures, in which one provision may contain multiple levels of subprovisions, and one subprovision may contain multiple requirements. For example, a statistical comparison of two random chapters in the International Building Code (IBC) 2012 and IECC 2012 shows that the average provision length in IECC is approximately 40% longer and that the percentage of the number of provisions containing one or multiple exceptions and/or complex requirement structures in IECC is more than three times.

Third, there is a lack of ACC systems and methods for checking the compliance of building designs with contract specifications. Project contracts, including contract specifications, are a major source of law – the source of private law; a contract represents a binding agreement imposing requirements on construction projects. Checking the compliance with the contract specifications is essential for energy compliance, since in addition to the energy codes the specifications also

prescribe energy requirements. However, automated specification compliance checking remains to be a challenge because of the specifications' "project-specific" nature; project specifications could vary widely from a project to project. Compared to automated energy code checking, automatically extracting requirements from contract specifications is more challenging because of the text complexities of the specifications: (1) hierarchically-complex text structures: text in specifications is usually organized in a more complex text hierarchy; the text is organized in wider and deeper text hierarchies. For example, a section in the specifications may contain many articles, where each article contains many paragraphs, each paragraph contains multiple levels of subparagraphs, and each subparagraph contains multiple requirements; (2) incomplete sentence structures: requirements in specifications are usually not represented in complete sentences; they are instead usually represented in short phrases. Such incomplete sentence structures hinder the ability to capture the dependency between the semantic information in the text. Such dependency information helps in reducing text ambiguities and enhancing the extraction performance; and (3) variety of levels of development (LODs): requirements in contract specifications may correspond to different BIM LODs. Extraction of requirements that go beyond the required BIM LOD may result in unnecessary processing efforts and potential compliance checking errors.

To address these gaps, this research aims to develop a set of methods and algorithms for text classification, information extraction, and information alignment for fully-automated compliance checking of BIM-represented building designs with energy requirements – specifically thermal

insulation requirements and lighting power requirements – in both energy codes and contract specifications. Semantic text classification is used for classifying the text in codes and specifications to filter out irrelevant and noisy text. Semantic, NLP-based information extraction is used to extract energy requirements from codes and specifications. Semantic information alignment is used to align the concept representations of the BIMs to the concept representations of the energy requirements.

1.2 Proposed Approach

1.2.1 Points of Departure

This research builds on the previous efforts by Zhang and El-Gohary (2013; 2014a; 2015a; 2017) and Salama and El-Gohary (2013a; 2013b). Zhang and El-Gohary (2013; 2014a; 2015a; 2017), especially, developed a novel semantic, NLP-enabled, and logic-based approach for automatically extracting both regulatory information (in regulatory documents) and design information [in building information models (BIMs)] and representing the extracted information into a semantic, logic-based representation for automated reasoning. Specifically, the outcomes from previous efforts include: (1) a deontic model (semantic model based on theory of rights and obligations) for ACC in construction for supporting normative automated reasoning (Salama and El-Gohary 2013b); (2) a machine learning-based text classification algorithm for classifying clauses of general conditions of project contracts into environmental and non-environmental clauses (Salama and El-Gohary 2013a); (3) a rule-based semantic information extraction algorithm for automated

extraction of quantitative requirements from building codes (Zhang and El-Gohary 2013); (4) a rule-based semantic information transformation algorithm for automated transformation of the extracted requirements into computer-processable logic rules (Zhang and El-Gohary 2015a); (5) an EXPRESS-based information extraction algorithm and a rule-based information transformation algorithm for automated extraction of design information from BIM models and transformation to logic facts (Zhang and El-Gohary 2014a); (6) a logic-based information representation and compliance reasoning schema for representation of regulatory requirements and design information for enabling automated logic reasoning for ACC (Zhang and El-Gohary 2014b); and (7) a semantic, natural language processing (NLP)-enabled, and logic-enabled system (a proof-of-concept prototype) for automatically checking the compliance of BIM-based building designs with building codes (Zhang and El-Gohary 2017). However, as mentioned in Section 1.1, despite the novelty and importance of these efforts, this thesis addresses additional, important knowledge gaps and research challenges. First, the approach in Zhang and El-Gohary (2017) is not entirely fully-automated, because the requirement-BIM matching component is semi-automated, although requiring minimal manual effort. New methods for fully-automated requirement-BIM matching component are thus needed. Second, their implementation and testing efforts only focused on the International Building Code 2009 (ICC 2009a). Automatically extracting requirements from energy codes and contract specifications is far more challenging because of the reasons outlined in Section 1.1. Major adaptation of the previous approach – and development of new methods and algorithms – are thus needed.

1.2.2 New Directions and Contributions

This research builds on the aforementioned previous efforts in four main aspects. First, the research fully automates the alignment of the concept representations of the BIM design information to the concept representations of the energy requirements so that they can "speak" the same language. Second, this research aims to study the practicality and feasibility of the NLP-enabled and logic-based approach in the energy compliance checking domain. Third, this research extends the compliance checking of BIM-represented building designs to different compliance domains and different kinds of regulatory documents – energy codes and specifications. Fourth, this research integrates the use of text classification as an initial step to ACC, which aims to support high ACC performance by filtering out irrelevant text to improve the efficiency and the performance of text processing and information extraction.

1.2.3 Proposed Framework and Scope

The proposed ACC framework for compliance checking of BIM-represented building designs with energy codes and contract specifications is illustrated in Figure 1.1. The ACC framework includes two types of elements: data and processes. The data, as input to the ACC framework, include: (1) a BIM-represented building design: an issue-for-construction version with a minimum LOD 350 [specifically minimum LOD 350 for the BIM architectural model and minimum LOD 400 for the BIM electrical model. LOD 350 is generally sufficient for code compliance checking (Solihin and Eastman 2015a)], in Industry Foundation Classes (IFC) format (.ifc file); (2) energy codes: in .txt

format; and (3) contract specifications: an issue-for-construction version, in .txt format. The ACC framework includes five main processes: (1) Text classification: using semantic text classification algorithms to filter out irrelevant text in energy codes and specifications to improve the efficiency and performance of information extraction; (2) Automated information extraction: using semantic, NLP-enabled, rule-based algorithms to extract the requirements from the classified text into a computer-processable rule-format. A combination of domain-specific preprocessing techniques, ontology-based pattern-matching extraction techniques, sequential dependency-based extraction methods, cascaded extraction methods, incompleteness-aware sequential dependency extraction methods, and detail-aware LOD extraction methods are used to deal with the complexities and challenges of the text (outlined in Section 1.1); (3) BIM information extraction: using EXPRESS-based information extraction to extract relevant design information from BIMs to an alignment-ready representation; (4) Automated information alignment: using semantic information alignment algorithm to match the concept representations of the extracted BIM design information to the concept representations of the extracted energy requirements. The aligned design information and energy requirements are transformed to logic facts and logic rules, respectively; and (5) Automated compliance reasoning: using logic-based reasoning to check the compliance of the logic facts with the logic rules and generate a compliance checking report, showing noncompliance cases with reasons of noncompliance.

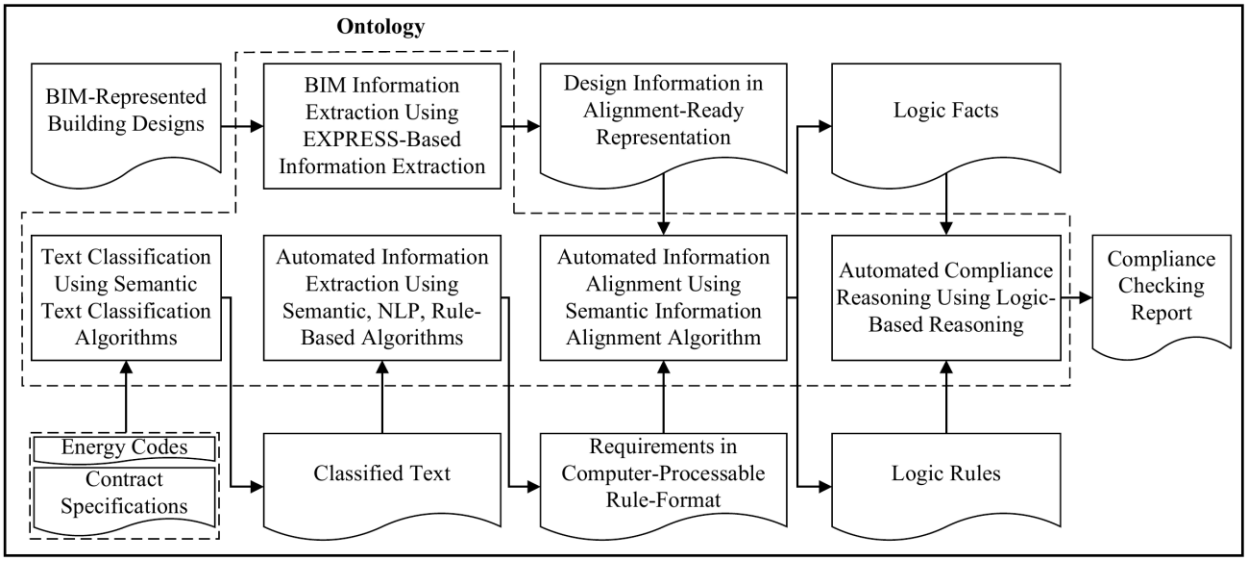


Figure 1.1. Proposed automated energy compliance checking framework

1.3 Knowledge Gaps in Text Classification, Information Extraction, and Information Alignment

1.3.1 Knowledge Gaps in Text Classification for ACC in Construction

In discussing the knowledge gaps in the area of text classification (TC), the knowledge gaps in existing machine learning (ML)-based (i.e., ML-based and non-ontology-based) TC efforts and existing ontology-based TC efforts. The knowledge gaps cover the main gaps inside and outside of the construction domain.

A variety of ML-based TC algorithms have been developed in the computer science (CS) domain. Despite of these enormous efforts in the CS domain, many challenges still exist in constructing classifiers that can be effective across different domains and, thus, TC models remain highly domain specific (Blitzer et al. 2007). There is no single best TC algorithm across all domains; the performance of one best performing ML algorithm tested on one dataset is not necessarily the best

one when tested on another dataset, especially when datasets from different domains are more dissimilar (Sebastiani 2002). It is difficult to reuse an existing classifier from one domain to another (e.g., medical versus construction), from one subdomain to another (e.g., safety versus environmental), or from one application to another (e.g., document management versus ACC), because text features vary across domains and subdomains, and performance requirements vary across applications (e.g., for ACC, unlike other applications, recall is more critical than precision) (Salama and El-Gohary 2013b). There is, thus, a need to identify the specific features of domain text and how to adapt or tune a classifier to those specific features and to the specific performance requirements of the domain or application.

A number of research efforts in the construction domain focused on ML-based TC (e.g., Caldas et al. 2002; Kovacevic et al. 2008; Mahfouz 2011; Salama and El-Gohary 2013b). However, hierarchical TC work in the construction domain is limited in three main ways. First, the performance of hierarchical TC tends to drop quickly when reaching a deeper level in the hierarchy. For example, Caldas and Soibelman (2003) addressed a three-level multilabel binary classification problem, but the accuracy dropped from 96% at the first level to 86% at the third level. Second, the algorithms can only handle single-label classification problems that were transformed from a multilabel problem using a binary classification approach. Dealing with a transformed multilabel classification problem as a binary instead of a multiclass classification problem may encounter data imbalance problems. A data imbalance problem occurs when the documents of one class are

much more than the documents of another class(es) (Sun et al. 2007). Third, the algorithms are not sufficiently adapted to the domain. It is important to utilize the features and methods that work best for each domain (Blitzer et al. 2007). More discussion of the state of the art and knowledge gaps is included in Section 3.1.1.

While generally successful, in contrast to ontology-based TC, ML-based TC usually discards semantic text information (e.g., meaning of words) although it is potentially very useful in identifying the correct label(s) of a document. In this regard, two main research gaps are identified in ontology-based TC efforts. First, there have been no research efforts for using ontology-based TC in the construction domain. This is a lost opportunity for exploring the use of domain semantics to improve the performance of TC-based applications in the construction domain. Second, outside of the construction domain, ontology-based TC efforts: (1) rely on supervised ML for training the classifier – using labelled training data – to learn the rules for labeling any given text (e.g., Vogrinčič and Bosnić 2011; Lee et al. 2009; He et al. 2004). This involves much manual effort in labeling the training data; (2) can only deal with single-label classification problems (e.g., Yang et al. 2008; Wei et al. 2006; Yu et al. 2006; Song et al. 2005; He et al. 2004) or are unable to deal with a multilabel TC problem directly (e.g., Waraporn et al. 2010). This requires transformation to multiple single-label problems; and/or (3) show inconsistent results for ontology-based TC in comparison with non-ontology-based, ML-based TC. Inconsistent results indicate that there is no single outperforming ontology-based method or algorithm, and, thus, that it is difficult to reuse an

existing ontology-based TC algorithm from one domain to the other. More discussion of the state of the art and knowledge gaps is included in Section 3.2.1.

1.3.2 Knowledge Gaps in Information Extraction for ACC in Construction

Information extraction efforts, especially ontology-based information efforts, are limited in the construction domain (e.g., Abuzir and Abuzir 2002; Al Qady and Kandil 2010; Zhang and El-Gohary 2013; Liu and El-Gohary 2017). Despite the importance of existing efforts, they are limited in six primary ways. First, existing methods extract information from unclassified text, which may result in unnecessary processing effort and may increase extraction errors due to processing irrelevant text. None of these efforts explored the use of text classification techniques to filter out irrelevant text prior to information extraction (IE) to improve the efficiency and performance of IE. Second, existing methods were not tested in deep information extraction from long provisions with multiple exceptions. For example, Abuzir and Abuzir (2002) and Al Qady and Kandil (2010) conducted shallow information extraction (extracting partial information from a sentence, whereas deep information extraction aims to extract all information expressed by a sentence based on a full analysis of the sentence); and Zhang and El-Gohary (2013) conducted deep information extraction, but tested their algorithms in extracting requirements from relatively shorter provisions with fewer exceptions, as stated in Section 1.1. Third, existing methods are limited in automatically dealing with text with hierarchically-complex sentence/text structures. For example, Al Qady and Kandil (2010) used a manual approach to break down American Institute of Architects (AIA) contract

sentences that contain enumerations and lists into separate sentences, each containing only one single component of the enumeration/list. This manual approach is time-consuming if there are a large number of sentences. Fourth, existing methods are limited in dealing with incomplete sentence structures. Information extraction from text with complete sentence structures is relatively easier, because complete sentence structures have regular grammatical patterns. Text with incomplete sentence structures, lacking such regular patterns, would thus likely to suffer from weak dependency relationships that would be insufficient to reduce ambiguities. Fifth, existing methods are not able to recognize and differentiate the LODs of the information in the contract specifications. Extraction of requirements in irrelevant LODs (i.e., information beyond the current/needed LOD) may result in potential compliance checking errors. Sixth, existing methods are not able to deal with tables inside textual documents. Many requirements in codes and specifications are represented in table format. Dealing with tables is expected to be easier than text because of its structured nature, but it needs algorithm adaptation and testing. More discussion of the state of the art and knowledge gaps is included in Sections 4.1 and 5.1.

1.3.3 Knowledge Gaps in Semantic Information Alignment for ACC in Construction

There are a significant number of regulatory compliance checking efforts in the architectural, engineering, construction, and facility management (AEC and FM) domain, in which different techniques were used to model the BIMs and regulations into the same concept representations. Examples of such techniques include the use of semantic web languages (e.g., Beach et al. 2015;

Pauwels et al. 2011), domain-specific modeling languages (e.g., Lee et al. 2015; Solihin and Eastman 2016), and predicate logic (e.g., Zhang and El-Gohary 2017; Solihin and Eastman 2016). Despite the importance of these efforts, their information alignment approaches are limited in one or more of the following three ways. First, all of these approaches require some degree of manual effort. For example, Dimyadi et al. 2016b; Lee et al. 2015; Nawari 2012; Lee et al. 2016; and Preidel and Borrmann 2016 require manual specification of the alignment, by domain experts, using predefined functions/languages. Manual approaches are typically time-consuming, costly, and unscalable (Beach et al. 2015; Eastman et al. 2009). Second, many of these efforts are somewhat rigid. For example, Beach et al. 2015; Pauwels et al. 2011; Tan et al. 2010; Delis and Delis 1995; Goel and Fenves 1969; Ding et al. 2006; See 2008; Liebich et al. 2002; and SMC 2009 use pre-defined mappings or mapping rules. Rigid approaches lack sufficient flexibility and adaptability to allow for successful implementation across BIM instances, different types of regulations, and changes/updates to the BIM or the regulations (Garrett et al. 2014; Dimyadi et al. 2016b). Third, several of these efforts – especially those by software vendors such as Ding et al. 2006; See 2008; Liebich et al. 2002; and SMC 2009 – use proprietary methods. Proprietary methods lack the needed transparency to enable the users to check the correctness of the alignment (Dimyadi et al. 2016b). More discussion of the state of the art and knowledge gaps is included in Section 6.1.

1.4 Problem Statement

Manual compliance checking is time-consuming, costly, and error-prone. Automated compliance checking (ACC) aims to address this practical gap by reducing the time, cost, and error of compliance checking. However, current ACC systems and methods are limited in three main ways. First, existing ACC systems and methods are not entirely automated; they require different degrees of manual effort to extract requirements from text into computer-processable representations, and match the concept representations of the extracted requirements to those of the BIM. Second, there is a lack of ACC systems and methods for checking the compliance of building designs with building energy codes. Third, there is a lack of ACC systems and methods for checking the compliance of building designs with contract specifications. Automatically extracting requirements from energy codes and contract specifications is more challenging than the extraction from building codes, because of the nature of the text in terms of longer provisions, more requirement exceptions in one provision, hierarchically-complex sentence/text structures, incomplete sentence structures, and variety of levels of development.

1.5 Research Objectives and Questions

The overall objective of this research is to develop a set of methods and algorithms for text classification, information extraction, and information alignment for supporting automated compliance checking of BIM-represented building designs with energy requirements (specifically thermal insulation requirements and lighting power requirements) in both energy codes and

contract specifications.

The scope of BIM is limited to an issue-for-construction version with a minimum LOD 350 (specifically minimum LOD 350 for the BIM architectural model and minimum LOD 400 for the BIM electrical model), in Industry Foundation Classes (IFC) format (.ifc file). The scope of documents is limited to energy codes and contract specifications. The scope of requirement formats is limited to requirements expressed in a text format (.txt format) and table format (.htm format), and excludes those expressed in equations, drawings, images, and references to other regulations/documents/sections. The scope of energy requirement types is limited to thermal insulation requirements and lighting power requirements. The scope of testing is limited to three energy codes [i.e., the 2012 International Energy Conservation Code (IECC), the 2013 Building Energy Efficiency Standards (known as the California Energy Code), and the Ontario Building Code Supplementary Standard SB-10] and contract specifications in MasterFormat.

Accordingly, five specific objectives are defined:

Objective 1: Develop construction-domain-specific semantic TC algorithms for classifying the text in energy codes and contract specifications to filter out irrelevant text for supporting EnergyACC in construction.

Research Questions: What are the domain-specific features of the text in energy codes and contract specifications? What techniques should be used to develop domain-specific TC algorithms using domain-specific features? How to deal with the hierarchical TC problem?

How to deal with the multilabel TC problem? How to further improve the TC performance for supporting high performance ACC? Would the use of semantics be effective in improving the performance of TC? How to best capture and utilize the semantics? Would a ML-based TC approach perform better or an ontology-based TC approach? Can a high-performing ontology-based algorithm, without supervised ML, be developed to reduce the manual effort in training?

Objective 2: Develop a semantic, NLP-enabled, rule-based information extraction algorithm for automated extraction of energy requirements from energy codes for supporting EnergyACC in construction.

Research Questions: How to deal with the long (and thus complex) provisions? How to deal with the exceptions (and the different ways of expressing exceptions)? How to deal with the hierarchically-complex sentence structures? How to deal with tables and extract requirements from tables? How to minimize the errors and achieve sufficient performance?

Objective 3: Develop a semantic, NLP-enabled, rule-based information extraction algorithm for automated extraction of energy requirements from contract specifications for supporting EnergyACC in construction.

Research Questions: How to deal with the variety of levels of development in requirements? How to deal with the hierarchically-complex text structures? How to deal with the incomplete sentence structures? How to minimize the errors and achieve sufficient performance?

Objective 4: Develop a semantic information alignment algorithm for automated alignment of the concept representations of the extracted BIM information (.ifc format, minimum LOD 350) to the concept representations of the extracted requirements for supporting EnergyACC in construction.

Research Questions: How to match the concept representations of the BIM to those of the requirements, so that they "speak the same language"? How to automatically interpret the meaning of concepts and recognize the candidate matches? How to capture the semantics behind the words and measure their semantic similarities? How to automatically identify and group the set of BIM instances that are linked to one regulatory requirement?

Objective 5: Implement the developed methods and algorithms in an EnergyACC prototype, and conduct a case study using the prototype to identify the feasibility and challenges for fully-automated and generalized compliance checking across different types of documents – particularly energy codes versus contract specifications in construction.

Research Questions: What are the performances of automated energy code checking and automated contract specification checking? What are the errors in both cases, and how do they compare? How do the errors propagate through the different prototype modules, in both cases?

1.6 Research Methodology and Tasks

The methodology is composed of six main tasks, as illustrated in Figure 1.2.

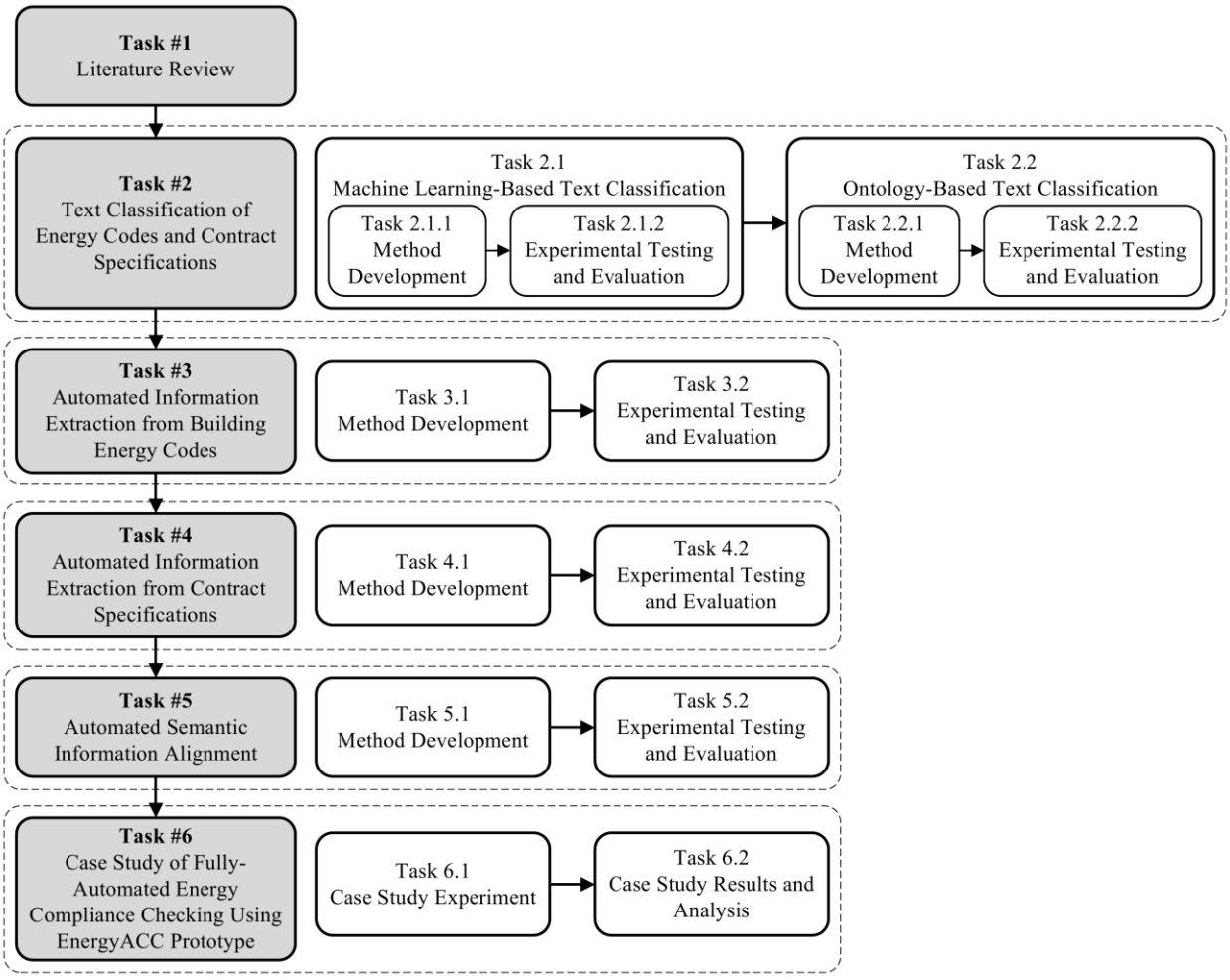


Figure 1.2. Research methodology and tasks

1.6.1 Task 1: Literature Review

The literature review was conducted in six primary areas related to this research: ACC in the construction domain, text classification, information extraction from text, contract specifications in the MasterFormat, industry foundation classes and buildingSMART Data Dictionary, BIM information extraction, and BIM-requirement alignment. The following points provide a summary of the literature review in each area:

- For automated compliance checking (ACC) in the construction domain, the literature review

focused on: (1) existing ACC efforts, in both academia and industry, in the construction domain, and (2) the state of the art in ACC and the practical gaps.

- For text classification (TC), the literature review focused on: (1) the definitions and categorization of the different types of TC problems, (2) the previous TC efforts in the general computing domain and in the construction domain, (3) the ML-based approach to deal with multilabel TC problems and its related techniques, and (4) the ontology-based approach to deal with multilabel TC problems and its related techniques.

- For information extraction (IE) from text, the literature review focused on: (1) the previous IE efforts in the general computing domain and in the construction domain, including named entity recognition, relation detection, event extraction, and full information extraction, and (2) the approaches of IE, including ontology-based approaches.

- For contract specifications in the MasterFormat, the literature review focused on the description of format (SectionFormat and PageFormat) and the analysis of the specification language in terms of sentence structure and writing style.

- For industry foundation classes (IFC) and buildingSmart Data dictionary (bSDD), the literature review focused on the definitions of IFC and bSDD, the mapping between bSDD and IFC, and the types of bSDD concepts and relationships.

- For BIM information extraction, the literature review focused on the approaches of BIM

information extraction and the previous efforts for supporting different applications.

- For BIM-requirement alignment, the literature review focused on the approaches of BIM-requirement alignment and the previous efforts using different approaches.

1.6.2 Task 2: Text Classification of Energy Codes and Contract Specifications

This task aimed to develop domain-specific semantic text classification (TC) methods and algorithms for classifying energy regulatory documents (including energy codes) and contract specifications for supporting EnergyACC in construction.

1.6.2.1 Task 2.1: Machine Learning-Based Text Classification

This task aimed to use a ML-based approach to develop a domain-specific hierarchical multilabel TC method and algorithm for classifying energy regulatory documents (including energy codes).

1.6.2.1.1 Task 2.1.1: Method Development

This task aimed to develop a domain-specific, ML-based hierarchical TC method and algorithm for classifying clauses in energy regulatory documents (including energy codes) into a number of hierarchically detailed topics. The method classifies clauses according to leaf topics at the fifth level of a semantic TC topic hierarchy. A flat approach was used to deal with the hierarchical TC problem. The multilabel classification problem was transformed into a multiclass classification problem. For preparing the training and testing data, approximately 1,200 clauses were collected from 10 energy regulatory documents, such as the 2012 IECC, and were classified into 10 leaf

subtopics of the energy efficiency topic (a subclass of environmental topic in the semantic topic hierarchy). In developing the TC algorithm, the following techniques were tested and evaluated in terms of average recall and precision and their standard deviation: (1) 10 popular ML algorithms; (2) two text representation methods [bag of words (BOW) model and bigram model]; and (3) three term weighting schemes – two supervised term weighting schemes ($TFRF_M$ and $TF_{max}RF_M$) that were modified to adapt them to multiclass classification and one unsupervised term weighting scheme (TFIDF) that is commonly used. For further performance enhancement, two performance improvement strategies were implemented: (1) feature selection: a number of methods were tested and, accordingly, K-best feature selection method and CHI feature scoring function were selected, and (2) domain-specific stopword removal: construction-domain-specific stopword lists were created and used to facilitate domain adaptation. The Scikit-learn ML tool (Pedregosa et al. 2011) in Python programming language was used to implement the selected ML algorithms.

1.6.2.1.2 Task 2.1.2: Experimental Testing and Evaluation

This task aimed to test and evaluate the developed ML-based hierarchical TC method and algorithm. The performance was evaluated using recall and precision, as per Equations 1.1 and 1.2, where true positive (TP) refers to the number of clauses labelled correctly as positive, false positive (FP) refers to the number of clauses labelled incorrectly as positive, and false negative (FN) refers to the number of clauses labelled incorrectly as negative. For this application, recall is more important than precision, because missing to recall one clause means overlooking a relevant clause,

which may affect the performance of the ACC system as a whole. Precision is not as critical, since irrelevant text could be filtered out during further IE. These measures were calculated based on a comparison of the experimental results with a manually-developed gold standard.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{1.1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{1.2}$$

1.6.2.2 Task 2.2: Ontology-Based Text Classification

This task aimed to enhance the TC performance by using an ontology-based approach to develop a domain-specific multilabel TC method and algorithm for classifying energy regulatory documents (including energy codes) and contract specifications.

1.6.2.2.1 Task 2.2.1: Method Development

This task aimed to develop an ontology-based, multilabel TC method and algorithm for classifying energy regulatory documents (including energy codes) and contract specifications for supporting ACC in construction. A domain ontology was developed for representing the hierarchy of environmental topics and the concepts and relationships associated with each topic. An unsupervised deep learning technique was used to learn the similarities between each clause (based on the terms in the clause) and each topic (based on the ontological concepts related to this topic) for classifying each clause into zero or more topics according to two experimentally set similarity thresholds. Specifically, a variant of three-layer feedforward neural network algorithm, the hierarchical softmax skip-gram, was used to learn the distributed representation of terms and

concepts in terms of real-valued vectors, and the similarities of such terms and concepts could be computed based on the cosine distance of their vectors. This hierarchical softmax skip-gram algorithm was selected because of its computational efficiency and accuracy on large datasets. The Generate Similar (Gensim) tool in Python programming language was used to implement the deep learning technique. In comparison to the previously-used non-ontology-based ML-based approach, in the ontology-based approach, (1) a document (or clause) is represented in terms of semantic concepts and relations, rather than just terms (words); (2) the multilabel classification problem is addressed in a direct way, instead of transforming the multilabel classification problem to multiple single-label classification problems (as commonly-used in ML-based TC); and (3) no human supervision is involved (i.e., training data are provided without labeling).

1.6.2.2.2 Task 2.2.2: Experimental Testing and Evaluation

This task aimed to test and evaluate the developed ontology-based multilabel TC method and algorithm. Since the proposed ontology-based TC algorithm can deal with multilabel classification problems directly, multilabel classification evaluation metrics were used. Four types of evaluation metrics were utilized: example-based metrics, macro metrics, micro metrics, and weighted metrics. Although the metrics are different, they all use redefined recall and precision measures to evaluate the overall performance. The four metrics are explained in Section 3.2.2.4. These measures were calculated based on a comparison of the experimental results with a manually-developed gold standard.

1.6.3 Task 3: Automated Information Extraction from Building Energy Codes

This task aimed to develop a semantic, NLP-enabled, rule-based information extraction (IE) method and algorithm for automated extraction of energy requirements from energy codes for supporting EnergyACC in construction.

1.6.3.1 Task 3.1: Method Development

This task aimed to develop a semantic, NLP-enabled, rule-based IE method and algorithm for automatically extracting thermal insulation requirements and lighting power requirements from energy codes. Domain-specific preprocessing techniques, ontology-based pattern-matching extraction techniques, sequential dependency-based extraction methods, and cascaded extraction methods were used to extract requirements from the provisions. A Hypertext Markup Language (HTML)-based table processing and extraction method was used to extract requirements from tables. The information extraction algorithm captured and used dependency information to reduce the semantic ambiguities of the text for enhancing the performance of extraction. A conceptual dependency structure was built to identify target semantic information elements (SIEs) (e.g., subject of compliance checking such as the building element) and the dependency information among the target SIEs. The extraction sequence was thus defined based on the dependency relations of SIEs. Both syntactic features [e.g., part of speech (POS) tags] and semantics features (i.e., concepts from an ontology) were used in the extraction rules to define the patterns of the text. The dependency information was used to assist in constructing the patterns in the extraction rules.

Cascaded extraction methods were used to deal with the complex text in energy codes (long provisions, hierarchically-complex provisions, and provisions with exceptions), by breaking down a complex extraction task into a number of simple extraction tasks (i.e., a complex extraction task is cascaded on a number of simple extraction tasks). The IE algorithm was implemented in the "a nearly-new information extraction" (ANNIE) system of the General Architecture for Text Engineering (GATE). The ontology was built using the ontology editor of GATE. The extraction rules were coded as Java Annotation Patterns Engine (JAPE) rules. Java programming language was used to implement the domain-specific preprocessing techniques.

1.6.3.2 Task 3.2: Experimental Testing and Evaluation

This task aimed to test and evaluate the developed IE method and algorithm. The performance was measured in terms of recall and precision. Recall is the number of correctly extracted information element instances divided by the total number of information element instances that should be extracted. Precision is the number of correctly extracted information element instances divided by the total number of extracted information element instances. These measures were calculated based on a comparison of the experimental results with a manually-developed gold standard, for information extracted from a chapter in the 2012 International Energy Conservation Code.

1.6.4 Task 4: Automated Information Extraction from Contract Specifications

This task aimed to develop a semantic, NLP-enabled, rule-based information extraction (IE) method and algorithm for automated extraction of energy requirements from contract

specifications for supporting EnergyACC in construction.

1.6.4.1 Task 4.1: Method Development

This task aimed to develop a semantic, NLP-enabled, rule-based IE method and algorithm for automatically extracting thermal insulation requirements and lighting power requirements from contract specifications. The algorithm developed in Task 3.1 (Section 1.6.3.1) was adapted to address the different nature of the text, including hierarchically-complex text structures, incomplete sentence structures, and variety of levels of development (LODs). To deal with such challenging text complexities, a domain-specific text splitting and stitching method was used to automatically simplify the hierarchically-complex text structures using a regular expressions-based pattern matching technique. An incompleteness-aware sequential dependency extraction method was used to capture dependency information from incomplete sentence structures to reduce the text ambiguities. A detail-aware LOD extraction method was used to automatically differentiate the LODs of sentences based on analyzing their grammatical moods using syntactic text features. Similar to Task 3.1 (Section 1.6.3.1), the IE method and algorithm was implemented in the "a nearly-new information extraction" (ANNIE) system of the General Architecture for Text Engineering (GATE).

1.6.4.2 Task 4.2: Experimental Testing and Evaluation

This task aimed to test and evaluate the developed IE method and algorithm. Similar to Task 3.2 (Section 1.6.3.2), the performance was measured in terms of recall and precision. These measures

were calculated based on a comparison of experimental results with a manually-developed gold standard, for information extracted from the contract specifications of an educational building project in Illinois, the Electrical and Computer Engineering (ECE) Building at the University of Illinois at Urbana-Champaign. The same project was used for the evaluation in Tasks 5 and 6.

1.6.5 Task 5: Automated Semantic Information Alignment

This task aimed to develop a fully-automated semantic information alignment method and algorithm for aligning the concept representations used in building information models (BIMs) to the concept representations used in the regulations (energy codes and contract specifications) for supporting EnergyACC in construction.

1.6.5.1 Task 5.1: Method Development

This task aimed to develop a fully-automated semantic information alignment method and algorithm for aligning the concept representations of the BIMs to the concept representations of the requirements in the energy codes and contract specifications. Two submethods were developed and used for information alignment. First, a first-level simple alignment method was used to align single design information instances to single regulatory concepts. Domain knowledge was used to interpret the meaning of concepts to recognize potential matching design information instances. An empirical method was used to analyze the patterns of semantic similarity to select the matching instances, in which a deep learning technique was used to measure the semantic similarity. Second, a final complex alignment method was used to recognize the groups of instances that belong to a

regulatory requirement. Supervised searching and unsupervised searching were used to identify the instance pairs, and network modeling was used to group and link the identified instances pairs to the associated regulatory concepts in the regulatory requirement. Java programming language was used to implement the semantic information alignment method and algorithm.

1.6.5.2 Task 5.2: Experimental Testing and Evaluation

This task aimed to test and evaluate the developed semantic information alignment method and algorithm. The performance was evaluated in terms of recall and precision. Recall refers to the total number of correctly aligned instances divided by the total number of correct instances in the gold standard. Precision refers to the total number of correctly aligned instances divided by the total number of aligned instances. These measures were calculated based on a comparison of the experimental results with a manually-developed gold standard, for a number of matching design information instances in a BIM to a number of energy requirements from energy codes. Both this BIM and the contract specifications (Task 4.2) belong to the same project (i.e., the Electrical and Computer Engineering Building at the University of Illinois at Urbana-Champaign).

1.6.6 Task 6: Case Study of Fully-Automated Energy Compliance Checking Using EnergyACC Prototype

This task aimed to conduct a case study to identify the feasibility and challenges for fully-automated and generalized compliance checking across different types of documents – particularly energy codes versus contract specifications in construction.

1.6.6.1 Task 6.1: Case Study Experiment

This task aimed to conduct a case study experiment for checking a BIM for compliance with building energy efficiency requirements from energy codes and contract specifications. An energy compliance checking prototype, called EnergyACC, was used to conduct the case study. The developed methods and algorithms (i.e., text classification, information extraction, and information alignment) in Tasks 2-5 were implemented in the prototype. The EnergyACC prototype was implemented in Java programming language using the Eclipse OXYGEN (Eclipse Foundation 2017). Two test cases were prepared: one for the energy code checking and one for the contract specification checking. The test cases were prepared based on: (1) a BIM of an educational building project in Illinoi: an issue-for-construction version with a minimum LOD 350 (specifically minimum LOD 350 for the BIM architectural model and minimum LOD 400 for the BIM electrical model), in Industry Foundation Classes (IFC) format (.ifc file); (2) three energy codes [i.e., the 2012 International Energy Conservation Code (IECC), the 2013 Building Energy Efficiency Standards (known as the California Energy Code), and the Ontario Building Code Supplementary Standard SB-10]: in .txt format; and (3) contract specifications: an issue-for-construction version, in .txt format. The scope of the case study was limited to thermal insulation and lighting power requirements (i.e., two subtopics of energy requirements). The compliance checking performance was measured in terms of recall and precision of noncompliance detection. Recall refers to the number of correctly detected noncompliance instances divided by the total number of noncompliance instances that should be detected. Precision refers to the number of

correctly detected noncompliance instances divided by the total number of detected noncompliance instances. These measures were calculated based on a comparison of the noncompliance detection results with two manually-developed gold standards (for energy code checking and contract specification checking).

1.6.6.2 Task 6.2: Case Study Results and Analysis

This task aimed to analyze the experimental results to identify the feasibility and challenges for fully-automated and generalized compliance checking by answering three primary research questions. What are the performances of automated energy code checking and automated contract specification checking? What are the errors in both cases, and how do they compare? How do the errors propagate through the different prototype modules, in both cases? The first question aims to assess whether acceptable performance could be achieved across different types of documents (i.e., energy codes versus contract specifications) – and how would the performance compare to the state of the art – to assess the feasibility of generalized automated approaches. The second question aims to study the errors to identify the challenges to automation and generalizability. The third question aims to study the error propagation features to identify the most critical errors to avoid.

1.7 Intellectual Merit and Broader Impacts

1.7.1 Intellectual Merit

This thesis research contributes to the body of knowledge in six primary ways. First, this research offers a domain-specific, machine learning (ML)-based hierarchical text classification (TC)

method for classifying clauses in energy regulatory documents. It is key in enabling automated energy compliance checking in the construction domain by enhancing the efficiency of automated IE. It addresses a more challenging TC problem – hierarchical TC as opposed to nonhierarchical TC. Hierarchical TC allows for a more granular classification of text according to detailed subtopics and thus would result in further enhancement of automated IE efficiency. Second, this research offers an ontology-based multi-label TC method for classifying text in energy regulatory documents and contract specifications. It offers a leading initiative; it is the first ontology-based TC effort in the construction domain. It uses an unsupervised deep learning algorithm for capturing the semantics behind the words and addresses the multilabel classification problem in a direct way without transformation to multiple single-label ones. Third, this research offers a semantic, NLP-enabled, rule-based information extraction (IE) method for automated extraction of energy requirements from energy codes. It uses a combination of domain-specific preprocessing techniques, sequential dependency-based extraction method, and cascaded extraction method to deal with the challenging text complexities in energy codes (i.e., longer provisions, requirement exceptions, and hierarchically-complex sentence structures). Fourth, this research offers a semantic, NLP-enabled, rule-based IE method for automated extraction of energy requirements from contract specifications. It uses a domain-specific text splitting and stitching method, an incompleteness-aware sequential dependency extraction method, and a detail-aware level of development (LOD) extraction method to deal with the challenging text complexities in contract specifications (i.e., hierarchically-complex text structures, incomplete sentence structures, and

variety of LODs). Fifth, this research offers a fully-automated semantic information alignment method for aligning BIM information to regulatory information. It captures domain knowledge to automatically interpret the meaning of concepts and recognize the candidate design information instances that are potentially matched to the regulatory concepts, and uses deep learning to capture the semantics behind the words and accordingly measure semantic similarity and select the matches. It uses supervised and unsupervised searching algorithms to automatically identify the relationships that create instance pairs, and uses network modeling to model and group the instance pairs that are linked to the associated concepts in a regulatory requirement. Sixth, this research offers new knowledge on the feasibility and challenges for fully-automated and generalized compliance checking across different types of documents – energy codes and contract specifications – including sources of errors and error propagation patterns. It provides important insights on the generalizability of fully-automated energy compliance checking methods, and sheds important light on the sources of errors in automated compliance checking and how these errors propagate – or not propagate – from an intermediate step to the other. Such insights are very important – they are pointers to limitations, future research directions, and paths for improvement.

More detailed discussions of the intellectual merit of each of the aforementioned methods and contribution to the body of knowledge are provided in Chapter 8.

1.7.2 Broader Impacts

The results of this research could bring the following significant benefits to the society at large:

- Reducing the time and cost of energy compliance checking in construction: It is estimated that ACC could reduce the plan review time from 60 days to 60 seconds (Fiatech 2013; Fiatech 2014; Fiatech 2015) Automated energy code checking is expected to enhance the efficiency of discovering, analyzing, and checking compliance of applicable energy codes, and consequently speed the energy code compliance verification process. Automated contract specification checking is expected to help ensure compliance with specification provisions, and promote the use of BIM for improved project delivery.

- Promoting energy compliance and energy savings: It is estimated that the compliance with building energy conservation codes and standards could potentially save $330 billion for the U.S. consumers by 2040 (Livingston et al. 2014). In addition, checking compliance with contract specifications will further encourage project participants to go beyond regulatory requirements (i.e., energy codes), and promote the adoption of voluntary and self-directed energy initiatives, thereby supporting energy efficient and sustainable construction.

- Providing insights on the generalizability of fully-automated energy compliance checking methods – across energy codes and contract specifications: Such insights are very important to the research community – they are pointers to limitations, future research directions, and paths for improvement.

1.8 Publications

The thesis contains material published in the following conference and journal papers:

- Zhou, P., and El-Gohary, N. (2014). "Semantic-based text classification of environmental regulatory documents for supporting automated environmental compliance checking in construction." *Proc., 2014 Construction Research Congress (CRC)*, ASCE, Reston, VA, 897–906.
- Zhou, P., and El-Gohary, N. (2014). "Ontology-based, multi-label text classification for enhanced information retrieval for supporting automated environmental compliance checking." *Proc., 2014 Int. Conf. on Computing in Civil and Building Engineering (ICCCBE)*, ASCE, Reston, VA, 2238–2245.
- Zhou, P., and El-Gohary, N. (2015). "Domain-specific hierarchical text classification for supporting automated environmental compliance checking." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000513, 04015057.
- Zhou, P., and El-Gohary, N. (2015). "Ontology-based information extraction from environmental regulations for supporting environmental compliance checking." *Proc., 2015 Int. Workshop on Computing in Civil Engineering (IWCCE)*, ASCE, Reston, VA, 190–198.
- Zhou, P., and El-Gohary, N. (2016). "Ontology-based multilabel text classification of construction regulatory documents." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000530, 04015058.
- Zhou, P., and El-Gohary, N. (2016). "Automated extraction of environmental requirements from contract specifications." *Proc., 16th Int. Conf. on Computing in Civil and Building Engineering (ICCCBE*), International Society for Computing in Civil and Building Engineering (ISCCBE), Osaka, Japan, 1669–1676.
- Zhou, P., and El-Gohary, N. (2017). "Ontology-based automated information extraction from building energy conservation codes." *Automat. Constr.*, 74, 103–117.
- Zhou, P., and El-Gohary, N. (2018). "Automated matching of design information in BIM to

regulatory information in energy codes." *Proc., 2018 Construction Research Congress (CRC)*, ASCE, Reston, VA, 75–85.

- Zhou, P., and El-Gohary, N. (2018). "Text and information analytics for fully automated energy code checking." *Proc., 2nd GeoMEast Int. Congress and Exhibition on Sustainable Civil Infrastructures*, Springer International Publishing, Cham, Switzerland, 196–208.

CHAPTER 2 – LITERATURE REVIEW

2.1 Automated Compliance Checking in the Construction Domain

2.1.1 Existing Automated Compliance Checking Systems and Methods in Construction

Manual compliance checking is time-consuming and costly (Eastman et al. 2009). Automated compliance checking (ACC) aims to address this practical gap by reducing the time and cost of checking the compliance of construction projects to regulatory requirements. There exists a considerable body of literature on ACC in construction, since the first ACC initiative in the 1960s, when Goel and Fenves modeled building structural design requirements into decision tables for computer-enabled compliance checking (Goel and Fenves 1969). Since then, various ACC methods have been developed for different applications. Examples of ACC efforts include checking of building designs (Eastman et al. 2009; Zhang et al. 2013; Solihin and Eastman 2016; İlal and Günaydın 2017), building accessibility and/or visibility (Yurchyshyna and Zarli 2009; Hjelseth and Nisbet 2011; Lee et al. 2015), building envelope performance (Tan et al. 2010), building acoustic performance (Pauwels et al. 2011), building safety design (Qi et al. 2011), building structural design (Nawari 2012), construction quality (Zhong et al. 2012), building safety design and planning (Melzner et al. 2013), building water network design (Martins and Monteiro 2013), building sustainability (Kasim et al. 2013; Beach et al. 2013; Beach et al. 2015), building energy design (Cheng and Das 2014), building evacuation (Choi et al. 2014), building fire safety (Dimyadi et al. 2014; Malsane et al. 2015; Dimyadi et al. 2016a; Preidel and Borrmann 2016),

deep foundation design (Luo and Gong 2015), and formwork constructability (Jiang and Leicht 2015). Larger ongoing research efforts that are led by industry organizations include the Solibri Model Checker (SMC) developed by Solibri to perform tasks like clash detection and building accessibility checking (Corke 2013); the Autocodes project that is led by Fiatech, which aims to automate the regulatory compliance review process with a focus on checking building accessibility and egress, fire and life safety, and mechanical and engineering (Fiatech 2014); the COMcheck/REScheck, which focuses on checking building energy conservation (U.S. DOE 2018b); the AVOLVE Electronic Plan Review, which focuses on checking building structural integrity (Avolve Software Corporation 2011); and the CORENET e-PlanCheck, which focuses on checking building design (Khemlani 2005).

All existing ACC systems and methods adopted rule-based checking mechanisms. According to Eastman et al. (2009), rule-based checking applies rules (i.e., a kind of constraints or conditions) on the checking target (e.g., a building design), and automatically evaluates whether the checking target complies with the applied rules. Generally, a rule-based ACC system and method is composed of four phases (Eastman et al. 2009): (1) a rule interpretation phase: human developers (e.g., a domain expert) are required to manually translate the rules written in human language (e.g., text, tables, equations) into a computer-understandable rule-format; (2) a building model preparation phase: building design information is captured in a computer-based representation (e.g., BIM model); (3) a rule execution phrase: the computer-understandable rules are applied on the

computer-represented building design to conduct checking; and (4) a checking results reporting phase: a checking report is generated showing both compliance cases and noncompliance cases. Additional information for the noncompliance cases may include the violation reasons and referenced source rules (i.e., rules in human language). Accordingly, this rule-based checking mechanism brings inherited common features to the existing ACC systems and methods: (1) manual rule interpretation: it is time-consuming to manually encode the rules from a large number of documents of different types (e.g., regulatory documents vs. contract documents), or from different domains (e.g., environment vs. safety); (2) proprietary checking rules: the rules are usually proprietary, which allows limited ability to modify/adapt the rules; and (3) frequent updates: it is time-consuming to update the rules in response to the changes/updates in the codes.

2.1.2 State of the Art in Automated Compliance Checking and Practical Gaps

Despite the importance of these ACC efforts, obstacles to reaching fully-automated ACC still remain in two primary areas: requirement extraction and requirement-BIM matching. Requirement extraction aims to automatically extract requirements from text into computer-processable representations. Requirement-BIM matching aims to automatically match the concept representations of the extracted requirements to those of the BIM. Existing ACC methods can be categorized into three groups, according to their levels of automation achieved – at requirement extraction and requirement-BIM matching. First, the majority of these methods have achieved a minimal level of automation in both requirement extraction and requirement-BIM matching. For

example, the methods in Khemlani (2005), Ding et al. (2006), See (2008), SMC (2009), Yurchyshyna and Zarli (2009), Tan et al. (2010), Nguyen and Kim (2011), Pauwels et al. (2011), Nawari (2012), Zhong et al. (2012), Melzner et al. (2013), Zhang et al. (2013), Cheng and Das (2014), Choi et al. (2014), Lee et al. (2015), Luo and Gong (2015), Lee et al. (2016), Dimyadi et al. (2016a, b), Solihin and Eastman (2016), and Preidel and Borrmann (2016), Mark et al. (2017), and Zhou et al. (2018) require different degrees of manual effort to model the requirements into various computer-interpretable representations, including Jena rules, semantic web rule language (SWRL) rules, conceptual graphs, SPARQL queries, language-integrated query (LINQ), regulatory knowledge query language (RKQL), visual code checking language (VCCL), building environment rule and analysis (BERA) language, eXtensible Markup Language (XML)-based decision tables, software API functions, custom-developed computer programs, and other proprietary representations. For requirement-BIM matching, they used different manual approaches: (1) using BIM terminologies to write requirements (e.g., Yurchyshyna and Zarli 2009; Nguyen and Kim 2011; Zhong et al. 2012; Melzner et al. 2013; Zhang et al. 2013; Cheng and Das 2014; Solihin and Eastman 2016); (2) developing mapping rulesets/algorithms/functions, or black box mapping files to translate the regulatory language to the BIM language (e.g., Goel and Fenves 1969; Delis and Delis 1995; Khemlani 2005; Ding et al. 2006; See 2008; SMC 2009; Tan et al. 2010; Pauwels et al. 2011; Mark et al. 2017); (3) developing the BIMs using the regulatory language (e.g., Choi et al. 2014; Luo and Gong 2015); and (4) relying on ACC users to conduct manual translation (e.g., Nawari 2012; Lee et al. 2015; Lee et al. 2016; Dimyadi et al. 2016a, b;

Preidel and Borrmann 2016; Zhou et al. 2018). Such ACC methods are time-consuming, costly, and hard to scale up.

Second, a few methods have reached a moderate level of automation. For example, Hjelseth and Nisbet (2011), Beach et al. (2015), and İlal and Günaydın (2017) used the RASE methodology to semi-automatically extract requirements from annotated text using four tags – "Requirement, Applicability, Selection, Exception (RASE)". However, manual effort is still required to conduct the tagging/annotation. In addition, substantial manual efforts are required to develop a separate mapping scheme [e.g., mapping ontology (Beach et al. 2015)] to map the requirements to the BIM representations. Such mapping schemes are rigid, thereby hard to generalize to different types of regulations/documents.

Third, a very limited number of methods have reached a full or nearly full level of automation in both requirement extraction and requirement-BIM matching. For example, Zhang and El-Gohary (2017) used NLP techniques to fully-automatically extract building design requirements from building codes, and used machine learning techniques to semi-automatically match the requirements to the Industry Foundation Classes (IFC)-represented design information. One main limitation of this work is that it has not been tested on energy codes and contract specifications, which contain challenging text complexities (i.e., longer provisions, requirement exceptions, hierarchically-complex sentence/text structures, incomplete sentence structures, and variety of LODs) (as discussed in Section 1.1). Also, their requirement-BIM matching is semi-automated,

although requiring minimal manual effort. The above analysis shows that there is still a lack of an ACC method that can achieve a full level of automation in both requirement extraction and requirement-BIM matching, and can be generalized across different types of regulations/documents.

In addition to the aforementioned knowledge gaps, existing ACC systems for compliance checking with building energy conservation requirements [e.g., COMcheck and REScheck developed by United States Department of Energy (U.S. DOE 2018b)], have three additional limitations: (1) procedural and rigid checking programs: For example, for checking interior lighting power requirements using the building area method (calculating the total allowed wattage by multiplying the prescribed unit area power allowance by the total area of the select type of building area like a convention center or school), the users are required to enter a large number of lighting fixtures information (e.g., type of fixture, lamps per fixture, number of fixture, fixture wattage) from lighting fixtures schedule to calculate the total proposed wattage. Then the compliance is determined by simply comparing the total proposed wattage with the total allowed wattage. This procedural and rigid checking process limits the reusability and extension of the developed checking program for other analysis; (2) long checking period: in conducting compliance checking, the users are required to manually enter all the related building elements with attribute values (e.g., type of lighting fixture with fixture wattage) for each checking topic. This would be time-consuming if checking a large project which may contain a huge number of elements. For example,

for checking a regular building project, it typically requires 3-5 business days to generate a compliance checking report (U.S. DOE 2018b); and (3) lacking the capability to check compliance with contract specifications: project contracts, including contract specifications, are a major source of law – the source of private law; a contract represents a binding agreement imposing requirements on construction projects. Checking the compliance with energy codes is vital, but not sufficient; it is also important to check the compliance with contract specifications since they also prescribe environmental/energy requirements. However, automated specification compliance checking remains to be a challenge because of the specifications' "project-specific" nature; project specifications could vary widely from a project to project.

2.2 Text Classification

2.2.1 Text Classification Problems

NLP is a subfield of artificial intelligence that aims to enable computers to process natural language in a human-similar way (Manning and Schütze 1999). TC is a subfield of NLP that aims to assign documents (or text units, such as paragraphs or clauses) to one or more predefined categories (Manning and Schütze 1999). The text is usually unstructured (i.e., does not have a clear computer-readable structure). A category is represented by a label, and may refer to a class or concept. TC problems can be categorized as multilabel or single-label classification problems (Tsoumakas and Katakis 2007; Ghamrawi and McCallum 2005). Multilabel classification aims to assign more than one label to a document. Single-label classification, on the other hand, aims to

predict only one label for each document. A single-label classification problem can be further categorized as: (1) a binary classification problem, if there are only two classes (usually as a positive class and a negative class) in the dataset; or (2) a multiclass classification problem, if the number of classes is more than two. In this research, a multilabel TC problem is addressed, since multiple labels could be assigned to one clause. For example, the following clause was assigned the labels “air leakage topic” and “thermal insulation topic”, because it contains requirements for high pressure ducts in terms of thermal insulation and sealing to prevent air leakage: “C403.2.7.1.2 Medium-pressure duct systems. All ducts and plenums designed to operate at a static pressure greater than 2 inches water gauge (w.g.) (500 Pa) but less than 3 inches w.g. (750 Pa) shall be insulated and sealed in accordance with Section C403.2.7. Pressure classifications specific to the duct system shall be clearly indicated on the construction documents in accordance with the International Mechanical Code” (ICC 2012).

There are two common methods to solve multilabel classification problems (Tsoumakas and Katakis 2007). The less commonly-used method is the direct approach – Algorithm Adaptation Method (AAM), which can cope with multilabel classification problems directly by modifying or extending some available algorithms. The advantage of AAM is that it can predict a set of labels at one time. However, its performance is still not good enough (Tsoumakas and Katakis 2007). The most commonly-used method is the indirect approach – Problem Transformation Method (PTM), where a multilabel classification problem can be transformed into two or more single-label

classification problems by assuming the independence of labels. If the number of labels in the original datasets is L, the transformation will result in L single-label classification problems (and thus L classifiers) and L number of datasets (one dataset for each label L_i). Each dataset is used to train one classifier on predicting the label L_i of that dataset. During testing, each test clause is processed by those L number of classifiers one by one, where each classifier decides whether to assign its corresponding label L_i or not. The total number of assigned labels during this process form the final label set of this test clause. Examples of TC work adopting PTM include Caldas et al. (2002), Kovacevic et al. (2008), and Mahfouz (2011). Using an AAM approach, only one classifier is built for all labels. Examples of TC work using AAM include Brinker and Hüllermeier (2007), Zhang and Zhou (2007), and Spyromitros et al. (2008). The advantages of AAM are: (1) the ability to predict a set of labels at one time; and (2) avoiding the assumption of label independence, which is not valid in many cases because labels are usually interrelated in real world (Manning et al. 2009). Considering interrelationships between labels may improve the TC performance (Sorower 2010).

A multilabel classification problem that was transformed to a single-label problem can be further addressed using a binary classification or a multiclass classification approach (Aly 2005). For each of the transformed datasets with label L_i, a binary classification approach defines the label L_i as the positive class and combines all other labels in a negative class, and then applies binary classification algorithms to address this binary classification problem. In contrast, a multiclass

classification approach does not combine the labels and directly uses multiclass classification algorithms. The advantage of a multiclass classification approach is that data imbalance problems resulting from the transformation (i.e., combining labels results in a relatively larger negative dataset in comparison to the smaller positive dataset) could be avoided.

2.2.2 Hierarchical Text Classification

TC could be flat (non-hierarchical) or hierarchical. Different from flat TC, in hierarchical TC, the labels are organized into a class hierarchy (usually represented as a tree structure or class taxonomy) (Silla and Freitas 2011; Fagni and Sebastiani 2010; Yoon et al. 2006; Sun et al. 2003; Sun and Lim 2001). Representing the labels in the form of a class hierarchy may support the TC process by offering a better description of the meanings of the labels in terms of its superclasses and subclasses. Hierarchical TC problems can be addressed using one of the following three approaches: flat approach, local classifier approach, and global classifier approach (Silla and Freitas 2011). The flat approach does not take the hierarchical information into account and only uses the labels of the leaf classes (Silla and Freitas 2011). When a leaf class is assigned to a document, all of its superclasses are also assigned to that document. This provides a simple but indirect solution to hierarchical TC.

The local classifier approach takes local hierarchical information into account (Silla and Freitas 2011; Yoon et al. 2006; Sun et al. 2003; Sun and Lim 2001). It takes a top-down approach in assigning documents to classes; for each document, the classifier assigns its first level class, then

proceeds to assign the document to the direct subclasses of that class, and so on, until reaching the leaf level of the hierarchy. The main disadvantage of the local classifier approach is that a misclassification of one class would propagate down the hierarchy to all subclasses. This may lead to low performance results at the lower levels of the hierarchy.

The global classifier approach takes the class hierarchy as a whole into account (Silla and Freitas 2011; Yoon et al. 2006; Sun et al. 2003). It tries to build a single classifier that can predict all classes from all levels of the hierarchy at one time. Global classifiers are relatively complex and their performances are usually inconsistent. This has limited the application of global classifiers.

In this research a flat approach is used; all labels used for classification are leaf classes. Although flat TC is used, retrieving documents on a parent level is easily achieved by aggregating the retrieved documents that have been retrieved on the leaf/children levels.

2.2.3 Text Classification Using Machine Learning Techniques

ML techniques are commonly used for TC. ML refers to a system learning from available data or previous experience (Manning and Schütze 1999). ML techniques can be categorized into three main types: (1) supervised ML: human guidance is provided in the form of labelled documents (all documents are given one or more predefined labels), where a training dataset is used to train the classifier to automatically classify a given document according to a predefined set of labels and a testing dataset is used to test the performance of the classifier; (2) unsupervised ML: documents are not labelled for training; and thus, instead of classifying given documents according to a

predefined set of labels, classifiers automatically (and without human guidance) cluster documents into potentially useful categories; and (3) semi-supervised ML: only a fraction of the training dataset is labelled, which provides partial human guidance.

In comparison to unsupervised and semi-supervised ML, supervised ML algorithms require higher manual effort for preparing the training dataset. However, their precision and recall are typically higher due to the benefit from human guidance. Some commonly-used supervised and semi-supervised ML algorithms include Support Vector Machines (SVM), Naïve Bayes (NB), k-Nearest Neighbors (kNN), and Decision Trees (DT) (Aggarwal and Zhai 2012). SVM is the most commonly-used supervised ML algorithm. It maps the labelled data into a feature space and tries to find the best separators that distinguish all categories. The testing data are then mapped to the feature space and classified by the found separators (Aggarwal and Zhai 2012). Some commonly-used unsupervised ML algorithms (Aggarwal and Zhai 2012) include k-Means and hierarchical algorithm (Aggarwal and Zhai 2012). Some less-commonly-used ML algorithms include labeled-(latent Dirichlet allocation) LDA (Ramage et al. 2009).

ML TC requires the representation of documents in terms of numerical features. The most commonly-used method for representing features of the text is the BOW model (Manning and Schütze 1999). In this model, a document is represented as an unordered set of words along with their corresponding frequencies of occurrence in this document, and the positions of the words are ignored. The words are all drawn from the vocabulary used in the document. The frequency of

each word is then normalized by the total occurrence of this word in the whole document collection. The advantages of the BOW model are simplicity and computational efficiency, though they come at the cost of discarding the relationships among words in terms of their relative positions in the document. Another, but less commonly-used, text representation method is the bigram model (Manning and Schütze 1999). In this model, the semantic relationships between any two adjacent words are captured. For example, the word-group "spring thermal radiation" is more likely to occur than "thermal spring radiation" for the "building energy efficiency topic". A document is represented by all such adjacent pairs of words along with their corresponding frequencies of occurrence in this document. A word-pair frequency in one document is then normalized by its total frequency in the entire set of documents.

Because different features have different powers in indicating a category, they should be assigned with different weights. There are two types of weighting schemes: unsupervised term weighting and supervised term weighting. Membership information refers to the known information about which category a training document belongs to. Unsupervised term weighting does not use this information. The most state-of-the-art unsupervised weighting scheme is term frequency inverse document frequency (TFIDF) (Manning et al. 2009). TF refers to the total occurrence frequency of a term in one document; DF refers to the number of documents in the entire document collection that contains this term; and IDF refers to the inverse of DF. TFIDF assumes that: (1) if a term occurs frequently in one document, then it is highly relevant to the category of this document, and

(2) if a term occurs frequently in many documents in the collection, then it is probably not discriminative of any category of documents. Accordingly, TFIDF aims to assign: (1) a higher weight to a term that appears frequently in one document, and (2) a lower weight to a term that appears frequently in many documents in a collection.

In contrast to unsupervised term weighting, membership information is used in supervised term weighting. Since not all categories have the same number of documents, supervised term weighting takes this statistical document distribution information into account when calculating the weight of a term in a document. Examples of newly-developed supervised term weighting schemes include term frequency relevance frequency (TFRF) (Man et al. 2009) and logarithmic term frequency maximum relevance frequency ($TF_{max}RF$) (Xuan and Quang 2014), where TF is same as that in TFIDF weighting and RF measures the relevance of a term to a category.

Because not all features contribute to the discrimination of a category, non-discriminative features need to be filtered out to enhance the power of those discriminative features. Feature selection (Manning et al. 2009) is the process of selecting a subset of the features in the training dataset and using this subset of features to represent the text. There are two main advantages of implementing feature selection. First, the computational efficiency can be improved by selecting a fraction of the features, especially in cases where the feature size can be in the order of millions and/or when using algorithms that require expensive computation like Naïve Bayes (NB) algorithms. Second, as mentioned above, performance can be improved by reducing non-discriminative features and

keeping the most discriminative features. There are two main approaches to selecting features: univariate feature selection (UFS) and recursive feature selection (RFS).

UFS tries to use univariate statistical tests to select features. UFS involves calculating a score for each feature using a scoring function, ranking features based on the scores, and then selecting the best features based on the ranking. To evaluate whether a feature is helpful in representing a category, a utility function is defined as U (feature, category) for scoring features. Feature scoring is the process of ranking features based on a utility function U (feature, category). All features ranked below a predefined threshold are discarded and only the features above the ranking threshold are used in classification. Common feature scoring functions used for multiclass classification include Chi-square (CHI), Information Gain (IG), and Mutual Information (MI). For the details of these feature scoring methods, the readers are referred to Aggarwal and Zhai (2012).

Instead of using a scoring function to rank and select features, RFS applies a ML algorithm to select features based on the ranking of features in terms of weights. The ML algorithm is used to assign weights to the features for ranking. The initial feature set is used as training data for the ML algorithm. The learned classifier assigns a weight to every feature. Then, a predefined number of features (N) with the lowest absolute weights are discarded. The remaining features are used as new training data for the ML algorithm. Then the weight of each feature is updated by applying the ML algorithm again on the new training data and another N features with the lowest weights are pruned. This recursive process terminates when the total number of remaining features reaches

another predefined number (M).

ML can also be classified into two main types: shallow learning and deep learning. Shallow learning can only learn simple functions with a linear combination of parameters from the training data (Bengio and LeCun 2007). Shallow learning algorithms (e.g., SVM) have been successful in TC, but their limited modeling and representational power make them unable to learn complex functions such as those involved in text semantics (Bengio and LeCun 2007). In contrast, deep learning can learn complex functions (cascaded by multiple single functions) with a non-linear combination of parameters from the training data (Bengio and LeCun 2007). Deep learning attempts to model the data based on the theory of distributed representations from ML. Distributed representations assume that the data are generated by some hidden factors. Deep learning further assumes that these hidden factors are organized into a multi-level hierarchy. Therefore, deep learning models the data in a multi-level hierarchy (Bengio et al. 2013). Examples of algorithms for implementing deep learning include neural networks, restricted boltzmann machines (RBM), deep belief networks (DBN), and stacked auto-encoders (Bengio 2009; Goodfellow et al. 2016). The most heavily-used algorithm for implementing deep learning is the neural network algorithm (Bengio et al. 2003; Goodfellow et al. 2016). A neural network algorithm models the iterative learning process of the human brain that learns from known information (i.e., unlabeled data) and infers new unknown information based on the learned knowledge (e.g., predicting the next word of a partial sentence based on the embedded linguistic characteristics/patterns learned from seeing

a large number of sentences) (Bengio et al. 2003). Examples of neural network algorithms include the convolutional neural networks (Bengio 2009), feedforward neural network algorithm (Bengio et al. 2003), and the recurrent neural network algorithm (Mikolov et al. 2010). The most state-of-the-art and best-performing algorithm is the hierarchical softmax skip-gram algorithm (Mikolov et al. 2013a; Mikolov et al. 2013b), which is a variant of the three-layer feedforward neural network algorithm. The hierarchical softmax skip-gram was developed to improve the computational efficiency and accuracy of distributed representations on large datasets. It tries to learn word vector representations from the training data and predict surrounding words of the current word in a sentence based on the corresponding learned word vectors (Mikolov et al. 2013a). The learned word vectors could predict semantic relationships of words/concepts (e.g., automatic-turn vs. manual-shut, lumen-luminaire vs. watts-lamp, weld-gasket vs. fasten-caulk) based on cosine distance of vectors.

2.2.4 Ontology-Based Techniques for Semantic Text Classification

Semantic TC refers to using the semantics of text to facilitate TC. An ontology is a knowledge conceptualization that captures the semantics of a domain in the form of concepts, relationships, and axioms (El-Gohary and El-Diraby 2010). An ontology can, thus, help in capturing the semantics of the text. In general, ontologies may support TC in two main ways: (1) Use an ontology to represent the features of the documents and then use a ML algorithm to classify documents based on their features. For example, in Lee et al. (2009), term features are extracted from

documents and mapped to the concepts of the ontology. Documents originally represented by term features get represented by ontology concept features instead. These concept features are then used for ML-based TC; and (2) Use an ontology to represent the categories in terms of concept features and then use the concept features of each category to classify the documents (represented in either concept features or term features) based on either concept-to-concept or concept-to-term semantic similarity scores. For example, Yu et al. (2006) use a combination of a linguistic ontology and statistical information (such as word frequency) for TC. The ontology covers concepts that describe the syntactic features [e.g., part of speech (POS) tag of a word] and semantic features of words (e.g., semantic tag of a word). These syntactic and semantic features of words (what Yu et al. call "linguistic ontology knowledge") are then learned based on a set of labelled training data. For TC, the keywords of documents are extracted and the documents are classified based on the linguistic ontology knowledge of its keywords. Yang et al. (2008) use concept vectors for TC. A category is represented in terms of a vector of concept-value pairs, where (a) the concept is derived from an ontology, and (b) the value is defined based on their term frequency inverse document frequency (TFIDF) scores [TFIDF aims to weight a word in a document in terms of the total count of that word in the document and the total number of documents in the whole document collection containing that word (Aggarwal and Zhai 2012)]. A testing document is represented in terms of a vector of keyword-TFIDF pairs. The documents are then classified based on the similarities of document vectors to category vectors. In contrast to the first example, the second example may be classified as an unsupervised ontology-based effort, because labelled training data are not needed.

Compared with supervised ML-based TC, unsupervised ontology-based TC thus provides the opportunity of eliminating the massive manual effort required for labeling training data.

2.3 Information Extraction from Text

2.3.1 Information Extraction

NLP is a subdiscipline of artificial intelligence that aims to enable computers to understand human language (Manning and Schütze 1999). IE applies NLP techniques [e.g., part-of-speech (POS) tagging, morphological analysis, etc.] to recognize information from unstructured data and formalize it into structured data (Jurafsky and Martin 2009). According to the level of complexity, IE can be categorized into four types: (1) named entity recognition, which aims to identify a particular entity (Jurafsky and Martin 2009); (2) relation detection, which aims to discern the relationships among the identified entities (Jurafsky and Martin 2009); (3) event extraction, which aims to identify events from text: each event has a trigger (i.e., the main word stating the event) and a number of associated arguments, and each event may be composed of a number of entities and their relationships (Grishman 2012; Piskorski and Yangarber 2013); and (4) full information extraction, which aims to extract all information expressed by a sentence based on a full analysis of the sentence (Zhang and El-Gohary 2013). Named entity recognition, relation detection, and event extraction can be classified as shallow IE because they aim to extract partial information from a sentence, whereas full information extraction could be classified as deep IE because it aims to extract all information from a sentence (Zhang and El-Gohary 2013).

There are two approaches to IE (Moens 2006; Jurafsky and Martin 2009; Moreno et al. 2013): a rule-based approach and a supervised machine learning (ML)-based approach. A rule-based approach requires human effort to analyze the text features in a relatively small set of text corpus (sometimes called developing data, which is analogous to training data in the case of ML), define the text patterns in terms of the text features, and then develop extraction rules based on the defined patterns. Text features may include (Moens 2006): (1) syntactic features, which refer to syntax-related features that are determined based on grammatical analysis, such as POS tags (e.g., tag "IN" represents a preposition like "for"); and/or (2) semantic features, which refer to concepts that capture the meaning of the information (e.g., "mass wall" is a concept that represents a type of wall). The patterns may be defined in terms of combinations of different syntactic and/or semantic features via regular expressions. Regular expressions is a language that is implemented by computers for pattern matching to characterize possible sequences of text (Jurafsky and Martin 2009).

A supervised ML-based approach requires human effort to collect a relatively large set of training data and annotate them with a large number of different types of text features and with the information that should be extracted. Then, a ML algorithm (e.g., Support Vector Machines, Hidden Markov Model, and Conditional Random Field) is used to automatically learn the extraction rules from the annotated training data. Compared with the rule-based approach, the ML-based approach (1) requires much larger size of annotated training data: because the performance

of a ML-based IE algorithm depends on the training data for learning, a sufficiently large size of training data is required to accurately learn text patterns and extraction rules; and (2) does not require manual efforts in pattern definition and extraction rule development: a ML algorithm automatically learns the patterns of text and the extraction rules.

2.3.2 Ontology-Based Information Extraction

Ontology-based information extraction (OBIE) is a subfield of IE. Comparing to general IE, which only depends on the lexical and/or syntactic information of the text, OBIE further relies on semantic information to extract information based on meaning. In many cases, OBIE is domain and application-oriented, when a domain and/or an application ontology is used to assist in extracting semantic information that is specific to a particular domain and/or application (Wimalasuriya and Dou 2010; Karkaletsis et al. 2011; Zhang and El-Gohary 2013). In this case, OBIE captures domain-specific semantic information as semantic features, which are then used in the patterns in the extraction rules. Compared with general IE, the domain-specific semantic information that is used in OBIE is promising in improving the IE performance for a specific domain (Wimalasuriya and Dou 2010; Zhang and El-Gohary 2013).

OBIE has been explored in different domains such as biology (e.g., Moreno et al. 2013), business (e.g., Arendarenko and Kakkonen 2012; Tao et al. 2014), law (e.g., Moens 2006), medicine (e.g., Soysal et al. 2010), mechanical engineering (e.g., Li and Ramani 2007), and civil engineering (e.g., Zhang and El-Gohary 2013; Liu and El-Gohary 2017). OBIE has also been explored in different

complexity levels of IE: named entity recognition (e.g., Moreno et al. 2013), relation detection (e.g., Li and Ramani 2007; Soysal et al. 2010; Tao et al. 2014), event extraction (e.g., Arendarenko and Kakkonen 2012), and full information extraction (e.g., Zhang and El-Gohary 2013). The most complex level (i.e., full information extraction) is the most challenging and the least explored. In terms of approach, all these efforts used a rule-based approach to deal with the OBIE problem.

2.4 Contract Specifications in the MasterFormat

Different types of documents are written in different "languages" to convey information for different purposes. Contract specifications are relatively well-organized – following standardized text organization and formatting – and are written in a highly concise language to reduce verbiage. The conciseness of the language can be illustrated in terms of sentence structure and writing style.

2.4.1 Specification Formatting

The MasterFormat is a hierarchical classification system developed by the Construction Specifications Institute (CSI) and the Construction Specifications Canada (CSC) for organizing project manuals (including contract specifications) (CSI and CSC 2014). There are a number of MasterFormat versions, with the most recent version being the MasterFormat 2018 (CSI and CSC 2018). In the MasterFormat 2018, contract specifications are divided into 50 divisions (numbered from 00 to 49). Each division is further broken down into a number of sections (numbered in a six-digit format like "072100"), and each section specifies the work results of a construction project in a certain stage (CSI and CSC 2014). To provide a further standardized text organization for each

section, the CSI and CSC developed the joint standards – SectionFormat/PageFormat (CSI and CSC 2017). The SectionFormat defines the overall text organization for each section to reduce possible omission or duplication of construction information (CSI and CSC 2009), while the PageFormat defines the text organization for each page of a section to provide consistent inner-section text formatting and numbering (CSI and CSC 2009).

In the SectionFormat, each section consists of three "parts" – "Part 1 General", "Part 2 Products", and "Part 3 Execution". Each part prescribes requirements corresponding to a different group of topics. Part 1 describes the administrative, procedural, and temporary requirements (e.g., references, definitions, submittals, quality assurance, delivery, storage, handling, warranty, commissioning, and maintenance) (CSI and CSC 2009, 2017). Part 2 describes the requirements for products, materials, equipment, systems, assemblies, accessories, fabrications, mixes, and factory finishing prior to installation or incorporation (CSI and CSC 2009, 2017). Part 3 describes field and site installation or application requirements such as preparatory actions and post-installation cleaning and protection (CSI and CSC 2009, 2017). The text in each part is organized in a hierarchical structure: each part consists of at least one article, each article contains at least one paragraph, and each paragraph may contain zero, one, or multiple levels of subparagraphs. The degree of detail of the requirements increases from the article level to the subparagraph level. Each article prescribes requirements about a major subject (e.g., mineral-wool board). Each paragraph in an article prescribes all related requirements for a particular subject (e.g., foil-faced

semi-rigid mineral-wool board) in that major subject category, while each subparagraph prescribes specific requirements for that particular subject (e.g., thermal resistivity for the foil-faced semi-rigid mineral-wool board).

In the PageFormat, formatting guidelines are described for numbering and naming the articles, paragraphs, and subparagraphs in a section. For example, each article is numbered by a "part" number, a decimal point, and one or two digits starting with either "1" or "01", and each article title is named in uppercase without ending punctuations. An illustrative example of an article in the SectionFormat/PageFormat is shown in Figure 2.1.

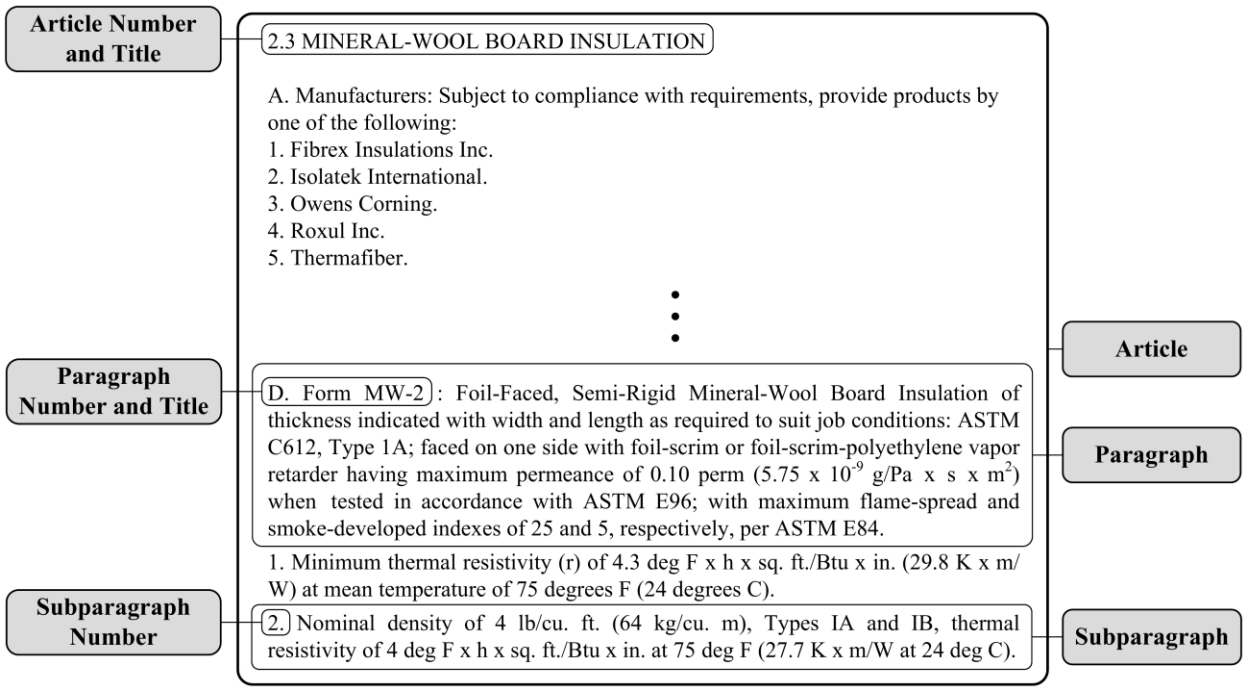


Figure 2.1. Example of an article in the SectionFormat/PageFormat

2.4.2 Sentence Structure

Sentence structures in contract specifications are different than those in energy codes in two ways.

First, shorter and incomplete sentences are used in the specifications. A sentence is usually composed of one or multiple phrases. A phrase is a piece of text that is separated by a delimiter such as a comma, colon, or semicolon (such delimiters are the text features of incomplete sentence structures, which are called "incompleteness features" thereafter). In this research, a phrase that is separated by a comma delimiter is called an information unit, while one or multiple phrases that are enclosed by colon/semicolon delimiters form an information group. An information unit is the minimum semantic information unit, which could represent a target information. An information group is a set of semantically related information units, which could represent a partial or full requirement. For example, S1 contains four information groups (each group is marked by a pair of angle brackets "<>"), in which the second information group has two information units (e.g., "ASTM C665", "Type I").

- S1: "<Un-faced, Glass-Fiber Blanket Insulation of thickness indicated with width and length as required to suit job conditions>: <ASTM C665, Type I>; <with maximum flame-spread and smoke-developed indexes of 25 and 50, respectively, per ASTM E84>; <passing ASTM E136 for combustion characteristics>."

Second, two grammatical moods – imperative and indicative – are used to write sentences in specifications, while only the indicative mood is used in codes. Imperative sentences are featured by a verb that explicitly defines an action (called "action verb" thereafter) as the beginning of a sentence, where the subject of an imperative sentence (e.g., "contractor") is implicit to improve conciseness and understandability. Imperative sentences are often used to prescribe requirements related to the installation of products or equipment (CSI 2004; Kalin et al. 2010), where these

requirements correspond to a BIM LOD of 400 or above. S2 shows an imperative sentence that contains a fabrication requirement, which is in LOD 400 or above. Indicative sentences are featured by a modal verb (e.g., shall) that precedes the main verb of a sentence. The usage of indicative sentences is minimized because they may result in unnecessary verbiage.

- S2: “Fabricate corners minimum 18 inches x 18 inches (450 mm x 450 mm) mitered and sealed as one piece.”

2.4.3 Writing Style

Streamlined writing is a writing technique recommended by CSI to reduce verbiage and grammatical issues in writing contract specifications (CSI 2004; Kalin et al. 2010). Using streamlined writing, paragraphs (including subparagraphs) are written in a 3-tuple format <topic, colon, content>, where “topic” is usually a keyword/phrase that refers to the primary subject of a paragraph, “colon” means “shall”, and “content” contains a number of sentences that prescribe detailed requirements related to the “topic”. Streamlined-written paragraphs have unique text capitalization features: the first letter of each word in the “topic”, and the first letter of the first word in the “content” are capitalized. Such capitalization features may be used to define text feature patterns to identify the “topic” of a paragraph because the “topic” usually contains target information that needs to be extracted. For example, the “topic” of P1, a streamlined-written subparagraph, contains the target information “Vapor Permeance”.

- P1: “6. Vapor Permeance: 0.01 perms (0.6 ng/Pa x s x sq. m); ASTM E96, Water Method.”

2.5 Industry Foundation Classes and buildingSMART Data Dictionary

The Industry Foundation Classes (IFC) is a widely accepted open specification for data exchange in the architectural, engineering, construction, and facility management (AEC and FM) domain (Eastman et al. 2010). The IFC specification is the "only existing public and non-proprietary, and well-developed data model for buildings and architecture existing today" (Eastman et al. 2011). The IFC schema is developed and maintained by the Model Support Group of buildingSMART (buildingSMART 2016a). The conceptual schema of IFC is written in EXPRESS data modeling language, registered as ISO 10303-11 (ISO 1994), and IFC now has become the official ISO standard – ISO 16739:2013 (ISO 2013). A number of versions of IFC specification have been developed, including IFC2x, IFC2x2, IFC2x3, IFC2x3 TC1, and IFC4 Add1. The latest version is IFC4 Add2 (Liebich et al. 2016).

The buildingSMART Data Dictionary (bSDD), formally known as the International Framework for Dictionaries (IFD), is an ISO 12006-3-based library that contains objects and their properties for the building and construction industry (buildingSMART 2016b). It aims to help participants identify and share objects and properties regardless of human language (buildingSMART 2016b). bSDD incorporates the mapping to the IFC specification so that searching a concept/relationship in the bSDD may return the corresponding IFC concepts (which may refer to an IFC entity, enumeration type, etc.) and relationships. For example, searching a concept "slab" in bSDD would return the IFC concepts "IfcSlab" and "IfcSlabType". Each bSDD concept is assigned a name as

a label (buildingSMART 2017). There are two types of names: long name and short name. The long name refers to the full name. There may exist multiple long names as synonyms (e.g., thermal insulance, thermal resistance, coefficient of thermal insulation, R value). The short name refers to the abbreviation of the concept, and there may exist zero or more short names (e.g., "meter" has a short name "m").

There are nine types of bSDD concepts: "activity", "actor", "classification", "document", "measure", "property", "subject", "unit", and "value". For the definition of each type, the readers are referred to (buildingSMART 2017). "Subject" may either refer to a physical object (e.g., a building element duct) or a logical object (e.g., space, submittal) (buildingSMART 2017). "Property" refers to an attribute of an object (e.g., R-value is an attribute of an object duct). bSDD concepts are connected by relationships. There are 25 relationships defined in the bSDD. The most commonly-used is the "specialization" relationship, which means one concept is a subconcept of another. For example, a duct is a specialized building element indicating that the concept "duct" is a subconcept of "building element". bSDD offers an open source Representational State Transfer (REST) model-based application programming interface (API) for parsing the bSDD concepts and relationships, and searching the matched concepts and relationships in other classifications (e.g., IFC) (buildingSMART 2017).

2.6 BIM Information Extraction

BIM information extraction aims to extract information from BIM models and prepare the

extracted information for different BIM uses (e.g., building energy analysis, regulatory compliance checking). Depending on the data representations of the BIM models, there are two main BIM information extraction approaches: (1) a proprietary software API-based approach that uses the software API to extract information from BIM models in proprietary softwares; and (2) an open standard data model parsing approach that uses either open source IFC toolboxes [e.g., OpenIFCTools, Java Standard Data Access Interface (JSDAI), xBIM, IFC Engine DLL] or custom-developed parsing algorithms to extract information from BIM models in open specifications (e.g., IFC, ifcXML, gbXML). Table 2.1 summarizes the BIM information extraction efforts in the recent five years, which used either approach for supporting different applications.

Table 2.1. BIM Information Extraction Efforts in the Recent Five Years

BIM extraction approach	Extraction tool	Application
Proprietary software API-based approach		**Example efforts in supporting general applications**
	Autodesk Dynamo API	Building energy performance visualization and management (Gerrish et al. 2017)
	Navisworks API	Construction risk knowledge management (Ding et al. 2016)
	Autodesk Revit API	Construction workface planning (Liu et al. 2016)
		Construction-specific information management (Nepal et al. 2013)
	Feature Manipulation Engine (FME)	Conversion of IFC schema to IndoorGML schema (Teo and Yu 2017)
		Example efforts in supporting regulatory compliance checking
	Autodesk Revit API	Building thermal envelope energy checking (Sinha et al. 2013)
	bim+ REST API	Fire safety checking (Preidel and Borrmann 2016)
Open standard data model parsing approach		**Example efforts in supporting general applications**
	OpenIFCTools	Partial building information model extraction (Zhang and Issa 2013)
	JSDAI	Dimensional quality assurance of full-scale precast concrete elements (Kim et al. 2016b), and mapping IFC schema to CityGML schema (Deng et al. 2016)
	xBIM	Indoor and outdoor combined route planning (Teo and Cho 2016)
	IFC Engine Dynamic Link Library (DLL)	BIM semantic information enrichment (Belsky et al. 2016), and indoor space path planning (Lin et al. 2013)
	Custom-developed parsing algorithms	Construction-specific information management (Nepal et al. 2013), building energy analysis (Lilis et al. 2016; Kim and Anderson 2013; Kim et al. 2013b; Kim et al. 2016a; Cemesova et al. 2015), partial model extraction (Won et al. 2013), safety risk identification (Zhang et al. 2016), automated construction schedules generation (Kim et al. 2013a), automated cost estimation (Lee et al. 2014), and interior utility network analysis (Hijazi et al. 2012)
		Example efforts in supporting regulatory compliance checking
	JSDAI	General building design checking (Zhang and El-Gohary 2015)
	IFC Engine DLL	High-rise and complex building evaluation checking (Choi et al. 2014)
	Custom-developed parsing algorithms	Compliance checking of fire safety (Dimyadi et al. 2016a), building sustainability (Beach et al. 2015), deep foundation design (Luo and Gong 2015), building accessibility and visibility (Lee et al. 2015), building energy efficiency (Cheng and Das 2014), general construction conformity (Yurchyshyna et al. 2008), general building design (Dhillon et al. 2014), and building envelope design (Nawari 2012; Tan et al. 2010)

2.7 BIM-Requirement Alignment

To check the regulatory compliance of a given instance of a BIM, the concept representations of

the BIMs should be aligned to the concept representations of the regulatory requirements so that they can "speak the same language" (or at least translate well). There are a significant number of regulatory compliance checking efforts in the AEC and FM domain, in which four main ways were mainly used to address the alignment problem. In the first approach, concepts and terms of the BIM (e.g., IFC concepts) are used in representing the regulatory requirements (i.e., "write the regulatory requirements using the BIM language/terminology"). The regulatory requirements may be represented as Jess rules (e.g., Zhong et al. 2015), Jena rules (e.g., Cheng and Das 2014; Baumgärtel et al. 2015), Semantic Web Rule Language (SWRL) rules (e.g., Zhong et al. 2012), conceptual graph-represented rules (e.g., Solihin and Eastman 2016), EXPRESS rules (e.g., Dimyadi et al. 2016c), BIM-server advanced queries (e.g., Qi et al. 2014), SPARQL Protocol and RDF Query Language (SPARQL) queries (e.g., Yurchyshyna et al. 2008; Yurchyshyna and Zarli 2009), BIM software API functions (e.g., Melzner et al. 2013; Zhang et al. 2013; Nguyen and Kim 2011), or custom-developed computer programs (e.g., Lee et al. 2016). In the second, a separate mapping scheme is developed to map the concepts and terms of the regulatory requirements to those of the BIM (i.e., "translate the regulatory language/terminology to the BIM language/terminology"). The mapping scheme may be represented as a mapping ontology (e.g., Beach et al. 2015), N3Logic rules (e.g., Pauwels et al. 2011), JBoss rules (e.g., Tan et al. 2010), procedural mapping algorithms/functions (e.g., Delis and Delis 1995; Goel and Fenves 1969), or a set of black box mapping files in the industrial efforts (Dimyadi et al. 2016b) [e.g., DesignCheck (Ding et al. 2006), SMARTCodes (See 2008), ePlanCheck (Liebich et al. 2002), Solibri Model

Checker (SMC 2009)]. In the third approach, the regulatory concepts and terms are used to develop the BIM models (e.g., Choi et al. 2014; Luo and Gong 2015) or extend the representations of the BIM models (e.g., Zhang and El-Gohary 2017) (i.e., "extend the BIM language/terminology with regulatory conceptualizations/terminology"). In the fourth, the users of the regulatory compliance checking systems are required to specify the alignment between the BIM information and the regulatory requirements using predefined functions/languages (i.e., "conduct a manual translation"), such as high-level query functions (Lee et al. 2016), Language-Integrated Query (LINQ) (Nawari 2012), Regulatory Knowledge Query Language (RKQL) (Dimyadi et al. 2016b), Visual Code Checking Language (VCCL) (Preidel and Borrmann 2016), and building environment rule and analysis (BERA) language (Lee et al. 2015).

CHAPTER 3 – TEXT CLASSIFICATION OF ENERGY CODES AND CONTRACT SPECIFICATIONS

3.1 Domain-Specific Hierarchical ML-Based Text Classification for Supporting Automated Energy Compliance Checking

3.1.1 Comparison to the State of the Art

A variety of ML-based TC algorithms (e.g., Aggarwal and Zhai 2012) have been developed in the computer science (CS) domain. Some common methods and popular algorithms implementing these methods include: (1) decision trees (DT) method implemented in algorithms of iterative dichotomiser3 (ID3), classifier4.5 (C4.5), classifier5 (C5) and classification and regression trees (CART) (Breiman et al. 1984); (2) probabilistic method implemented in NB algorithm; (3) linear and non-linear method implemented in support vector machine (SVM) algorithm with linear and radial basis function (rbf) kernel; (4) proximity-based method implemented in algorithms of nearest neighbor and nearest centroid; and (5) ensemble method implemented in algorithms of random forest (RF) and gradient boosted regression trees (GBRT). For the details of these methods, the readers are referred to Aggarwal and Zhai (2012), Breiman (2001), and Friedman (2001). Similar to other TC problems, a number of ML algorithms were explored for multiclass classification problems. For example, Malkani and Gillie (2012) used SVM and NB to classify tweets into a set of topics, Wu et al. (2007) used NB and k-Nearest Neighbors (kNN) to classify news stories, and Giorgetti and Sebastiani (2003) used NB and SVM to classify answers of open-ended questions in surveys.

Despite of these enormous efforts in the CS domain, many challenges still exist in constructing

classifiers that can be effective across different domains and, thus, TC models remain highly domain-specific (Blitzer et al. 2007). There is no single best TC algorithm across all domains; the performance of one best performing ML algorithm tested on one dataset is not necessarily the best one when tested on another dataset, especially when datasets from different domains are more dissimilar (Sebastiani 2002). As discussed in Salama and El-Gohary (2013b), it is difficult to reuse an existing classifier from one domain to another (e.g., medical versus construction), from one subdomain to another (e.g., safety versus environmental), or from one application to another (e.g., document management versus ACC), because text features vary across domains and subdomains, and performance requirements vary across applications (e.g., for ACC, unlike other applications, recall is more critical than precision). There is, thus, a need to identify the specific features of domain text and how to adapt or tune a classifier to those specific features and to the specific performance requirements of the domain or application. A construction-domain-specific TC algorithm is, thus, required for classifying construction documents.

A number of research efforts in the construction domain focused on TC (e.g., Caldas et al. 2002; Kovacevic et al. 2008; Mahfouz 2011; Salama and El-Gohary 2013b). However, hierarchical TC work in the construction domain is limited in the following ways: (1) the performance of hierarchical TC tends to drop quickly when reaching a deeper level in the hierarchy. For example, Caldas and Soibelman (2003) addressed a three-level multilabel binary classification problem, but the accuracy dropped from 96% at the first level to 86% at the third level; (2) the algorithms can

only handle single-label classification problems that were transformed from a multilabel problem using a binary classification approach. Dealing with a transformed multilabel classification problem as a binary instead of a multiclass classification problem may encounter data imbalance problems. A data imbalance problem occurs when the documents of one class are much more than the documents of another class(es) (Sun et al. 2007); (3) the algorithms are not sufficiently adapted to the domain. It is important to utilize the features and methods that work best for each domain (Blitzer et al. 2007). For example, domain-specific stopwords could be removed to make domain content-bearing words more discriminative; (4) the types of ML algorithms that were tested and evaluated are limited. For example, to the best of the author's knowledge, the performance of ensemble methods in classifying construction text were not tested; and (5) the types of term weighting schemes that were tested and evaluated are also limited. Some newly-developed supervised term weighting schemes that showed effectiveness in some domains (e.g., Xuan and Quang 2014) were not tested in classifying construction text.

To address these gaps, this research explores the following: (1) the use of multiclass classification approach to deal with multilabel classification problems; (2) the use of a domain-specific stop word list as an approach for domain adaptation; (3) the testing of a number of ML algorithms (e.g., RF and GBRT algorithms that implement the ensemble method) and term weighting schemes that were not commonly evaluated in the construction domain; and (4) the effect of feature selection and domain-specific stopword removal on the performance of hierarchical classification of

environmental regulatory documents.

3.1.2 Proposed Method for Domain-Specific Hierarchical Text Classification of Energy Regulatory Documents

To address the aforementioned knowledge gaps, this research proposes a domain-specific, ML-based hierarchical TC method for classifying clauses in energy regulatory documents (including energy codes) into a number of hierarchically detailed topics for supporting EnergyACC in construction. The method classifies clauses according to leaf topics at the fifth level of a semantic TC topic hierarchy. A flat approach was used to deal with the hierarchical TC problem. The multilabel classification problem was transformed into a multiclass classification problem. The TC methodology is summarized in Figure 3.1. Step 4 and Step 5 are iterative. Feature selection and domain-specific stopword removal are tested as potential performance improvement strategies.

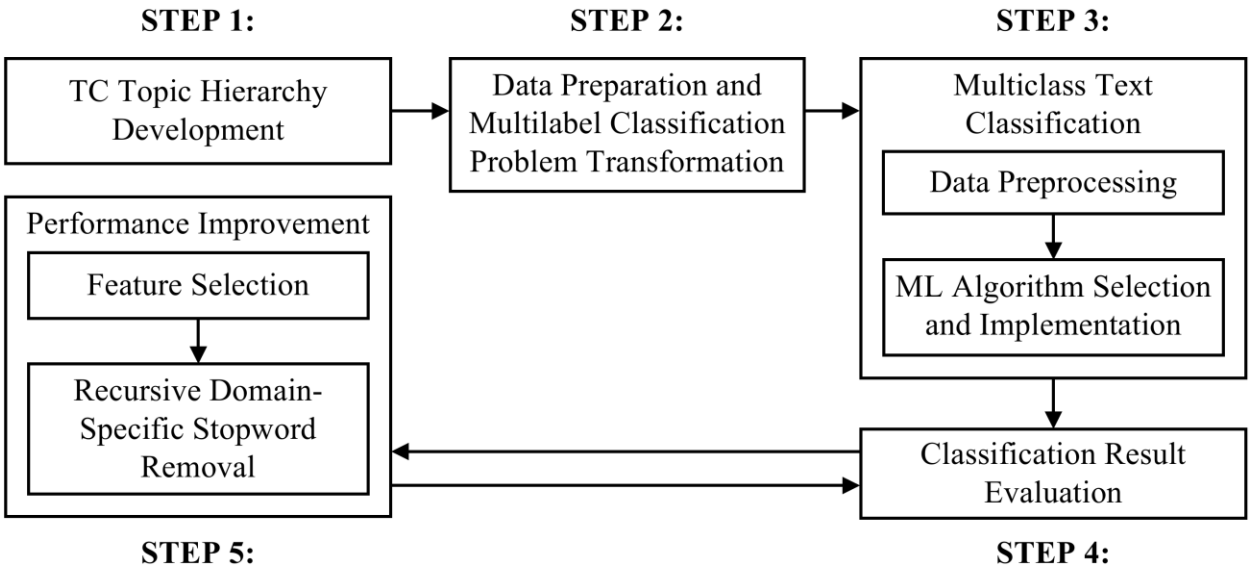


Figure 3.1. Methodology for domain-specific hierarchical text classification

3.1.2.1 TC Topic Hierarchy Development

This research focuses on analyzing the "energy efficiency topic", which is a subtopic of

"environmental topic" (as per Figure 3.2). In order to develop the topic hierarchy, the established methodologies for taxonomy development (e.g., El-Gohary and El-Diraby 2010) were followed. The concepts were extracted based on a review of the main relevant documents in the domain (e.g., environmental codes and standards such as the 2012 International Energy Conservation Code and 2010 ASHRAE Energy Standard for Buildings Except Low-Rise Residential Buildings). Subsequently, the concepts were structured into a taxonomy using a combination of top-down and bottom-up approaches. The "commercial building energy efficiency topic" subhierarchy is shown in Figure 3.2. All the leaf nodes (ten subtopics) were used as labels of classification.

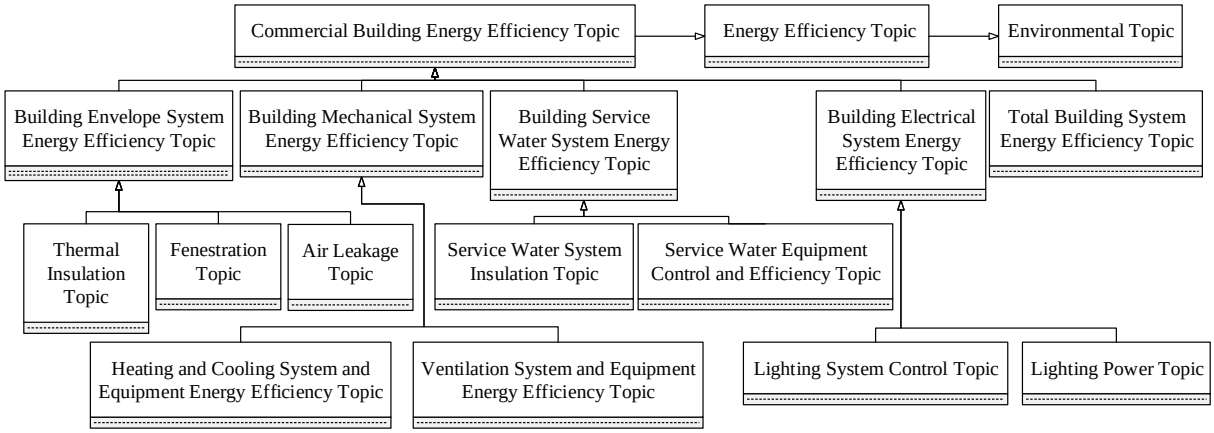


Figure 3.2. Text classification topic hierarchy

3.1.2.2 Data Preparation and Multilabel Classification Problem Transformation

Around 1,200 clauses were collected from ten regulatory documents (see Figure 3.3). These documents were selected because they all cover energy efficiency requirements, which is the focus of this research. In dividing a document into clauses, the document was split to the most granular subheading level. One problem that is generally faced in automatically splitting data is data noise

elimination. Usually, original data are in different formats (.txt, .pdf, .doc, etc.) and/or different encodings (ANSI, Unicode, etc.), while a software system can only process files in a certain format. The collected set of clauses were transformed into .txt format, as required by the developed TC system. However, noise like unknown characters that occur during transformation to .txt format could undermine the performance of TC. The noise was reduced by automatically transforming different encodings to the UTF-8 encoding.

Document
2012 International Energy Conservation Code
2010 California Energy Code
ANSI/ASHRAE/IES Standard 90.1-2010 Energy Standard for Buildings Except Low-Rise Residential Buildings
2013 Nonresidential Compliance Manual for the 2013 Building Energy Efficiency Standards
2007 National Green Building Standard, Chapter 7 Energy Efficiency
ANSI/ASHRAE/USGBC/IES Standard 189.1-2009 Standard for the Design of High-Performance Green Buildings Except Low-Rise Residential Buildings
2009 LEED Reference Guide for Green Building Design and Construction
2013 Energy Policy and Conservation Act, Section 342
Energy Independence and Security Act of 2007
2011 North American Fenestration Standard/Specification for Windows, Doors and Skylights

Figure 3.3. List of regulatory documents

Data sufficiency is also, generally, another challenge for ML-based TC. There is no set definition of how much data are considered sufficient. In the construction domain, especially, there is no benchmark of what is a sufficient data size. However, generally, the more data collected, the more confident it is believed that the data are sufficient. A series of popular datasets in the CS domain include “kdd 2010” and “20 Newsgroups” (Chang and Lin 2011). The main properties (number of classes, data size, and number of features) for popular datasets have varied, for example, from 2 to 105 classes, 44 to 10,000 data pieces, and around 7,200 to 55,000 text features (Chang and Lin

2011). After data preprocessing (Step 3 in Section 3.1.2.3), the used dataset was composed of 10 classes (or topics), around 1,200 data pieces (or clauses), and 4,200 text features. Compared with popular datasets in other applications, the number of features in this set is relatively small. However, it is considered sufficient for the following reasons: (1) the vocabulary used in environmental regulations is relatively standardized; and, thus, the number of distinctive features (e.g., "wattage", "daylight", "switch", "insulation", "leakage", "ventilation") for each class (topic) is relatively small. As a result, a small feature size would result in including sufficient features to identify a text; and (2) the length (number of words) of a clause is relatively small; and, typically, the number of distinctive features of a data piece is proportional to its length. The number of clauses collected in the experiment is also considered sufficient because of the relatively high performance that the classifier achieved.

After data collection, each clause was manually labelled with one or more of the ten topics, which were identified in Section 3.1.2.1. Which labels should be assigned to a clause is based on analyzing the content of that clause. For example, the following clause was assigned the labels "air leakage topic" and "thermal insulation topic", because it contains requirements for high pressure ducts in terms of thermal insulation and sealing to prevent air leakage: "Ducts designed to operate at static pressures in excess of 3 inches water gauge (w.g.) (750 Pa) shall be insulated and sealed in accordance with Section C403.2.7.". For convenience of computer processing, each of the ten labels was represented by a unique serial number from 1-10. The labeling of the dataset was

reviewed by two other researchers and 100% agreement on labeling was achieved.

After data labeling, the multilabel classification problem was transformed to ten (since L = 10 in the original dataset) single-label multiclass classification problems.

3.1.2.3 Multiclass Text Classification

3.1.2.3.1 Data Preprocessing

Data preprocessing is the process of transforming the raw text into the format required by the ML algorithm(s). Most ML algorithms require fixed size numerical feature vectors as the input. Therefore, raw text documents need to be represented by numerical value features. In developing the proposed algorithm, both the BOW model and the Bigram model were tested to determine the best method for representing environmental regulatory text.

In order to represent text using either the BOW model or the Bigram model, three commonly-used techniques for data preprocessing were implemented: (1) Tokenization: Tokenization is the task of segmenting the text into pieces called tokens (words, punctuation, etc.), eliminating certain characters such as punctuation, and transforming words to their lowercase forms (e.g., “building , Thermal Insulation” is tokenized into “building thermal insulation”); (2) Stopword removal: Stopwords refer to those high-frequency and low-content words that are not discriminative in classification, such as function words like “am”, “is”, “a”, “the”, and “of”. According to Zipf’s law in NLP (Manning and Schütze 1999), medium and low-frequency words are usually content-bearing and thus have higher discriminating power, while high-frequency words are low content-

bearing and thus have lower discriminating power. Removing stopwords can, thus, help eliminate non-discriminative high-frequency words, thereby reducing the number of features and revealing the discriminative words; and (3) Stemming: Stemming is the process of stripping off word suffixes (in some cases prefixes) to map a word to its root or stem. For example, "insulation" and "insulated" can both be mapped to "insul". Stemming reduces the number of features by combining words sharing the same stem. It is usually effective in improving the performance of classification (e.g., Liao et al. 2003). In the proposed TC algorithm, stemming was implemented because the experimental work in Salama and El-Gohary (2013b) showed improved performance with stemming. A Python implementation of Porter2 stemming algorithm (Porter 2006) for English stemming was used.

Even if all contentless or low-content features are filtered out, heuristically, not all remaining content-bearing features would have the same power in predicting a label for a clause. Feature weighting is, therefore, used to differentiate between features that are important for classification and those that are not. In developing the proposed algorithm, one unsupervised (TFIDF) and two supervised term weighting schemes (TFRF and $TF_{max}RF$) were tested (Man et al. 2009).

There are many variances of TFIDF weighting schemes. In this research, Equation 3.1 was selected as it can prevent overweighting a high TF and DF by using a logarithmic function, where tf_d is the frequency of a term in one document/clause d, N is the total number of documents/clauses in the collection, and tf_N represents the total frequecy of this term in all documents/clauses.

$$\text{TFIDF} = \log(\text{tf}_\text{d} + 1) * [1 + \log(\frac{\text{N}}{\text{tf}_\text{N}})] \quad (3.1)$$

For the two supervised weighting schemes, because both are developed for binary classification, the weighting equations were extended/modified to adapt them to multiclass classification (and were called $TFRF_M$ and $TF_{max}RF_M$, where the subscript M means "Modified"). In both TFRF and $TF_{max}RF$, TF measures the term frequency in the same way as in TFIDF, while RF and $_{max}RF$ involve supervised effort in contrast to the unsupervised IDF. In TFRF, relevance frequency (RF) measures how relevant a term is to a category. In binary classification, the RF of a term T in the positive category is the ratio of the number of documents (DF) containing term T in the positive category to that in the negative category (as per Equation 3.2). Since this research deals with multiclass classification, this original RF was extended/modified (and was called RF_M). RF_M of a term T in a category C is the ratio of the number of documents (DF) containing term T in category C to that in all other categories (as per Equation 3.3). Accordingly, TFRF was modified to $TFRF_M$, as per Equations 3.4 and 3.5, respectively. In $TF_{max}RF_M$ (Equation 3.7), $_{max}RF_M$ is the maximum RF_M of a term T in each category C_j (as per Equation 3.6), where the upper bound of j represents the total number of categories. For implementing $TFRF_M$ and $TF_{max}RF_M$ (Equations 3.5 and 3.7, respectively) in multiclass classification, a logarithmic TF function [to prevent overweighting of common, non-discriminative terms (same as in Equation 3.1)] and logarithmic RF_M and $_{max}RF_M$ functions [based on the original equations (Man et al. 2009; Xuan and Quang 2014)] were used. Accordingly, Equations 3.8 and 3.9 were used for implementing these two extended/modified

supervised term weighting schemes, where tf_d is the frequency of a term in one document (i.e., clause) d, a is the number of clauses in category C_i containing this term, c is the number of clauses of all other categories containing this term, and the upper bound of i represents the total number of categories.

$$\mathrm{RF} = \frac{\text{DF in positive category containing T}}{\text{DF in negative categoriy containing T}} \tag{3.2}$$

$$\mathrm{RF_M} = \frac{\text{DF in category C containing T}}{\text{DF in all categories except C containing T}} \tag{3.3}$$

$$\mathrm{TFRF} = \mathrm{TF} * \mathrm{RF} = \mathrm{TF} * \frac{\text{DF in positive category containing T}}{\text{DF in negative categoriy containing T}} \tag{3.4}$$

$$\mathrm{TFRF_M} = \mathrm{TF} * \mathrm{RF_M} = \mathrm{TF} * \frac{\text{DF in category C containing T}}{\text{DF in all categories except C containing T}} \tag{3.5}$$

$$_{\max}\mathrm{RF_M} = \text{maximum of set } \left(\frac{\text{DF in category } C_j \text{ containing T}}{\text{DF in all categories except } C_j \text{ containing T}}\right) \tag{3.6}$$

$$\mathrm{TF}\ _{\max}\mathrm{RF_M} = \mathrm{TF} * \ _{\max}\mathrm{RF_M} = \mathrm{TF} * \text{maximum of set } \left(\frac{\text{DF in category } C_j \text{ containing T}}{\text{DF in all categories except } C_j \text{ containing T}}\right) \tag{3.7}$$

$$\mathrm{TFRF_{M_{c_i}}} = \log(\mathrm{tf_d} + 1) * \log\left(10 + \frac{a}{c}\right) \tag{3.8}$$

$$\mathrm{TF_{max_{C_i}}RF_M} = \log(\mathrm{tf_d} + 1) * \max_{C_i}\left[\log\left(10 + \frac{a}{c}\right)\right] \tag{3.9}$$

A preprocessing program for executing the above-mentioned data preprocessing subtasks was coded in Python programming language. The input to the program are raw text files (collected clauses in .txt format), and the output are two datasets (training dataset and testing dataset) in the Library for Support Vector Machine (LIBSVM) format. Each line in the training and testing dataset files represents one clause in feature-numeric value pairs and its corresponding topic serial

number (topics are numbered from 1 to 10). Since the number of clauses for each topic varies very differently (from about 30 to 180, see Figure 3.4), the input files (1,215 collected clauses) were randomly split into training and testing datasets, with a ratio of 2:1, respectively, in order to avoid a very small testing dataset size. Since one classifier needs to be built for each class, the program was implemented ten times to obtain ten pairs of training and testing datasets. These training and testing datasets were used for classifier training and performance evaluation, respectively.

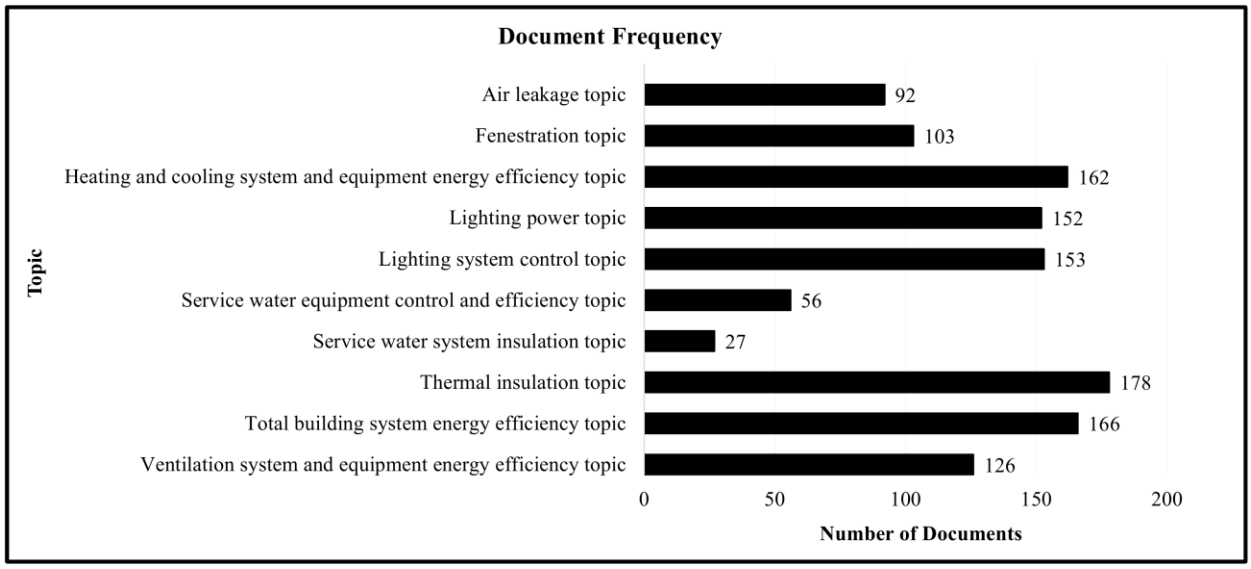


Figure 3.4. Document frequency of topics

3.1.2.3.2 ML Algorithm Selection and Implementation

A variety of ML algorithms have shown reasonable performance in TC. However, no single algorithm has demonstrated to consistently outperform the others across various applications and domains (Sebastiani 2002). In this research, ten popular ML algorithms were tested, including SVM (implemented in both linear and rbf kernel), DT (implemented by CART algorithm), NB (implemented by three variances of algorithms: Gaussian NB, Multinomial NB, Bernoulli NB),

kNN, Radius-based Neighbors (RBN), Nearest Centroid (NC), RF and GBRT (Aggarwal and Zhai 2012; Breiman 2001; Friedman 2001).

Each algorithm has some important parameters that were tuned/optimized by trial and error based on experimental results. Tuning/optimizing parameters refers to the process of looking for the best parameters to maximize the performance. Experimental results refer to the performance yielded when the parameters are tested. For example, parameter C in SVM with linear kernel can control the weight of positive and negative clauses as the number of them could be unbalanced, which may influence the performance of the classifier. To tune the parameter C in SVM, an initial range of values (e.g., 10^{-3}, 10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}, 10^{3} etc.) was tested to identify the approximate magnitude of C. Then, a range of specific values (e.g., 10^{-1} - 1, etc.) in that magnitude was tested to identify the approximately-best C value. The above testing steps were implemented using loops in the Python programming language. The above-mentioned ML algorithms were implemented using the Scikit-Learn ML algorithm(s) package written in Python programming language (Pedregosa et al. 2011). The parameters of each algorithm were tuned/optimized to find the closest-to-best parameters that result in the highest performance. Closest-to-best parameters are "good-enough", because it is infeasible to enumerate all possible values to find the exactly-best parameters (like finding the exact value of π).

3.1.2.4 Classification Result Evaluation

The performance of the above-mentioned ML algorithms was evaluated using recall and precision,

as per Equations 3.10 and 3.11, where true positive (TP) refers to the number of clauses labelled correctly as positive, false positive (FP) refers to the number of clauses labelled incorrectly as positive, and false negative (FN) refers to the number of clauses labelled incorrectly as negative. For this application, recall is more important than precision, because missing to recall one clause means overlooking a relevant clause, which may affect the performance of the ACC system as a whole. Precision is not as critical, since irrelevant text could be filtered out during further IE.

In addition, confusion matrix (CM) was used to analyze the results. CM is a very useful tool to analyze the performance of classifiers (Manning et al. 2009). It is a number-of-classes × number-of-classes matrix, in which the diagonal shows how many testing clauses are labelled correctly and other positions show how many testing clauses are misclassified from one class to another class. CM can thus help reveal misclassification causes like human errors in labelling.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{3.10}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{3.11}$$

3.1.2.5 Performance Improvement

Initial testing and evaluation was conducted without implementing any performance improvement strategies, in order to establish the baseline for comparison. Feature selection and recursive stopword removal were then implemented to explore their effect on improving the performance.

3.1.2.5.1 Feature Selection

The effect of two main approaches of feature selection, UFS and RFS, on performance improvement was empirically tested.

As mentioned in the Section 2.2.3, common feature scoring functions used for UFS include CHI, IG, and MI. Although there is no systematical performance difference among these three feature scoring functions, CHI tends to select those more low-frequency features/words (Aggarwal and Zhai 2012; Manning et al. 2009). Since in this application the text is characterized by a relatively small number of features, some low-frequency features may be significant in identifying the class. Therefore, CHI scoring function was used for testing UFS. After scoring and ranking the features, two types of feature selection methods were used: (1) K-best: A K number of features are selected; and (2) Percentage: A certain percentage of features are selected. Fifty (50) to 3,000 features were tested for selecting K, with a 50-feature increasing step size; and 3% to 75% were tested for selecting the percentage, with a 3% increasing step size.

For testing RFS, the best performing ML algorithm (defined in Section 3.1.2.5, as further discussed in Section 3.1.3) was used to assign the weights and different combinations of M and N were tested to select the best feature set.

3.1.2.5.2 Recursive Domain-Specific Stopword Removal

For each topic, a domain-specific stopword list was created and tested in a recursive manner. As

mentioned in Section 3.1.2.3, the standard English stopword list was initially used to remove those high-frequency but low-content words (e.g., “a” and “the”), which tend to be non-discriminative for general text. Stopwords are commonly removed using a standard stopword list. However, a domain-specific stopword list might be more descriptive of a specific domain (or subdomain). Domain-specific stopwords are those words which have no discriminative power within a specific domain or context (Makrehchi and Kamel 2008). Since construction domain stopword lists are not available, for each topic, a list was created by manually adding domain-specific, non-discriminative words (such as “include”, “allow”, and “install”) to the original standard English stopword list, in a recursive manner. All words were counted and then high-frequency, low-content words were identified based on domain knowledge and using trial and error.

After removing the stopwords using the general stopword list, the remaining total number of distinct words were around 4,000. After sorting these words according to their term frequencies in the whole document collection, it was found that the term frequencies decreased from the levels of 5,000 to 1,000, 1,000 to 500, and 500 to 400 for the first 25, second 25, and third 50 words, respectively. This means that the term frequencies of the remaining 3,900 words were all below 400. Because there is no benchmark to indicate the cut-off term frequency except using trial and error (Manning et al. 2009; Rijsbergen 1979), in this research, those top-100 term frequency words were considered as potential stopwords. These 100 words were then checked and classified into two groups: (1) words that are discriminative of specific topics and thus should be excluded from

the stopword list (e.g., “control” is highly related to the “lighting control topic”, although it appeared over 3,300 times in the dataset), and (2) words that are non-discriminative of any topic and thus are good potential candidate stopwords (e.g., “include” appeared over 1,300 times in the dataset, and is non-discriminative and non-predictive of any topic). To determine the final stopwords, these non-discriminative words (i.e., words in the second group, such as “include, “allow”, “according”, “foot”, “addition”, “install”, “function”, “percent”, etc.) were tested one by one for each topic: if removing a word from the features improved performance for a topic, it was added to the domain-specific stopword list of that topic.

3.1.3 Experimental Results and Analysis

The experiments were conducted in a performance-boosting manner; for each step, the technique that yielded the best performance was optimized and selected. The final combination of techniques that were selected for all steps forms the best TC algorithm.

3.1.3.1 Performance of Different of ML Algorithms

The best performance result of each of the ten tested algorithms for each category is summarized in Table 3.1. kNN, RF, and SVM showed the top three recall results with 91.60%, 91.50%, and 89.90% recall values, respectively. They were selected for further comparative evaluation, after implementing feature selection, for the following reasons: (1) RF inherently implements partial feature selection due to its internal algorithm design. So, an “apple-to-apple” comparison requires further comparison after implementing feature selection for kNN and SVM; and (2) the three

algorithms have yielded similar much higher average recall and precision with the least standard deviation, in comparison to the rest of the algorithms. So, there was no need to further evaluate the other algorithms. The results of the comparative evaluation are shown in Table 3.2. SVM (with linear kernel) was selected as the optimal algorithm for further performance improvement because it showed relatively robust performance in terms of average and standard deviation of recall and precision. Although the recall of kNN is 0.2% higher than that of SVM, it comes at a high precision cost (over 10% reduction in precision).

The relative high performance results of SVM could be explained by the following reasons: (1) SVM is especially suitable for handling environmental regulatory text, because environmental topics can usually be represented by a small set of key, discriminative features (e.g., "fenestration topic" achieved 100% recall and 82% precision using only 100 features); and (2) these key features usually occur together (e.g., "service", "water", "heatingD", and "control" occurred together for the "service water equipment control and efficiency topic"). These properties enable those support vectors to be easily identified for classification, which helps reduce FN errors, thereby improving recall. The relative high performance of SVM with a linear kernel might also indicate that environmental regulatory text does not contain as much ambiguity as other types of text which would require more complex nonlinear kernels (e.g., rbf, polynomial) for classification.

The use of SVM is also consistent with recent TC research studies in the construction domain which used SVM, such as in classifying contract documents (Salama and El-Gohary 2013b),

project correspondences and meeting minutes (Mahfouz 2011), and safety documents like the U.S. Occupational Safety and Health Administration (OSHA) standards (Chi et al. 2014).

Table 3.1. Performance of Different ML Algorithms (Before Feature Selection)

Topic	Performance of ML algorithm																			
	SVM		DT		Gaussian NB		Multinomial NB		Bernoulli NB		kNN		RBN		NC		RF		GBRT	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Air leakage topic	87%	90%	85%	76%	70%	72%	95%	62%	86%	62%	85%	90%	96%	38%	84%	91%	89%	81%	77%	85%
Fenestration topic	84%	90%	89%	72%	72%	83%	85%	86%	81%	60%	84%	91%	96%	43%	85%	86%	89%	93%	91%	90%
Heating and cooling system and equipment energy efficiency topic	87%	89%	39%	83%	82%	77%	80%	82%	73%	83%	84%	83%	83%	11%	79%	87%	82%	85%	92%	75%
Lighting power topic	89%	94%	34%	96%	78%	86%	79%	96%	86%	81%	85%	95%	80%	40%	89%	89%	87%	96%	80%	93%
Lighting system control topic	82%	95%	65%	95%	80%	84%	75%	95%	82%	91%	79%	96%	86%	47%	79%	91%	74%	99%	80%	95%
Service water equipment control and efficiency topic	74%	80%	64%	84%	58%	60%	64%	36%	75%	12%	71%	88%	75%	12%	71%	80%	82%	92%	82%	72%
Service water system insulation topic	76%	100%	71%	92%	67%	77%	0%	0%	75%	23%	68%	100%	100%	46%	69%	85%	92%	92%	71%	92%
Thermal insulation topic	98%	89%	51%	93%	88%	80%	59%	97%	31%	98%	92%	94%	21%	100%	90%	90%	84%	97%	90%	93%
Total building system energy efficiency topic	91%	89%	62%	67%	76%	74%	66%	91%	81%	75%	90%	92%	82%	18%	85%	88%	90%	91%	79%	84%
Ventilation system and equipment energy efficiency topic	91%	83%	84%	76%	79%	74%	91%	81%	93%	69%	92%	87%	100%	20%	89%	81%	91%	89%	91%	88%
Average	**85.9%**	**89.9%**	**64.4%**	**83.4%**	**75.0%**	**76.7%**	**69.4%**	**72.6%**	**76.3%**	**65.4%**	**83.0%**	**91.6%**	**81.9%**	**37.5%**	**82.0%**	**86.8%**	**86.0%**	**91.5%**	**83.3%**	**86.7%**
Standard deviation	**7.22%**	**5.70%**	**18.89%**	**10.37%**	**8.54%**	**7.50%**	**27.01%**	**31.70%**	**17.03%**	**28.02%**	**8.21%**	**4.93%**	**23.17%**	**26.13%**	**7.39%**	**3.88%**	**5.54%**	**5.46%**	**7.24%**	**7.83%**

Table 3.2. Performance of Different ML Algorithms (After Feature Selection)

Topic	Performance of ML algorithm					
	SVM		kNN		RF	
	Precision	Recall	Precision	Recall	Precision	Recall
Air leakage topic	86%	93%	71%	88%	89%	81%
Fenestration topic	84%	93%	90%	93%	89%	93%
Heating and cooling system and equipment energy efficiency topic	86%	90%	82%	86%	82%	85%
Lighting power topic	85%	97%	44%	100%	87%	96%
Lighting system control topic	84%	97%	54%	99%	74%	99%
Service water equipment control and efficiency topic	77%	96%	82%	92%	82%	92%
Service water system insulation topic	76%	100%	100%	100%	92%	92%
Thermal insulation topic	95%	94%	75%	98%	84%	97%
Total building system energy efficiency topic	88%	92%	56%	97%	90%	91%
Ventilation system and equipment energy efficiency topic	91%	83%	94%	84%	91%	89%
Average	**85.2%**	**93.5%**	**74.8%**	**93.7%**	**86.0%**	**91.5%**
Standard deviation	**5.71%**	**4.70%**	**18.52%**	**6.02%**	**5.54%**	**5.46%**

3.1.3.2 Performance of Different Text Representation Models

As shown in Table 3.3, the Bigram model showed zero precision and recall for half of the topics, indicating that capturing semantic information of environmental regulatory text statistically in terms of word positions in a sentence (as in the Bigram model) results in a much decreased performance in comparison to the use of unordered words (as in the BOW model). These results are similar to those reported in other domains and applications (e.g., classifying news articles and medical abstract) that show that the BOW model could perform better than the Bigram model despite the fact that it discards all word association information (Moschitti and Basili 2004). This can be attributed to the following reason: individual relevant features in the BOW model may

become irrelevant when associated and combined as new features in the Bigram model (Boulis and Ostendorf 2005). Because the relatively small number of features in environmental regulatory text may make the majority of new features in the Bigram model unique, during term weighting, these unique features would not contribute to the differentiation of topics. Accordingly, the BOW model was empirically selected for text representation.

Table 3.3. Performance of Different Text Representation Models

Topic	Performance of text representation model (no feature selection, SVM with linear kernel)					
	BOW			Bigram		
	C[1]	Precision	Recall	C[1]	Precision	Recall
Air leakage topic	1	87%	90%	0.4	26%	13%
Fenestration topic	0.8	84%	90%	150	0%	0%
Heating and cooling system and equipment energy efficiency topic	2	87%	89%	0.1	0%	0%
Lighting power topic	1	89%	94%	0.3	28%	48%
Lighting system control topic	0.7	82%	95%	0.1	0%	0%
Service water equipment control and efficiency topic	2	74%	80%	0.2	0%	0%
Service water system insulation topic	1	76%	100%	0.1	0%	0%
Thermal insulation topic	3	98%	89%	9	13%	31%
Total building system energy efficiency topic	2	91%	89%	9	15%	12%
Ventilation system and equipment energy efficiency topic	0.7	91%	83%	40	17%	2%

[1]C is a penalty parameter used in SVM that adjusts the data unbalance problem.

3.1.3.3 Performance of Different Term Weighting Schemes

TFIDF, $TFRF_M$, and $TF_{max}RF_M$ weighting schemes were tested using the previously selected BOW model and SVM algorithm. The performance results are shown in Table 3.4. In order to ensure

that the performance of different weighting schemes is not affected by feature selection, comparative experiments were conducted for all weighting schemes both with and without feature selection. Although different weighting schemes show different performances for different topics, only one single term weighting scheme must be selected for all topics to avoid the use of multiple term weighting schemes in one classifier. Thus, the average recall of all topics and the corresponding standard deviation (SD) were used for weighting scheme selection. Two selection criteria were used: (1) highest recall, and (2) lowest SD, which indicates robust performance across topics. Prior to feature selection, TFIDF achieved the best average recall of 89.9% with the least SD of 5.7%, compared with 89% and 85.3% recall and 6.88% and 8.25% SD for $TF_{max}RF_M$ and $TFRF_M$, respectively. After feature selection, TFIDF still outperformed in terms of recall (93.5%), but without achieving the least SD (4.7%). $TF_{max}RF_M$ achieved 92.1% recall and lowest SD of 3.96%, while $TFRF_M$ still yielded the lowest recall of 91.1% and highest SD of 5.59%. Accordingly, TFIDF was selected as the optimal term weighting scheme because of the desired high recall.

Additionally, the following three observations were made. First, $TF_{max}RF_M$ consistently outperformed $TFRF_M$ in terms of both the average and the standard deviation of recall and precision. These findings are similar to those reported in the news domain (as tested on the "20 Newsgroups" and "Reuters News" datasets) (Xuan and Quang 2014). Second, $TF_{max}RF_M$ outperformed TFIDF in terms of precision. Not only it yielded the best precision at seven out of

the ten topics, but it also yielded the best average precision among the three weighting schemes. Third, feature selection did not affect the recall ordering of the three weighting schemes.

Table 3.4. Performance of Different Term Weighting Schemes (Before and After Feature Selection)

Topic	Performance of term weighting scheme											
	TFIDF				$TF_{max}RF_M$				$TFRF_M$			
	Before FS*		After FS*		Before FS*		After FS*		Before FS*		After FS*	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Air leakage topic	87%	90%	86%	93%	97%	97%	97%	97%	89%	78%	90%	84%
Fenestration topic	84%	90%	84%	93%	95%	86%	91%	93%	93%	87%	96%	96%
Heating and cooling system and equipment energy efficiency topic	87%	89%	86%	90%	77%	86%	79%	88%	83%	80%	83%	83%
Lighting power topic	89%	94%	85%	97%	89%	91%	89%	93%	87%	83%	87%	85%
Lighting system control topic	82%	95%	84%	97%	90%	83%	92%	85%	96%	92%	96%	92%
Service water equipment control and efficiency topic	74%	80%	77%	96%	75%	75%	78%	90%	75%	71%	76%	94%
Service water system insulation topic	76%	100%	76%	100%	82%	90%	82%	90%	58%	100%	64%	100%
Thermal insulation topic	98%	89%	95%	94%	88%	98%	91%	98%	88%	89%	90%	91%
Total building system energy efficiency topic	91%	89%	88%	92%	88%	90%	92%	93%	91%	91%	91%	91%
Ventilation system and equipment energy efficiency topic	91%	83%	91%	83%	94%	94%	94%	94%	80%	82%	84%	95%
Average	**85.9%**	**89.9%**	**85.2%**	**93.5%**	**87.5%**	**89.0%**	**88.5%**	**92.1%**	**84.0%**	**85.3%**	**85.7%**	**91.1%**
Standard deviation	**7.22%**	**5.70%**	**5.71%**	**4.70%**	**7.41%**	**6.88%**	**6.52%**	**3.96%**	**11.05%**	**8.25%**	**9.74%**	**5.59%**

*FS =Feature selection

3.1.3.4 Baseline Performance

Prior to the implementation of performance improvement strategies, the initially selected combination of techniques (BOW model, TFIDF weighting, and SVM algorithm with linear kernel) was used as a baseline for comparison. The results before and after implementing the improvement strategies are summarized in Table 3.5. Table 3.5 shows the corresponding best parameters found when the highest recall is achieved. For the baseline condition, an average performance of 89.9% and 85.9% recall and precision, respectively, was achieved.

Table 3.5. Effects of Feature Selection and Recursive Domain-Specific Stopword Removal on Performance

Topic	Performance before and after using feature selection and domain-specific stopword removal													
	No feature selection, SVM with linear kernel			Feature selection using K best, SVM with linear kernel					Feature selection using K-best, domain-specific stopword removal, SVM with linear kernel					
	C[1]	Precision	Recall	C[1]	K[2]	Precision	Recall	Increase in Recall[3]	C[1]	K[2]	Stopwords[4]	Precision	Recall	Increase in Recall[3]
Air leakage topic	1	87%	90%	2	750	86%	93%	+3%	4	700	allow, include	82%	98%	+5%
Fenestration topic	0.8	84%	90%	9	350	84%	93%	+3%	6	100	allow, include, according	82%	100%	+7%
Heating and cooling system and equipment energy efficiency topic	2	87%	89%	2	2,050	86%	90%	+1%	2	1,100	allow, include, section, area, multiply	83%	97%	+7%
Lighting power topic	1	89%	94%	5	1,150	85%	97%	+3%	5	1,150	N/A	85%	97%	+0%
Lighting system control topic	0.7	82%	95%	2	2,850	84%	97%	+2%	2	2,850	N/A	84%	97%	+0%
Service water equipment control and efficiency topic	2	74%	80%	9	550	77%	96%	+16%	9	550	N/A	77%	96%	+0%
Service water system insulation topic	1	76%	100%	7	650	76%	100%	+0%	5	700	according, addition	88%	100%	+0%
Thermal insulation topic	3	98%	89%	6	850	95%	94%	+5%	6	850	N/A	95%	94%	+0%
Total building system energy efficiency topic	2	91%	89%	3	1,000	88%	92%	+3%	7	550	include, install, foot	83%	97%	+5%
Ventilation system and equipment energy efficiency topic	0.7	91%	83%	0.7	2,500	91%	83%	+0%	2	1,450	include, allow, percent	84%	97%	+14%
Average		**85.9%**	**89.9%**			**85.2%**	**93.5%**	**+3.6%**				**84.3%**	**97.3%**	**+3.8%**
Standard deviation		**7.22%**	**5.7%**			**5.71%**	**4.7%**	**-1.01%**				**4.67%**	**1.77%**	**-2.93%**

Note:

[1]**C** is a penalty parameter used in SVM that adjusts the data unbalance problem.

[2]**K** means the number of features selected to achieve the best performance, at a unit of 50.

[3]Shows percentage increase in recall compared to previous performance.

[4]**N/A** means there are no effective stopwords found to improve performance.

3.1.3.5 Effect of Feature Selection

CHI and K-best were empirically selected as the optimal methods for feature selection, because based on the experimental results they together outperformed RFS as well as CHI and percentage feature selection. Based on the results, feature selection has shown to be effective in improving the performance in terms of average recall (see Table 3.5). The average recall and precision have reached 93.5% and 85.2%, respectively. The results also demonstrate the expected trend that an increase in recall (3.6% in this case) decreases precision (0.7% in this case). The highest improvement was achieved for the "service water equipment control and efficiency topic", at a 16% increase in recall (reaching 96% recall) using 550 features. Only one of the ten topics did not show any improvement in recall ("ventilation system and equipment energy efficiency topic"), which indicates that feature selection does not necessarily improve recall for all classes.

The number of features selected at the highest recall (K value shown in Table 3.5) provides some insight about the differences across classes. For example, on one hand, the "fenestration topic" used 350 features only to achieve maximum recall, which indicates that clauses belonging to the "fenestration topic" can be easily classified. On the other hand, the "ventilation system and equipment energy efficiency topic" used 2,500 features but still gained no improvement in recall, which indicates that clauses belonging to this topic are harder to differentiate. Similarly, other topics using a relatively large K value (relative to other topics in this dataset) to achieve best recall showed less recall improvement. For example, the "heating and cooling system and equipment energy efficiency topic" used 2,050 features, but achieved only 1% recall increase. In addition, a

relatively large K value may also imply that potential subtopics may exist for that topic, thereby needing to aggregate more features from each subtopic to better represent their parent topic. Overall, none of the ten topics used more than 2,900 features to achieve its best recall. This shows the effectiveness of feature selection in enhancing recall, even if the original feature size is relatively small.

3.1.3.6 Effect of Recursive Domain-Specific Stopword Removal

The performance was significantly improved after using the proposed domain-specific stopword lists (see Table 3.5). This indicates that the use of domain-specific text characteristics is effective in improving the performance of classification. The final performance shows an average 97.3% and 84.3% recall and precision, respectively. The average recall increased by 3.8% at the expense of a 0.9% decrease in precision. The standard deviation of both recall and precision continued to decrease and finally dropped to 1.77% and 4.67%, respectively, which indicates that the proposed TC algorithm is relatively robust on all topics. The results also show the following two findings. First, a change of stopwords caused a variation in the best parameters. Taking the parameter K as an illustration, all those topics that achieved improvement in terms of recall used fewer features (e.g., the “ventilation system and equipment energy efficiency topic” used 1,450 features instead of 2,500 features to gain 14% increase in recall), while topics that gained improvement in terms of precision used more features (e.g., the “service water system insulation topic” used 50 more features to reach 12% increase in precision meanwhile still maintaining 100% recall). This observation may also substantiate the counteractive recall-precision relationship in the aspect of

number of features: selecting more features may help identify more feature differences among topics which reduces FP errors, but may increase FN errors thereby undermining the recall, and vice versa. Second, stopwords varied across classes. This indicates that different topics (and subtopics) may require different stopword lists.

3.1.3.7 Final Performance and Error Analysis

An error analysis was conducted to identify the sources of errors for the final performance. Precision errors come from incorrectly assigning false labels to some clauses. For example, for the following clause, in addition to the correct label "thermal insulation topic", the label "heating and cooling system and equipment energy efficiency topic" was incorrectly assigned to the clause: "6.4.4.1.5 Radiant Floor Heating. The bottom surfaces of floor structures incorporating radiant heating shall be insulated with a minimum of R-3.5. Adjacent envelope insulation counts toward this requirement. Exception: Requirements for heated slab-on-grade floors incorporating radiant heating are in Chapter 5." (ASHRAE 2010). This is probably because the clause contains "heating" four times, which is a representative feature of the label "heating and cooling system and equipment energy efficiency topic".

Recall errors come from missing assignment of correct labels to some clauses. For example, in the following clause, the correct label "air leakage topic" was assigned, but another correct label "thermal insulation topic" was missing: "C403.2.7.1.3 High pressure duct systems. Ducts designed to operate at static pressures in excess of 3 inches water gauge (w.g.) (750 Pa) shall be insulated

and sealed in accordance with Section C403.2.7. In addition, ducts and plenums shall be leak tested in accordance with the SMACNA HVAC Air Duct Leakage Test Manual with the rate of air leakage (CL) less than or equal to 6.0 as determined in accordance with Equation 4-5." (ICC 2012). This is probably because the representative features (e.g., "sealed", "leak", "leakage") of the "air leakage topic" dominated those of the "thermal insulation topic" (e.g., "insulated") in terms of term frequency.

3.2 Ontology-Based Multilabel Text Classification of Construction Regulatory Documents

3.2.1 Comparison to the State of the Art

ML techniques have commonly been used for TC (e.g., Caldas et al. 2002; Kovacevic et al. 2008; Mahfouz 2011; Salama and El-Gohary 2013b). While generally successful, non-ontology-based ML-based TC usually discards semantic text information (e.g., meaning of words) although it is potentially very useful in identifying the correct label(s) of a document. Some non-ontology-based ML-based TC algorithms try to partially and indirectly capture some semantic text information (e.g., using Bigram model to capture relationships of adjacent words in a sentence in terms of conditional probability). However, the probabilistic and statistical methods usually achieve unsatisfactory performance in capturing the semantics of the text (see Section 3.1). Semantic-based TC has, thus, been introduced to capture and take advantage of the semantics of the text for improving the TC performance. The use of ontologies in TC has, therefore, recently attracted much research effort.

In this regard, two main research gaps are identified. First, there has been no research efforts for using ontology-based TC in the construction domain. This is a lost opportunity for exploring the use of domain semantics to improve the performance of TC-based applications in construction. Second, outside of the construction domain, ontology-based TC efforts: (1) rely on supervised ML for training the classifier – using labelled training data – to learn the rules for labelling any given text (e.g., Vogrinčič and Bosnić 2011; Lee et al. 2009; He et al. 2004). This involves much manual effort in labelling the training data; (2) can only deal with single-label classification problems (e.g., Yang et al. 2008; Wei et al. 2006; Yu et al. 2006; Song et al. 2005; He et al. 2004) or are unable to deal with a multilabel TC problem directly (e.g., Waraporn et al. 2010). This requires transformation to multiple single-label problems; and/or (3) show inconsistent results for ontology-based TC in comparison with non-ontology-based ML-based TC. Some efforts (e.g., Fang et al. 2007) compared ontology-based TC with SVM-ML-based TC and showed that SVM-ML-based TC outperformed. Some efforts (e.g., Yang et al. 2008; Song et al. 2005; He et al. 2004) compared ontology-based TC with multiple ML algorithms for TC and showed that ontology-based TC outperformed only some of these ML algorithms [e.g., only Naïve Bayes in Yang et al. (2008)]. Other efforts (e.g., Yu et al. 2006) showed that the ontology-based approach outperformed the non-ontology-based ML-based approach using multiple algorithms like NB, kNN and SVM, but only reported enhanced performance in terms of precision (Yu et al. 2006). These inconsistent results indicate that there is no single outperforming ontology-based method/algorithm, and, thus, that it is difficult to reuse an existing ontology-based TC algorithm from one domain to the other.

3.2.2 Proposed Method for Ontology-Based Text Classification of Energy Regulatory Documents

To address the aforementioned knowledge gaps, this research proposes an ontology-based multilabel TC for classifying energy regulatory documents (including energy codes) and contract specifications for supporting EnergyACC in construction. A domain ontology was developed for representing the hierarchy of environmental topics and the concepts and relationships associated with each topic. An unsupervised deep learning technique was used to learn the similarities between each clause (based on the terms in the clause) and each topic (based on the ontological concepts related to this topic) for classifying each clause into zero or more topics according to two experimentally set similarity thresholds. Specifically, a variant of three-layer feedforward neural network algorithm, the hierarchical softmax skip-gram, was used to learn the distributed representation of terms and concepts in terms of real-valued vectors, and the similarities of such terms and concepts could be computed based on the cosine distance of their vectors. This hierarchical softmax skip-gram algorithm was selected because of its computational efficiency and accuracy on large datasets.

A four-phase methodology for ontology-based domain-specific TC is proposed, as shown in Figure 3.5. The labels are defined based on a hierarchy of topics. For each topic, a subontology is built to model the concepts and relationships that are related to this topic. Then, a deep learning algorithm is applied to learn the similarities between each clause (based on the terms in the clause) and each topic (based on the ontological concepts related to this topic) for classifying each clause into zero

or more topics.

In comparison to existing ontology-based TC methods, the proposed method is different in three primary ways. First, instead of using supervised ML (e.g., He et al. 2004; Lee et al. 2009; Vogrinčič and Bosnić 2011; Wijewickrema and Gamage 2013) for training the classifier to learn the rules of labelling, unsupervised ML is used for learning the semantic similarity between a term of a clause and a concept related to a topic from a set of training clauses. As a result, only the testing data are labelled, which saves much manual effort that would have been required to label the training data. Second, instead of using a problem transformation approach (e.g., He et al. 2004; Lee et al. 2009; Wijewickrema and Gamage 2013), a direct multilabel ontology-based TC method is used. As a result, (1) only one pair of training and testing data needs to be prepared; and (2) only one classifier needs to be built. This reduces the data preparation and classifier building effort. Third, instead of using shallow learning (e.g., Lee et al. 2009), deep learning is used to better represent the complexity that exists in text semantics. This aims to enhance the performance of TC.

In comparison to the method in Section 3.1, in this ontology-based approach, (1) a document (or clause) is represented in terms of semantic concepts and relations, rather than just terms (words); (2) the multilabel classification problem is addressed in a direct way, instead of transforming the multilabel classification problem to multiple single-label classification problems (as commonly-used in ML-based TC); and (3) no human supervision is involved (i.e., training data are provided without labeling).

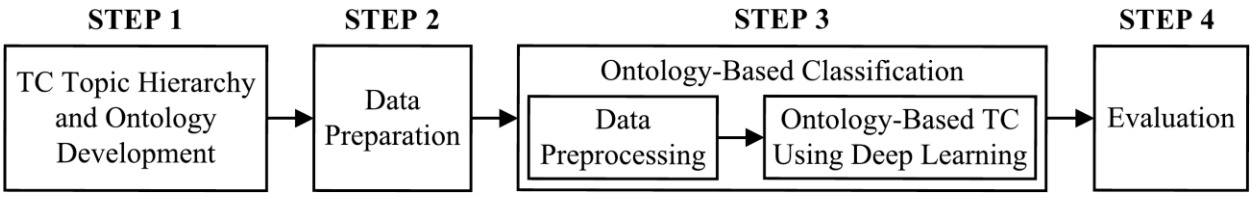


Figure 3.5. Proposed ontology-based text classification methodology

3.2.2.1 TC Topic Hierarchy and Ontology Development

A topic hierarchy was first developed to identify the labels that will be used for TC. In this research, the analysis is focused on the "energy efficiency topic", which is a subtopic of "environmental topic" (as per Figure 3.6). For developing the topic hierarchy, the established methodologies for taxonomy development (e.g., El-Gohary and El-Diraby 2010) were followed. The methodology includes two primary steps: (1) identification of the main concepts in the domain of interest: the concepts were extracted based on a review of the main relevant environmental regulatory documents (e.g., the 2012 International Energy Conservation Code and the 2010 California Energy Code); and (2) organization of the identified concepts into a hierarchy of concepts: the concepts were structured into a taxonomy using a combination of a top-down (starting by defining the most abstract concepts) and a bottom-up approach (starting by defining the most specific concepts). The "commercial building energy efficiency topic" subhierarchy is shown in Figure 3.6. Six of the ten leaf nodes (subtopics in the taxonomical topic hierarchy) were used as labels for classification.

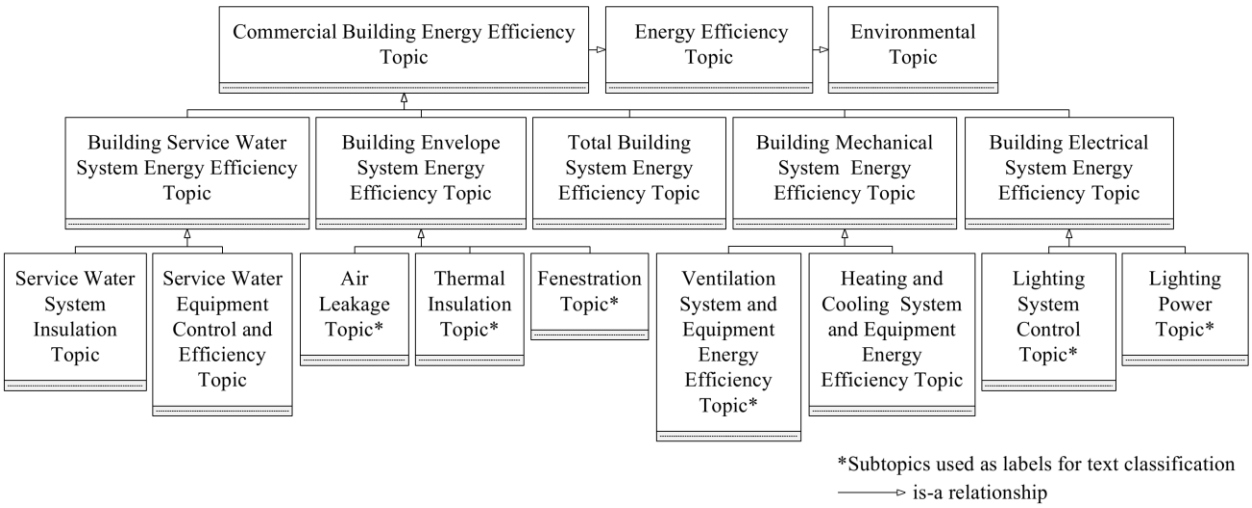


Figure 3.6. Text classification topic hierarchy

For modeling the semantic information associated with each topic, the hierarchy was extended into an application ontology. For each topic, a subontology was built to model the concepts and relationships that are related to this topic. For developing the ontology, the ontology development methodology by El-Gohary and El-Diraby (2010) was benchmarked. The main steps that were used to develop the ontology include: (1) purpose and scope definition: the purpose of the ontology is to support semantic TC and the scope is limited to "commercial building energy efficiency"; (2) taxonomy building: the same methodology as that used for building the TC topic hierarchy was followed (as described above). For the identification of the main concepts (the first step in taxonomy development), the scope was focused on identifying the main concepts that are related to each of the six leaf node concepts in the TC topic hierarchy. For example, the concepts related to the "lighting control topic" include "multilevel lighting control", "daylighting control", and "demand responsive control"; (3) relation modeling: the non-hierarchal relationships between concepts were identified and modeled to describe the semantic links between concepts. For

example, “is_controlled_by” links the concepts “luminaire control” and “motion sensor”; and (4) ontology coding: the concepts and relations were represented using Unified Modeling Language (UML) class diagrams. For example, Figure 3.7 and Figure 3.8 show the subontologies for the “lighting system control topic” and “lighting power topic”, respectively.

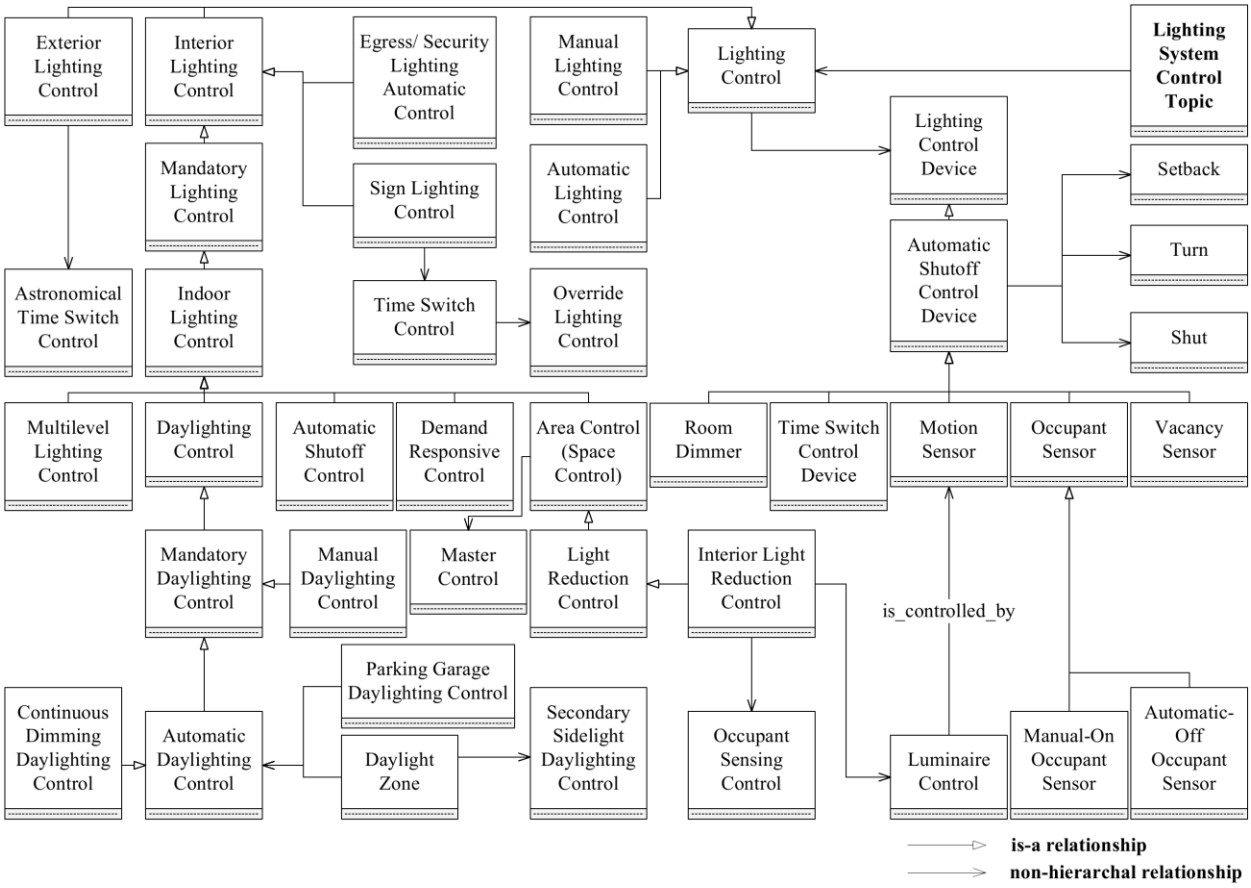


Figure 3.7. Partial subontology for “Lighting System Control Topic”

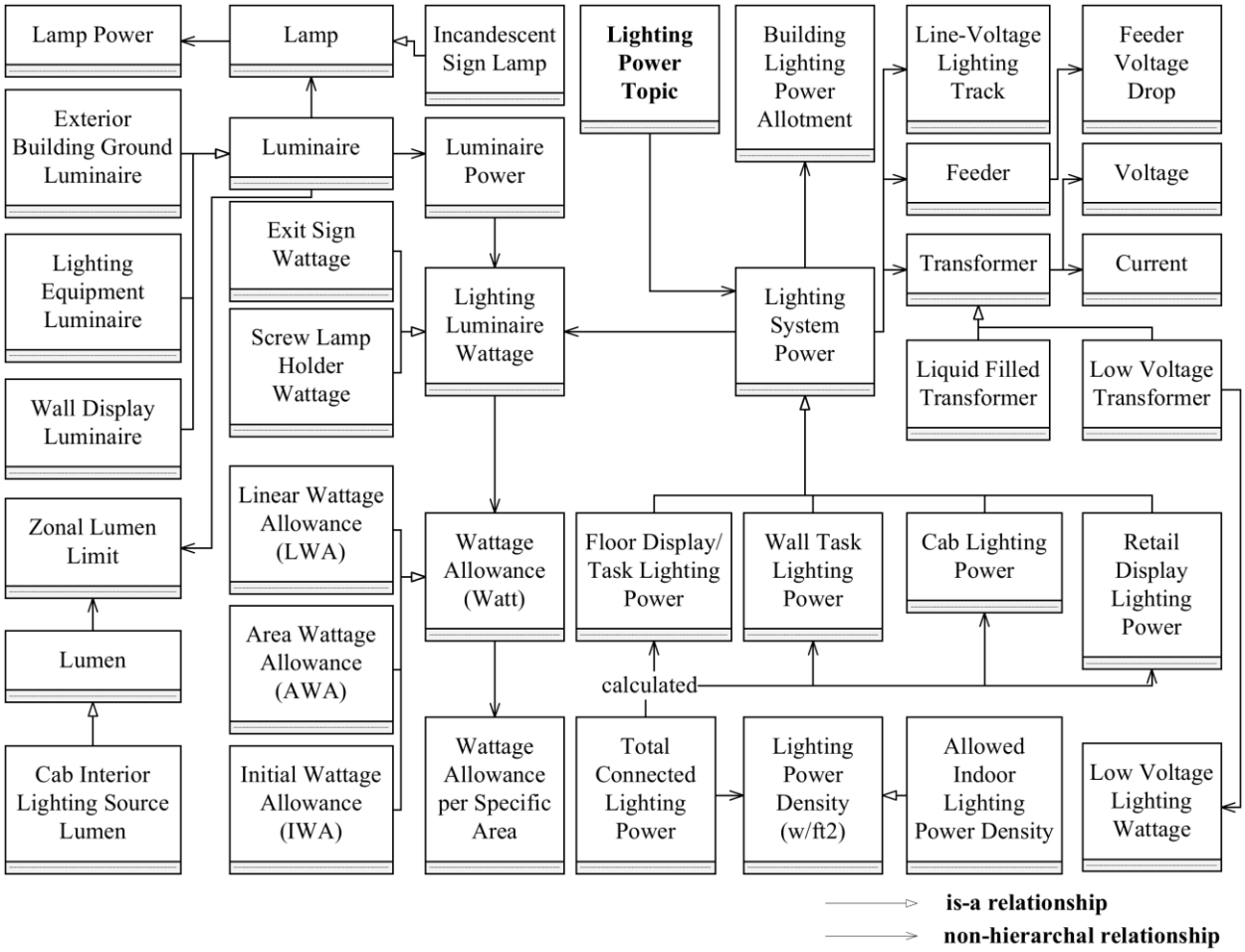


Figure 3.8. Partial subontology for "Lighting Power Topic"

3.2.2.2 Data Preparation

Around 2,400 clauses were collected from 25 regulatory documents (see Figure 3.9). The documents were manually selected because they all contain energy efficiency requirements for commercial buildings, which is the scope of this research. All original documents were downloaded in .pdf format. The clauses were then extracted manually from the documents and each clause was represented in a separate .txt format file. A clause is defined as a paragraph of text that contains at least one requirement. For example, the following is a clause that was extracted from the 2012 International Energy Conservation Code (ICC 2012): "C403.2.7.1.2 Medium-

pressure duct systems. All ducts and plenums designed to operate at a static pressure greater than 2 inches water gauge (w.g.) (500 Pa) but less than 3 inches w.g. (750 Pa) shall be insulated and sealed in accordance with Section C403.2.7. Pressure classifications specific to the duct system shall be clearly indicated on the construction documents in accordance with the International Mechanical Code".

In collecting the data (clauses), data sufficiency in terms of quantity and quality was taken into account. The performance of the proposed methodology highly depends on the quantity and quality of data (Mikolov et al. 2013a) due to the use of deep learning (in Section 3.2.2.3). The use of large quantities of data can facilitate accurate learning of semantics. Good quality data refers to the words in the sentences being logical and coherent. Based on the experimental results, the good performance results indicate that data are sufficient in terms of quantity and quality.

The collected clauses were split into two sets, a training set and a testing set, at a ratio 5:1. Because the proposed methodology uses unsupervised ML, only the testing set was manually labelled for use in performance evaluation (in Section 3.2.2.4). A gold standard (the labeled testing set) was manually developed. Based on content, the author assigned each testing clause zero or more of the six labels. The labeling was then checked by two other researchers. Full agreement on labeling was achieved.

Document
2012 International Energy Conservation Code
2010 California Energy Code
ANSI/ASHRAE/IES Standard 90.1-2010 Energy Standard for Buildings Except Low-Rise Residential Buildings
2013 Nonresidential Compliance Manual for the 2013 Building Energy Efficiency Standards
2007 National Green Building Standard, Chapter 7 Energy Efficiency
ANSI/ASHRAE/USGBC/IES Standard 189.1-2009 Standard for the Design of High-Performance Green Buildings Except Low-Rise Residential Buildings
2009 LEED Reference Guide for Green Building Design and Construction
2013 Energy Policy and Conservation Act, Section 342
Energy Independence and Security Act of 2007
2011 North American Fenestration Standard/Specification for Windows, Doors and Skylights
2008 District of Columbia Construction Code
2009 New Hampshire State Building Code
2011 Vermont Commercial Building Energy Standards
2007 Oregon Structural Specialty Code, Chapter 13 Energy Conservation
2010 Oregon Energy Efficiency Specialty Code
2009 Massachusetts Stretch Energy Code
2009 Virginia Energy Conservation Code
2010 Florida Building Code, Energy Conservation
2012 North Carolina Energy Conservation Code
2009 New Mexico Energy Conservation Code
2011 Houston Commercial Energy Conservation Code
2010 Energy Conservation Construction Code of New York State
2006 Phoenix Green Construction Code, Chapter 6 Energy Conservation, Efficiency and Atmospheric Quality
2006 International Energy Conservation Code as Amended by the City of Phoenix
2012 Washington State Energy Code

Figure 3.9. Regulatory document list

3.2.2.3 Ontology-Based Classification

3.2.2.3.1 Data Preprocessing

Data preprocessing is the process of transforming the raw text into the required format. Three steps of data preprocessing were implemented: (1) Tokenization: Tokenization aims to segment the text into words or tokens, meanwhile eliminating characters such as punctuation and transforming words to their lowercase form. For example, “building, Thermal Insulation” are tokenized to

"building thermal insulation"); (2) Stemming: Stemming aims to strip off word suffixes (in some cases prefixes) to its root or stem. For example, "insulation" and "insulated" can both be mapped to "insul". Stemming reduces the number of features by combining words sharing the same stem. It is usually effective in improving the performance of classification (e.g., reaching 5% gain in average precision for English) (Manning et al. 2009). In the proposed ontology-based methodology, stemming was implemented using the Porter2 stemming algorithm in Python programming language; and (3) Stopword removal: Stopwords refer to those high-frequency and low-content words that are not discriminative in classification like "am", "is", "a", "the", "of". Removing stopwords from a document can, thus, reveal the content-bearing, discriminative words. The similarity between the discriminative terms of a document and the concepts of each subontology can then be measured for classification. A Python preprocessing program was coded for implementing the above-mentioned three subtasks. The input to the program is two sets of raw text (.txt) files (training and testing sets, see Section 3.2.2.2), and the output is two pre-processed datasets (training dataset and testing dataset). The training and testing datasets are the input to the ontology-based TC algorithm (discussed in the Section 3.2.2.3.2).

3.2.2.3.2 Ontology-Based Text Classification Using Deep Learning: Proposed Method

The proposed classification method is similarity-based. After the datasets are preprocessed, a deep learning algorithm is applied on the training dataset to learn the distributed representations of terms and concepts for capturing the similarities between each term in a clause and each concept in the

ontology. Accordingly, the similarity between a clause and a topic is quantified. The assignment of a label to a clause is then determined based on similarity values and two experimentally-defined similarity thresholds.

The hierarchical softmax skip-gram algorithm is used for deep learning and computing the similarities between each term in a clause and each concept in the ontology; it was selected because of its best-reported performance (Mikolov et al. 2013a, b). The algorithm learns the distributed representations of terms (in the clauses that exist in the training data) and concepts (in the ontology) from training data. A distributed representation of a term (or concept) is a real-valued vector of features that characterize the meaning of the term (or concept). The feature vector includes syntactic features (e.g., morphological category like gender of noun) and semantic features (e.g., hypernymy relation like room-bedroom) (Mikolov et al. 2013b). After learning the distributed representations of terms and concepts, the similarity between each term in a testing clause (i.e., a clause in the testing dataset) and each concept in the ontology is measured by the cosine similarity of their vectors (Mikolov et al. 2013c). Cosine similarity measures the angle between vectors and a smaller angle indicates higher similarity (Harispe et al. 2013).

The similarities between each term in a testing clause and each concept related to a topic are then summed up for each clause-topic pair to compute the total similarity (TS) between a clause and a topic. All topics with a positive TS with a clause are selected as potential labels for that clause; and the topics with a negative TS are filtered out. Zero similarity is used as the threshold for

selecting potential labels by assuming that, in this application, any clause that is relevant to a topic must have a positive TS to that topic. Under this assumption, all true labels of a clause are a subset of its potential labels. The experimental results show that this assumption is valid in this application.

If there is exactly one topic with a positive TS with a clause, then that topic is assigned as the only label for that clause. If there are no topics with a positive TS with a clause, then no topics are assigned to that clause. If there are more than one topic with a positive TS with a clause, then the topic with the highest TS is assigned as the primary label for that clause (the corresponding topic is then referred to as a primary topic). In this case, based on the primary topic, two thresholds are used for assigning the remaining labels (referred to as secondary labels) for that clause: (1) the total similarity difference (TSD) (see Equation 3.12) between the primary topic and the other topic(s) (potential secondary topics) are measured. Then, a TSD threshold is used to further identify the secondary labels. Topics with a TSD equal to or less than the threshold are assigned to the clause as its secondary labels. The larger the TSD threshold, the more secondary labels are assigned to the clause. In this case, more true labels are likely to be recalled, but incorrect labels may also be assigned to the clause; and (2) the total similarity percentage difference (TSPD) (see Equation 3.13) between the primary topic and the potential secondary topic(s) are measured. Then, a TSPD threshold is used to further identify the secondary labels. Topics with a TSPD equal to or less than the threshold are assigned to the clause as its secondary labels. Both thresholds values

are set experimentally for maximizing the overall performance. TSD and TSPD values are calculated as per Equation 3.12 and Equation 3.13, where TSD_{sd} is the TSD of topic *s* for clause *d*, $TSPD_{sd}$ is the TSPD of topic *s* for clause *d*, TS_{pd} is the TS of topic *p* for clause *d*, and TS_{sd} is the TS of topic *s* for clause *d*, topic *s* is a potential secondary topic of clause *d*, and topic *p* is the primary topic of clause *d*.

$$\mathrm{TSD_{sd}} = \mathrm{TS_{pd}} - \mathrm{TS_{sd}} \tag{3.12}$$

$$\mathrm{TSPD_{sd}} = \frac{\mathrm{TS_{pd}} - \mathrm{TS_{sd}}}{\mathrm{TS_{pd}}} \tag{3.13}$$

In using both threshold values, two assumptions are made. First, the strength/weakness of relevance of a clause to multiple topics could be reflected and ordered by their corresponding TS values. The topic with the strongest relevance/highest TS is the primary label, and all other relevant topics are the secondary labels. Second, for a certain clause, the TS values of all relevant secondary topics should be closer to the TS value of the primary topic than those of irrelevant topics, thereby having these relevant topics falling in a certain TSPD range. The final labels assigned to a clause are selected based on one of the two threshold values, whichever is the strictest. Since each topic addresses a different aspect of energy efficiency, specific threshold values should be set for each topic.

An illustrative example showing label assignments for a given clause based on similarity and threshold values is provided in Table 3.6. The TS value of each topic was computed, as shown in Table 3.6. Accordingly, “air leakage topic” was assigned as the primary label of that clause,

because of its highest TS value (1723.4). Accordingly, the TSD and TSPD values of each potential secondary topic were computed, as shown in Table 3.6. For example, the “thermal insulation topic” has TSD and TSPD values of 396.5 and 23.0%, respectively. Based on these values, the “thermal insulation topic” was assigned as a secondary label of that clause because both of its TSD and TSPD values fall below the threshold values (486.9 and 37.8% TSD threshold and TSPD threshold, respectively). All other potential secondary topics were not assigned because they did not meet the threshold values.

Table 3.6. Example of Classifying a Testing Clause

Topic*	Total similarity*	Total similarity difference*	Total similarity percentage difference*	Total similarity difference threshold*	Total similarity percentage difference threshold*	Result*
Air leakage topic	1723.4	0.0	0.0%	486.9	37.8%	Assigned (primary)
Thermal insulation topic	1326.9	396.5	23.0%			Assigned (secondary)
Fenestration topic	406.0	1317.4	76.4%			Not assigned
Ventilation system and equipment energy efficiency topic	304.2	1419.2	82.4%			Not assigned
Lighting power topic	222.0	1501.4	87.1%			Not assigned
Lighting system control topic	48.7	1674.7	97.2%			Not assigned

***For the following clause:** "503.2.7 Duct and plenum insulation and sealing. All supply and return air ducts and plenums shall be insulated with a minimum of R-5 insulation when located inside the building thermal envelope and a minimum of R-8 insulation when located outside the building thermal envelope in accordance with Table 503.2.7. When located within a building envelope assembly, the duct or plenum shall be separated from the building exterior or unconditioned or exempt spaces by a minimum of R-8 insulation. Exceptions: 1. When located within equipment. 2. When the design temperature difference between the interior and exterior of the duct or plenum does not exceed 15 °F (8° C).

All ducts, air handlers and filter boxes shall be sealed in accordance with the Mechanical Code and SMACNA Method A." (Houston City Council 2011).

3.2.2.3.3 Implementation of the Proposed Method

To implement the proposed method, Generate Similar (Gensim) (Rehurek and Sojka 2010), a Python programming language version of the softmax skip-gram algorithm (by Mikolov et al. (2013b), was used for deep learning and for computing the similarities between each term in a testing clause and each concept in the ontology. The input to the tool includes (1) the training data,

split as a set of sentences as required by the tool, (2) testing clauses, and (3) ontology concepts related to each topic. Some input parameters that were experimentally set include: (1) the dimensionality of word vectors was set as 200; (2) the maximum distance between the current and predicted word in a sentence was set as two; (3) after removing stopwords, all remaining words have a frequency lower than five were ignored; and (4) two worker threads (a parameter that is related to the training speed of multicore machines) were used. For a more detailed description of these parameters, the readers are referred to Rehurek and Sojka (2010). The output of the tool is the similarity values between each term in a testing clause and each concept in the ontology.

Another program was developed to (1) compute the TS value between each testing clause and each topic based on the similarity values (from the previous program); (2) assign the primary labels of the testing clauses; (3) compute the TSD and TSPD values for each potential secondary topic of a testing clause; and (4) assign the secondary labels to the testing clauses based on whether their TSD and TSPD values meet the threshold values. The program was coded in Python.

3.2.2.4 Evaluation

Since the proposed ontology-based TC algorithm can deal with multilabel classification problems directly, multilabel classification evaluation metrics were used. Four types of evaluation metrics were utilized. Although the metrics are different, they all use redefined recall and precision measures to evaluate the overall performance. In general, recall measures the number of correctly predicted true labels [(true positive (tp)] as a percentage of total number of true labels [tp plus false

negative (fn)]; and precision measures the number of correctly predicted true labels (tp) as a percentage of total number of predicted labels [tp plus false positive (fp)]. Typically, there is a tradeoff between recall and precision, because the more true labels are recalled, the higher the risk of making precision errors. In the subject application, recall is given a higher priority than precision because missing to recall one relevant clause – and thus missing to check compliance of the project with this clause – might result in noncompliance detection errors.

Multilabel evaluation metrics can be categorized into two main types: example-based metrics and label-based metrics (Tsoumakas et al. 2010; Madjarov et al. 2012). Using example-based metrics, the performance (recall and precision) of classification for each test document (clause, in this research) is calculated, and the overall performance is obtained by calculating the mean performance over all test documents. Example-based recall and precision are calculated using Equation 3.14 and Equation 3.15 (Madjarov et al. 2012), where tp_i is the number of labels predicted correctly as positive for a testing document i, fp_i is the number of labels predicted incorrectly as positive for a testing document i, fn_i is the number of labels predicted incorrectly as negative for a testing document i, N is the total number of test documents.

$$\text{Example} - \text{based Recall} = \frac{1}{N}\sum_{i=1}^{N}\frac{|tp_i|}{|tp_i+fn_i|} \tag{3.14}$$

$$\text{Example} - \text{based Precision} = \frac{1}{N}\sum_{i=1}^{N}\frac{|tp_i|}{|tp_i+fp_i|} \tag{3.15}$$

Using label-based metrics, the classification performance is calculated for each category, and the overall performance is obtained by calculating the mean performance across all categories. Six

label-based metrics are used: micro-recall, micro-precision, macro-recall, macro-precision, weighted-recall, and weighted-precision. The six metrics are calculated using Equations 3.16-21 (Madjarov et al. 2012; Pedregosa et al. 2011), where tp_j is the number of test documents labelled correctly as positive for a category j, fp_j is the number of test documents labelled incorrectly as positive for a category j, fn_j is the number of test documents labelled incorrectly as negative for a category j, C is the total number of categories (in this application, $C = 6$ since there are six topics in total), and $(tp_j + fn_j)$ is the total number of true labels for all test documents for category j. During the evaluation of each category, the label-based metrics temporarily treat the category as positive and all other categories as negative. Micro-recall and micro-precision measure the overall performance by counting the total number of *tp*, *fp*, and *fn* across all categories. Macro-recall and macro-precision calculate the recall and precision by counting the total number of *tp*, *fp*, and *fn* for each category, and then use the arithmetic mean performance across all categories to obtain the overall performance. The weighted-based metrics are very similar to the macro-based metrics, except that the former use the total number of true labels for all test documents across each category as a weight to obtain a weighted mean performance.

$$\text{Micro} - \text{Recall} = \frac{\sum_{j=1}^{C} tp_j}{\sum_{j=1}^{C} tp_j + \sum_{j=1}^{C} fn_j} \tag{3.16}$$

$$\text{Micro} - \text{Precision} = \frac{\sum_{j=1}^{C} tp_j}{\sum_{j=1}^{C} tp_j + \sum_{j=1}^{C} fp_j} \tag{3.17}$$

$$\text{Macro} - \text{Recall} = \frac{1}{C}\sum_{j=1}^{C} \frac{tp_j}{tp_j + fn_j} \tag{3.18}$$

$$\text{Macro} - \text{Precision} = \frac{1}{C}\sum_{j=1}^{C}\frac{tp_j}{tp_j+fp_j} \quad (3.19)$$

$$\text{Weighted} - \text{Recall} = \frac{1}{\sum_{j=1}^{C}(tp_j+fn_j)}\sum_{j=1}^{C}(tp_j + fn_j)\frac{tp_j}{tp_j+fn_j} \quad (3.20)$$

$$\text{Weighted} - \text{Precision} = \frac{1}{\sum_{j=1}^{C}(tp_j+fn_j)}\sum_{j=1}^{C}(tp_j + fn_j)\frac{tp_j}{tp_j+fp_j} \quad (3.21)$$

These different types of metrics can be used in combination to indicate performance from multiple perspectives. Although related, these metrics measure the performance in different ways (Madjarov et al. 2012; Pedregosa et al. 2011). Example-based metrics treat all documents with equal weight regardless of the different number of labels the documents may have. For example, a two-label document with only one correctly predicted label and a six-label document with three correctly predicted labels have equal example-based recall (i.e., 1/2 = 3/6), although the two documents contribute to the absolute number of errors differently. In contrast to the example-based metrics, label-based metrics take this difference (i.e., number of labels for each document) into account. Among the three types of label-based metrics, micro-based metrics do not consider which category the incorrect labels come from but instead consider all errors from all categories equally in performance assessment. In contrast, macro-based metrics consider which category the incorrect labels come from and assesses the overall performance in terms of the performance for each category. In comparison to macro-based metrics, weighted-based metrics further weights the performance for each category in terms of total number of true labels in that category. Micro-based metrics are, thus, the most stringent, because an error from any category contributes equally to the total performance, whereas in example-based, macro-based, and weighted-based metrics an error

could be discounted during averaging/weighting. A high performance variance across macro-based metrics and weighted-based metrics may indicate that the dataset suffers from a label imbalance problem. Label imbalance problems are common in multilabel classification; they occur when some categories have much more documents than other categories (e.g., 1 versus 100) (Chawla et al. 2004; Charte et al. 2013).

3.2.3 Experimental Results and Analysis

The proposed ontology-based TC algorithm was tested on the six topics using the four types of evaluation metrics. The overall performance under each metric and their corresponding thresholds are summarized in Table 3.7. Among the four types of metrics, the example-based metric yielded the highest performance at 98.69% recall and 92.70% precision, while the micro-based metric showed the least performance at 97.32% recall and 86.51% precision. These results are consistent with the fact that micro-based metrics are the most stringent.

Table 3.7. Performance of Ontology-Based TC Approach

Topic	Ontology-based approach										Machine learning-based approach	
	Total similarity difference threshold	Total similarity percentage difference threshold	Example-based metric		Micro-based metric		Macro-based metric		Weighted-based metric			
			Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
Air leakage topic	486.9	37.8%	98.69%	92.70%	97.32%	86.51%	97.65%	90.44%	97.32%	89.01%	97.30%	84.30%
Fenestration topic	120.2	44.0%										
Lighting power topic	79.8	4.3%										
Lighting system control topic	79.8	17.2%										
Thermal insulation topic	370.1	0.5%										
Ventilation system and equipment energy efficiency topic	140.1	18.6%										

3.2.3.1 Evaluation of the Proposed Ontology-Based Text Classification Algorithm

The overall performance under each multilabel evaluation metric may (1) illustrate the strengths and weaknesses of the proposed methodology in classifying environmental regulatory documents, thereby providing clues for refining the methodology for further performance improvement; and (2) indicate some characteristics of environmental regulatory documents for future comparison with those of documents of other types (e.g., contract documents) and other domains (e.g., safety).

3.2.3.1.1 Threshold Analysis

A threshold analysis was conducted. The thresholds (TSD threshold and TSPD threshold) of each topic are keys in determining the assignment of multiple labels to a clause. Analyzing the thresholds of the topics may thus help in understanding the characteristics/relationships of the topics and may reveal some clues for further performance improvement. Based on the analysis, it is observed that thresholds of topics may reflect the existing semantic overlaps/relationships among the topics and, thus, the organization of topics in the hierarchy. Three observations are made. First, a semantic overlap/relationship between two topics may be reflected by the closeness of their threshold values. For example, the semantic relationships between the "air leakage topic" and the "thermal insulation topic" (e.g., that air leakage in a building envelop directly influences its quality of thermal insulation) was reflected by the closeness of their TSD threshold values (486.9 and 370.1, respectively). Similarly, both "lighting system control topic" and "lighting power topic" used the same TSD threshold value (79.8). The close/same threshold values are

expected due to the existing semantic relationships between the two topics. For example, installing an automatic shut-off in the lighting system can save energy to help meet the requirements of lighting power. This may be substantiated by the fact that these two topics are located in the same branch of the topic hierarchy. Second, the semantic distinctiveness (i.e., less overlap) of a topic may be reflected by its small threshold values compared to other topics. For example, the relatively small TSD thresholds (the smallest at 79.8) of the "lighting system control topic" and the "lighting power topic" indicate that these two topics are likely not semantically related to the other topics. This may be reflected by the fact that these two topics are the only two topics in their hierarchy branch. Similarly, the TSD and TSPD thresholds of the "ventilation system and equipment energy efficiency topic" (140.1 and 18.6%) were small compared to the other topics, indicating that this topic is likely a semantically isolated topic. This may be substantiated by the fact that it is a sole topic in its hierarchy branch, thereby sharing less semantic overlaps with the other topics. Less semantic overlaps with other topics makes the classification of clauses related to this topic easier, which is partially reflected by its high performance, namely 100% and 97.8% macro-based recall and precision, respectively. Third, the semantic dominance of a primary topic, compared to secondary topics, may be reflected by its large TSD and TSPD thresholds. For example, the largest TSD threshold (486.9) and the second largest TSPD threshold (37.8%) were used for the "air leakage topic". The large thresholds indicate that the total similarities of the secondary topics to the labeled clauses are relatively much smaller than that of the primary topic. This further indicates that the "air leakage topic" is relatively more relevant to these clauses than their secondary topics.

3.2.3.1.2 Performance Analysis

The proposed ontology-based TC methodology achieved overall recall values from 97.32% to 98.69% and overall precision values from 86.51% to 92.70%. First, compared to other ontology-based TC efforts, the proposed methodology: (1) outperforms some efforts (e.g., Fang et al. 2007; Wei et al. 2006) in both recall and precision; (2) outperforms some efforts in a limited way. For example, the proposed algorithm outperforms some efforts (Song et al. 2005; Yang et al. 2008) in recall under all four metrics but in precision under only example-based and macro-based metrics; and (3) is outperformed by some efforts in a limited way. For example, He et al. (2004) only addressed a single-label binary classification problem, though they achieved both recall and precision values of over 97%. Second, in terms of performance improvement compared with the non-ontology-based, supervised ML-based TC in Section 3.1, the proposed approach shows improvement in both recall and precision (average improvement of 0.5% in recall and 5.4% in precision), rather than a trade-off improvement [e.g., recall was improved at the expense of precision in Wei et al. (2006), Fang et al. (2007), and Yang et al. (2008)]. The proposed algorithm shows more improvement in precision compared with recall because many incorrect labels are filtered out during the assignment of secondary labels.

Among all four types of evaluation metrics, (1) the example-based metrics showed the highest performance, at 98.69% recall and 92.70% precision. This indicates a high performance level; (2) the most stringent metric – the micro-based metrics – showed 97.32% recall and 86.51% precision.

This provides the most conservative performance estimate for comparison with the non-ontology-based ML-based approach; (3) the macro-based metrics showed 97.65% recall and 90.44% precision by evaluating the performance in terms of each category; and (4) the weighted-based metrics showed 97.32% recall and 89.01% precision. The small difference between the macro-based and weighted-based metrics (a difference of 0.33% in recall and 1.43% in precision) may indicate that the dataset does not suffer from label imbalance problems.

An error analysis was conducted to identify the sources of errors. Precision errors come from incorrectly assigning false labels to some clauses. A semantic relationship between two or more topics may result in misclassification among these topics. For example, the following clause was labelled incorrectly with "thermal insulation topic", in addition to the correct label "air leakage topic", since any air leakage in the building thermal envelope may compromise the performance of thermal envelope insulation: "C402.4.8 Recessed lighting. Recessed luminaires installed in the building thermal envelope shall be sealed to limit air leakage between conditioned and unconditioned spaces" (ICC 2012).

Recall errors come from incorrectly missing to assign true labels to some clauses. A semantic dominance of a primary topic, compared to secondary topics, may result in missing secondary labels. For example, one secondary label ("thermal insulation topic") was missed for both of the following two clauses, because their primary topics ("air leakage topic" and "fenestration topic" for Clauses 1 and 2, respectively) dominated semantically: (1) "C403.2.7.3.3 High-pressure duct

systems. Ducts designed to operate at static pressures in excess of 3 inches water gauge (w.g.) (750 Pa) shall be insulated and sealed in accordance with Section C403.2.7." (ICC 2012); and (2) "502.3.2 Maximum U-factor and SHGC. For vertical fenestration and skylights, the maximum U-factor and solar heat gain coefficient (SHGC) shall be as specified in Table 502.3." (ICC 2009b).

3.2.3.2 Performance Comparison: Ontology-Based Approach vs. Machine Learning-Based Approach

The performance of the proposed ontology-based approach was further compared to that of the non-ontology-based, supervised ML-based approach [proposed in Section 3.1], in terms of recall and precision, as illustrated in Figure 3.10. In Section 3.1, SVM was selected based on performance after testing ten commonly-used ML algorithms, including SVM (implemented in both linear and rbf kernel), DT [implemented by classification and regression trees algorithm (Breiman et al. 1984)], NB (implemented by three variances of algorithms: Gaussian NB, Multinomial NB, Bernoulli NB), kNN, Radius-based Neighbors, Nearest Centroid, Random Forest and Gradient Boosted Regression Trees (Aggarwal and Zhai 2012; Breiman 2001; Friedman 2001).

Under the four evaluation metrics, the ontology-based approach achieved recall values from 97.32% to 98.69% and precision values from 86.51% to 92.70%. Thus, based on the testing data, it consistently outperformed the non-ontology-based, supervised ML-based approach that had achieved 97.30% and 84.30% overall average recall and precision, respectively. This shows that the proposed ontology-based approach is potentially successful in utilizing the semantics of the text for improving the performance of TC.

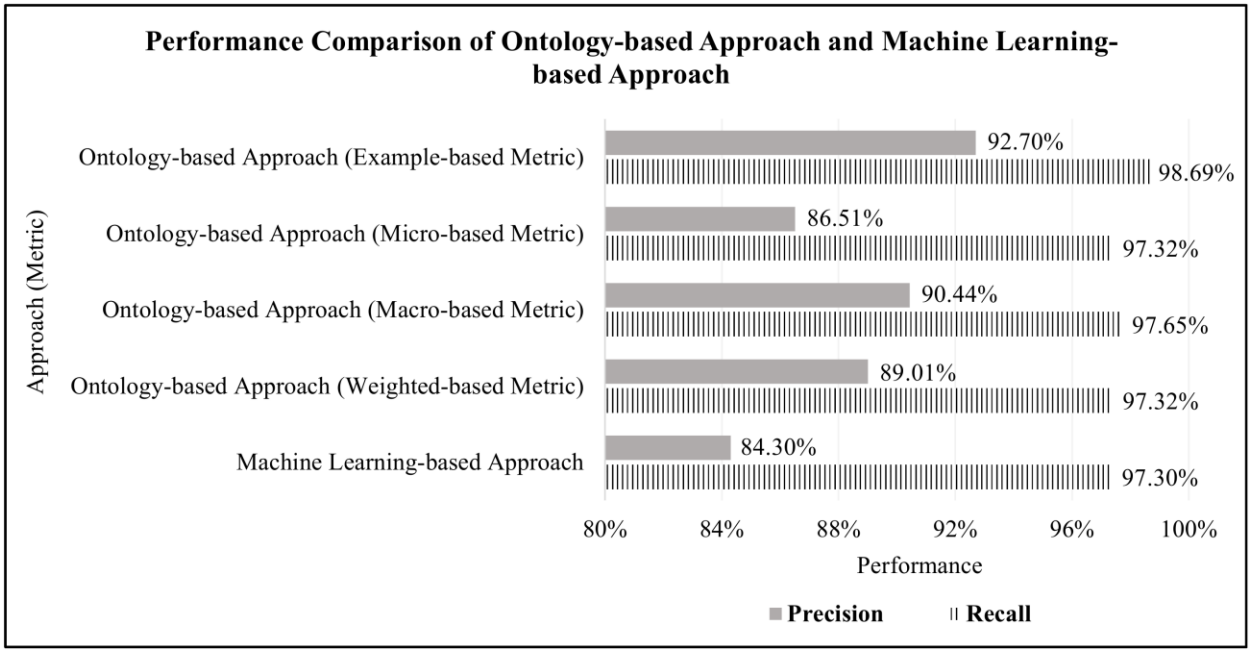


Figure 3.10. Performance comparison of ontology-based approach and machine learning-based approach

CHAPTER 4 – AUTOMATED INFORMATION EXTRACTION FROM BUILDING ENERGY CODES

4.1 Comparison to the State of the Art

Despite the large number of IE efforts outside the construction domain, the number of IE efforts, especially OBIE efforts, are limited in the construction domain. For non-ontology-based IE efforts, Al Qady and Kandil (2010) used limited syntactic features [i.e., specific phrases like VP (i.e., verb phrase) segment and its role ACTIVE_VERB] to extract concepts and relations from contract documents, with the aim to improve construction document management (e.g., document classification and retrieval). Abuzir and Abuzir (2002) used document structure features (i.e., HTML tags) and simple lexico-syntactic features (e.g., "such as" is one lexico-syntactic feature that is used to extract the terms following it because it usually indicates a synonym relationship among these terms) to extract terms and their relations from web pages, with the aim to construct a thesaurus of civil engineering. For ontology-based IE efforts, Zhang and El-Gohary (2013) used a combination of syntactic and semantic features to extract regulatory requirements from building codes for supporting automated code compliance checking, where the semantic features are extracted using a building ontology. Despite the importance of these efforts, they are still limited in one or more of the following four main ways. First, existing efforts extract information from unclassified text, which may result in unnecessary processing effort and may increase extraction errors due to processing irrelevant text. None of these efforts explored the use of text classification techniques to filter out irrelevant text prior to IE to improve the efficiency and performance of IE.

Second, existing efforts were not tested in deep IE from long provisions with multiple exceptions. For example, Abuzir and Abuzir (2002) and Al Qady and Kandil (2010) conducted shallow IE (extracting partial information from a sentence, whereas deep IE aims to extract all information expressed by a sentence based on a full analysis of the sentence). Zhang and El-Gohary (2013), on the other hand, conducted deep IE, but tested their algorithms in extracting requirements from international building codes, which include relatively shorter provisions with fewer exceptions in comparison to energy conservation codes; energy conservation codes include relatively long provisions with several exceptions. Third, existing efforts are limited in automatically dealing with text that includes hierarchically-complex sentence structures. For example, Al Qady and Kandil (2010) used a manual approach to break down AIA contract sentences that contain enumerations and lists into separate sentences, each containing only one single component of the enumeration/list. This manual approach is time-consuming, if there are a large number of sentences. Fourth, there is still a need in improving OBIE performance to support high performance ACC. For example, there are no IE efforts that explored building a conceptual dependency structure to capture the dependency information among the target information and using this dependency information when defining the patterns in the extraction rules, in order to reduce text ambiguities for enhancing the extraction performance. Similarly, it is important to explore the use of a deeper (more detailed) ontology [e.g., deeper in compared to that used in Zhang and El-Gohary (2013)] in improving the extraction performance in the environmental regulatory domain.

4.2 Proposed Method for Ontology-Based Information Extraction from Building Energy Conservation Codes

To address the aforementioned knowledge gaps, this research proposes a rule-based ontology-based information extraction (OBIE) method for automatically extracting thermal insulation requirements and lighting power requirements from energy codes for supporting EnergyACC in construction. Domain-specific preprocessing techniques, ontology-based pattern-matching extraction techniques, sequential dependency-based extraction methods, and cascaded extraction methods were used to extract requirements from the provisions. An HTML-based table processing and extraction method was used to extract requirements from tables. The information extraction algorithm captures and uses dependency information to reduce the semantic ambiguities of the text for enhancing the performance of extraction. A conceptual dependency structure was built to identify target semantic information elements (SIEs) (e.g., subject of compliance checking such as the building element) and the dependency information among the target SIEs. The extraction sequence was thus defined based on the dependency relations of SIEs. Both syntactic features (e.g., POS tags) and semantics features (i.e., concepts from an ontology) were used in the extraction rules to define the patterns of the text. The dependency information was used to assist in constructing the patterns in the extraction rules. Cascaded extraction methods were used to deal with the complex text in energy codes (long provisions, hierarchically-complex provisions, and provisions with exceptions), by breaking down a complex extraction task into a number of simple extraction tasks (i.e., a complex extraction task is cascaded on a number of simple extraction tasks).

Although a ML-based approach can save the manual efforts in pattern definition and extraction rule development, a rule-based approach is adopted in this research for two main reasons. First, a rule-based approach tends to yield higher performance, because human expertise usually results in more accurate patterns and extraction rules (Moens 2006). The performance of ML in a complex task such as IE is usually inconsistent and insufficient (Ireson et al. 2005). In this specific application, the level of complexity in IE is even much higher, compared to the state-of-the-art IE, which makes a rule-based approach especially suitable in this case. Deep IE is needed to extract all information that describes a regulatory requirement and high performance is needed to support high performance ACC – both making the IE problem quite challenging. Second, in this application, the manual effort in pattern definition and extraction rule development in the rule-based approach is expected to be much less than that required for manually annotating a sufficiently large size of training data if taking a ML-based approach.

The proposed IE method is composed of six primary steps (see Figure 4.1): preprocessing, feature selection, identification of target semantic information elements and their conceptual dependency structure, development of extraction rules for sequential dependency-based extraction and cascaded extraction, implementation of the extraction algorithm, and evaluation. An illustrative example of the inputs and outputs of the main processing steps (i.e., Steps 1-5) is shown in Figure 4.2.

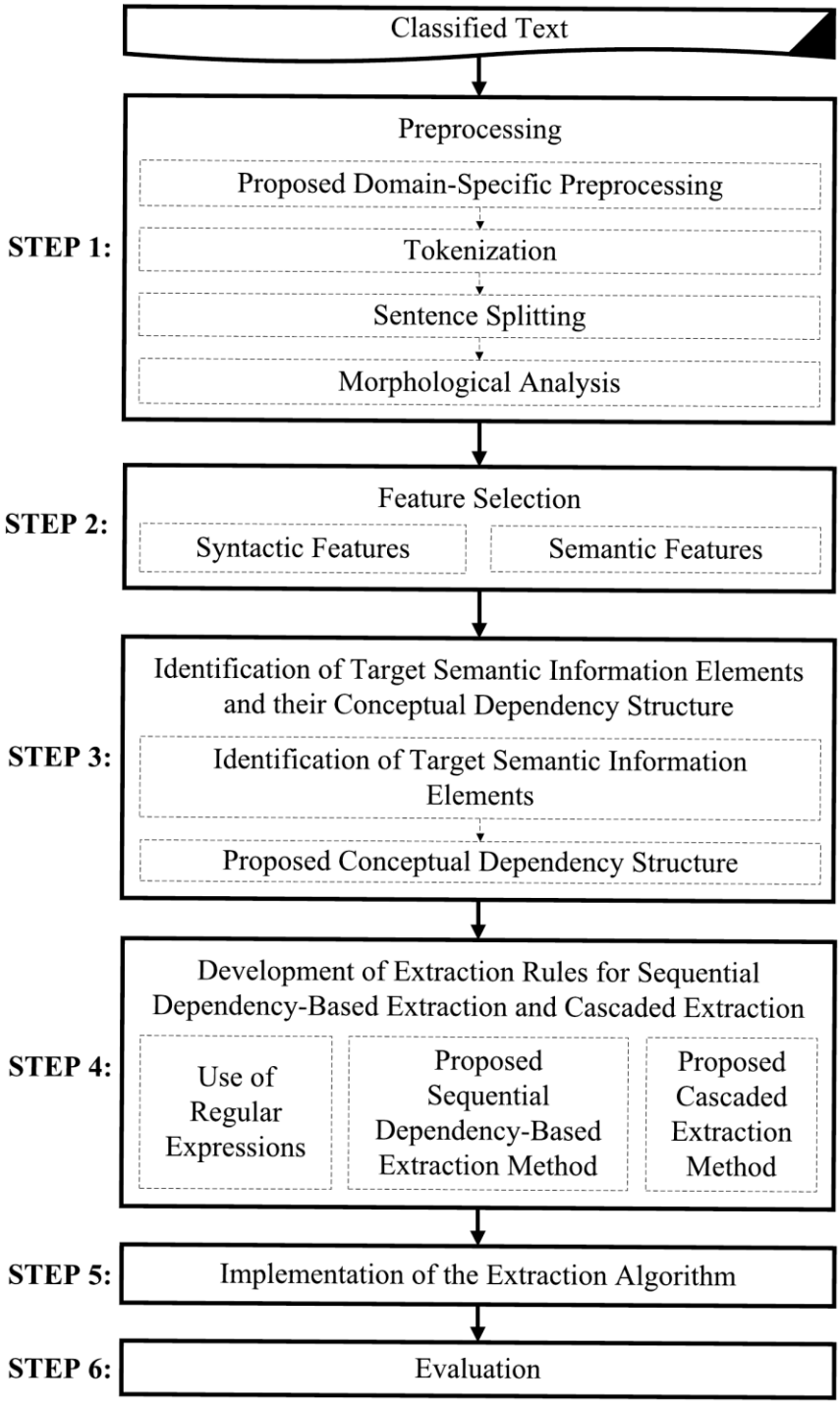


Figure 4.1. Proposed ontology-based information extraction methodology

Classified Text

Original Text (Relevant)

C402.3.2 Minimum skylight fenestration area. In an enclosed space greater than 10,000 square feet (929 m2), directly under a roof with ceiling heights greater than 15 feet (4572 mm), and used as an office, lobby, atrium, concourse, corridor, storage, gymnasium/exercise center, convention center, automotive service, manufacturing, nonrefrigerated warehouse, retail store, distribution/sorting area: transportation, or workshop, the total daylight zone under skylights shall be not less than half the floor area and shall provide a minimum skylight area to daylight zone under skylights of either:
1. Not less than 3 percent with a skylight VT of at least 0.40; or
2. Provide a minimum skylight effective aperture of at least 1 percent determined in accordance with Equation 4-1.
Exception: Skylights above daylight zones of enclosed spaces are not required in:
1. Buildings in climate zones 6 through 8.
2. Spaces where the designed general lighting power densities are less than 0.5 W/ft2 (5.4 W/m2).
3. Areas where it is documented that existing structures or natural objects block direct beam sunlight on at least half of the roof over the enclosed area for more than 1,500 daytime hours per year between 8 am and 4 pm.
4. Spaces where the daylight zone under rooftop monitors is greater than 50 percent of the enclosed space floor area.

Step 1: Preprocessing

Preprocessed Text (Partial preprocessed text shown, whitespace tokens not shown)

<Sentence>……</Sentence><Sentence>……
<Token>1</Token><Token>.</Token><Token>Not</Token><Token>less</Token><Token>than</Token><Token>3</Token><Token>percent</Token><Token>with</Token><Token>a</Token><Token>skylight</Token><Token>VT</Token><Token>of</Token><Token>at</Token><Token>least</Token><Token>0</Token><Token>.</Token><Token>40</Token><Token>;</Token><Token>or</Token></Sentence>

Step 2: Feature Selection

Syntactic Features

- **POS Tags (Shown in parenthesis as examples)**

e.g., 1(**LS**) .(**.**) Not(**RB**) less(**JJR**) than(**IN**) 3(**CD**) percent(**NN**) with(**IN**) a(**DT**) skylight(**NN**) VT(**NNP**) of(**IN**) at(**IN**) least(**JJS**) 0(**CD**).(**.**)40(**CD**);(**:**) or(**CC**)

- **Gazetteers**

Negation gazetteer (e.g., “not”, “no”)
Measurement unit gazetteer (e.g., “square feet” and “cfm/ft2”)

- **Auxiliary Tags (Partial)**

ListNumber (e.g., “1.”), CD (e.g., “3 percent”, “0.40”)

Semantic Features

- **Ontological Concepts** [e.g., “skylight”, “visible transmittance (VT)”]

Step 3: Identification of Target Semantic Information Elements and their Conceptual Dependency Structure

Target Semantic Information Elements

“Subject”, “Subject Restriction”, “Compliance Checking Attribute”, “Deontic Operator Indicator”, “Quantitative Relation”, “Comparative Relation”, “Quantity Value”, “Quantity Unit/Reference”, and “Quantity Restriction”

Conceptual Dependency Structure

- Extraction sequence
- Dependency information

Step 4: Development of Extraction Rules for Sequential Dependency-Based Extraction and Cascaded Extraction

Extraction Rules

e.g., “**(IN JJS):cr + QuantityValue → cr.ComparativeRelation**”: This rule states that if information (e.g., “at least”) matches the pattern “**IN JJS**” and the information is followed by an instance of “**Quantity Value**” (e.g., “0.40”), the information is extracted as an instance of “**Comparative Relation**” (i.e., “at least” is an instance of “Comparative Relation”)

Step 5: Implementation of the Extraction Algorithm

Extracted Instances of the Target Semantic Information Elements

Subject	skylight
Subject Restriction	in an enclosed space greater than 10,000 square feet, under a roof with ceiling heights greater than 15 feet, used as an office, lobby, atrium, concourse, corridor, storage, gymnasium/exercise center, convention center, automotive service, manufacturing, nonrefrigerated warehouse, retail store, distribution/sorting area: transportation, or workshop
Compliance Checking Attribute	VT
Deontic Operator Indicator	shall (obligation)
Quantitative Relation	N/A
Comparative Relation	at least
Quantity Value	0.4
Quantity Unit/Reference	N/A
Quantity Restriction	N/A

Figure 4.2. An illustrative example of the of the main algorithm steps

4.2.1 Preprocessing

Prior to extracting requirements from documents, the documents were first classified (using the method explained in Section 3.2) to filter out the text that is not related to building energy requirements. The documents assigned with a zero label were filtered out. For example, the following sentence shows an example of text that was filtered out, because it describes a document administration requirement rather than a building energy requirement: "The construction documents shall specify that the documents described in this section be provided to the building owner within 90 days of the date of receipt of the certificate of occupancy" (ICC 2012).

The raw classified text was preprocessed for preparation for the following processing and analysis steps. Two primary types of preprocessing were conducted: (1) domain-specific text preprocessing: preprocessing techniques for addressing the specific complexity of the text in energy conservation codes were proposed and used; and (2) general text preprocessing: three commonly-used text preprocessing techniques were used, including tokenization, sentence splitting, and morphological analysis.

4.2.1.1 Proposed Domain-Specific Preprocessing

Two main domain-specific preprocessing techniques were proposed and used: provision splitting and meaning-based stitching. In addition, parenthesis removal and quotation marks removal were proposed and used. Provision splitting and stitching were proposed and used to deal with the following types of text complexities in energy conservation codes: provisions with hierarchically-

complex sentence structures and provisions with exceptions. Two levels of splitting were proposed and used: (1) if a provision has exception(s), then the exception(s) is(are) split from the provision; and (2) based on the splitting results, if the provision and/or exception(s) contain(s) a list of sublevel provisions/exceptions, then the provision and/or exceptions are further split to the lowest level in which each resulting provision/exception contains a component from the list. During stitching, (1) the heading of the provision gets extracted and stitched to each split provision/exception to form the complete provision/exception; and (2) the relationship indicators, which indicate the conjunctive/disjunctive relationships among those split provisions/exceptions, are recognized based on key words such as “and”, “or”, “one of the following”, “all of the following”, and are extracted and stitched to each split provision/exception to retain the overall meaning of the requirement – if these split provisions (i.e., subprovisions) are conjunctive obligations or alternative obligations. For example, “or” is a disjunctive relationship indicator meaning that each of those split provisions in one set are alternative obligations. An alternative obligation is an obligation that allows the obligor to choose which of a number of things to follow (Civil Law Dictionary, 2015), where the compliance with any of them would achieve compliance with the main provision. An illustration of provision splitting and stitching is shown in Figure 4.3.

Figure 4.3. An illustration of provision splitting and stitching

Parenthesis removal aims to remove all parenthesis and the content in parenthesis for simplifying the extraction problem in terms of pattern definition, because in energy conservation codes' text, the content usually contains equivalent information to the information preceding the parenthesis. For example, as per Figure 4.3, "(929 m2)" is removed from the following text, because it represents the same quantitative information but in a different unit: "In an enclosed space greater than 10,000 square feet (929 m2)…" (ICC 2012). Removing such information can both simplify the following information extraction steps (by avoiding extracting multiple semantically-repetitive information) and ensure the consistency of information (e.g., all quantitative information is in the same metric). Quotation marks removal aims to remove all quotation marks because they may interrupt the identification and extraction of specific domain concepts.

4.2.1.2 Tokenization

Tokenization splits the English raw text into tokens (e.g., words, numbers, punctuations, symbols, whitespace) (Manning and Schütze 1999; Moens 2006). For example, the text "1. Not less than 3 percent with a skylight VT of at least 0.40; or" is tokenized into "'1' '.' 'Not' 'less' 'than' '3' 'percent' 'with' 'a' 'skylight' 'VT' 'of' 'at' 'least' '0' '.' '40' ';' 'or'" (the whitespace tokens are not shown). This task aims to identify the boundary of sentences (e.g., periods) and prepare for the following POS tagging task (Moreno et al. 2013).

4.2.1.3 Sentence Splitting

This task aims to split the text into sentences for future processing by detecting sentence boundary indicators like question marks, exclamation points, and periods (Jurafsky and Martin 2009). Unlike question marks and exclamation points, periods are ambiguous in delimiting sentences. For example, the period in "C402.4" is part of the name of a regulatory provision and is not a sentence boundary indicator. A set of domain-specific sentence splitting rules were, thus, developed for domain adaptation, because existing sentence splitters [e.g., the "a nearly-new information extraction" (ANNIE) Sentence Splitter] are domain and application-independent (Cunningham et al. 2011); and, thus, caused errors in splitting the text in the energy conservation codes domain. For example, in partial provision PP1 (which is the result of provision splitting and stitching shown in Figure 4.3), existing sentence splitters mistakenly recognized the period in the list number "1." as full stop of a sentence. The developed sentence splitting rules helped address this issue. This set of domain-specific sentence splitting rules are potentially reusable for similar information extraction applications from similar construction regulatory text.

- PP1: "C402.3.2 Minimum skylight fenestration area. In an enclosed space greater than 10,000 square feet, directly under a roof with ceiling heights greater than 15 feet … the total daylight zone under skylights shall be not less than half the floor area and shall provide a minimum skylight area to daylight zone under skylights of either: 1. Not less than 3 percent with a skylight VT of at least 0.40; or" (ICC 2012).

4.2.1.4 Morphological Analysis

Morphological analysis collapses different derivational (e.g., affixes like "ly", "ion") and inflectional forms (e.g., plural, progressive) of a word to their base form (Manning and Schütze

1999). For example, “balances”, “balancing”, “balance”, and “balanced” are all mapped to “balance”. This task aims to help recognize the semantic features of the text by mapping the morphologically-analyzed text to the ontology concepts. For example, through morphological analysis, “balancing valves” in the natural text is recognized and mapped to the concept “balance valve” in the ontology.

4.2.2 Feature Selection

After preprocessing the text, the syntactic features and semantic features were selected for further extraction rule development (Section 4.2.4). In this research, POS tags, gazetteers, and auxiliary tags were used as syntactic features, while concepts from the ontology were used as semantic features. Semantic features were used to facilitate the extraction of domain-specific semantic information, which would be hard to extract using syntactic features only; semantic features are essential to recognize domain-specific meaning. For example, “building thermal insulation” and “lacking initial test” have exactly the same syntactic features in terms of POS tags (i.e., “VBG JJ NN”, representing gerund, adjective, and singular noun), but the former is recognized to be an instance of “Subject” (a semantic information element) based on the concepts in the ontology. Both syntactic and semantic features were used in the patterns in the extraction rules.

4.2.2.1 Syntactic Features

Three main syntactic features were used: POS tags, gazetteers, and auxiliary tags. POS tagging assigns a tag to each word based on its syntactic word class (e.g., noun, verb, adjective) (Moens

2006). For example, the tags “VBG”, “JJ”, and “NN” were assigned to the gerund, adjective, and singular noun in a sentence, respectively (Jurafsky and Martin 2009).

A gazetteer refers to a list of words that share a common category (e.g., list of countries) (Wimalasuriya and Dou 2010). In this research, a number of words/symbols that represent similar meanings were collected as gazetteers. Each gazetteer was assigned with a tag and each tag was used as a syntactic feature. Accordingly, two gazetteers were manually developed and used: (1) a negation gazetteer, which includes negation words like “no” and “not”; and (2) a measurement unit gazetteer, which includes unit words/symbols like “square feet” and “cfm/ft^2”. Words/symbols belonging to the first and second gazetteers were assigned “neg” and “unit” tags, respectively.

A total of 15 auxiliary tags were also defined and used. Tagging with auxiliary tags (because they are newly-defined tags) was conducted using a set of tagging rules (as explained in Section 4.2.5). Examples of auxiliary tags that appeared frequently when tagging energy conservation codes include: (1) “ListNumber”: assigned to the serial number of each split provision/exception such as “1.” and “2.1”, and (2) “CD” (short for cardinal number): assigned to the numbers in the text that are potential quantity values of a regulatory requirement. For example, in partial provision PP1, both “3 percent” and “0.40” are potential quantity values of the requirement. However, not all numbers should be annotated with the tag “CD”. For instance, in PP1, the numbers “402”, “3”, and “2” in “C402.3.2” are part of the provision designation number, which is not a potential quantity value, and thus should not be tagged with “CD”. Since this ambiguity may result in

potential errors in extracting quantity values, a number of CD tagging rules were developed to reduce such ambiguity based on the patterns of adjacent syntactic features for a cardinal number. For instance, in PP1, if a number has a preceding capital letter "C" and is followed by one or more repetitive patterns (period + number), then all these numbers should not be annotated with the tag "CD".

4.2.2.2 Semantic Features

An ontology was developed to help recognize the semantic features of the text by capturing the concepts related to commercial building energy conservation. The ontology was developed into the ninth level, including 335 concepts in total. A partial view of the ontology is shown in Figure 4.4. The ontology was built/edited using the web ontology language in-memory (OWLIM) Ontology Editor in the General Architecture for Text Engineering (GATE) (Cunningham et al. 2011). The ontology was then inputted into the OntoRoot Gazetteer module to: (1) create a gazetteer of all concepts for using each concept as a semantic feature; and (2) parse the hierarchical "is-a" relationship among concepts to facilitate pattern definition for extraction rule development (as discussed in Section 4.2.4).

For developing the ontology, the ontology development methodology by El-Gohary and El-Diraby (2010) was benchmarked. Accordingly, the methodology for developing the ontology included four main steps: (1) purpose and scope definition, (2) taxonomy building, (3) relation modeling, and (4) ontology coding. The purpose of the ontology is to support OBIE. The scope of the

ontology is limited to the "commercial building energy efficiency" domain. For taxonomy building, the main concepts in the domain of interest were identified based on a review of the main relevant environmental regulatory documents (e.g., the 2012 IECC), and then the identified concepts were organized into a hierarchy of concepts using a combination of a top-down (starting by defining the most abstract concepts) and a bottom-up approach (starting by defining the most specific concepts). For example, concepts related to the "building mechanical system energy efficiency concept" (a subconcept of "commercial building energy efficiency concept") such as "HVAC system", "air economizer system", and "water economizer system" were identified; and then, for this specific example, "air economizer system" and "water economizer system" were modeled as subconcepts of "HVAC system". For relation modeling, the non-hierarchal relationships between concepts were identified and modeled to describe the semantic links between the concepts. For example, "is_controlled_by" is a non-hierarchical relationship that links "lamp" with "occupant sensor". As mentioned above, the concepts and relations of the ontology were coded in OWLIM.

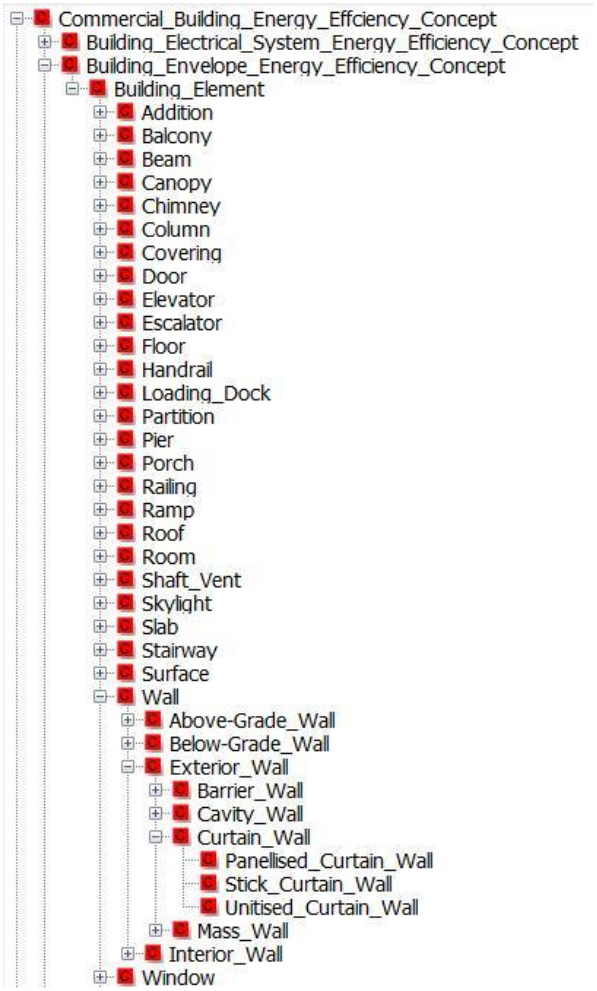


Figure 4.4. Partial view of the ontology

4.2.3 Identification of Target Semantic Information Elements and their Conceptual Dependency Structure

4.2.3.1 Identification of Target Semantic Information Elements

Before developing the extraction rules, the target information that needs to be extracted should be identified based on the specific requirements of the application and the domain. Nine types of target semantic information elements (SIEs) for representing quantitative regulatory requirements were identified [following Zhang and El-Gohary (2013)] and used: including “Subject”, “Subject Restriction”, “Compliance Checking Attribute”, “Deontic Operator Indicator”, “Quantitative

Relation", "Comparative Relation", "Quantity Value", "Quantity Unit/Reference", and "Quantity Restriction".

"Subject" refers to the primary entity that is regulated in a requirement, and corresponds to a concept in the ontology. For example, the concept "skylight" from ontology could be an instance of "Subject". "Compliance Checking Attribute" refers to a specific property of a "Subject" that is checked for compliance, and corresponds to a concept in the ontology. For example, the concept "minimum skylight area" in the ontology could be an instance of "Compliance Checking Attribute". "Deontic Operator Indicator" is a word or phrase that indicates the deontic type of the requirement (Salama and El-Gohary 2013; Zhang and El-Gohary 2013): obligation, permission, or prohibition. For example, in the following sentence "shall" indicates obligation: "The minimum thermal resistance of the insulating material installed in, or continuously on, the below-grade walls shall be as specified in Table C402.2, and shall extend to a depth of 10 feet below the outside finished ground level, or to the level of the floor, whichever is less." (ICC 2012). "Quantitative Relation" refers to the type of semantic relationship between the "Compliance Checking Attribute" and "Quantity Value". In the example above, "extend" is an instance of "Quantitative Relation". "Comparative Relation" refers to a relationship, such as "less than" or "equal to", for stating a quantitative range of a quantity value. "Quantity Value" refers to the quantitative measure of the requirement, while "Quantity Unit/Reference" refers to an explicit measurement unit or an implicit reference unit for the "Quantity Value" (e.g., 10 feet, 35 percent of its rated power). "Subject

Restriction" and "Quantity Restriction" refer to constraints that are placed on the "Subject" and "Quantity Value", respectively, where a restriction may consist of multiple ontology concepts and/or relationships. In this research, for one quantitative requirement: (1) there must be only one "Subject", only one "Comparative Relation", and only one "Quantity Value". For "Comparative Relation", a default "greater_than_or_equal" was used if the relation in a requirement is implicit (e.g., "…shall extend to a depth of 10 feet…"); (2) there could be at most one "Compliance Checking Attribute", at most one "Deontic Operator Indicator", at most one "Quantitative Relation", and at most one "Quantity Unit/Reference"; and (3) there could be zero, one, or multiple "Subject Restrictions" and "Quantity Restrictions". An illustrative example showing the SIEs for a requirement, after splitting and stitching (Figure 4.3), is shown in Figure 4.5.

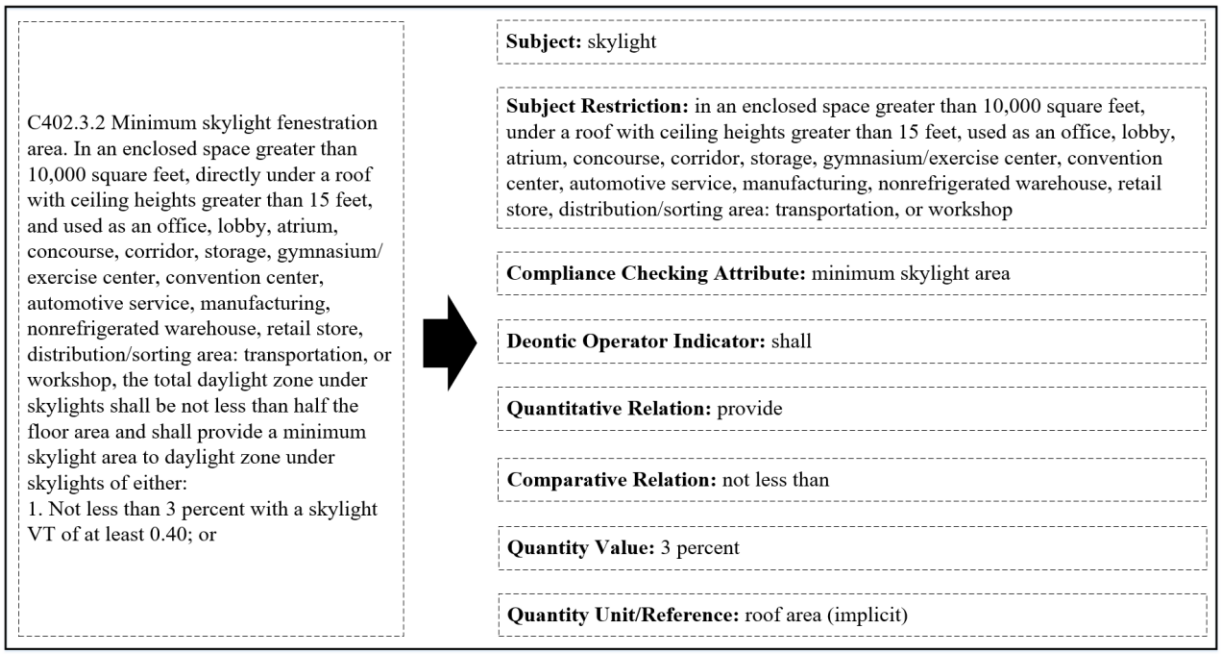


Figure 4.5. Example semantic information element instances

4.2.3.2 Proposed Conceptual Dependency Structure

The conceptual dependency structure of the SIEs was developed based on conceptual dependency theory. According to conceptual dependency theory, any two linguistic structures of identical meaning should have the same conceptual dependency structure (Moens 2006). In this research, the proposed information extraction algorithm is used to extract requirements containing quantitative information of energy conservation. Since all instances of quantitative information express the same meaning in terms of requirements expressed in numerical values on an entity, these instances of quantitative information could be represented by the same conceptual dependency structure. The conceptual dependency structure is composed of inter-dependent primary concepts and relations (Moens 2006). Since a sentence is usually composed of multiple concepts and relations, the sentences were analyzed to identify those primary concepts and relations that correspond to the target SIEs.

After analyzing the dependencies among the target SIEs, the conceptual dependency structure of the SIEs was built, as per Figure 4.6. The conceptual dependency structure indicates that: (1) there exists an extraction sequence that an SIE should be extracted only after all its preceding SIEs are extracted, which is called the sequential dependency extraction method in this research; and (2) developing the extraction rules to extract an SIE may use the preceding SIEs to reduce ambiguities/errors. Comparing to extracting an SIE isolatedly (i.e., extraction rules are developed without using dependency information), the use of dependency information in developing

extraction rules imposes more stringent conditions on matching information, thus ruling out information that does not match the conditions.

In Figure 4.6, the arrow represents the dependency relationship and the serial number of each SIE indicates the extraction sequence. For example, “Subject” depends on both “Deontic Operator Indicator” and “Comparative Relation”, and “Deontic Operator Indicator” also depends on “Comparative Relation”. Therefore, “Comparative Relation” should be extracted first, and “Subject” should be extracted only after its two preceding SIEs (i.e., “Deontic Operator Indicator” and “Comparative Relation”) have been extracted. For the interdependent SIEs, they should be extracted together after all their preceding SIEs have been extracted. For example, the SIEs “Deontic Operator Indicator” and “Quantitative Relation” are interdependent and thus should be extracted together after their preceding “Comparative Relation” SIEs have been extracted.

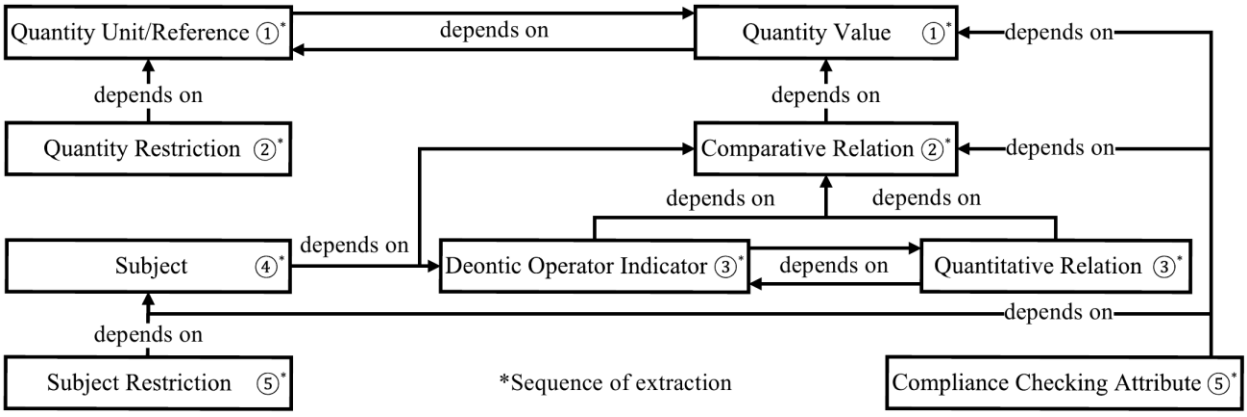


Figure 4.6. Conceptual dependency structure

4.2.4 Development of Extraction Rules for Sequential Dependency-Based Extraction and Cascaded Extraction

After identifying the target SIEs and their conceptual dependency structure, the extraction rules were manually developed to help extract the instances of the target SIEs. The extraction rules were developed after reviewing a number of energy regulatory documents [e.g., ANSI/ASHRAE/IES Standard 90.1-2010 (ASHRAE 2010)] – excluding the IECC which was used for testing – and manually analyzing the text features and patterns in these documents. These documents are the developing data, which – as mentioned above – are analogous to the training data in the case of machine learning. The left side of an extraction rule models the pattern of the text in terms of syntactic features (i.e., POS tags, gazetteers, and/or auxiliary tags) and/or semantic features (i.e., concepts from the ontology), while the right side defines the information that should be extracted when this pattern is matched. In developing the rules, regular expressions were used. Methods for sequential dependency-based information extraction and cascaded information extraction were proposed and considered when developing the rules.

4.2.4.1 Use of Regular Expressions

In defining those patterns, regular expressions were used to define the most simplified (but generalized) patterns so that a rule can deal with a variety of text sharing similar regularities in terms of syntactic and semantic features, regardless of the length and content of the text. As such, regular expressions may facilitate the extraction of complex SIEs (e.g., “Subject Restriction” and “Quantity Restriction”) because they usually contain such feature regularities; they are usually

composed of a number of repetitive semantic and/or syntactic features in certain patterns. For example, “designed for sensible heating of an indoor space through heat transfer from the thermally effective panel surfaces” is an instance of “Subject Restriction”. The semantic features (i.e., concepts) and syntactic features (i.e., POS tags, in this example) of this instance can be analyzed as follows: “sensible heating”, “indoor space”, “heat transfer”, and “thermally effective panel surface” are four ontology concepts, “VBN” is the POS tag for the past participle “designed”, “DT” is for the determiners “an” and “the”, and “IN” is for the prepositions “for”, “of”, “through”, and “from”. Thus, this instance is just a repetition of four prepositional phrases starting with a past participle (i.e., “designed”), and each prepositional phrase may contain an optional determiner. Accordingly, with the help of regular expressions, the pattern for extracting such similar instances could be defined as: “VBN (IN (DT)? commercial_building_energy_efficiency_concept)+”, where the “?” indicates that there is at most one determiner in a prepositional phrase, the “commercial building energy efficiency concept” is a semantic feature representing a concept, the subpattern “(IN (DT)? commercial_building_energy_efficiency_concept)” represents a prepositional phrase, and the “+” indicates that there is at least one such kind of prepositional phrase.

For the use of semantic features in pattern definition, only the top concept of all the possibly matched subconcepts was used as the semantic feature. The top concept is the highest-level relevant concept in the ontology which subsumes all possibly matched subconcepts. For example,

in the above instance, the semantic feature “commercial building energy efficiency concept” is the top concept of all possibly matched subconcepts such as “sensible heating”, “indoor space”, “heat transfer”, and “thermally effective panel surface”. For more details on the use of regular expressions in pattern definition, the readers are referred to Cunningham et al. (2011).

4.2.4.2 Proposed Sequential Dependency-Based Extraction Method

In developing the rules, the dependency information among the SIEs assisted in defining the patterns. For example, the following rule was developed to extract the instances of “Comparative Relation” (a target SIE): “(JJR IN):cr + QuantityValue → cr.ComparativeRelation”. “JJR” and “IN” are POS tags for comparative adjective and preposition. “JJR IN” is a pattern that matches information like “less than”. When “JJR IN” is followed by “QuantityValue”, which is the dependency information, the information matching “JJR IN” should probably be an instance of “Comparative Relation”. Therefore, a pointer “cr” was set to pattern “JJR IN”, and the information (which the pointer refers to) matching this pattern was extracted as an instance of “ComparativeRelation”. Similarly, Rule 7 “(commercial_building_energy_efficiency_concept):sj + VBZ + ComparativeRelation → sj.Subject” was used to extract “radiant panel” as an instance of “Subject”, where “radiant panel” is a subconcept of “commercial building energy efficiency concept”.

4.2.4.3 Proposed Cascaded Extraction Method

In developing the rules for extracting complex SIEs (e.g., "Subject Restriction" and "Quantity Restriction"), which usually appear in longer provisions, a cascaded information extraction method was proposed and used to break down a complex extraction task into a number of simple extraction tasks (i.e., a complex extraction task is cascaded on a number of simple extraction tasks). As such, simple SIEs (or simpler SIEs, e.g., "Quantity Restriction" is simpler than "Subject Restriction") are used as features in the rules that extract complex SIEs. The extraction of such complex SIEs can, thus, be broken down into two steps: (1) extracting the simple SIEs; and (2) extracting the complex SIEs based on the simple SIEs. For example, in partial provision PP1, "…under a roof with ceiling heights greater than 15 feet..." (ICC 2012) is an instance of "Subject Restriction" that contains quantitative information corresponding to five SIEs. Accordingly, the instances of the five SIEs were first extracted: "roof", "ceiling height", "greater than", "15", and "feet" were extracted as instances of "Subject", "Compliance Checking Attribute", "Comparative Relation", "Quantity Value", and "Quantity Unit/Reference", respectively. Then these extracted instances were used to extract the instance of the "Subject Restriction" (i.e., these five simple SIEs were used as features in the rule that extracted the "Subject Restriction").

4.2.5 Implementation of the Extraction Algorithm

Steps 1-4 (Section 4.2.1-4) were implemented in the "a nearly-new information extraction" (ANNIE) system of GATE suite of tools (Cunningham et al. 2011), including the following

modules: ANNIE English Tokeniser, ANNIE Sentence Splitter, GATE Morphological Analyser, ANNIE POS Tagger, ANNIE Gazetteer, OntoRoot Gazetteer, and JAPE Transducer. Each of these modules may have initialization parameters. For example, "caseSensitive" is a parameter having either "true" or "false" values, which indicates whether matching should be conducted in a case-sensitive manner or not. For the details of the parameters for all these modules in the ANNIE system, the readers are referred to Cunningham et al. (2011).

The two gazetteers (see Section 4.2.2) were added to the ANNIE Gazetteer module in GATE, along with other existing gazetteers (e.g., location, currency, etc.). The 15 auxiliary tags (see Section 4.2.2) were added to JAPE Transducer, where the tagging was conducted using a set of tagging rules.

All extraction rules (see Section 4.2.4) were developed in the grammar of Java Annotation Patterns Engine (JAPE) required by GATE (Cunningham et al. 2011) using a JAPE editor – Vim (Vi IMproved) (Robbins et al. 2008). The JAPE grammar has five control styles to assist the extraction in terms of rule matching. The most commonly used is the "applet" control style (Cunningham et al. 2011). For a region of text starting from a fixed location of a sentence, under the "applet" control style, only the rule that matches the longest text starting from the fixed location will be fired. For instance, in extracting the instance of "Quantity Unit/Reference" from the following sentence, Rule 12 can match both the text "Btu" and "Btu per inch/h × ft^2 × °F": "For automatic-circulating hot water and heat-traced systems, piping shall be insulated with not less than 1 inch

of insulation having a conductivity not exceeding 0.27 Btu per inch/h × ft^2 × °F." (ICC 2012). According to the matching mechanism of "applet" control style (i.e., longest matching) used in Rule 12, it is the "Btu per inch/h × ft^2 × °F" that was extracted as an instance of "Quantity Unit/Reference". For further details on the other control styles and the JAPE grammar, the readers are referred to Cunningham et al. (2011). All the developed extraction rules were inputted into the "JAPE Transducer" module for executing the extraction. The outputs (Section 4.2.5), as illustrated in Figure 4.2, are the extracted instances of the target SIEs, which were used for performance evaluation (Section 4.2.6).

4.2.6 Evaluation

The OBIE algorithm was tested in extracting commercial building energy efficiency regulatory requirements from Chapter 4 of the 2012 IECC (ICC 2012). IECC was selected because it is the most widely-adopted building energy conservation code in the U.S. The performance was evaluated by comparing the extraction results to a gold standard.

The gold standard for Chapter 4 of the 2012 IECC (ICC 2012) was manually developed. It was developed by three researchers – the author and two other researchers. Although it is a good strategy to use domain experts to develop a gold standard to ensure its validity and reliability, in many cases this is not feasible (Kilicoglu et al. 2011), because domain experts are usually not easily available to participate in such time-intensive activities and their time is highly expensive (Li et al. 2015). It is, therefore, a common practice to have researchers with domain knowledge

develop the gold standard (Tateisi et al. 2014; Ganu et al. 2010). In this research, the author was involved in developing the gold standard because of his familiarity with all of the following three areas, which helps ensure the correctness – and thus the validity – of the gold standard annotations: energy conservation codes, civil engineering domain, and the NLP domain. Two other civil engineering researchers participated in developing the gold standard, in order to (1) avoid confirmation bias for validity, and (2) have multiple annotators annotate the same sentence and measure inter-annotator agreement to evaluate the reliability of the gold standard. Typically, two or three annotators annotate the same text for information extraction work (Li et al. 2015), which indicates that using three annotators is sufficient.

The annotation was conducted in three main steps: (1) A short 15-minute presentation was given to the annotators to explain the annotation objective, the target semantic information elements, and illustrate the instances of all semantic information elements using examples from the development text [e.g., ANSI/ASHRAE/IES Standard 90.1-2010 (ASHRAE 2010)]; (2) another warm-up and question and answer (Q&A) session was conducted for training the annotators and clearing any confusion using example sentences from the development text; and (3) the annotators conducted the annotation independently. The inter-annotator agreement was calculated. The initial inter-annotator agreement was 86% in F-measure, which indicates the reliability of the gold standard. "An F-measure of 0.80 or above is generally considered sufficient inter-annotator agreement" (Pestian et al. 2012). Any discrepancies were then discussed and resolved until a consensus was

reached, thereby achieving final full annotator agreement. So, overall, the use of this team of three annotators and the annotation process aimed to balance reliability and validity.

An illustrative example of three provisions and their corresponding target SIEs is shown in Table 4.1. The performance was evaluated by comparing the extraction results with the gold standard. The performance was measured in terms of recall and precision (Maynard 2006; Moens 2006). Recall is the percentage of correctly extracted instances out of the total number of instances that should be extracted. Precision is the percentage of correctly extracted instances out of the total number of extracted instances.

Table 4.1. Examples of Provisions and their Corresponding Semantic Information Elements in the Gold Standard

Semantic information element	**Partial provision (requirement)**		
Provision	In an enclosed space greater than 10,000 square feet, directly under a roof with ceiling heights greater than 15 feet, and used as an office, lobby, atrium, concourse, corridor, storage, gymnasium/exercise center, convention center, automotive service, manufacturing, nonrefrigerated warehouse, retail store, distribution/sorting area: transportation, or workshop, the total daylight zone under skylights shall be not less than half the floor area and shall provide a minimum skylight area to daylight zone under skylights of either: 1. Not less than 3 percent with a skylight VT of at least 0.40; or (ICC 2012)		
Requirement	R1	R2	R3
Subject	total daylight zone	skylight	skylight
Subject restriction	under skylights, in an enclosed space greater than 10,000 square feet, under a roof with ceiling heights greater than 15 feet, used as an office, lobby, atrium, concourse, corridor, storage, gymnasium/exercise center, convention center, automotive service, manufacturing, nonrefrigerated warehouse, retail store, distribution/sorting area: transportation, or workshop	in an enclosed space greater than 10,000 square feet, under a roof with ceiling heights greater than 15 feet, used as an office, lobby, atrium, concourse, corridor, storage, gymnasium/exercise center, convention center, automotive service, manufacturing, nonrefrigerated warehouse, retail store, distribution/sorting area: transportation, or workshop	in an enclosed space greater than 10,000 square feet, under a roof with ceiling heights greater than 15 feet, used as an office, lobby, atrium, concourse, corridor, storage, gymnasium/exercise center, convention center, automotive service, manufacturing, nonrefrigerated warehouse, retail store, distribution/sorting area: transportation, or workshop
Compliance checking attribute	area (implicit)[1]	minimum skylight area	VT[2]
Deontic operator indicator	shall (obligation)	shall (obligation)	shall (obligation)
Quantitative relation	N/A	provide	N/A
Comparative relation	not less than	not less than	at least
Quantity value	half	3 percent	0.4
Quantity unit/reference	floor area	roof area (implicit)[1]	N/A
Quantity restriction	N/A	N/A	N/A

1. “implicit” means the instance is not explicitly stated in the text.

2. VT=Visible transmittance

4.3 Experimental Results and Analysis

4.3.1 Performance Results

The experimental results are summarized in Table 4.2. The number of patterns used to extract the "Subject", "Subject Restriction", "Compliance Checking Attribute", "Deontic Operator Indicator", "Quantitative Relation", "Comparative Relation", "Quantity Value", "Quantity Unit/Reference", and "Quantity Restriction" instances are 25, 15, 8, 11, 11, 9, 14, 14, and 9, respectively. In addition, ten patterns were defined for the cascaded extraction of "Subject Restriction" instances, while one pattern was used for the "Quantity Restriction" instances. The gold standard includes 127, 87, 53, 56, 52, 87, 87, 87, and 23 instances of "Subject", "Subject Restriction", "Compliance Checking Attribute", "Deontic Operator Indicator", "Quantitative Relation", "Comparative Relation", "Quantity Value", "Quantity Unit/Reference", and "Quantity Restriction", respectively, at a total of 659 instances. A performance of 97.4% recall and 98.5% precision was achieved, which indicates that the proposed information extraction algorithm is potentially effective in extracting regulatory requirements from energy conservation codes.

Table 4.2. Experimental Results of Extracting Requirements from Energy Conservation Codes

Total number of instances	Subject	Subject restriction	Compliance checking attribute	Deontic operator indicator	Quantitative relation	Comparative relation	Quantity value	Quantity unit/reference	Quantity restriction	Total
In gold standard	127	87	53	56	52	87	87	87	23	659
Extracted	124	85	53	55	51	87	88	87	22	652
Correctly extracted	115	85	53	55	51	87	87	87	22	642
Precision	92.7%	100.0%	100.0%	100.0%	100.0%	100.0%	98.9%	100.0%	100.0%	98.5%
Recall	90.6%	97.7%	100.0%	98.2%	98.1%	100.0%	100.0%	100.0%	95.7%	97.4%

4.3.2 Effects of Sequential Dependency-Based Extraction

The results show that the use of dependency information was effective in reducing semantic ambiguities. This can be illustrated by the extraction results for the instances of "Quantity Unit/Reference" and "Comparative Relation"; both achieved 100% precision. For example, "above", which is a potential instance of "Comparative Relation", is a word that can create semantic ambiguity. For example, "above" could either be followed by a "Quantity Value" [e.g., "the pavement temperature is above 50 °F" (ICC 2012)] or a location [e.g., "skylights are installed above daylight zone" (ICC 2012)], but only in the former case "above" would mean a "Comparative Relation". The proposed algorithm was able to avoid such semantic ambiguities because of utilizing dependency information. For example, after extracting the SIE "Quantity Value", it is used as dependency information to assist in the extraction of other SIEs. Thus, to correctly extract "above" as an instance of "Comparative Relation", the pattern can be defined as "IN + QuantityValue", where "IN" is the POS tag of preposition (i.e., above) and "QuantityValue"

is the dependency information, indicating that "above" should be extracted as an instance of "Comparative Relation" only in the case when it is followed by a quantity value. However, sometimes there is no dependency information to help resolve ambiguities, especially when extracting a target information at the top of the conceptual dependency structure. This could be illustrated by the errors in the extraction of "Quantity Value" instances, which is one of the top SIEs in the conceptual dependency structure. For example, in the following sentence, the number "15" was incorrectly extracted as an instance of "Quantity Value", because there is no dependency information: "…Materials in Items 1 through 15 shall be deemed to comply with this section provided joints are sealed and materials are installed as air barriers in accordance with the manufacturer's instructions…" (ICC 2012).

"Compliance Checking Attribute", "Deontic Operator Indicator", and "Quantitative Relation" all showed 100% precision. This could be partially attributed to their unique semantic and/or syntactic features. For example, the concepts corresponding to "Compliance Checking Attribute" all semantically represent some properties like "air leakage rate" and "U-factor". The syntactic features corresponding to "Deontic Operator Indicator" all include the POS tag "MD" for a modal verb (e.g., shall, must), while the syntactic features for "Quantitative Relation" all correspond to verbs in different tenses. But, the perfect precision for "Deontic Operator Indicator" and "Quantitative Relation" may also be partially attributed to the use of dependency information: utilizing dependency information helped avoid errors that result from independent extraction.

One recall error, however, occurred due to sequential extraction. Failure to extract the "Subject" led to failure in extracting its related "Subject Restriction". This indicates that the use of dependency information may sometimes over-constrain the matching conditions, thus, ruling out instances that should be extracted.

4.3.3 Effects of Cascaded Extraction

The results show that the proposed domain-specific preprocessing techniques and cascaded information extraction methods are effective in dealing with long provisions, hierarchically-complex sentence structures, and exceptions. This can be illustrated by the extraction results for the instances of "Subject Restriction". Only two out of the 87 subject restrictions showed recall errors, and only one of them was due to errors in cascaded extraction. The following instance (which is an exception consisting of a complex restriction) showed a recall error because of missing uncommon patterns: "Exception: Economizers are not required for the systems listed below. 2. Where more than 25 percent of the air designed to be supplied by the system is to spaces that are designed to be humidified above 35 F dew-point temperature to satisfy process needs." (ICC 2012). In extracting the information for this complex restriction, in a cascaded way, only partial information ("more than 25 percent of the air" and "spaces that are designed to be humidified above 35 F dew-point temperature") is correctly extracted, whereas the complex relationship "designed to be supplied by the system is to" was not extracted.

4.3.4 Sources of Extraction Errors

Three sources of errors were identified: missing uncommon patterns, conflict resolution errors, and NLP tool errors. Extraction of "Subject" both achieved the lowest recall (90.55%) and lowest precision (92.74%) among the nine SIEs because of the following two reasons: missing uncommon patterns and conflict resolution errors. There are two interesting cases of missing uncommon patterns. First, the subject is prescribed in the provision heading, not the provision itself. For example, in the following provision, although the "fan" was extracted as an instance of "Subject", it is the "heat rejection equipment fan" that should have been extracted as the subject: "C403.4.4 Heat rejection equipment fan speed control. Each fan powered by a motor of 7.5 hp or larger shall have the capability to operate that fan at two-thirds of full speed or less…" (ICC 2012). Second, the subject is implicitly prescribed. For example, in the following sentence, the subject that corresponds to "minimum skylight area" (i.e., the "Compliance Checking Attribute") is "skylight", which is implicitly prescribed: "In an enclosed space greater than 10,000 square feet, …the total daylight zone under skylights shall be not less than half the floor area and shall provide a minimum skylight area to daylight zone under skylights of either: 1. Not less than 3 percent with a skylight VT of at least 0.40;" (ICC 2012).

For conflict resolution errors, few conflict resolution rules caused errors in extraction as a result of resolving conflicts (e.g., a conflict occurs when multiple instances of a "Subject" are extracted) incorrectly. For example, in the following sentence, both "supply air systems" and "VAV systems"

were initially extracted as instances of “Subject”, and after conflict resolution “VAV system” was finally extracted, which is incorrect: “Supply air systems serving multiple zones shall be VAV systems which, during periods of occupancy, are designed and capable of being controlled to reduce primary air supply to each zone to one of the following before reheating, recooling or mixing takes place: 1. Thirty percent of the maximum supply air to each zone.” (ICC 2012).

Both “Deontic Operator Indicator” and “Quantitative Relation” showed recall errors (98.21% and 98.08% recall, respectively) resulting from missing uncommon patterns and NLP tool errors. For “Deontic Operator Indicator”, the recall error comes from missing uncommon patterns. For example, one extraction rule for “Deontic Operator Indicator” states that besides matching the syntactic feature tag “MD”, the instance should also be preceded with a semantic feature (i.e., a concept). However, in the following sentence, the time adverbial “when initiated” is a very uncommon pattern which led to failure in extracting “shall” as an instance of “Deontic Operator Indicator”: “…4. The override switch, when initiated, shall permit the controlled lighting to remain on for a maximum of 2 hours;” (ICC 2012). For “Quantitative Relation”, the recall error comes from the tokenization errors in the inner tool, which mistakenly assigned adjective tag “JJ” to a past participle verb; and, thus, the verb was not extracted as an instance of “Quantitative Relation”.

The recall error for “Quantity Restriction” (95.7% recall) comes from missing uncommon patterns. The precision error for “Quantity Value” (98.9% precision) occurred due to conflict resolution errors. Other than that, as discussed above, the two recall errors for “Subject Restriction” (97.7%

recall) occurred due to errors in cascaded extraction and sequential extraction.

CHAPTER 5 – AUTOMATED INFORMATION EXTRACTION FROM CONTRACT SPECIFICATIONS

5.1 Comparison to the State of the Art

Information extraction (IE) efforts are limited in the construction domain. These efforts used different approaches – rule-based and ML-based – to support different applications. Example efforts using a rule-based approach include Al Qady and Kandil (2010), which used syntactic features (e.g., phrase segments) to extract concepts and relations from contract documents for enhanced document management; and Abuzir and Abuzir (2002), which used lexico-syntactic features and document structure features (i.e., HTML tags) to extract terms and their relations from web pages for constructing a thesaurus of civil engineering. Example efforts using a ML-based approach include Liu and El-Gohary (2017), which proposed an ontology-based, semi-supervised conditional random fields-based IE method to extract information entities that describe bridge deficiencies and maintenance actions from bridge inspection reports for improved bridge deterioration prediction.

Two main efforts used rule-based IE to support ACC. First, Zhang and El-Gohary (2013) proposed an IE method to extract design requirements from building codes to support automated building code checking. They used a building ontology to capture the semantic features in the building-code text, and developed a set of pattern-matching-based IE rules for the extraction. Second, an IE method was proposed in Chapter 4 to extract energy requirements from energy codes to support automated energy code checking. Compared to the first effort, a more detailed ontology to improve

the extraction performance was used. A combination of methods to deal with long provisions, hierarchical sentences, and exceptions were also proposed.

Despite the contributions of the aforementioned efforts, there is a lack of methods that can deal with the following three text complexities that characterize contract specifications: hierarchically-complex text structures, incomplete sentence structures, and variety of LODs. First, the methods in Chapter 4 addressed the hierarchical complexity of the text on the sentence level. However, specifications exhibit paragraph-level complexity, which is more challenging: the text is hierarchically deep and wide, as well as dynamic. An article in the specifications may consist of dozens of paragraphs, where each paragraph may consist of up to ten levels of subparagraphs and each level may consist of dozens of subparagraphs. In contrast, a provision in the International Energy Conservation Code (IECC) usually has few levels of subprovisions, where each level has only several sentences. In addition, the paragraph-level complexities of specifications usually increase with project size. In contrast, different versions of the IECC tend to have similar complexity levels. Second, previous efforts (Zhang and El-Gohary 2013) and the methods in Chapter 4 did not address text with incomplete sentence structures, which unlike codes, is common in specifications. IE from text with complete sentence structures is relatively easier, because complete sentence structures have regular grammatical patterns. The experimental results in Chapter 4 showed that regular grammatical patterns usually implied strong dependency relationships among the target information, which was sufficient to reduce most of the text

ambiguities. Text with incomplete sentence structures, lacking such regular patterns, would thus likely to suffer from weak dependency relationships that would be insufficient to reduce ambiguities. Third, existing methods are not able to recognize and differentiate the LODs of the information in the contract specifications. Extraction of requirements in irrelevant LODs (i.e., information beyond the current/needed LOD) may result in potential compliance checking errors.

5.2 Proposed Method for Semantic Information Extraction from Contract Specifications

To address the aforementioned knowledge gaps, this research proposes a semantic, NLP-enabled, rule-based IE method for automatically extracting thermal insulation requirements and lighting power requirements from contract specifications for supporting EnergyACC in construction. The method developed in Chapter 4 was adapted to address the different nature of the text, including hierarchically-complex text structures, incomplete sentence structures, and variety of levels of development (LODs). To deal with such challenging text complexities, a domain-specific text splitting and stitching method was used to automatically simplify the hierarchically-complex text structures using a regular expressions-based pattern matching technique. An incompleteness-aware sequential dependency extraction method was used to capture dependency information from incomplete sentence structures to reduce the text ambiguities. A detail-aware LOD extraction method was used to automatically differentiate the LODs of sentences based on analyzing their grammatical moods using syntactic text features. The proposed IE method is composed of six primary steps (see Figure 5.1): text preprocessing, feature selection, identification of the

conceptual dependency structure of the target information, extraction rule development, extraction implementation, and performance evaluation.

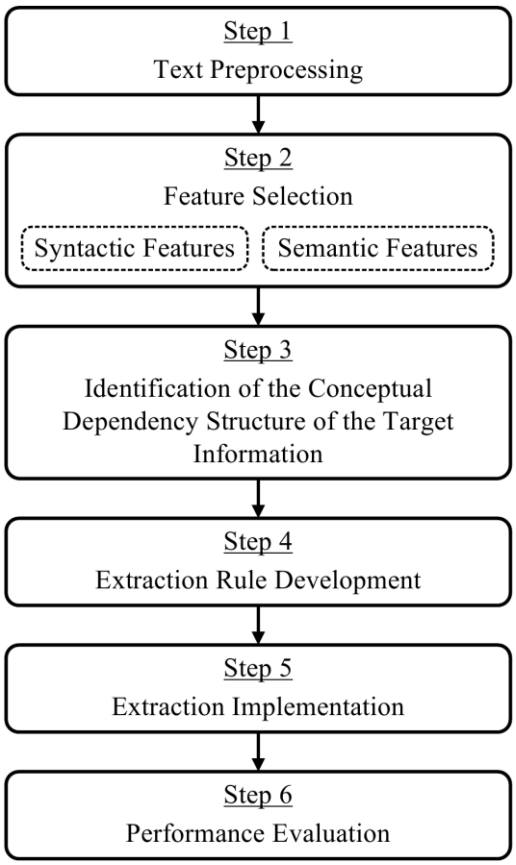


Figure 5.1. Proposed semantic information extraction methodology

5.2.1 Proposed Methods for Addressing the Challenging Text Complexities in Contract Specifications

5.2.1.1 Proposed Domain-Specific Text Splitting and Stitching Method

A domain-specific text splitting and stitching method is proposed and used to simplify the hierarchically-complex text structures in contract specifications prior to information extraction. The proposed method uses a regular expression-based pattern matching technique to automatically recognize the text features that signal splitting, and split the text to the most granular level. The

proposed method includes two steps: splitting and stitching. First, each article is split to the lowest subparagraph level, with each resulting split text containing the text from a lowest-level subparagraph, each super-level subparagraph, and the highest-level paragraph. Each resulting split text is then stitched with the numbers and titles of the corresponding section and article.

During splitting, the articles, paragraphs, and subparagraphs are automatically recognized based on the text feature patterns of their numbers and titles defined in the PageFormat, where such numbers and titles signal the splitting. Regular expressions are used to define the patterns. During stitching, the numbers and titles of sections/articles are stitched for two reasons. First, the titles may contain target information that needs to be extracted. For example, the article title "MINERAL WOOL BOARD INSULATION" contains the target information "mineral wool board", which is an instance of the "Subject" of a requirement. Second, the numbers and titles are used as a reference for the requirement during the reporting of the compliance checking results. As an example, Figure 5.2 shows the splitting and stitching for the article in Figure 2.1.

Figure 5.2. An example to illustrate the proposed domain-specific text splitting and stitching method

5.2.1.2 Proposed Incompleteness-Aware Sequential Dependency Extraction Method

An incompleteness-aware sequential dependency extraction method is proposed and used to extract target information from text with incomplete sentence structures. The proposed method adapts the sequential dependency extraction method in Section 4.2.4.2 to text with incomplete sentence structures. The key of sequential dependency extraction is to use the dependency information among target information to assist in defining text feature patterns and developing extraction rules to reduce ambiguities and enhance extraction performance. The target information, in the context of ACC, is the nine semantic information elements (SIEs) of ACC (Zhang and El-Gohary 2013): ″Subject", "Subject Restriction", "Compliance Checking Attribute", "Deontic Operator Indicator", "Quantitative Relation", "Comparative Relation", "Quantity Value", "Quantity Unit/Reference", and "Quantity Restriction For the detailed definitions of these SIEs, the readers are referred to Zhang and El-Gohary (2013) and Section 4.2.3.1. Table 5.1 shows an example of an SIE-represented energy requirement in a project's contract specifications. The dependency information is the conceptual dependency information among the SIEs that helps identify the extraction sequence of the SIEs, where the use of dependee SIEs to extract a depender SIE reduces ambiguities.

The proposed method is novel because it further uses incompleteness features (i.e., features of incomplete sentence structures), along with dependency information, to help define the feature patterns and develop extraction rules that reduce ambiguities in text with incomplete sentence structures. The incompleteness features can reduce ambiguities that cannot be solely addressed by

dependency information. For example, both “temperature” and “thermal resistivity” are potential candidates of the SIE “Compliance Checking Attribute”, but in the context of P2, only “thermal resistivity” (in bold) should be extracted as a “Compliance Checking Attribute” of a requirement; the “temperature” (in bold) refers to a testing condition. In rule R1, the comma “,”, which is an incompleteness feature that signals the beginning of a phrase, reduces this ambiguity. It imposes an additional matching condition that a candidate “Compliance Checking Attribute” should be preceded with an incompleteness feature (in addition to being succeeded by “Quantity Value” and a “Quantity Unit/Reference”, the dependency information).

- P2: “C. Form MW-1: Un-faced, Semi-Rigid Mineral-Wool Board Insulation of thickness indicated with width and length as required to suit job conditions: ASTM C612, Type 1A; with maximum flame-spread and smoke-developed indexes of 15 and zero, respectively, per ASTM E84; passing ASTM E136 for combustion characteristics.
  1. Minimum thermal resistivity (r) of 4.3 deg F x h x sq. ft./Btu x in. (29.8 K x m/W) at mean **temperature** of 75 degrees F (24 degrees C).
  2. Nominal density of 4 lb/cu. ft. (64 kg/cu. m), Types IA and IB, **thermal resistivity** of 4 deg F x h x sq. ft./Btu x in. at 75 deg F (27.7 K x m/W at 24 deg C).”
- R1: (Token.string == “,”) + (potentialCommercialBuildingEnergyEfficiencyAttribute):cr + IN + QuantityValue + QuantityUnit/Reference –> cr.ComplianceCheckingAttribute.,
  where,
  the pattern “(Token.string == “,”)” matches a comma, which is an incompleteness feature; “potentialCommercialBuildingEnergyEfficiencyAttribute” is a semantic feature that corresponds to a commercial building energy efficiency property (e.g., thermal resistivity), a concept in the ontology; “IN” is the POS tag for prepositions (e.g., of); and “QuantityValue” and “QuantityUnit/Reference”, the dependency information, match “4” and “deg F x h x sq. ft./Btu x in.”, respectively.

Table 5.1. Example of an SIE-Represented Energy Requirement

Split text	Semantic information element (SIE)	Energy requirement
Section 07 21 00 – Building Insulation 2.3 MINERAL-WOOL BOARD INSULATION D. Form MW-2: Foil-Faced, Semi-Rigid Mineral-Wool Board Insulation of thickness indicated with width and length as required to suit job conditions: ASTM C612, Type 1A; faced on one side with foil-scrim or foil-scrim-polyethylene vapor retarder having maximum permeance of 0.10 perm (5.75 x 10^{-9} g/Pa x s x m^2) when tested in accordance with ASTM E96; with maximum flame-spread and smoke-developed indexes of 25 and 5, respectively, per ASTM E84. 1. Minimum thermal resistivity (r) of 4.3 deg. F x h x sq. ft./Btu x in. (29.8 K x m/W) at mean temperature of 75 degrees F (24 degrees C).	Subject	foil-faced semi-rigid mineral-wool board
	Subject restriction	N/A
	Compliance checking attribute	thermal resistivity
	Deontic operator indicator	N/A
	Quantitative relation	N/A
	Comparative relation	minimum
	Quantity value	4.3
	Quantity unit/reference	deg. F x h x sq. ft./Btu x in.
	Quantity restriction	N/A

5.2.1.3 Proposed Detail-Aware Level of Development (LOD) Extraction Method

A detail-aware LOD extraction method is proposed and used to extract requirements that are in the target BIM LOD for compliance checking of BIMs. In this research, an LOD 350, as defined in the 2017 LOD specification (BIMForum 2017), is the target for both the requirements and the BIMs. The proposed method is novel because the target LOD is automatically differentiated based on analyzing the sentence grammatical moods in terms of syntactic text features. The proposed method uses an indirect way to extract information in LOD 350 from the text: (1) the extraction rules (see Section 5.2.2.4) are first used to extract all information regardless of LOD, (2) the information in LOD 400 or above is then recognized and removed from all the extracted information, leaving only information in LOD 350.

A rule-based pattern matching approach is used to recognize the information in LOD 400 or above. As discussed in Section 2.4.2, imperative sentences are used to convey information in LOD 400 or above, and the key feature of imperative sentences is action verb. As such, a set of tagging rules are used to recognize and tag action verbs. Subsequently, sentences that begin with action verbs, except those in an exclusion list, are recognized as containing information in LOD 400 or above. Action verbs that are poor indicators of information in LOD 400 or above (such as "provide") are included in a list of exclusion verbs. POS tags are used to define the feature patterns in the tagging rules. Figure 5.3 shows an example of a tagging rule, where (1) the feature pattern is shown in Lines 1-10, (2) the recognized action verbs are tagged with "actionVerb", as per Line 12, and (3) the list of exclusion verbs is shown in Line 7. This tagging rule was able, for example, to recognize that "hem" in S3 is an action verb, and accordingly S3 was removed.

- S3: "Hem exposed edges of flashing on underside 1/2 inch (13 mm)."

A new domain-specific POS tagger is proposed and used to support the recognition of action verbs, because a general POS tagger is limited in tagging domain-specific documents. In general, a POS tagger relies on two components to tag a word – a lexicon and a ruleset. The lexicon is a dictionary of words, where each word has a list of related POS tags (i.e., a word may have multiple word classes). The first tag in the list is the most likely word class for that word in one type of documents, and is the default tag assigned to a word. For example, "check NN VB VBP" is an item in the lexicon of the GATE Hepple POS Tagger (Cunningham et al. 2011), where the related POS tags "NN VB VBP" indicate that the word "check" has three word classes: it can be a singular or mass

noun (“NN”, which is the default tag), a based form verb (“VB”), or a non-3rd person singular present verb (“VBP”). The ruleset contains a set of conversion rules to convert the default tag to another tag, if its adjacent words in the text match certain feature patterns. For example, Hepple’s conversion rule “NN VB PREVTAG TO” (Cunningham et al. 2011) would convert the tag of “check” from its default “NN” to “VB”, if the previous word of “check” in the text has the POS tag “TO” (i.e., if “check” is preceded by “to”). This rule would still fail to correctly tag “check” as a verb in cases where it is not preceded by a “to”, such as this case: “Check the movement of the doors at both limits of travel”. As this example shows, a general POS tagger, like the Hepple’s, does not perform well in tagging action verbs because of missing domain-specific information in its lexicon and ruleset. For example, Hepple’s lexicon is built based on the text in the Wall Street Journal (Cunningham et al. 2011), in which “check” is more likely to be a noun referring to a written order directing a bank to pay money, hence “NN” is its default tag. In contrast, in the context of contract specifications, “check” is more likely to be a verb meaning “verify”. Therefore, the default for “check” in the new tagger is “VB”. The new domain-specific tagger was developed by building a domain-specific lexicon and ruleset based on Hepple’s.

Line number	LOD tagging rule No. 1	Explanation
1:	(	Beginning of feature pattern;
2:	({Token.string == "."}|{Token.string == ":"})	The first token must be a period or a colon, which is an incompleteness feature;
3:	({CC})?	The next token may be an optional coordinating conjunction;
4:	({Token.string == "do"}{Token.string == "not"})?	The next two tokens may be an optional phrase “do not”;
5:	({RB})?	The next token may be an optional adverb;
6:	({NN})?	The next token may be an optional singular or mass noun;
7:	(({VB, Token.string != "provide", Token.string != "include", Token.string != "refer", Token.string != "comply", Token.string != "conform", Token.string != "furnish", Token.string != "see", Token.string != "use", Token.string != "do", Token.string != "be", Token.string != "pass", !VB within paragraphTitle} | {VBP, Token.string != "provide", Token.string != "include", Token.string != "refer", Token.string != "comply", Token.string != "conform", Token.string != "furnish", Token.string != "see", Token.string != "use", Token.string != "do", Token.string != "be", Token.string != "pass", !VBP within paragraphTitle})):tempTag	The candidate action verb must (1) be a base form verb or a non-3rd person singular present verb; and (2) not be in the list of exclusion verbs: “provide, include, refer, comply, conform, furnish, see, use, do, be, pass”; and (3) not be part of a paragraph title. A tag “tempTag” is assigned to each candidate;
8:	{Token.kind != punctuation, Token.category != MD, !paragraphNumber}	The token immediately after a candidate must not be a punctuation, modal verb or paragraph number;
9:	({Token, Token.string != ":", Token.category != MD, Token.string != "are", Token.string != "is", !paragraphNumber})[4]	Any of the next four consecutive tokens must not be a colon, modal verb, “are”, “is”, or paragraph number;
10:	)	End of feature pattern;
11:	-->	The boundary of feature pattern;
12:	:tempTag.actionVerb = {rule = "LOD tagging rule No. 1"}	The candidates that match the feature pattern are action verbs, and are tagged as “**actionVerb**”;

Figure 5.3. Example of a tagging rule for recognizing action verbs

5.2.2 Implementation of the Proposed Semantic Information Extraction Algorithm

The proposed semantic IE method was implemented and tested in extracting building energy requirements from the contract specifications of an educational building project. The scope of testing was limited to two subtopics of commercial building energy efficiency: thermal insulation and lighting power. Only text from "Part 2 Products" of the specifications was used in testing because that part contains all related design requirements for compliance checking (see Section 2.4).

5.2.2.1 Text Preprocessing

The text was first classified, using the method explained in Section 3.2, to filter out articles in contract specifications that are not related to building energy efficiency. Figure 5.4 shows an example of an irrelevant article that was filtered out, because it prescribes fireproofing requirements.

2.2 SPRAYED FIRE-RESISTIVE MATERIALS
A. SFRM: Manufacturer's standard, factory-mixed, lightweight, dry formulation, complying with indicated fire resistance design, and mixed with water at Project site to form a slurry or mortar before conveyance and application.
1. Products: Subject to compliance with requirements, provide one of the following:
a. Grace, W. R. & Co. - Conn.; Grace Construction Products; Monokote MK-6 Series.
b. Isolatek International; Cafco 300.
c. Southwest Fireproofing Products Co.; Type 5GP.

•
•
•

6. Combustion Characteristics: ASTM E136.
7. Surface-Burning Characteristics: Comply with ASTM E84; testing by a qualified testing agency. Identify products with appropriate markings of applicable testing agency.
a. Flame-Spread Index: 10 or less.
b. Smoke-Developed Index: 10 or less.

Figure 5.4. Example of an irrelevant article

The proposed domain-specific text splitting and stitching method (Section 5.2.1.1) was first implemented to deal with the hierarchically-complex text structures. The proposed method was used to split the classified articles. The resulting split text was then preprocessed using three common text processing techniques: tokenization, sentence splitting, and morphological analysis.

Tokenization aims to split the raw text into tokens (e.g., words, numbers, punctuations, symbols, whitespace) for identifying sentence boundaries and preparing for POS tagging (Manning and Schütze 1999; Moens 2006). For example, the text "2. Nominal density of 4 lb/cu. ft." was tokenized into "'2' '.' 'Nominal' 'density' 'of' '4' 'lb' '/' 'cu' '.' 'ft' '.'" (whitespace tokens are not shown).

Sentence splitting aims to split the text into individual sentences based on the sentence boundary indicators (e.g., periods, exclamation marks, and question marks) (Jurafsky and Martin 2009). Sentence boundary indicators in contract specifications have similar ambiguities to those in energy codes. For example, both the periods in the subparagraph number "2." and in the measurement unit "lbs. per cubic foot" are not sentence boundary indicators. Therefore, a set of domain-specific sentence splitting rules were manually developed to correctly recognize and split sentences in specifications.

Morphological analysis aims to convert the words in derivational (e.g., affixes like "ly", "ion") or inflectional forms (e.g., plural, progressive) to their base form (Manning and Schütze 1999). For example, both "limiting" and "limiter" were converted to the base form "limit". Morphological analysis helped in recognizing and selecting the semantic features (Section 5.2.2.2.2) by mapping the morphologically-analyzed text to the ontology concepts. For example, through morphological analysis, "current limiting circuit breakers" in the natural text was mapped to the concept "current limit circuit breaker" in the ontology. The concept was then used as a semantic feature.

5.2.2.2 Feature Selection

The target information is recognized based on the features of the text (as per Section 5.2.2.4). Two types of text features were used: syntactic and semantic features.

5.2.2.2.1 Syntactic Features

Three types of syntactic features were used: POS tags, gazetteers, and domain-specific tags. POS tagging aims to assign a POS tag to each word based on its word class (e.g., adjective, noun, preposition) (Moens 2006). For example, the tag "JJ" was assigned to adjectives (e.g., "thermal"). In this research, a domain-specific POS tagger was developed and used (as per Section 5.2.1.3). Each POS tag was used as a syntactic feature.

A gazetteer refers to a list of words belonging to the same category (e.g., country names) (Wimalasuriya and Dou 2010). In this research, two gazetteers were manually developed: a comparison gazetteer and a measurement unit gazetteer. The first includes words/phrases that indicate comparison relationships (e.g., "exceed", "less than"). The second includes words/symbols that represent quantity units (e.g., "watt", "deg F x h x sq. ft./Btu x in."). Each gazetteer was assigned a tag, and each tag was used as a syntactic feature.

Fourteen domain-specific tags were defined in this research, and a set of tagging rules were developed to tag the text with these tags. An example of such tags is "domainSpecificCD", which refers to domain-specific cardinal numbers (e.g., "1-1/8"). Each tag was used as a syntactic feature.

5.2.2.2.2 Semantic Features

The semantic features of the text were captured using the building energy ontology (Section 4.2.2.2), which covers concepts that are related to commercial building energy efficiency. A partial

view of the energy ontology is shown in Figure 5.5. The web ontology language in-memory (OWLIM) Ontology Editor of the General Architecture for Text Engineering (GATE) (Cunningham et al. 2011), a text processing platform, was used to build the ontology. The ontology was then processed by the GATE "OntoRoot Gazetteer" module to create a concept gazetteer, as well as parse the hierarchical is-a relationships among the concepts. Each concept in the gazetteer was used as a semantic feature. The is-a relationships helped recognize subconcept relationships for the extraction (Section 5.2.2.4).

Figure 5.5. Partial view of the commercial building energy ontology

5.2.2.3 Identification of the Conceptual Dependency Structure of the Target Information

A conceptual dependency structure (CDS) was developed to represent the dependency information among the target information – the nine SIEs (see Section 4.2.3.1). Such dependency information helps to identify the extraction sequences of the SIEs. The CDS of Section 4.2.3.2 was adapted; the text in contract specifications was analyzed to identify the dependency relationships among the SIEs and accordingly the CDS of Section 4.2.3.2 was modified. Figure 5.6 shows the developed CDS for specifications, in which arrows represent the dependency relationships and numbers indicate the extraction sequence. Compared to the CDS in Section 4.2.3.2, the CDS for specifications is different in two main ways. First, it contains 50% more dependency relationships among all SIEs, where most of these extra relationships involve two SIEs – "Deontic Operator Indicator" and "Quantitative Relation". For example, "Deontic Operator Indicator" and "Quantitative Relation" depend on two new SIEs – "Quantity Value" and "Compliance Checking Attribute". Second, "Subject" is moved up in the CDS, before "Deontic Operator Indicator", indicating that "Deontic Operator Indicator" is no longer used as a dependency information in the extraction of "Subject". These two dependency differences are due to the differences in sentence structures and writing style across codes and specifications (as discussed in Section 2.4). For example, modal verbs (which are instances of "Deontic Operator Indicator") are less likely used in specification sentences (e.g., the modal verb "shall" is replaced by a colon in streamlined writing), which results in the need to use other information (e.g., incompleteness features) to extract "Subjects".

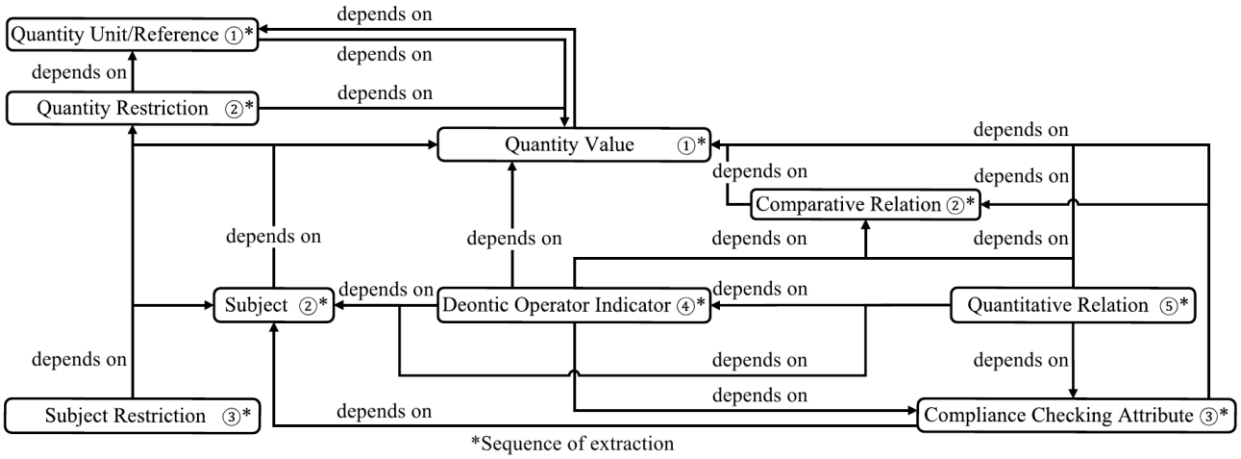


Figure 5.6. Conceptual dependency structure

5.2.2.4 Extraction Rule Development

The extraction rules were manually developed after analyzing the text feature patterns in a sample of text (called development data). The development data included 400 sample sentences from five contract specifications. Each extraction rule has two sides: the left side models the feature patterns using regular expressions, and the right side indicates the target information that needs to be extracted. For example, R2 extracted "permeance" from S4, as an instance of "Compliance Checking Attribute". The rule states that if a commercial building energy efficiency property is preceded by a "Comparative Relation" and is followed by "IN", "Quantity Value", and "Quantity Unit/Reference", the property should be extracted as an instance of "Compliance Checking Attribute".

- R2: ComparativeRelation + (potentialCommercialBuildingEnergyEfficiencyAttribute):cr + IN + QuantityValue + QuantityUnit/Reference –> cr.ComplianceCheckingAttribute.,
  where,
  "potentialCommercialBuildingEnergyEfficiencyAttribute" is a semantic feature that matches a concept in the ontology; "IN" is a POS tag that matches prepositions like "of"; "ComparativeRelation", "QuantityValue", and "QuantityUnit/Reference" are three SIEs that

are used as dependency information (matching to "maximum", "0.10", and "perm", respectively, in S4); and "cr" is the pointer set to the semantic feature "potentialCommercialBuildingEnergyEfficiencyAttribute".

- S4: "Faced on one side with foil-scrim or foil-scrim-polyethylene vapor retarder having maximum permeance of 0.10 perm (5.75×10^{-9} g/Pa x s x m^2) when tested in accordance with ASTM E96."

During the development of extraction rules, the proposed incompleteness-aware sequential dependency extraction method (Section 5.2.1.2) and detail-aware LOD extraction method (Section 5.2.1.3) were considered and implemented to deal with incomplete sentence structures and variety of LODs. In the LOD extraction, (1) 71 domain-specific words, with their related POS tags, and six conversion rules were added to the lexicon and ruleset of the GATE Hepple POS Tagger; and (2) five tagging rules were developed to recognize the action verbs of imperative sentences.

To address conflicts that may arise in the extraction, twelve domain-specific conflict resolution (CR) rules were developed. These rules can be categorized into two types – global and local CR rules. Global rules are applicable to all SIEs, while local rules are only applicable to specific SIEs. For example, CR1 is a global rule, which applies to all SIEs. It extracted "2.0" (rather than "0") from S5, as an instance of "Quantity Value", because "2.0" is the longest matching instance. In contrast, CR7 is a local rule, which applies to "Subjects" only. It extracted the more specific "fluorescent electronic ballast" (rather than "ballast") from S6, as an instance of "Subject".

- S5: "Sprayed Polyurethane Foam Sealant: 1- or 2-component, foamed-in-place, polyurethane foam sealant, 1.5 to 2.0 lb/cu. ft. (24 to 32 kg/cu. m) density."
- CR1: "If there are multiple instances that match the text feature patterns, only the longest matching instance should be extracted."

- S6: “2.9 FLUORESCENT ELECTRONIC BALLAST F. Ballast Requirements: 5. Ballast power factor shall be greater than 95%.”
- CR7: “If there are multiple matching instances for “Subject”, the instance that corresponds to the lower-level concept (i.e., the more specific concept in the ontology) should be extracted.”

5.2.2.5 Extraction Implementation

The proposed semantic IE method was implemented in a Java-based platform. The application programming interfaces (APIs) of the following modules in the “a nearly-new information extraction” (ANNIE) system of GATE (Cunningham et al. 2011) were used to implement the methods in Sections 5.2.2.1-5.2.2.4, including the ANNIE English Tokeniser, ANNIE Sentence Splitter, GATE Morphological Analyser, ANNIE POS Tagger, ANNIE Gazetteer, OntoRoot Gazetteer, and JAPE Transducer. Each of the above modules may have a set of initialization parameters, and each parameter was experimentally set in this research. For example, the values of the parameters “lexiconURL” and “rulesURL” in the ANNIE POS Tagger were set to the developed domain-specific lexicon and rules. The two gazetteers (Section 5.2.2.2) were added to the ANNIE Gazetteer. The domain-specific sentence splitting rules (Section 5.2.2.1), the tagging rules for the fourteen domain-specific tags (Section 5.2.2.2), the tagging rules for recognizing the action verbs in the detail-aware LOD extraction method (Section 5.2.2.4), and the extraction rules (Section 5.2.2.4) were developed in the grammar of Java Annotation Patterns Engine (JAPE) (Cunningham et al. 2011) using the JAPE editor – Vim (Vi IMproved) (Robbins et al. 2008), and were added to the JAPE Transducer. The domain-specific text splitting and stitching (Section 5.2.2.1) and the CR rules (Section 5.2.2.4) were implemented in separate Java programs.

5.2.2.6 Performance Evaluation

A testing dataset was used to evaluate the performance. The dataset was prepared based on the contract specifications of an educational building project in Illinois. The sections from the divisions 07 and 26 of the specifications were selected, because they contain thermal insulation and lighting power requirements. A total of 393 requirements were manually collected from these sections and a ratio of 2/5 was used to randomly sample a number of requirements for testing, resulting in 148 requirements in the testing dataset. In order to develop the gold standard, the requirements were annotated by three annotators – the author and two other researchers. An initial inter-annotator agreement of 88% in F-measure was achieved, which is considered a sufficient score. "An F-measure of 0.8 or above is generally considered sufficient inter-annotator agreement" (Pestian et al. 2012). The discrepancies were then discussed and resolved until consensus was reached, thereby achieving final full annotator agreement. As an illustration, Table 5.2 shows three SIE-represented requirements from the gold standard, which come from one article after domain-specific text splitting and stitching in Figure 5.2.

Recall and precision were used for measuring the performance (Maynard et al. 2006; Moens 2006). Recall refers to the percentage of the total number of correctly extracted instances out of the total number of instances in the gold standard. Precision refers to the percentage of the total number of correctly extracted instances out of the total number of extracted instances. The confidence interval (p) was further calculated for recall and precision to test the statistical significance of the results (Goutte and Gaussier 2005). The Wilson score without continuity correction (Wilson 1927) was

used for measuring the confidence intervals because it is both computationally simple and satisfactory (Newcombe 1998). It was calculated using Equation 5.1, where p_0 refers to the values of precision or recall, $q_0 = 1 - p_0$, λ is the critical value for the confidence interval, n refers to either the total number of extracted instances (for calculating p for precision) or the total number of instances in the gold standard (for calculating p for recall), and $t = \lambda^2/n$.

$$p = \frac{p_0 + t/2}{1+t} \pm \frac{\sqrt{p_0 q_0 t + t^2/4}}{1+t} \quad (5.1)$$

Table 5.2. Examples of Three SIE-Represented Requirements in the Gold Standard

Semantic information element	**Article after domain-specific text splitting and stitching (requirement)**		
Split text	Section 07 21 00 – Building Insulation 2.3 MINERAL-WOOL BOARD INSULATION D. Form MW-2: Foil-Faced, Semi-Rigid Mineral-Wool Board Insulation of thickness indicated with width and length as required to suit job conditions: ASTM C612, Type 1A; faced on one side with foil-scrim or foil-scrim-polyethylene vapor retarder having maximum permeance of 0.10 perm (5.75×10^{-9} g/Pa x s x m^2) when tested in accordance with ASTM E96; with maximum flame-spread and smoke-developed indexes of 25 and 5, respectively, per ASTM E84. 2. Nominal density of 4 lb/cu. ft. (64 kg/cu. m), Types IA and IB, thermal resistivity of 4 deg F x h x sq. ft./Btu x in. at 75 deg F (27.7 K x m/W at 24 deg C).		
Requirement	R1	R2	R3
Subject	foil-scrim or foil-scrim-polyethylene vapor retarder	foil-faced semi-rigid mineral-wool board	foil-faced semi-rigid mineral-wool board
Subject restriction	N/A	N/A	N/A
Compliance checking attribute	permeance	nominal density	thermal resistivity
Deontic operator indicator	N/A	N/A	N/A
Quantitative relation	N/A	N/A	N/A
Comparative relation	maximum	greater than or equal[1]	greater than or equal[1]
Quantity value	0.10	4	4
Quantity unit/reference	perm	lb/cu. ft.	deg F x h x sq. ft./Btu x in.
Quantity restriction	N/A	N/A	N/A

1. “greater than or equal” was used as a default comparative relation if it is implicit in text.

5.3 Experimental Results and Analysis

5.3.1 Performance Results

The experimental results are summarized in Table 5.3. A total of 99 extraction rules were developed. The gold standard contained a total of 787 instances for all nine SIEs. An overall performance of 96.8% recall [with confidence interval as (95.4%, 97.8%) at 95% confidence level] and 97.6% precision [with confidence interval as (96.2%, 98.4%) at 95% confidence level] was achieved. This promising performance indicates that the proposed semantic information extraction method is successful in extracting building energy requirements from contract specifications.

Table 5.3. Experimental Results of Extracting Building Energy Requirements from Contract Specifications

Parameter/measure	Subject	Subject restriction	Compliance checking attribute	Deontic operator indicator	Quantitative relation	Comparative relation	Quantity value	Quantity unit/reference	Quantity restriction	Total	Average
Information extraction rules	20	4	11	4	9	6	20	20	5	99	
Instances in gold standard	148	35	108	15	29	148	148	148	8	787	
Instances extracted	137	33	107	15	29	147	156	149	8	781	
Instances correctly extracted	132	33	104	15	29	147	147	147	8	762	
Precision	96.4%	100.0%	97.2%	100.0%	100.0%	100.0%	94.2%	98.5%	100.0%		97.6%
Recall	89.2%	94.3%	96.3%	100.0%	100.0%	99.3%	99.3%	99.2%	100.0%		96.8%

5.3.2 Effects of Incompleteness-Aware Sequential Dependency Extraction

The experimental results show that the use of incompleteness features along with dependency information was effective in reducing ambiguities in contract specifications. Dependency

information alone may not be sufficient in reducing ambiguities. For example, as shown in Figure 5.7, the letter "A" (in bold), which is a potential instance of "Quantity Unit/Reference", may create ambiguity. It may be a paragraph number (e.g., "A. Item 1: Busway:"), a part of a designation number (e.g., "ASTM C612, Type 1A & 1B."), or the abbreviation of "ampere" (e.g., "C. Auxiliary contacts, rated 10 A, 250 VAC,"). But, only in the last case it should be extracted as an instance of "Quantity Unit/Reference". The proposed method was able to avoid such ambiguities using incompleteness features, in addition to the dependency information.

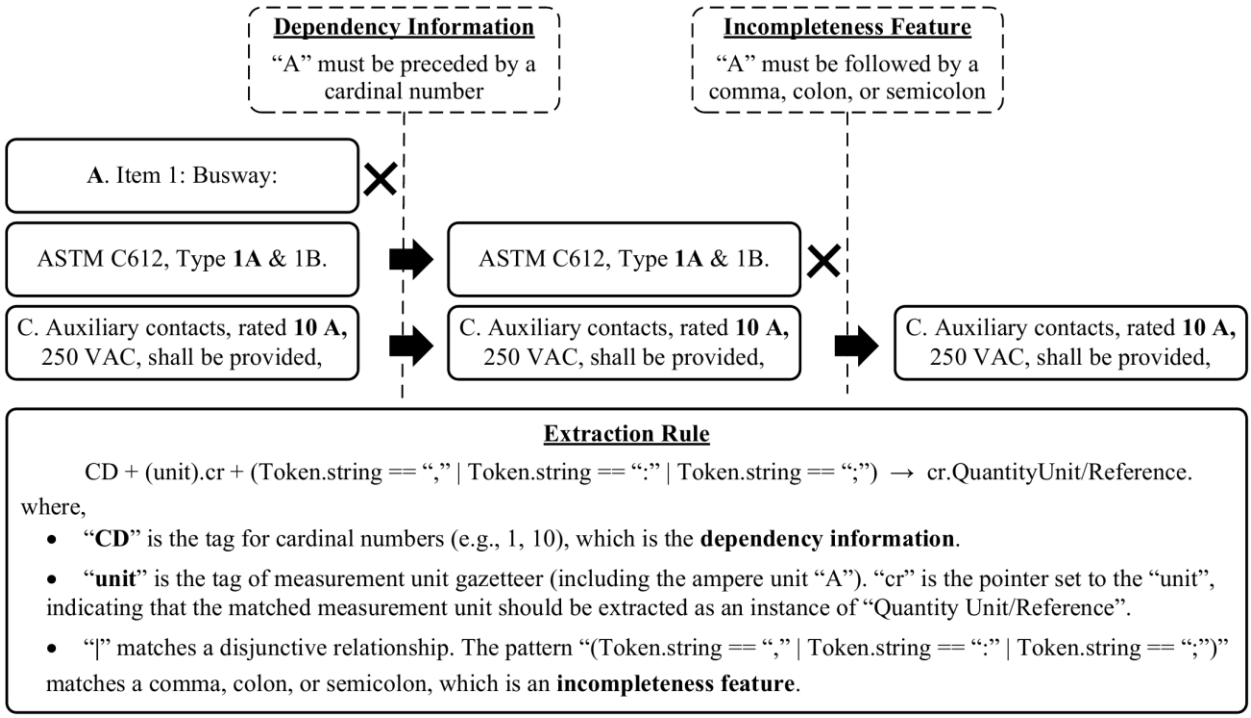


Figure 5.7. An example of using incompleteness features along with dependency information in reducing ambiguities

5.3.3 Effects of Detail-Aware LOD Extraction

The experimental results show that no extraction errors were caused by the detail-aware LOD

extraction, which indicates the success of the proposed method in extracting information in LOD 350. However, a few errors occurred in tagging the action verbs for two reasons. First, the incomplete sentence structures make it challenging to tag words that have word-class ambiguity. As discussed in Section 5.2.1.3, (1) some words may have multiple word classes, which may result in word-class ambiguity, and (2) the POS tagger relies on the context of a word (i.e., its adjacent words in text) to determine its word class. However, text with incomplete sentence structures may lack the sufficient contextual information to determine the correct word class, which may result in POS tagging errors. For example, the word "safeguard" has word-class ambiguity; it could be either a noun or a verb. It was incorrectly tagged as an action verb in T1 because it is the first word of the sentence and has only partial context information (e.g., "Gravel", "Stop", "System") to determine its word class. A semantic POS tagger could have helped address this word-class ambiguity; it could have used domain-specific knowledge (e.g., in the form of ontology or gazetteer) to disambiguate the word classes. For example, "Safeguard" is the trademark name for a gravel stop system, thereby more likely to be a noun.

- T1: "a. Safeguard Gravel Stop System by W.W. Hickman Co."

Second, a few verbs that were included in the verb exclusion list were occasionally used in imperative sentences that prescribe requirements in LOD 400 or above. For example, the verb "provide" was added to the verb exclusion list, but it should have been tagged as an action verb in T2, because the imperative sentence contains a fabrication requirement for flat lock seams (i.e., a requirement in LOD 400 or above).

- T2: “C. Provide flat lock seams, except corners. Fabricate corners minimum 18 inches x 18 inches (450 mm x 450 mm) mitered and sealed as one piece.”

5.3.4 Error Analysis

An error analysis was conducted to identify the sources of extraction errors. Six sources of errors were identified: dependency information, uncommon patterns, conflict resolution errors, extraction tool errors, ambiguous concept representation, and coreference ambiguity. First, missing the extraction of the dependee may result in missing the extraction of the depender. For example, in T3, “maximum” was not extracted as a “Comparative Relation”, because its dependees, “4” (“Quantity Value”) and “"” (the unit symbol for feet, a “Quantity Unit/Reference”), were not extracted.

- T3: “2. Board size: 4'-0" x 4'-0" maximum.”

Second, uncommon patterns may result in missing the extraction of target information. In comparison to energy codes, uncommon patterns are more likely to occur in contract specifications due to their special text representations. For example, “4” and “"” were not extracted as instances of “Quantity Value” and “Quantity Unit/Reference” from T3 because of this rare pattern (“4'-0" x 4'-0"”) for representing values and units.

Third, conflict resolution (CR) errors are due to missing a CR rule or due to an error caused by a CR rule. Missing a CR rule may result in extracting irrelevant information. For example, in T4, four instances of “Quantity Value” (i.e., “0.10”, “25”, “2.25”, “4.3”) and three instances of “Quantity Unit/Reference” (i.e., “perm”, “lb/cu. ft.”, “deg F x h x sq. ft./Btu x in.”) were extracted,

in which “25” was incorrect, because it refers to the “flame spread index”, an attribute for an irrelevant fire safety requirement. The current CR rules were not able to deal with such errors. On the other hand, errors caused by the CR rules may result in missing the extraction of target information. For example, CR10 caused an extraction error for T5. “Impedance” was incorrectly deleted from T5 because it is contained in “impedance tolerance”. Both “impedance” and “impedance tolerance” should have been extracted as “Compliance Checking Attribute” instances, as each one refers to a requirement.

- T4: “C. Form FG-4: Foil-Faced, Glass-Fiber Board Insulation of thickness indicated with width and length as required to suit job conditions: ASTM C612, Type IA; faced on one side with foil-scrim-kraft or foil-scrim-polyethylene vapor retarder having maximum permeance of 0.10 perm (5.75×10^{-9} g/Pa x s x m^2) when tested in accordance with ASTM E96, with maximum flame-spread and smoke-developed indexes of 25 and 50, respectively, per ASTM E84.
  1. Nominal density of 2.25 lb/cu. ft. (36 kg/cu. m), thermal resistivity of 4.3 deg F x h x sq. ft./Btu x in. at 75 deg F (29.8 K x m/W at 24 deg C).”
- T5: “H. Impedance: Minimum 5.75 percent. The impedance tolerance shall be plus or minus 7.5 percent.”
- CR10: “If there are two instances for the SIE “Compliance Checking Attribute” and one instance is contained in another, keep the longer instance and delete the shorter one.”

Fourth, few errors are caused by the extraction tool errors. For example, in T6, the “time delay” was not extracted as a “Compliance Checking Attribute” because the GATE Morphological Analyzer failed to recognize the semantic feature “time delay”.

- T6: “g. When required by National Electrical Code or indicated on Project Drawings, the control system shall include a ground fault monitoring relay. The relay shall be adjustable from 100-1200 amps, and include adjustable time delay of 0-1.0 seconds.”

Fifth, a concept may have an ambiguous representation in the text, which may result in failure to

recognize its semantic features and consequently failure in the extraction. For example, in T7, the concept “recessed mounted panelboard” was represented as “Panelboard, recessed mount”, which resulted in failure to recognize the semantic feature and extract “Panelboard, Recessed Mount” as a “Subject” instance. This indicates that additional processing may be required, in addition to morphological analysis, to help recognize the semantic features of the text with ambiguous concept representation.

- T7: “1. Panelboard, recessed mount, 208/120 volt, 3 phase, 4 wire, S/N, ground bus, copper bus, bolt-on breakers, NEMA 1 enclosure, see schedules for size and configuration, see Specifications for additional information.”

Sixth, coreference ambiguity may result in the extraction of inaccurate information. For example, in T6, “relay” was incorrectly extracted as an instance of “Subject”, which is inaccurate – because “relay” actually refers to “ground fault monitoring relay”, referenced in an earlier sentence. Coreference ambiguity is currently out of the research scope.

5.3.5 Performance Comparison: Contract Specifications versus Energy Codes

The performance of extracting information from contract specifications was further compared to that from energy codes (in Chapter 4), as per Figure 5.8. For energy codes, a 97.4% recall [with confidence interval as (95.9%, 98.4%) at 95% confidence level] and 98.5% precision [with confidence interval as (97.2%, 99.2%) at 95% confidence level] was achieved. This is a comparable level of performance to that achieved for contract specifications (Section 5.3.1). This indicates that a semantic rule-based approach is potentially scalable across different types of

regulatory documents, after the necessary adaptations are conducted. Some level of adaptation, such as using a new or extended ontology and/or modifying the conceptual dependency structure (CDS), is essential to address the change in the text characteristics as the type of document changes.

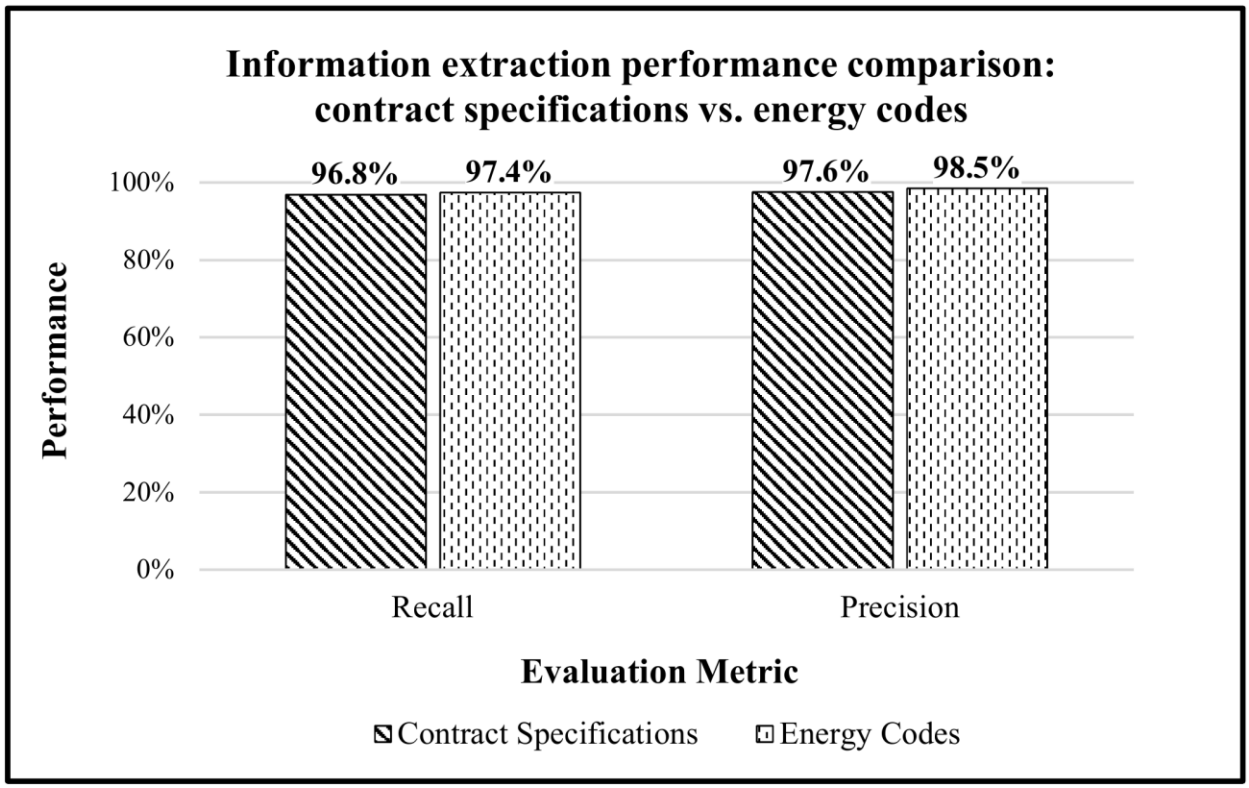


Figure 5.8. Information extraction performance comparison: contract specifications vs. energy codes

CHAPTER 6 – AUTOMATED SEMANTIC INFORMATION ALIGNMENT: BIM-REQUIREMENT ALIGNMENT

6.1 Comparison to the State of the Art

Despite the importance of those efforts in BIM-requirement alignment (as per Section 2.7), their information alignment approaches are limited in one or more of the following three ways. First, all of these approaches require some degree of manual effort. For example, Dimyadi et al. 2016b; Lee et al. 2015; Nawari 2012; Lee et al. 2016; and Preidel and Borrmann 2016 require manual specification of the alignment, by domain experts, using predefined functions/languages. Manual approaches are typically time-consuming, costly, and unscalable (Beach et al. 2015; Eastman et al. 2009). Second, many of these efforts are somewhat rigid. For example, Beach et al. 2015; Pauwels et al. 2011; Tan et al. 2010; Delis and Delis 1995; Goel and Fenves 1969; Ding et al. 2006; See 2008; Liebich et al. 2002; and SMC 2009 use pre-defined mappings or mapping rules. Rigid approaches lack sufficient flexibility and adaptability to allow for successful implementation across BIM instances, different types of regulations, and changes/updates to the BIM or the regulations (Garrett et al. 2014; Dimyadi et al. 2016b). Third, several of these efforts [especially those by software vendors such as Ding et al. (2006), See (2008), Liebich et al. (2002), and SMC (2009)] use proprietary methods. Proprietary methods lack the needed transparency to enable the users to check the correctness of the alignment (Dimyadi et al. 2016b).

6.2 Proposed Method for Semantic Information Alignment of BIMs to Energy Requirements

6.2.1 Proposed Method for Fully-Automated Semantic Information Alignment

To address the aforementioned gaps, this research proposes a fully-automated semantic information alignment method to align the concept representations of the BIMs to the concept representations of the requirements in the energy codes and contract specifications for supporting EnergyACC in construction. The proposed method aims to align the IFC-represented design information instances to the regulatory information. The proposed method is novel in two ways. First, it captures domain knowledge to automatically interpret the meaning of concepts and recognize the candidate design information instances that are potentially matched to the regulatory concepts, and uses deep learning to capture the semantics behind the words and accordingly measure semantic similarity and select the matches. Second, it uses supervised and unsupervised searching algorithms to automatically identify the relationships that create instance pairs, and uses network modeling to model and group the instance pairs that are linked to the associated concepts in a regulatory requirement. The proposed method includes two primary submethods: (1) a method for first-level, simple alignment (individual-individual matching): matching single design information instances to single regulatory concepts, and (2) a method for final, complex alignment (group-group matching): recognizing the regulatory concepts that belong to one requirement, and linking the matched design information instances to these associated regulatory concepts. First, the first-level simple alignment method is used to align single design information instances to

single regulatory concepts. Domain knowledge is used to interpret the meaning of concepts to recognize potential matching design information instances. An empirical method is used to analyze the patterns of semantic similarity to select the matching instances, in which a deep learning technique is used to measure the semantic similarity. Second, the final complex alignment method is used to recognize the groups of instances that belong to a regulatory requirement. Supervised searching and unsupervised searching is used to identify the instance pairs, and network modeling is used to group and link the identified instances pairs to the associated regulatory concepts in the regulatory requirement.

6.2.1.1 First-Level Simple Alignment

The first-level simple alignment method aims to align single design information instances (i.e., the instances of the IFC entities in an instance of a BIM, which is called thereafter "BIM instances") to single regulatory concepts. There are two types of regulatory concepts: object concepts and property concepts. An object concept refers to an object such as a building element (e.g., duct). A property concept refers to a property of an object (e.g., thermal resistance is a property of a duct). Accordingly, two types of BIM instances are defined: object instances and property instances. An object instance refers to an instance of an IFC entity that matches or aligns to an object concept, including instances of IfcProduct, IfcProductType, IfcSystem, and IfcMaterial. A property instance refers to an instance of an IFC entity that matches or aligns to a property concept, including instances of IfcSimpleProperty, IfcPhysicalSimpleQuantity, IfcMaterialProperties, and IfcMaterialLayer.

The proposed method includes two primary steps: concept interpretation and matching, and semantic similarity analysis. Concept interpretation and matching aims to interpret the meaning of regulatory concepts and accordingly to select an initial set of candidate matches. A match is defined, in this research, as a BIM instance that is matched or aligned to a concept. Semantic similarity analysis aims to assess the semantic similarity for each candidate pair (BIM instance and regulatory concept), and accordingly to select the matches for regulatory concepts. A domain ontology (called commercial building energy ontology) is used to support both steps. Figure 6.1 illustrates the method for first-level simple alignment.

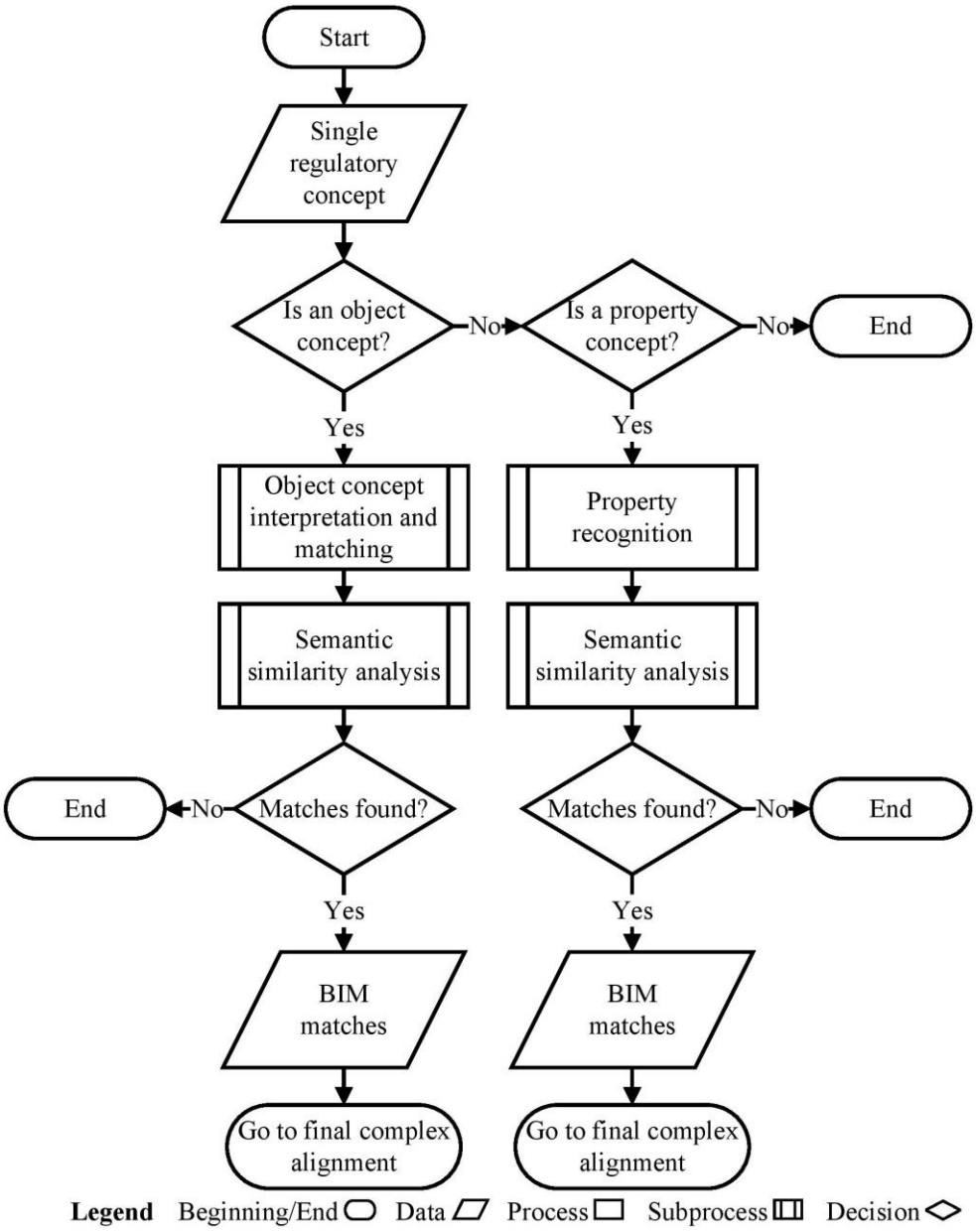


Figure 6.1. Method for first-level simple alignment

6.2.1.1.1 Concept Interpretation and Matching

Concept interpretation and matching aims to interpret the meaning of a regulatory concept to recognize all candidate matches (BIM instances). The proposed approach uses ontology and bSDD to capture domain knowledge for the interpretation of the meaning of concepts to automatically

recognize the candidate matches. This step includes: object concept interpretation and matching, and property recognition. The “subject” and “property” type bSDD concepts, as discussed in Section 2.5, were used in this research to search for the matching IFC entities.

6.2.1.1.1.1 Object Concept Interpretation and Matching

Object concept interpretation aims to recognize all candidate object instances (candidate matches) for a regulatory object concept. Three methods for object concept interpretation and matching are proposed and used, as shown in Figure 6.2: (1) concept interpretation using bSDD direct searching for finding perfect matches; (2) concept interpretation using ontology-based concept decomposition for finding parent matches; and (3) concept interpretation using superconcept information for finding parent matches. The three methods are conditionally dependent: if a preceding method fails to find a match, the following method is used. If all three methods fail, no match to that concept has been found.

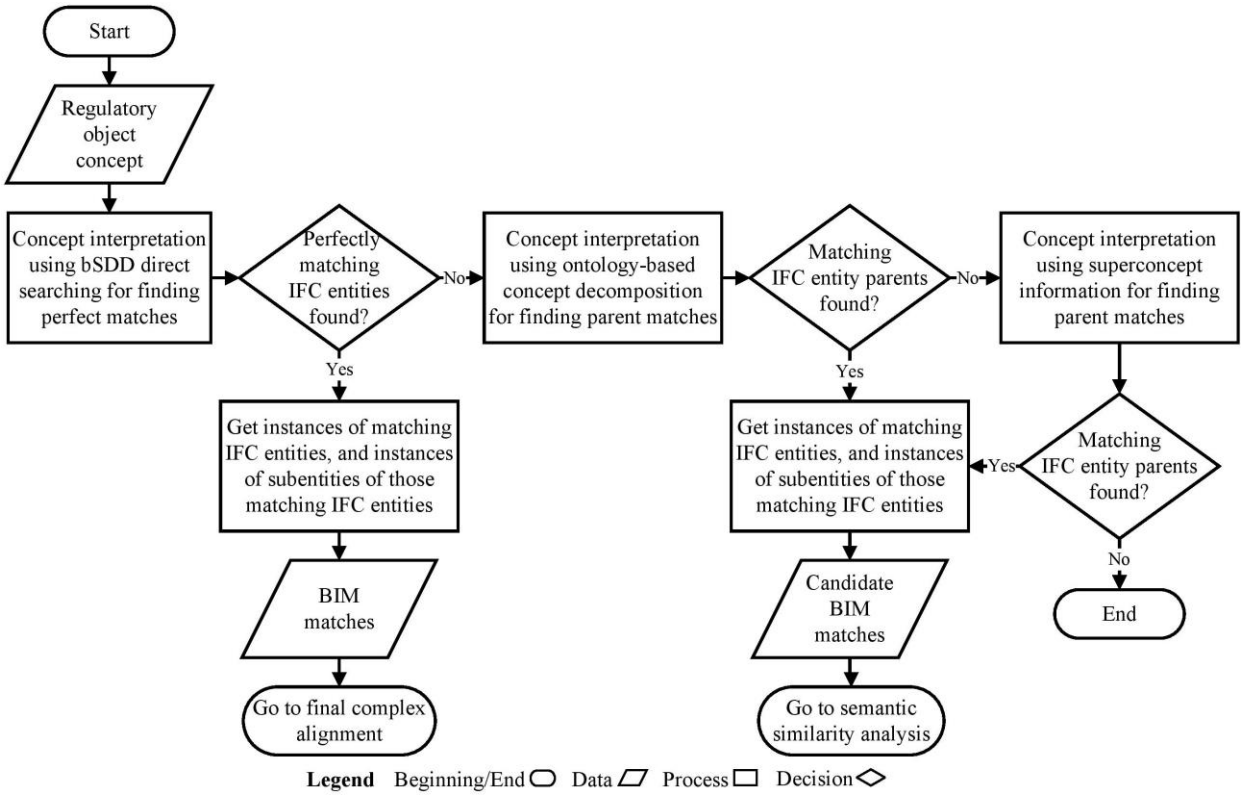


Figure 6.2. Method for object concept interpretation and matching

Method 1: Concept interpretation using bSDD direct searching for finding perfect matches. This method uses the bSDD API to directly search the bSDD to find the perfectly matching IFC entities (100% match to the morphologically-analyzed regulatory concept). The instances of these matching IFC entities (including instances of their subentities) are the matches for the regulatory object concepts. First, the object concepts (i.e., the terms in the concept name) are morphologically analyzed to collapse the different derivational (e.g., affixes like “ly”, “ion”) and inflectional forms (e.g., plural, progressive) of each term to its base form. Then, three techniques are used for searching: (1) using potentially-equivalent concept names: a set of potentially-equivalent concept names are defined based on the original and base forms of the terms. For example, the following

four concept names are potentially equivalent to the original concept name “lighting fixtures”: “lighting fixtures”, “lighting fixture”, “light fixtures”, and “light fixture”. Using this technique, instances of the IfcLightFixture and IfcLightFixtureType entities would be recognized as perfect matches to “lighting fixtures”; (2) using bSDD synonyms: synonyms of bSDD concept names are used in the search for matching IFC entities. For example, the synonyms of the bSDD concept “lighting fixture” include “luminaire”. Using this technique, instances of the IfcLightFixture and IfcLightFixtureType entities would be recognized as perfect matches to “luminaire”; and (3) using ontology-based equivalent concepts: similar to bSDD synonyms, equivalent concepts in the ontology are used in the search for matching IFC entities. For example, in the ontology, the concept “beam” is equivalent to the concept “girder”. Using this technique, instances of the IfcBeam and IfcBeamType entities would be recognized as perfect matches to “girder”.

Method 2: Concept interpretation using ontology-based concept decomposition for finding parent matches. If Method 1 fails to find any matching IFC entities, Method 2 is used. This method uses ontology-based concept decomposition to find matching IFC entity parents. The instances of those IFC entity parents (including instances of their subentities) are the candidate matches for the regulatory object concepts. A given object concept (e.g., “metal-framed roof”) is decomposed into two parts: (1) a core inner concept carrying the most important meaning of the given concept (e.g., “roof”); and (2) the remaining part of that given concept (e.g., “metal-framed”), which could be viewed as the property information of that core inner concept. The core inner

concept name is then used to recognize the matching parents using Method 1.

Two steps are proposed and used to decompose a concept based on the ontology. First, ontology parsing is used to find all potential inner concepts for a given concept (using morphological analysis as necessary). Second, conflict resolution methods are used to select the core inner concept from those potential ones. Three conflict resolution methods are proposed and used: (1) iterative concept name reduction: if there are two (or more) equivalent concepts among those inner concepts, the longer concept(s) (i.e., concept with the longer name) is replaced by the shorter one. This is based on the hypothesis that if a concept uses more terms to express the same meaning, it contains redundant information. For example, the first round of ontology parsing finds the following inner concepts for the concept "metal building roof assembly": "metal", "metal building", "metal building roof", "building", "roof," "roof assembly", among which "roof assembly" and "roof" are equivalent. Accordingly, the concept "metal building roof assembly" is reduced to "metal building roof"; (2) concept removal: any inner concept that is inside another inner concept is removed. For example, among the four inner concepts of "metal building roof" – "metal", "metal building", "building", and "roof" – the concepts "metal" and "building" are removed; and (3) concept selection: if there are multiple inner concepts remaining (after iterative concept name reduction and concept removal), the rightmost inner concept (with reference to its position in the given concept) is selected, because the rightmost terms in a concept usually carry the most important meaning. For example, among the remaining inner concepts, "metal building" and "roof", the

rightmost concept “roof” is selected as the core one. Accordingly, searching for the concept “roof” in the bSDD (using Method 1) would return IfcRoof and IfcRoofType. As such, using Method 2, the IfcRoof and IfcRoofType entities would be recognized as the matching parents of “metal building roof assembly”.

Method 3: Concept interpretation using superconcept information for finding parent matches. If Method 2 fails to find any matching IFC entities, Method 3 is used. This method uses superconcept information (in the ontology and the bSDD) to find matching IFC entity parents. The instances of these IFC entity parents (including instances of their subentities) are the candidate matches for the object concepts. The “specialization” relationship, as discussed in Section 2.5, was used in this research to search for the superconcepts of a bSDD concept. The search is conducted in a recursive manner: if a lower-level superconcept fails to find a matching parent, a higher-level superconcept is used in the search – until the root concept is reached. For example, using Method 3, the IfcMaterial entity would be recognized as the matching parent of “radiant panel” (using the search term “material”, where “material” is a superconcept of “radiant panel” in the ontology).

6.2.1.1.1.2 Property Recognition

Property recognition aims to recognize all candidate property instances (candidate matches) for a regulatory property concept. The recognition of candidates is conducted in an indirect way, as per Figure 6.3: (1) the object concepts associated with a given property concept are identified, (2) the matching object instances are recognized, and (3) all the property instances that are associated with

each of the matching object instances are captured as candidate property instances. Property instances without any quantitative property values are excluded from the candidate list, because this research only focuses on checking the compliance with quantitative requirements.

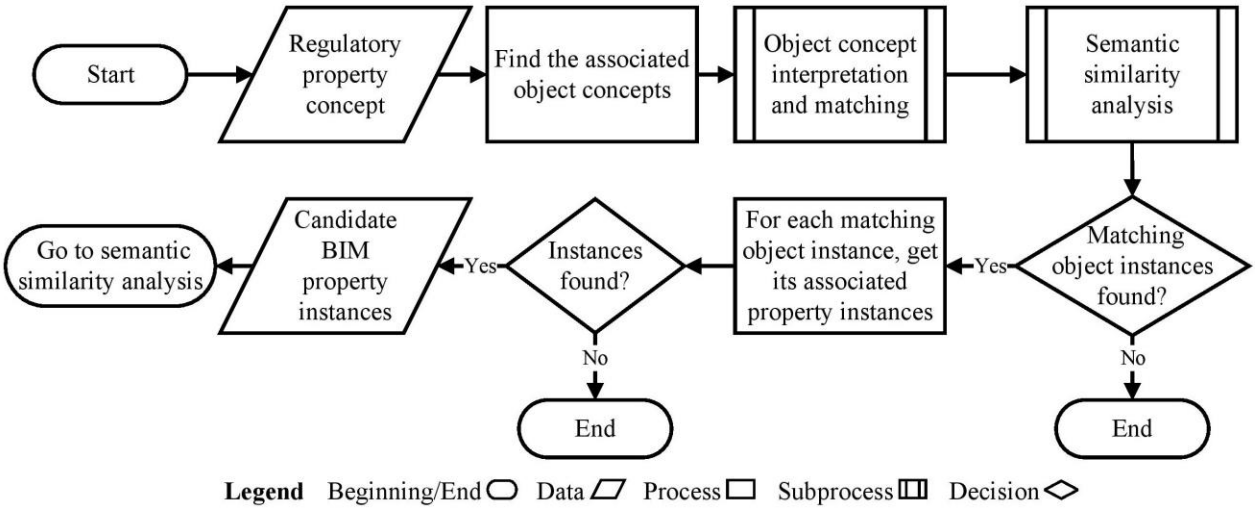


Figure 6.3. Method for property recognition

6.2.1.1.2 Semantic Similarity Analysis

Semantic similarity analysis aims to assess the semantic similarity for each candidate pair (BIM instance and regulatory concept), and accordingly to select the aligned BIM instances (i.e., matches) for regulatory concepts. The proposed approach is novel in two ways. First, it uses deep learning to capture the semantics behind the words for enhanced assessment of the semantic similarity. Second, it uses an empirical way to analyze the patterns of semantic similarities for enhanced recognition of concept matches. The semantic similarity analysis includes three sequential steps, as per Figure 6.4: semantic similarity scoring, semantic similarity weighting and ranking, and semantic similarity threshold analysis.

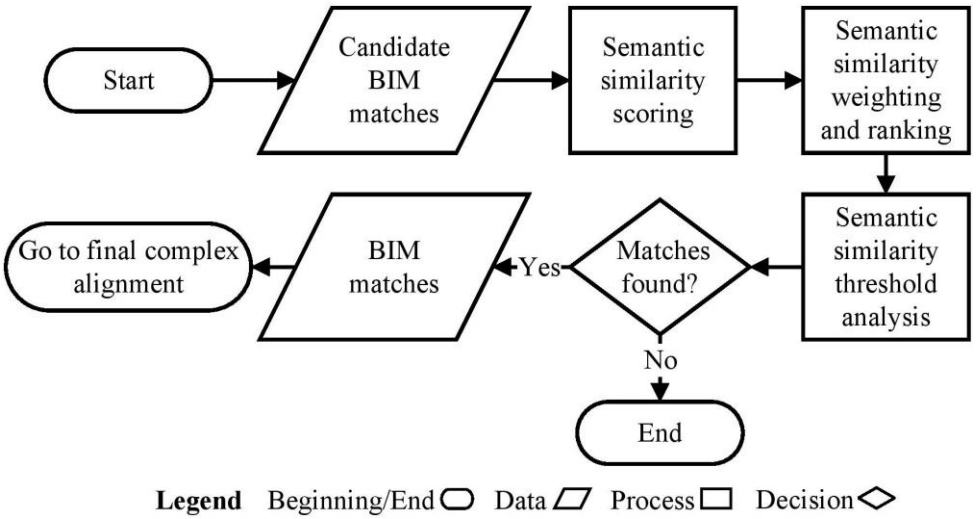


Figure 6.4. Method for semantic similarity analysis

6.2.1.1.2.1 Semantic Similarity Scoring

Semantic similarity scoring aims to assess the semantic similarity between a BIM instance (a candidate match) and a regulatory concept. The proposed scoring method: (1) describes each instance by its entity, property, and material information, all which are called 'instance descriptors' thereafter, (2) describes each concept by its name, regulatory definition, and its equivalency/synonym information, all which are called 'concept descriptors' thereafter, (3) assesses term-to-term semantic similarity (i.e., similarity between a term in a BIM instance descriptor and a term in a regulatory concept descriptor) using a deep learning technique, and (4) assesses instance-concept similarity based on all term-to-term similarities.

The following types of instance descriptors are proposed and used for similarity assessment of object pairs (object instance and object concept): (1) corresponding IFC entity information (e.g., the entity attribute "name" and its value "Basic Roof:EPDM - 4 1/2" - lsf 3:9773671" for an

IfcRoof instance); (2) property information (e.g., property name “Heat Transfer Coefficient (U)” and value “0.374015748031496” for the IfcRoof instance); (3) material information (e.g., “Insulation / Thermal Barriers - Batt insulation #ThermalAssetName: Glass Fiber Batt” is the name of a material layer for the IfcRoof Instance). If the object instance is an aggregated type instance, the material information of the aggregated instances is used as the material information of the object instance; and (4) the instance name and corresponding IFC entity name (without the prefix “Ifc”) of the instances that are spatially located inside the object. For example, if an IfcLamp instance is spatially located inside an IfcSpace instance, the IfcLamp instance name “LED lamp” and its entity name “Lamp” are used as descriptors for the IfcSpace instance. For property pairs (property instance and property concept), the name of the property instance [e.g., “Heat Transfer Coefficient (U)”] is used as its descriptor.

The following types of concept descriptors are proposed and used for similarity assessment of candidate pairs (both object pairs and property pairs): (1) concept name (e.g., u factor), (2) names of its equivalent concepts in the ontology (e.g., u value), (3) names of its synonyms in the bSDD (e.g., heat transfer), (4) names of its acronyms (e.g., LED is an acronym of light emitting oxide), and (5) its quantitative definitions. The construction domain-specific acronyms were manually collected from a number of regulatory documents. The quantitative definitions were extracted from the regulatory documents using the proposed ontology-based information extraction method in Chapter 4 and represented using four semantic information elements (comparative relation,

compliance checking attribute, quantity value, and quantity unit/reference). For example, the concept “low-sloped roof” is described using the following semantic information elements, which were extracted from the 2012 International Energy Conservation Code (ICC 2012): “slope” (compliance checking attribute), “less than” (comparative relation), and “2/12” (quantity value).

The term-to-term semantic similarity is assessed using deep learning. A deep learning technique (i.e., Word2vec) (Bengio 2009) is first used to learn a vector representation for each term (excluding stopwords like “at”, “the”, “am”, and terms with a frequency less than five) from a number of energy regulatory documents. Then, term-to-term semantic similarity is calculated as the cosine similarity of their corresponding vectors, as per Equation 6.1. The similarity between two same terms is always 1 regardless of their term frequencies. The similarity is set to zero if at least one term has a frequency less than five. A positive cosine similarity value indicates two terms are similar, while a negative value means two terms are dissimilar. To avoid negative similarity values, the semantic similarity scoring function in Equation 6.1 is further transformed to Equation 6.2 using an exponential function. The value 100 was empirically selected as the base of the exponential function to give more power to similar terms, while discounting the semantic similarity of dissimilar terms.

The total instance-concept similarity (TS_{ic}) is assessed by aggregating all possible pairs of term-to-term similarities, as per Equation 6.3. Duplicate terms (after stemming, morphological analysis, and removing stopwords) are not double-counted to avoid potential TS_{ic} inflation.

$$\mathrm{sim}(\mathrm{term_m}, \mathrm{term_n}) =$$

$$\begin{cases} 1, \text{ if } \mathrm{term_m} = \mathrm{term_n} \\ \mathrm{cosine}(\mathrm{term_m}, \mathrm{term_n}), \text{ if frequencies of both } \mathrm{term_m} \text{ and } \mathrm{term_n} \geq 5, \mathrm{term_m} \neq \mathrm{term_n} \\ 0, \text{ if } 1 \leq \text{frequency of either } \mathrm{term_m} \text{ or } \mathrm{term_n} < 5, \mathrm{term_m} \neq \mathrm{term_n} \end{cases} \quad (6.1)$$

$$\mathrm{S_{tr}}(\mathrm{term_m}, \mathrm{term_n}) = 100^{\mathrm{sim}(\mathrm{term_m}, \mathrm{term_n})-1} \quad (6.2)$$

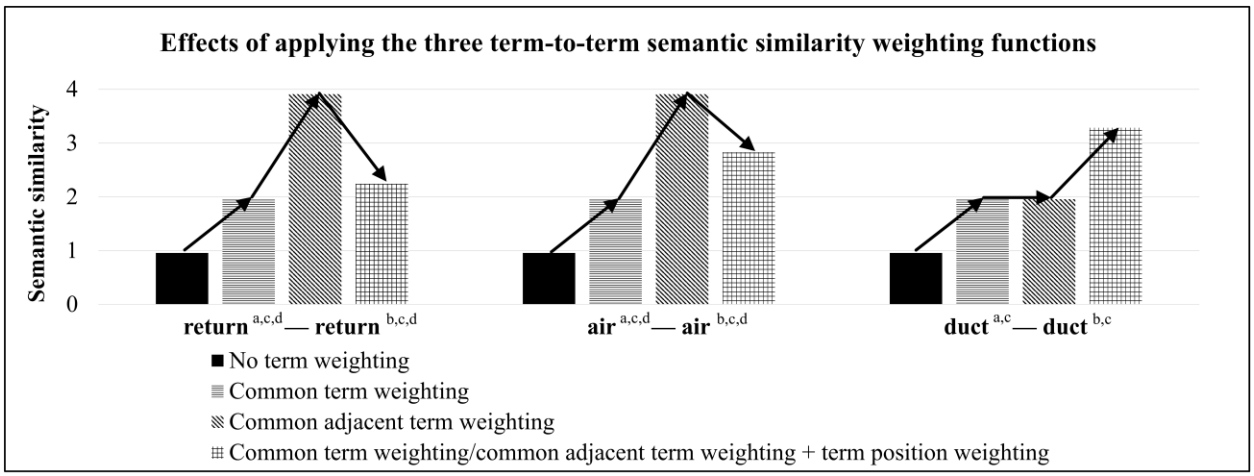
$$\mathrm{TS_{ic}} = \sum_{\mathrm{term_m} \in \text{instance}, \ \mathrm{term_n} \in \text{concept}} \mathrm{S_{tr}}(\mathrm{term_m}, \mathrm{term_n}) \quad (6.3)$$

6.2.1.1.2.2 Semantic Similarity Weighting and Ranking

Semantic similarity weighting and ranking aims to weight the term-to-term semantic similarity for different terms, weight the total instance-concept similarity for different candidate matches (i.e., BIM instances) for a regulatory concept, and accordingly rank the candidate matches. The weighting aims to capture the factors that impact the degree of similarity such as positions of matching terms and lengths of instance descriptors.

Three term-to-term semantic similarity weighting functions are proposed and used to capture the fact that different terms may have different powers in indicating the matching degree between a BIM instance and a regulatory concept – based on the degree of match and position. First, if the descriptors of a concept share common terms (100% match) with the descriptors of a BIM instance, it is likely that the BIM instance is related to that concept. Thus, higher term-to-term semantic similarity should be given to those common terms. Equation 6.4 shows the weighted term-to-term semantic similarity for the common terms (S_{ct}), in which one bonus point is empirically given to the term-to-term semantic similarity (S_{tr}) of those common terms. Second, if there are two or more

common terms and these common terms are adjacent in both descriptors, this indicates more confidence that the BIM instance is related to that concept. Thus, the term-to-term semantic similarity of those common adjacent terms should be further increased. Equation 6.5 shows the weighted term-to-term semantic similarity for the common adjacent terms (S_{cat}), which is double of the S_{ct}. Third, since the carried meaning of terms in a concept decreases from right to left, the term-to-term semantic similarity should be further adjusted based on the term positions in the concept. Equation 6.6 shows the proposed term position weighting (TPW) function, which is logarithmic to avoid overweighting the rightmost terms in the concept. Figure 6.5 shows an example of applying those three term-to-term semantic similarity weighting functions for assessing the matching degree between the BIM instance "IfcDuctSegment (#4946670)" and the regulatory concept "supply and return air duct".



[a]This term comes from the concept descriptors below.
[b]This term comes from the BIM instance descriptors below.
[c]Common term between concept descriptors and BIM instance descriptors.
[d]Common adjacent term between concept descriptors and BIM instance descriptors.

Concept	BIM instance	Concept descriptors	BIM instance descriptors
supply and return air duct	#4946670= IFCDUCTSEGMENT('1VrzP4okL1AOes_uCjegtj',#42,'Rectangular Duct:Rectangular Duct - 1":9757602',$,'Rectangular Duct:Rectangular Duct - 1":9757654',#4946649, #4946665,'9757602',$);	supply and **return air duct**	**IFC entity information:** {name,Rectangular Duct:Rectangular Duct - 1":9757602,object type, Rectangular Duct:Rectangular Duct - 1":9757654} **Property information(partial):** {Insulation,#4946674,Insulation Thickness,0.025399999999999992,{{null, metre, '1'}}}; {Insulation,#4946675,Insulation Type,Batt Insulation 1",{{null, one, '1'}}}; {Mechanical,#4946472,System Type,**Duct** System: **Return Air,**{{null, one, '1'}}}; {Mechanical,#4946686,Area,1.623749010560001,{{null, metre, '2'}}}; {Mechanical,#4946687,Bottom Elevation,4.1148,{{null, metre, '1'}}}; {Mechanical,#4946688,Equivalent Diameter,0.4460769616132811,{{null, metre, '1'}}}; {Mechanical,#4946690,System Classification,Return Air,{{null, one, '1'}}}; {Mechanical,#4946691,System Name,Mechanical Return Air 1,{{null, one, '1'}}};

Figure 6.5. Example of applying the three term-to-term semantic similarity weighting functions

An instance-concept semantic similarity weighting function is proposed to capture the impact of the instance descriptor lengths on the total instance-concept similarity (TS_{ic}). Since longer instance descriptors tend to result in higher TS_{ic} than shorter ones (i.e., longer descriptors result in aggregating more pairs of term-to-term semantic similarities), the TS_{ic} should be weighted by the average length of all instance descriptors. Equation 6.7 shows the weighted total instance-concept

semantic similarity ($TS_{weighted}$), where the TS_{ct} is the weighted total term-to-term semantic similarity of common terms and the TS_{nct} is the total term-to-term semantic similarity of noncommon terms. Only the TS_{nct} is weighted by the average length of all instance descriptors to avoid discounting the TS_{ct}. Equations 6.8 and 6.9 show the equations used for calculating the TS_{ct} and the TS_{nct}, respectively, where k is the total number of common terms.

$$S_{ct}(term_n, term_n) = S_{tr}(term_n, term_n) + 1 \quad (6.4)$$

$$S_{cat}(term_n, term_n) = 2 * S_{ct}(term_n, term_n) \quad (6.5)$$

$$TPW(term_n) = \frac{\log(n+1)}{\sum_{m=1}^{\text{concept length}} \log(m+1)}, term_n \text{ is the } n^{th} \text{ term from left in a concept} \quad (6.6)$$

$$TS_{weighted} = TS_{ct} + \frac{TS_{nct}}{\text{instance descriptor length}} * \text{average length of all instance descriptors} \quad (6.7)$$

$$TS_{ct} = \left[\sum_{term_n \in \text{ common nonadjacent terms}} TPW(term_n) * S_{ct}(term_n, term_n) + \sum_{term_m \in \text{ common adjacent terms}} TPW(term_m) * S_{cat}(term_m, term_m)\right] *$$

$$k, k \text{ is the total number of common terms} \quad (6.8)$$

$$TS_{nct} = \sum_{term_m \in \text{instance } term_n \in \text{concept, } term_m \neq term_n} S_{tr}(term_m, term_n) \quad (6.9)$$

After semantic similarity weighting, the candidate matches (BIM instances) for a given regulatory concept are ranked by the $TS_{weighted}$, in a decreasing order.

6.2.1.1.2.3 Semantic Similarity Threshold Analysis

Semantic similarity threshold analysis aims to analyze the degree of instance-concept similarity, based on a set of thresholds, in order to select the matches from the set of ranked candidate matches

for a regulatory concept. Two sets of thresholds were experimentally set: one for the object concepts and one for the property concepts, as shown in Table 6.1, respectively.

Table 6.1. Threshold Types and Values for Regulatory Concepts

Concept type	Threshold type	Value
Object concept	Minimum TS_{Oct}[1]	TS_{ct} (t^2)
	Minimum $TS_{Oweighted}$	0.5
	Minimum normalized $TS_{Oweighted}$	0.5
	Minimum normalized $TS_{Oweighted}$ difference	0.06
Property concept	Minimum $TS_{Pweighted}$	$(1/\sum_{n=1}^{\text{Concept length}} n) * 2$
	Minimum normalized $TS_{Pweighted}$	0.5
	Minimum normalized $TS_{Pweighted}$ difference	0.05

[1]Only applies to the BIM instances recognized using Method 2 (Section 6.2.1.1.1.1).

[2]Calculated using Equation 6.8, where common terms $\in$ core inner concept. A core inner concept carries the most important meaning of a given concept (e.g., "roof" is the core inner concept for the given concept "metal-framed roof").

Threshold analysis for object pairs: In order to select the matches (from a set of candidate matches) for an object concept, the following four threshold types are proposed and used together to indicate the similarity cut-off (i.e., the level of similarity that indicates a match):

- The minimum TS_{Oct} threshold defines the threshold for matching – a candidate pair must have a TS_{ct} larger than the minimum TS_{Oct} threshold to be a match. This threshold helps define the cutoff of similarity indicated by the degree of sharing common terms (e.g., a pair sharing 3 adjacent common terms would have a higher total similarity than a pair sharing 2 non-adjacent common terms).

- The minimum $TS_{Oweighted}$ threshold helps define the cutoff of similarity indicated by the weighted total instance-concept similarity ($TS_{weighted}$), where a $TS_{weighted}$ lower than the threshold is probably not reflecting a large-enough degree of similarity to indicate a match.

- The minimum normalized $TS_{Oweighted}$ threshold is a normalized version of the former threshold – normalized to the range of (0, 1) by dividing the $TS_{weighted}$ of each by the highest $TS_{weighted}$ among all pairs. The normalization helps adjust the threshold values to a common scale.

- The minimum normalized $TS_{Oweighted}$ difference threshold helps define the cutoff of similarity indicated by the difference in similarity between two adjacently-ranked pairs. The higher the difference, the larger the similarity jump from one pair to the other. The threshold helps indicate whether this jump is large enough to mean that the pair with the higher normalized $TS_{weighted}$ is similar but the following pair is not similar enough to be a match. Given a set of ranked pairs, a difference value larger than the threshold indicates that the pairs before the cutoff are sufficiently different than the pairs after – the ones before are similar (i.e., matches), but the ones after are dissimilar.

The four thresholds are used in the following manner to collectively indicate which candidate pairs meet the different types of similarity cutoffs – the pairs that meet all four thresholds, in the following way, are likely to be matches. First, all candidates that do not meet both the minimum TS_{Oct} threshold and the minimum $TS_{Oweighted}$ threshold are filtered out. Second, the remaining candidates that meet both the minimum normalized $TS_{Oweighted}$ threshold and the minimum

normalized $TS_{Oweighted}$ difference threshold cutoff qualify as matches. For example, Figure 6.6 shows that there are 72 candidate BIM instances for the concept “supply and return air duct”, 46 of them (ranked 19 to 64) meet both the minimum TS_{Oct} threshold and the minimum $TS_{Oweighted}$ threshold. All 72 instances meet the minimum $TS_{Oweighted}$ threshold, which indicates that they all have a large-enough degree of total similarity to indicate a potential match – they are all duct instances. However, only 46 of those instances (ranked 19 to 64) meet the minimum TS_{Oct} threshold, which indicates that they all have a large-enough degree of common-term similarity. These 46 instances have a higher degree of sharing terms, compared to the other 26 that only share the term “duct”. Figures 6.7 and 6.8 show that only 41 of the 46 candidate instances meet both the minimum normalized $TS_{Oweighted}$ threshold and the minimum normalized $TS_{Oweighted}$ difference threshold cutoff. Figure 6.7 shows that all 46 instances meet the minimum normalized $TS_{Oweighted}$ threshold, which indicates they all have a large-enough degree of normalized total similarity. Figure 6.8 further shows that there are two large-enough similarity jumps: a larger jump (between rank 41 and 42) and a smaller jump (between rank 30 and 31). The 41 instances share two common adjacent terms “supply air” or “return air” and share another common non-adjacent term “duct”, while the other five instances (42 to 46) have relatively lower common-term similarity; they only share two non-adjacent common terms “air” and “duct”.

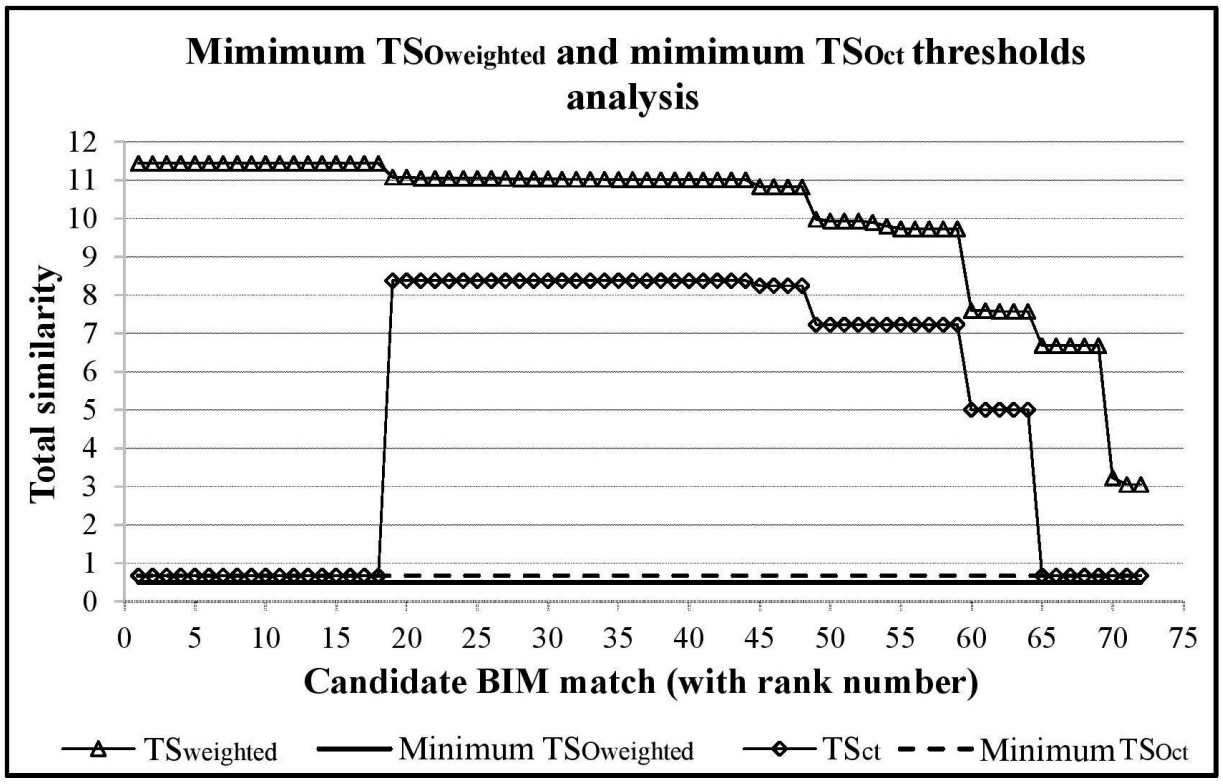


Figure 6.6. Example of minimum $TS_{Oweighted}$ and minimum TS_{Oct} thresholds analysis

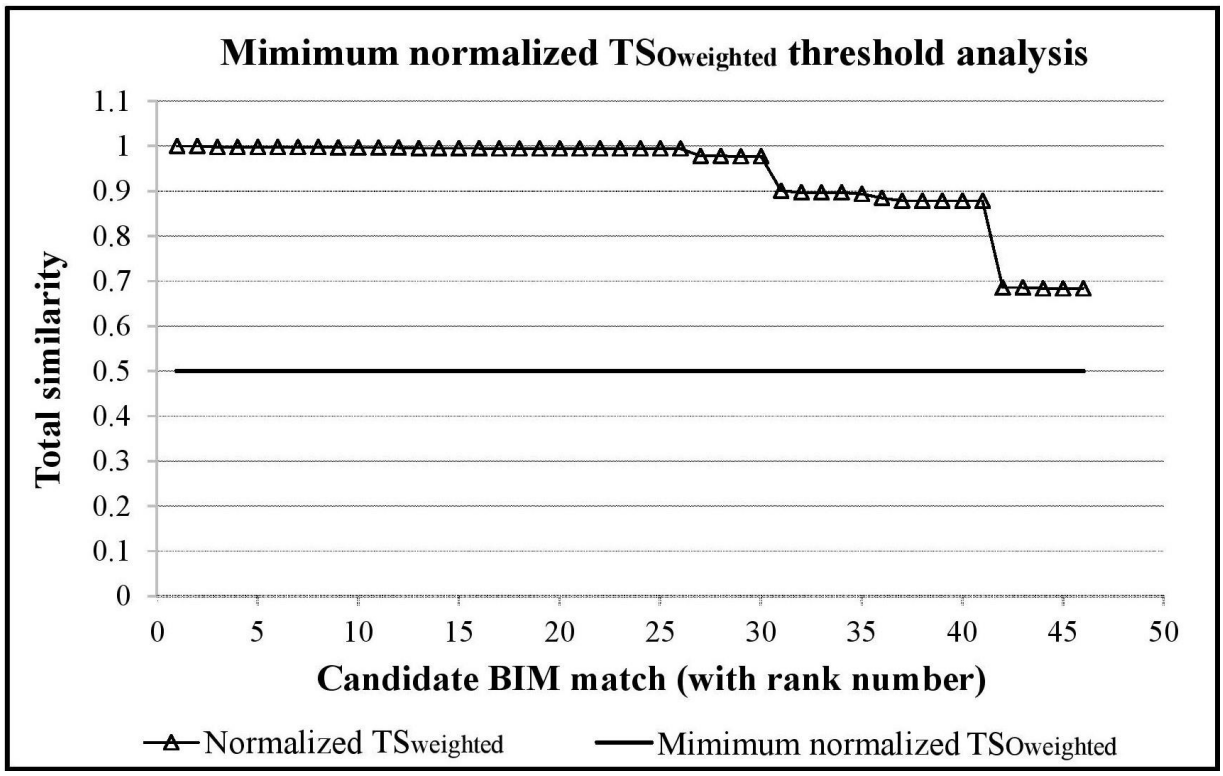


Figure 6.7. Example of minimum normalized $TS_{Oweighted}$ threshold analysis

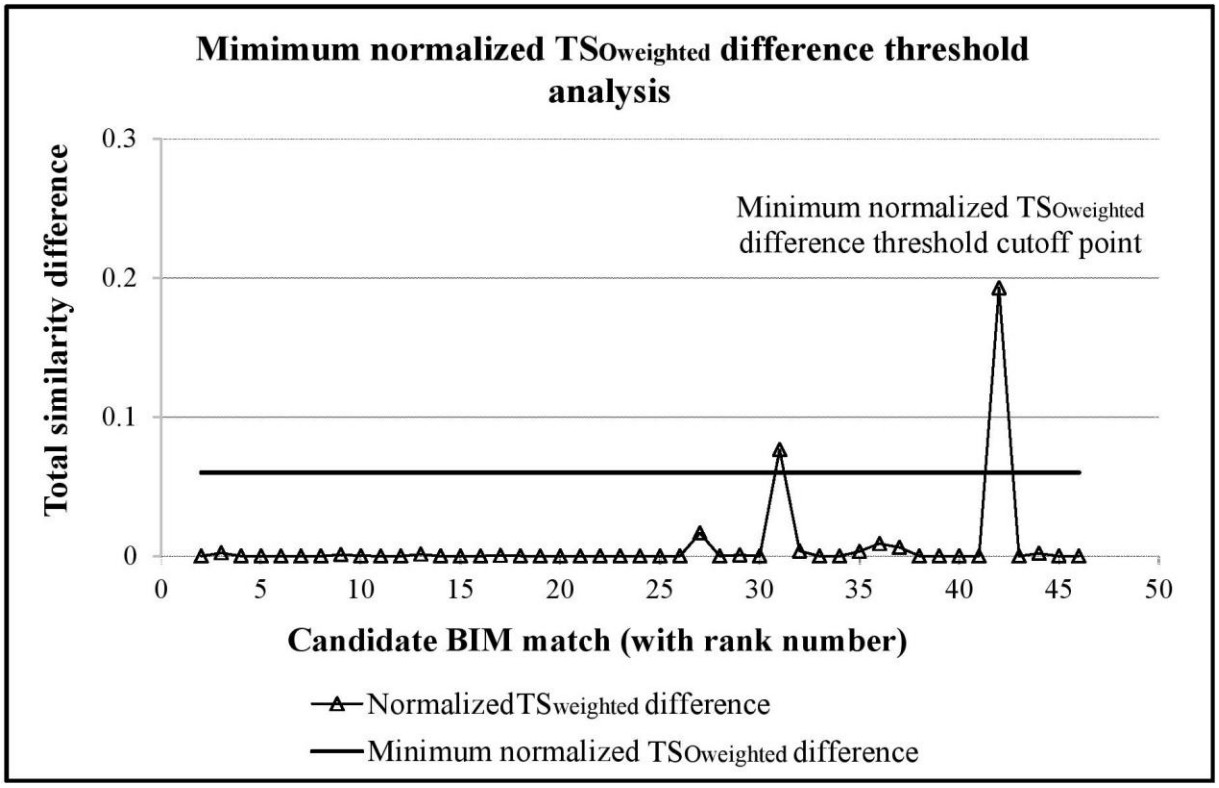


Figure 6.8. Example of minimum normalized $TS_{Oweighted}$ difference threshold analysis

Threshold analysis for property pairs: The threshold analysis for property pairs is similar to that for object pairs. Three similar threshold types are proposed and used to select the matching BIM instances for a property concept: minimum $TS_{Pweighted}$, minimum normalized $TS_{Pweighted}$, and minimum normalized $TS_{Pweighted}$ difference thresholds. All property candidates must meet all three thresholds to qualify as matches.

6.2.1.2 Final Complex Alignment

Final complex (group-group) alignment aims to recognize the object concepts that belong to one requirement (called thereafter 'concept group'), find the matches to each concept in that concept group (using the methods in Section 6.2.1.1), and accordingly recognize the instance groups (which could be one or more) that are linked to that concept group (i.e., recognize the instance

groups that belong to one requirement). The proposed final complex alignment approach is novel in two ways. First, it uses supervised and unsupervised searching to find the relationships that create instance pairs. Second, it uses network modeling to model and link concept groups and their associated instances, where each concept group and its associated instance groups are modeled as a network of linked concept pairs and instance pairs. The proposed final complex alignment method is, thus, composed of three main steps, illustrated in Figure 6.9: supervised searching, unsupervised searching, and network construction.

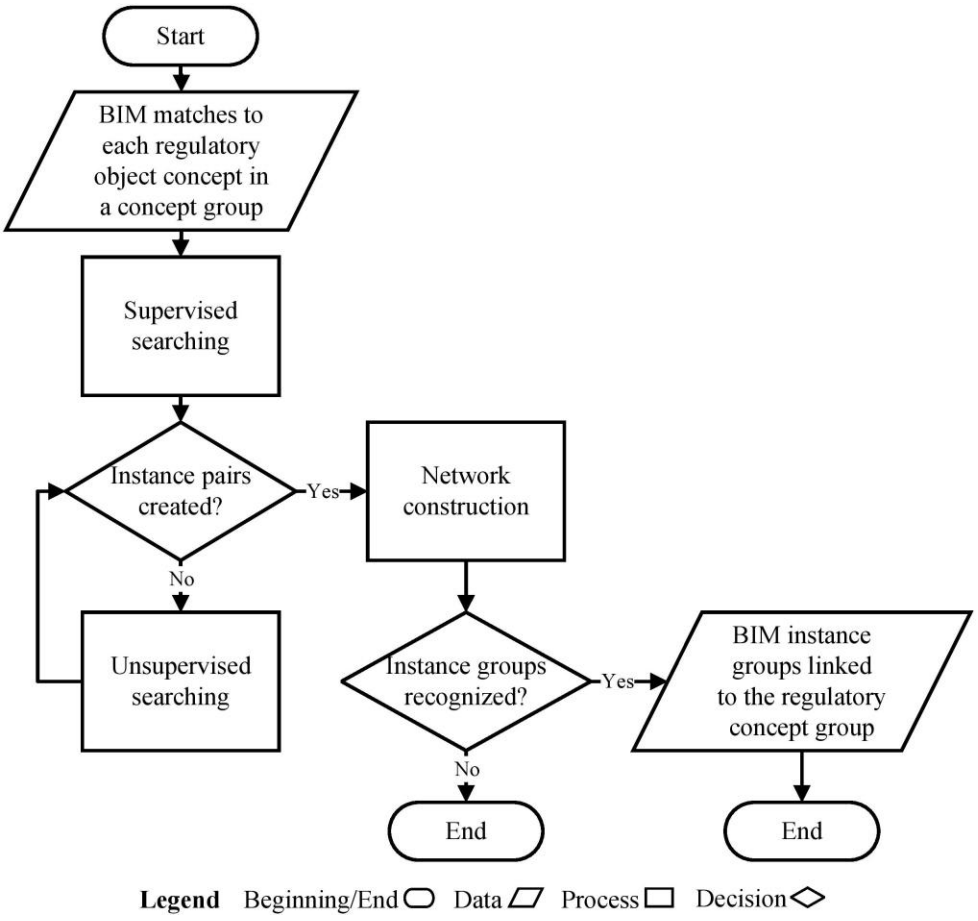


Figure 6.9. Method for final complex alignment

6.2.1.2.1 Supervised Searching

Supervised searching aims to recognize, in a supervised manner (i.e., using known or predefined relationships), the relationships that create instance pairs. Two types of relationships were empirically predefined and used for searching: (1) object-material usage relationship, which is defined as a relationship (an instance of the IfcRelAssociatesMaterial entity) that links two IfcMaterial object instances, or links one IfcMaterial object instance and one IfcProduct/IfcTypeProduct object instance; and (2) spatially-contained relationship, which is defined as a relationship (an instance of the IfcRelContainedInSpatialStructure entity or an instance captured using bounding box geometric assessment) that links two IfcProduct/IfcTypeProduct object instances, where one object instance is spatially contained in the other (e.g., a lighting fixture is spatially contained in a space). Bounding box geometric assessment indirectly captures the spatially-contained relationship between two object instances by assessing whether the bounding box of one instance is entirely inside the bounding box of the other instance, where bounding box refers to the geometric orthogonal box representation of an object instance. For example, Figure 6.10 shows that the bounding box of an IfcLightFixure instance is inside the bounding box of an IfcSpace instance.

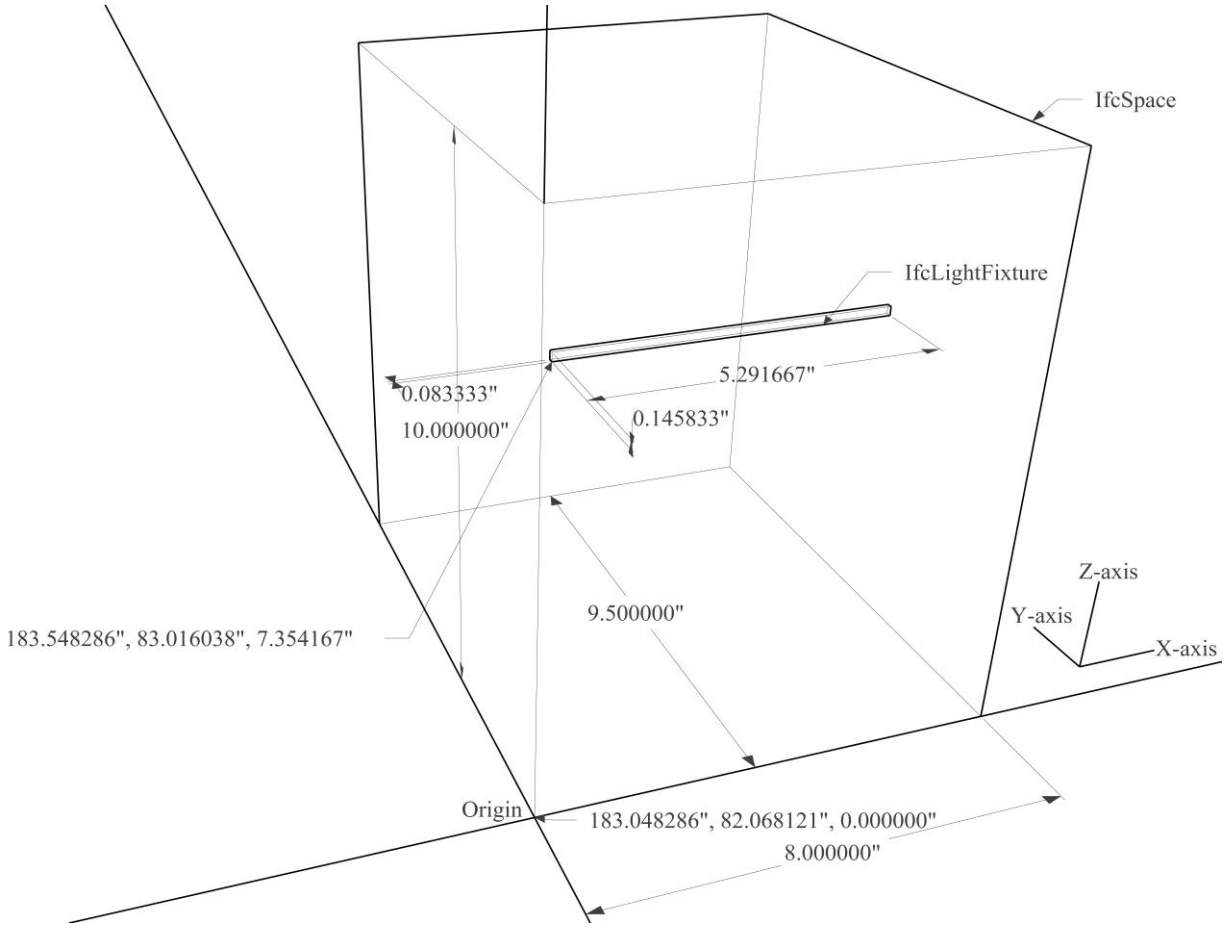


Figure 6.10. The bounding box of an IfcLightFixture instance is inside the bounding box of an IfcSpace instance

6.2.1.2.2 Unsupervised Searching

Unsupervised searching aims to recognize, in an unsupervised manner (i.e., without any known or predefined relationships), the relationships that create instance pairs. A graph-based searching algorithm is used to automatically find the relationships. A relationship, in this case, is an instance of the IfcRelationship entity (including the instances of all its subentities except IfcRelAssociatesMaterial and IfcRelContainedInSpatialStructure, which are used in the supervised searching) that links object instances of IfcProduct/IfcTypeProduct and IfcSystem –

using one-to-many association (i.e., one IfcRelationship instance may link one relating object instance to many related object instances). This research assumes that two object instances are linked if there exists at least one path between the two instances. For example, Figure 6.11 shows that (1) an IfcRelAssignsToGroup (a subentity of the IfcRelationship) instance links an IfcSystem instance (a relating instance) to IfcValve and IfcPipeSegment instances (two related instances); and (2) an IfcValve instance and an IfcPipeSegment instance are linked by two paths: a longer path linked by instances of IfcRelConnectsPortToElement, IfcDistributionPort, and IfcRelConnectsPorts; and a shorter path linked by an instance of IfcRelAssignsToGroup. To automatically find the path(s) between two object instances, an IFC instance graph is built by parsing the instances of IfcRelationship, in which an edge represents an IfcRelationship instance and a node represents a relating or a related object instance. Then, a graph searching algorithm is used to search the IFC instance graph for a path between the two object instances. Because there may exist multiple paths between two instances, it was assumed that the shortest path is the one that determines if two instances are linked. For example, the IfcValve instance and the IfcPipeSegment instance (in Figure 6.11) were linked as a pair by the IfcRelAssignsToGroup instance, because they occurred on the shortest path. The commonly-used Dijkstra searching algorithm (Dijkstra 1959) was used in this research for shortest path finding.

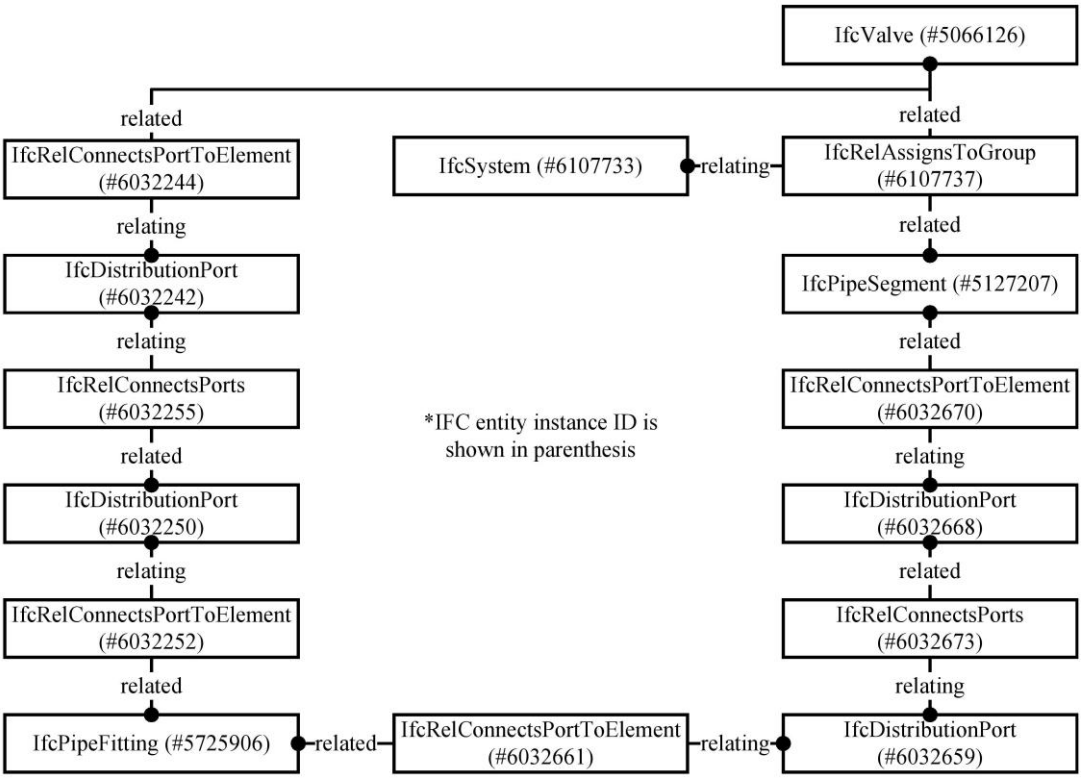


Figure 6.11. Examples of relationships between IFC instances that were found using unsupervised searching

6.2.1.2.3 Network Construction

Network construction aims to link concept groups and their associated instances into a set of networks. Instance pairs (identified in 6.2.1.2.1 and 6.2.1.2.2) are grouped and linked to their concept groups, where each concept group is related to a regulatory requirement. A set of instance pairs belongs to one instance group, if those instance pairs (1) have one-to-one correspondence to the concept pairs, and (2) can be linked in the same way as the concept pairs in the concept group. An example of final complex alignment is shown in Figure 6.12, where eight instance pairs were identified using unsupervised searching, and two instance groups were accordingly formed and linked to a concept group using network construction.

Regulatory sentence	Regulatory concept	Matched BIM instances for each regulatory concept
5. Strainers, control valves, and balancing valves associated with piping 1 inch (25 mm) or less in diameter.	strainer	#5104797= IFCVALVE('1PPJnN55XBtfL1p8yTtwW$',#42,'Pipe-Valve-Crane-F277-Class_125_Strainer:Pipe-Valve-Crane-F277-Class_125_Strainer 2" 2:9825022',$,'Pipe-Valve-Crane-F277-Class_125_Strainer 2" 2',#5104796,#5104785,'9825022',$);
		#5177753= IFCVALVE('3vy3SF$z97QO6JV1$hT9tP',#42,'Pipe-Valve-Crane-F277-Class_125_Strainer:Pipe-Valve-Crane-F277-Class_125_Strainer 2" 3:9826238',$,'Pipe-Valve-Crane-F277-Class_125_Strainer 2" 3',#5177752,#5177741,'9826238',$);
	control valve	#5111538= IFCVALVE('1PPJnN55XBtfL1p8yTtwdC',#42,'Motor Control Valve - 0.5-2 Inch:Motor Control Valve - 0.5-2 Inch - 3/4":9825037',$,'Motor Control Valve - 0.5-2 Inch - 3/4"',#5111537,#5111526,'9825037',$);
		#5197536= IFCVALVE('3vy3SF$z97QO6JV1$hT9st',#42,'Motor Control Valve - 0.5-2 Inch:Motor Control Valve - 0.5-2 Inch - 2":9826256',$,'Motor Control Valve - 0.5-2 Inch - 2"',#5197535,#5197524,'9826256',$);
	balancing valve	#5119147= IFCVALVE('1PPJnN55XBtfL1p8yTtwdF',#42,'Balancing Valve - Angle - 2.5-12 Inch - Flanged:Balancing Valve - Angle - 2.5-12 Inch - Flanged - 2 1/2":9825038',$,'Balancing Valve - Angle - 2.5-12 Inch - Flanged - 2 1/2"',#5119146,#5119135,'9825038',$);
		#5189724= IFCVALVE('3vy3SF$z97QO6JV1$hT9sf',#42,'Balancing Valve - Angle - 2.5-12 Inch - Flanged:Balancing Valve - Angle - 2.5-12 Inch - Flanged - 4":9826254',$,'Balancing Valve - Angle - 2.5-12 Inch - Flanged - 4"',#5189723,#5189712,'9826254',$);
	piping	#5105649= IFCPIPESEGMENT('1PPJnN55XBtfL1p8yTtwd4',#42,'Pipe Types:Pipe 3/4":9825029',$,'Pipe Types:Pipe 3/4":8996945',#5105627,#5105645,'9825029',$);
		#5178437= IFCPIPESEGMENT('3vy3SF$z97QO6JV1$hT9sa',#42,'Pipe Types:Pipe 2":9826243',$,'Pipe Types:Pipe 2":9560571',#5178415,#5178433,'9826243',$);
	diameter	#5105665= IFCPROPERTYSINGLEVALUE('Inside Diameter',$,IFCLENGTHMEASURE(0.0686666666666667),$);
		#5178455= IFCPROPERTYSINGLEVALUE('Outside Diameter',$,IFCLENGTHMEASURE(0.197916666666667),$);

Unsupervised searching

Concept pair	Instance pair
strainer —— piping	IfcValve (#5104797) —— IfcPipeSegment (#5105649)
	IfcValve (#5177753) —— IfcPipeSegment (#5178437)
control valve —— piping	IfcValve (#5111538) —— IfcPipeSegment (#5105649)
	IfcValve (#5197536) —— IfcPipeSegment (#5178437)
balancing valve —— piping	IfcValve (#5119147) —— IfcPipeSegment (#5105649)
	IfcValve (#5189724) —— IfcPipeSegment (#5178437)
diameter —— piping	IfcPropertySingleValue (#5105665) —— IfcPipeSegment (#5105649)
	IfcPropertySingleValue (#5178455) —— IfcPipeSegment (#5178437)

Network construction

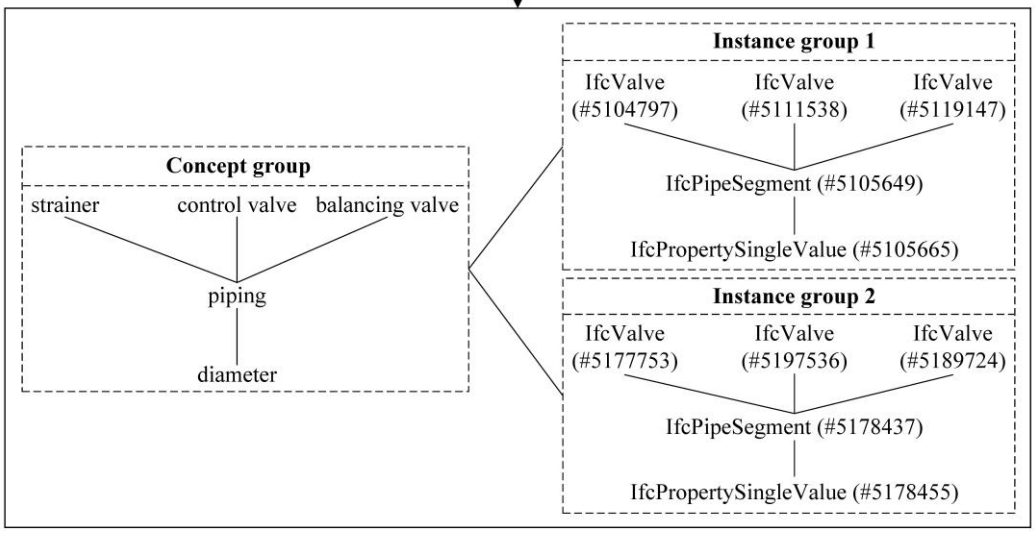


Figure 6.12. Example of final complex alignment

6.2.2 Implementation of the Proposed Semantic Information Alignment Algorithm

The proposed information alignment method was implemented for testing and evaluation. A Revit model of an educational building project in Illinois, which was created using Autodesk Revit 2016 (Autodesk 2016a), was used for testing. Regulatory requirements were extracted from three energy regulatory documents – the 2012 International Energy Conservation Code (ICC 2012), the 2013 Building Energy Efficiency Standards (California Energy Commission 2013) (known as the California Energy Code), and the Ontario Building Code Supplementary Standard SB-10 (Ontario Ministry of Municipal Affairs 2011), which represent energy regulatory documents developed by the international council, specific US states, and other countries, respectively. The scope of the testing was limited to commercial building thermal insulation requirements and lighting power requirements (i.e., two subtypes of energy requirements). The implementation included four steps: BIM information extraction, regulatory information extraction, semantic information alignment (using the proposed method, as per Section 6.2.1), and evaluation.

6.2.2.1 BIM Information Extraction

The BIM information was extracted from the .ifc file of the BIM model and processed into an intermediate representation for further alignment to the regulatory information. The open standard data model parsing approach is used in this research because an open standard BIM model (e.g., IFC) can ensure platform independency of the developed semantic information alignment method and algorithm. An open source toolbox, rather than a custom-developed parsing algorithm, is used

to extract the BIM information because open source toolboxes have better scalability across different versions of the IFC specification. The open source toolbox JSDAI is selected because it is Java-based, and fits better with the Java-implementation of the developed semantic information alignment method and algorithm. The BIM information was extracted using three steps: IFC export, information extraction from IFC data file, and post-processing of the extracted information.

6.2.2.1.1 IFC Export

To ensure the platform independency of the implementation, the information in the Revit model was exported to an IFC data file (.ifc file) using the Revit IFC exporter. Since the current Revit IFC exporter (Autodesk 2016b) is limited in exporting commercial building energy efficiency design information (e.g., export of material thermal properties such as thermal conductivity is not supported), the exporter was extended in the C# programming language using the Microsoft Visual Studio 2015 (Microsoft 2016). The IFC4 specification (Liebich 2013) was selected because it is the most recent version supported by Revit, and has extended support for exporting energy domain entities (e.g., lighting entities).

6.2.2.1.2 Information Extraction from IFC Data File

The instances of the entities were extracted from the .ifc file using an EXPRESS-based information extraction method, which is an adaptation of the method in Zhang and El-Gohary (2015a). An EXPRESS-based data access method was used to extract all IFC entity instances into an intermediate representation: [IFC Entity Name, IFC Entity Instance ID, IFC Entity Attribute

Names, IFC Entity Attribute Values]. Since not all IFC entities are relevant to regulatory compliance checking (e.g., the IfcProcess entity defines an individual activity or event, which may be used in construction scheduling), only instances of the IFC entities from the following six categories were extracted in this research: (1) relationship entities, (2) object-related entities, (3) material-related entities, (4) property-related entities, (5) unit-related entities, and (6) geometry-related entities. The specific IFC entities in each category are summarized in Table 6.2. If an IFC entity was an abstract entity (i.e., which is a non-instantiable entity), all instances of all its non-abstract subentities were extracted. Table 6.3 shows an example that includes two entity instances that were extracted in the intermediate representation, which are related to one regulatory requirement.

A special processing method was used in extracting instances of the property-related entities IfcPreDefinedPropertySet and IfcSimpleProperty. If the "Unit" attribute value of an instance was not available, the IfcValue measure type for the "NominalValue" attribute of that instance was used as a temporary unit. The measure type was used to deduce an explicit unit in the next post-processing step (see Section 6.2.2.1.3). For example, Table 6.3 shows that the measure type "IFCTHERMALRESISTANCEMEASURE" was used as a temporary unit for the property instance "Thermal Resistance", because the "Unit" attribute value for the IfcPropertySingleValue instance is "$" (i.e., not available).

Table 6.2. List of IFC Entities Used in the BIM Information Extraction

Relationship entities	Object-related entities	Material-related entities	Property-related entities	Unit-related entities	Geometry-related entities
IfcRelationship*	IfcProduct*	IfcMaterialList	IfcPropertySet*	IfcDerivedUnit	IfcProductDefinitionShape
	IfcTypeProduct*	IfcMaterial	IfcPreDefinedPropertySet*	IfcDerivedUnitElement	IfcShapeRepresentation
	IfcSystem	IfcMaterialLayerSetUsage	IfcSimpleProperty*	IfcConversionBasedUnit	IfcStyledRepresentation
		IfcMaterialProfileSetUsage	IfcPhysicalSimpleQuantity*	IfcSIUnit	IfcBoundingBox
		IfcMaterialLayerSet	IfcQuantitySet*	IfcDimensionalExponents	IfcCartesianPoint
		IfcMaterialLayer	IfcMaterialProperties	IfcMeasureWithUnit	IfcLocalPlacement
					IfcAxis2Placement3D
					IfcDirection

*Abstract entity.

Table 6.3. Examples of Extracted Entity Instances in the Intermediate Representation

IFC entity instance in .ifc file	IFC entity instance in the intermediate representation			
	IFC entity name	IFC entity instance ID	IFC entity attribute names	IFC entity attribute values
#4940907= IFCDUCTSEGMENT('3GeO_XJYnD0R2X24BqtWKm',#42, 'Oval Duct:Oval Duct – 2":9746348',$, 'Oval Duct:Oval Duct – 2":9746591',#4940833,#4940902, '9746348',$);	IFCDUCTSEGMENT	#4940907	GlobalId	3GeO_XJYnD0R2X24BqtWKm
			OwnerHistory	IFCOWNERHISTORY#42
			Name	Oval Duct:Oval Duct – 2":9746348
			Description	$(N/A)
			ObjectType	Oval Duct:Oval Duct – 2":9746591
			ObjectPlacement	IFCLOCALPLACEMENT#4940833
			Representation	IFCPRODUCTDEFINITIONSHAPE#4940902
			Tag	9746348
			PredefinedType	$(N/A)
#4940943= IFCPROPERTYSINGLEVALUE('Thermal Resistance',$,IFCTHERMALRESISTANCEMEASURE(1.7611016132736),$);	IFCPROPERTYSINGLEVALUE	#4940943	Name	Thermal Resistance
			Description	$(N/A)
			NominalValue	1.761101613273600
			Unit	IFCTHERMALRESISTANCEMEASURE

6.2.2.1.3 Post-Processing of Extracted Information

Post-processing was conducted to transform the intermediately-represented information to an alignment-ready representation, in which irrelevant design information (e.g., GlobalId,

OwnerHistory) is filtered out to avoid unnecessary future processing efforts. After post-processing, the design information was represented in a number of hashmaps. These hashmaps were categorized into four groups according to their roles in information alignment: (1) the hashmaps "relationship", "product bounding box", and "spatially-contained product" were used for final complex alignment; (2) the hashmaps "material property", "object property", "product quantity", and "material layer-based property" contain the descriptors for property instances; (3) the hashmaps "object property", "object associated material information", "object entity information", "spatially-contained product", "material entity information", and "material category" contain the descriptors for object instances; and (4) the hashmaps "object ID-to-name", "object ID-to-entity name", "object entity name-to-ID", "relAggregates", "material ID-to-name", "material entity name-to-ID", and "material ID-to-object ID" are auxiliary hashmaps that were used to support the overall semantic information alignment. Table 6.4 provides examples to illustrate the hashmaps.

Table 6.4. Examples of Post-Processed BIM Design Information Represented in Hashmap Format

Hashmap name	Key	Value	Example ({Key : Value})	Note
Object property	IfcProduct/ IfcTypeProduct/ IfcSystem instance ID	List of properties, each property is represented in: (Property set name, Property instance ID, Property name, Value, Unit)	{#4975576 : Analytical Properties(Type), #4975907, Heat Transfer Coefficient (U), 0.374015748031496, {{null, kelvin, '-1'},{kilo, gram, '1'},{null, second, '-3'}}}	Properties of an IfcTypeObject instance are assigned to each occurrence IfcObject instance of that type, based on the assignment rules specified in the IfcRelDefinesByType entity.
Object entity information	IfcProduct/ IfcTypeProduct instance ID	The entity attribute values of the Key IfcProduct/IfcTypeProduct instance	{#4975576 : Basic Roof:EPDM – 4 1/2" – lsf 3:9773671, Basic Roof:EPDM – 4 1/2" – lsf 3, 9773671, notdefined}	Exclude values of the following entity attributes: 1. Ownerhistory, Globalid. 2. Entity attribute value is null. 3. Entity attribute value is other entity instances.

Special post-processing was conducted for measurement units, for the property instances in the following hashmaps: "material property", "object property", "product quantity", and "material layer-based property". Four post-processing methods were used to transform the IFC-represented units of the property instances into an intermediate representation for alignment to the units used in regulatory requirements. The intermediate representation: (1) is composed of one or multiple unit elements, (2) uses a 3-tuple representation – [prefix, name, exponent] – for each unit element, and (3) uses the International System of Units (SI) unit system, where each unit was represented using only the SI base unit (i.e., meter, kilogram, second, ampere, kelvin, candela, and mole). The four methods that were used to post-process the unit instances into the 3-tuple representation correspond to the four types of unit instances that were considered in this research: IfcSIUnit, IfcDerivedUnit, IfcConversionBaseUnit, and the temporary unit IfcValue measure type (see Section 6.2.2.1.2).

6.2.2.2 Regulatory Information Extraction

The regulatory information was extracted from the energy regulatory documents using the proposed ontology-based information extraction method in Chapter 4. The extracted regulatory information (i.e., a regulatory requirement/exception) was represented using nine semantic information elements (SIEs), including "subject", "subject restriction", "compliance checking attribute", "deontic operator indicator", "quantitative relation", "comparative relation", "quantity value", "quantity unit/reference", and "quantity restriction" as discussed in Section 4.2.3.1. Table 6.5 shows examples of the SIEs for a regulatory requirement and a regulatory exception.

Table 6.5. Examples of Semantic Information Elements (SIEs) for Regulatory Requirement and Exception

Sentence type	Subject	Subject restriction	Compliance checking attribute	Deontic operator indicator	Quantitative relation	Comparative relation	Quantity value	Quantity unit/ reference	Quantity restriction
Regulatory requirement	supply and return air ducts	where located outside the building	R-value	shall	insulated	minimum	R-8	N/A	N/A
Regulatory exception	strainers, control valves, and balancing valves	associated with piping 1 inch or less in diameter	N/A	N/A	N/A	N/A	N/A	N/A	N/A

6.2.2.3 Semantic Information Alignment

The semantic information alignment was conducted to align the BIM information to the SIE-represented regulatory requirements and exceptions, following the method described in Section 6.2.1. The object concept interpretation and matching was conducted, as per Section 6.2.1.1.1.1, to recognize the candidate BIM object instances (candidate matches) for the regulatory object concepts. These object concepts refer to "subjects" and to objects referenced in the "subject

restrictions" and "quantity restrictions", where object concepts that belong to one SIE tuple (i.e., SIEs that belong to one requirement/exception) form one concept group. The property recognition was conducted, as per Section 6.2.1.1.1.2, to recognize the candidate BIM property instances (candidate matches) for the regulatory property concepts. These property concepts refer to "compliance checking attributes" and to properties referenced in the "subject restrictions", "quantity units/references", and "quantity restrictions". The semantic similarity analysis was conducted, as per Section 6.2.1.1.2, to select the matches from the candidates. The instance pairs were then identified from the matches using the supervised and the unsupervised searching, and then grouped and linked to the requirements using the network construction, as per Sections 6.2.1.2.1 to 6.2.1.2.3. The measurement units in the "quantity units/references" were converted to the 3-tuple representation using a set of conversion rules.

The proposed semantic information alignment method was implemented in a Java-based platform. The platform used the following public APIs to accomplish some specific tasks. The JSDAI API (LKSoftWare GmbH 2016) was used for EXPRESS-based BIM information extraction. The bSDD API (buildingSMART 2016b) was used to search the bSDD for finding the matching IFC entities (or entity parents), synonyms of bSDD concepts, and superconcept information. The Protégé (Musen 2015) was used to build the commercial building energy ontology, while the Apache Jena Ontology API (Apache Jena 2016) was used for parsing the ontology to find superconcepts and equivalent concepts. The deeplearning4j API (Deeplearning4j Development Team 2016), a Java

implementation of Word2vec, was used for learning and computing all term-to-term similarities. The Stanford CoreNLP API (Manning et al. 2014) was used for morphological analysis.

6.2.2.4 Evaluation

A gold standard was manually built to test and evaluate the proposed semantic information alignment method. The gold standard includes 33 regulatory requirements and 10 regulatory exceptions (from the three regulatory documents), and 744 corresponding matches (i.e., BIM instances in the Revit model). The types of matches and their numbers are shown in Table 6.6.

Recall and precision were used to calculate the alignment performance. Recall was calculated as the total number of correctly aligned instances over the total number of instances in the gold standard. Precision was calculated as the total number of correctly aligned instances over the total number of instances aligned. To further test the statistical significance of the results, the confidence interval (*p)* was calculated for both recall and precision using the Wilson score without continuity correction (Goutte and Gaussier 2005; Wilson 1927), which is a computationally simple and satisfactory method for measuring confidence intervals (Newcombe 1998). Equation 6.10 (Wilson 1927) was used, where p_0 refers to the values of precision or recall, λ is the critical value for the confidence interval, *n* refers to either the total number of instances aligned (for calculating *p* of precision) or the total number of instances in the gold standard (for calculating *p* of recall).

$$p = \frac{p_0 + t/2}{1+t} \pm \frac{\sqrt{p_0 q_0 t + t^2/4}}{1+t},\ where\ q_0 = 1 - p_0,\ t = \lambda^2/n \quad (6.10)$$

6.3 Experimental Results and Analysis

6.3.1 Overall Performance Results

The overall performance results are summarized in Table 6.6. An overall performance of 93.4% recall [with confidence interval as (91.4%, 95.0%) at 95% confidence level] and 94.7% precision [with confidence interval as (92.8%, 96.1%) at 95% confidence level] was achieved. This indicates that the proposed semantic information alignment method is promising.

Table 6.6. Performance of Aligning BIM Instances to Semantic Information Elements (SIEs)

Total number of BIM instances	Subject	Subject restriction	Compliance checking attribute	Deontic operator indicator[a]	Quantitative relation[a]	Comparative Relation[a]	Quantity Value	Quantity unit/reference	Quantity restriction	Total
In gold standard	176	133	159	N/A	N/A	N/A	138	138	0	744
Aligned	190	109	159	N/A	N/A	N/A	138	138	0	734
Correctly aligned	174	109	140	N/A	N/A	N/A	134	138	0	695
Precision	91.6%	100.0%	88.1%	N/A	N/A	N/A	97.1%	100.0%	N/A	94.7%
Recall	98.9%	82.0%	88.1%	N/A	N/A	N/A	97.1%	100.0%	N/A	93.4%

[a]SIE that is only used for representation of regulatory information.

An example of the implementation of the proposed semantic information alignment method is shown in Figure 6.13, where the BIM instance “IfcDuctSegment#4940907” is a match to the regulatory object concept “supply and return air duct” (an SIE “subject”), because the duct instance has the property value “supply air” (i.e., it is a supply air duct), “IfcBuilding#163” is a match to the object concept “building” (an object referenced in the SIE “subject restriction”), “Thermal Resistance” is a match to the property concept “R-value” (an SIE “compliance checking attribute”), and both the unit of “Thermal Resistance” and the unit referring to the SIE “quantity unit” are represented in the 3-tuple representation.

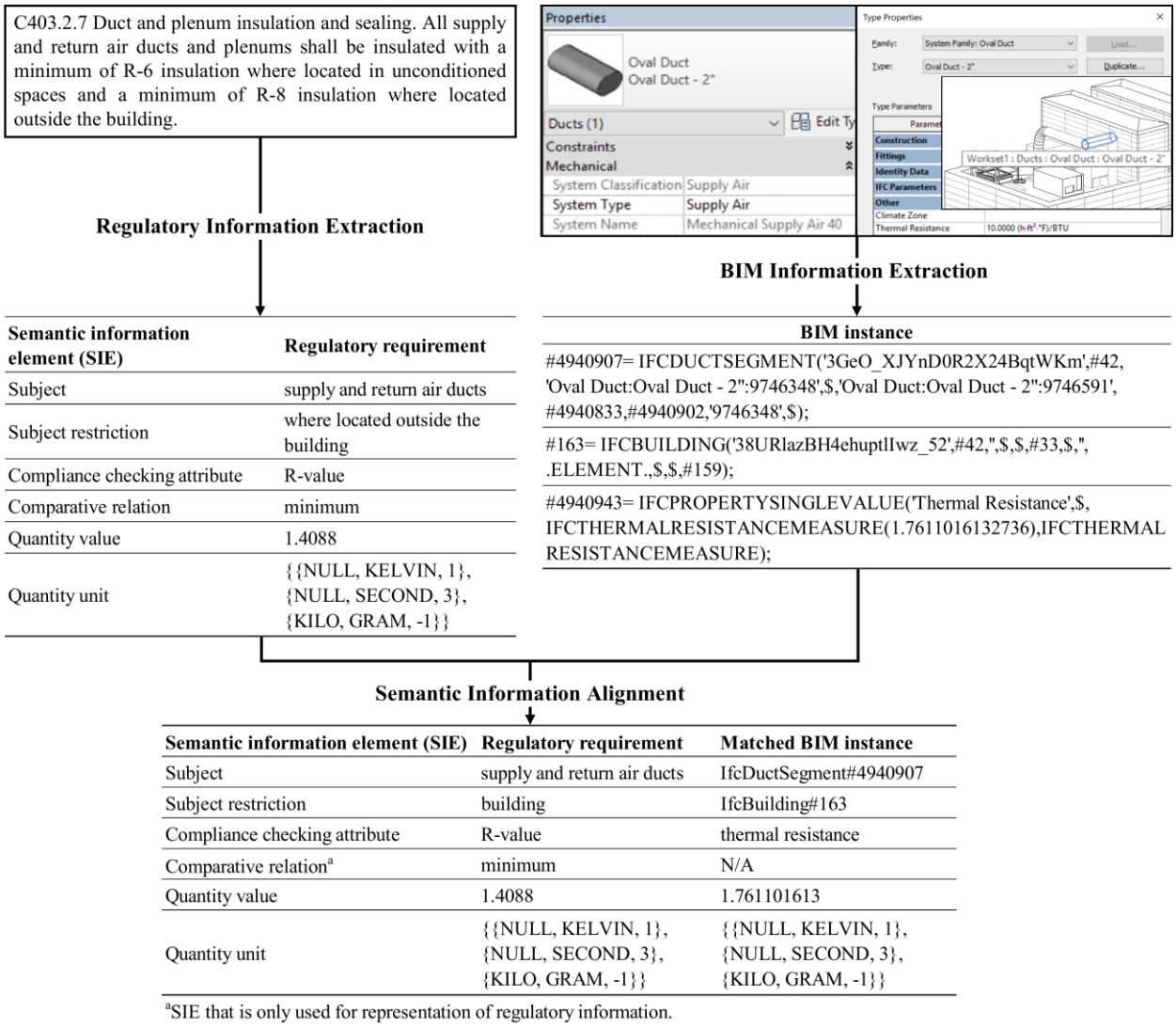


Figure 6.13. Example of matched BIM information to a regulatory requirement

6.3.2 Error Analysis for First-Level Simple Alignment

Three main sources of errors in first-level simple alignment were identified: ambiguity of regulatory concepts, noise of BIM instances, and errors in semantic similarity analysis. Concepts may be ambiguously expressed in the text; they may not be easily mapped to the BIM instances (Garrett et al. 2014), which may result in incorrect or missing recognition of matches. For example,

(1) the property concept "diameter" was ambiguously stated in sentence S1, which resulted in recognizing both the property instances "outside diameter" and "inside diameter" as matches (only one is correct); and (2) the object concept "conditioned space" in S2 has a special definition in the energy regulatory documents, which resulted in failure of interpretation using only the concept descriptors to find the matches.

- S1: "…associated with piping 1 inch or less in diameter." (ICC 2012)
- S2: "An area or room within a building being heated or cooled, containing uninsulated ducts, or with a fixed opening directly into an adjacent conditioned space." (ICC 2012)

The noise in BIM instances may result in incorrectly recognizing the matches for the regulatory concepts. For example, two BIM instances for the concept "balancing valve" were incorrectly recognized as matches for the concept "control valve", because their instance descriptors – "Adjusting/Controlling Valves for Liquid Services" – share two common adjacent terms "controlling valves", which resulted in a large-enough degree of similarity to the concept "control valve".

The errors in semantic similarity analysis come from the semantic similarity scoring function and term position weighting function. First, the semantic similarity scoring function excludes the rare terms (i.e., term with a frequency less than five) in calculating the total instance-concept semantic similarity to indicate the matching, which may result in failing to distinguish incorrect matches. For example, both "aged solar reflectance index" and "solar reflectance index" were incorrectly recognized as matches to the concept "solar reflectance", because the rare terms "aged" and "index"

were excluded, which resulted in equivalent total instance-concept similarities to that of the correct match “solar reflectance”. This indicates that some rare terms may also carry meaningful information in the energy domain. Second, the term position weighting function overweights the rightmost terms in a concept, resulting in giving those terms excessive power to indicate the matching. For example, the BIM instance “Glass-Selux-Heat Tempered Convex Lens” was incorrectly recognized as a match to the concept “radiant heating”, because the common term “heat” is the rightmost term in the concept and was therefore given an overweighted common-term similarity to indicate the matching.

6.3.3 Error Analysis for Final Complex Alignment

Two main sources of errors in final complex alignment were identified. First, supervised searching may fail to identify the instance pairs created by the spatially-contained relationships that are captured using the bounding box geometric assessment, because some object instances may not be represented in bounding boxes. For example, two IfcSlab instances were not linked to two IfcSpace instances as instance pairs by the bounding box geometric assessment, because the two IfcSlab instances were represented in the swept solid geometric representation rather than bounding boxes. Second, unsupervised searching may not recognize the correct instance pairs, because the relationships that are found by searching may not match the relationship that creates a concept pair. For example, the unsupervised searching incorrectly recognized the IfcDuctSegment (#4940907) and the IfcBuilding (#163) as an instance pair corresponding to the concept pair “supply and return

air duct" and "building". The relationship that creates the concept pair is "the duct is outside the building" (as per S3), but instance #4940907 is inside instance #163.

- S3: "C403.2.7 Duct and plenum insulation and sealing. All supply and return air ducts and plenums shall be insulated with a minimum of R-6 insulation where located in unconditioned spaces and a minimum of R-8 insulation where located outside the building." (ICC 2012)

CHAPTER 7 – A CASE STUDY OF FULLY-AUTOMATED ENERGY COMPLIANCE CHECKING USING THE ENERGYACC PROTOTYPE

7.1 EnergyACC Prototype

A fully-automated energy compliance checking prototype, called "EnergyACC", was used to conduct a case study experiment for checking a BIM for compliance with building energy efficiency requirements from energy codes and contract specifications. The developed methods and algorithms (Chapters 3 to 6) were implemented in the prototype. The prototype, thus, includes four main modules: text classification, information extraction, information alignment, and compliance reasoning. An overview of the EnergyACC prototype is illustrated in Figure 7.1.

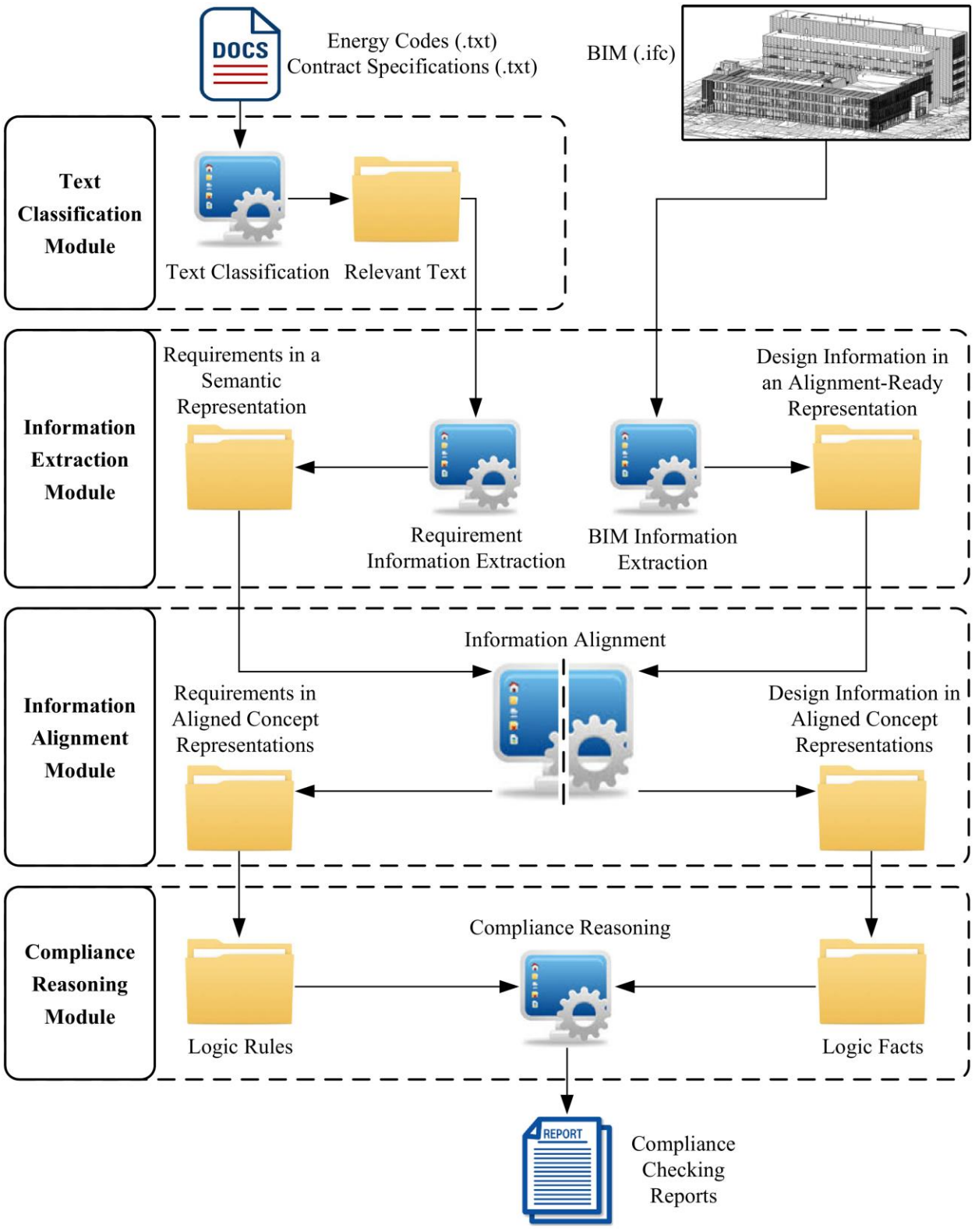


Figure 7.1. Overview of the fully-automated energy compliance checking prototype (EnergyACC)

The text classification (TC) module uses the ontology-based TC method and algorithm developed in Section 3.2 to filter out irrelevant text in the energy codes and contract specifications to avoid unnecessary processing effort and potential errors from the subsequent modules, thereby improving the efficiency and performance of checking.

The information extraction (IE) module uses the ontology-based IE methods and algorithms developed in Chapters 4 and 5 to automatically extract requirements from energy codes and contract specifications to a computer-interpretable rule format, and uses the BIM IE method from Section 6.2.2.1 to automatically extract design information from BIMs to an alignment-ready representation.

The information alignment module uses the semantic information alignment method and algorithm developed in Chapter 6 to automatically match the concept representations of the extracted energy requirements to those of the design information, so that the requirements and the BIM "speak" the same language.

The compliance reasoning module performs the check automatically and reports the results. First, the requirements and the matched design information are automatically transformed into logic rules and logic facts, respectively. Second, using logic reasoning, the logic facts are checked for compliance with the logic rules. The logic rules and facts are represented using the semantic logic representations proposed in Zhang and El-Gohary (2016), with necessary adaptations for considering the complex conjunctive/disjunctive relationships among multiple requirements in a

provision (i.e., clause/article). Each logic rule represents a requirement, and is composed of three logic clauses (as shown in Figure 7.2): (1) a primary logic clause, which is the core representation of a requirement (e.g., as shown in bold in Figure 7.2) and represents the compliant case. The noncompliant case is inferred from the compliant case following a close-world assumption (i.e., design information is noncompliant with the requirement if it is not compliant with the primary logic clause); and (2) two secondary logic clauses for supporting the primary logic clause, in which one (e.g., secondary logic clause 1 in Figure 7.2) represents the activation conditions for checking the requirement, and the other (e.g., secondary logic clause 2 in Figure 7.2) serves as a reporting of the checking results. An example of a logic rule and its corresponding logic facts, in B-prolog language, is shown in Figure 7.2. A complete compliance checking example, to illustrate the outputs of the four modules, is shown in Figure 7.3.

Logic Rule

Primary Logic Clause	Secondary Logic Clause 1	Secondary Logic Clause 2
compliance_three_year_aged_solar_reflectance_index_of_roofs(Three_year_aged_solar_reflectance_index, Roofs) :- **roofs(Roofs), relation(Roofs,Climate_zones), climate_zones(Climate_zones), (relation(Climate_zones,1); relation(Climate_zones,2); relation(Climate_zones,3)), three_year_aged_solar_reflectance_index(Three_year_aged_solar_reflectance_index), relation(Roofs,Three_year_aged_solar_reflectance_index), greater_than_or_equal(Three_year_aged_solar_reflectance_index, quantity(64, {{null, one, '1'}}))**.	check_5_5_3_1_1_roof_solar_reflectance_and_thermal_emittance(X) :- (roofs(Roofs), relation(Roofs,Climate_zones), climate_zones(Climate_zones), (relation(Climate_zones,1); relation(Climate_zones,2); relation(Climate_zones,3)), three_year_aged_solar_reflectance_index(X), relation(Roofs,X)) -> check_three_year_aged_solar_reflectance_index_of_roofs(X,Roofs);true.	check_three_year_aged_solar_reflectance_index_of_roofs(X,Y) :- compliance_three_year_aged_solar_reflectance_index_of_roofs(X,Y) -> (writeln((X,of,Y,is,compliant,with,'5.5.3.1.1,Roof Solar Reflectance and Thermal Emittance',requirement2_1))); (writeln((X,of,Y,is,noncompliant,with,'5.5.3.1.1, Roof Solar Reflectance and Thermal Emittance', requirement2_1,it,should,be, 'greater_than_or_equal, 64')), false).

Logic Facts

```
roofs('basic_roof_epdm_4_1_2_lsf_11_9839622#5328658').
relation('basic_roof_epdm_4_1_2_lsf_11_9839622#5328658','climate_zone#5328997').
climate_zones('climate_zone#5328997').relation('climate_zone#5328997',3).
relation('basic_roof_epdm_4_1_2_lsf_11_9839622#5328658','aged_sri#4202728').
three_year_aged_solar_reflectance_index('aged_sri#4202728').
get_quantity('aged_sri#4202728', 10.0, {{null, one, '1'}}).
```

Figure 7.2. An example of a logic rule and its corresponding logic facts in B-prolog language

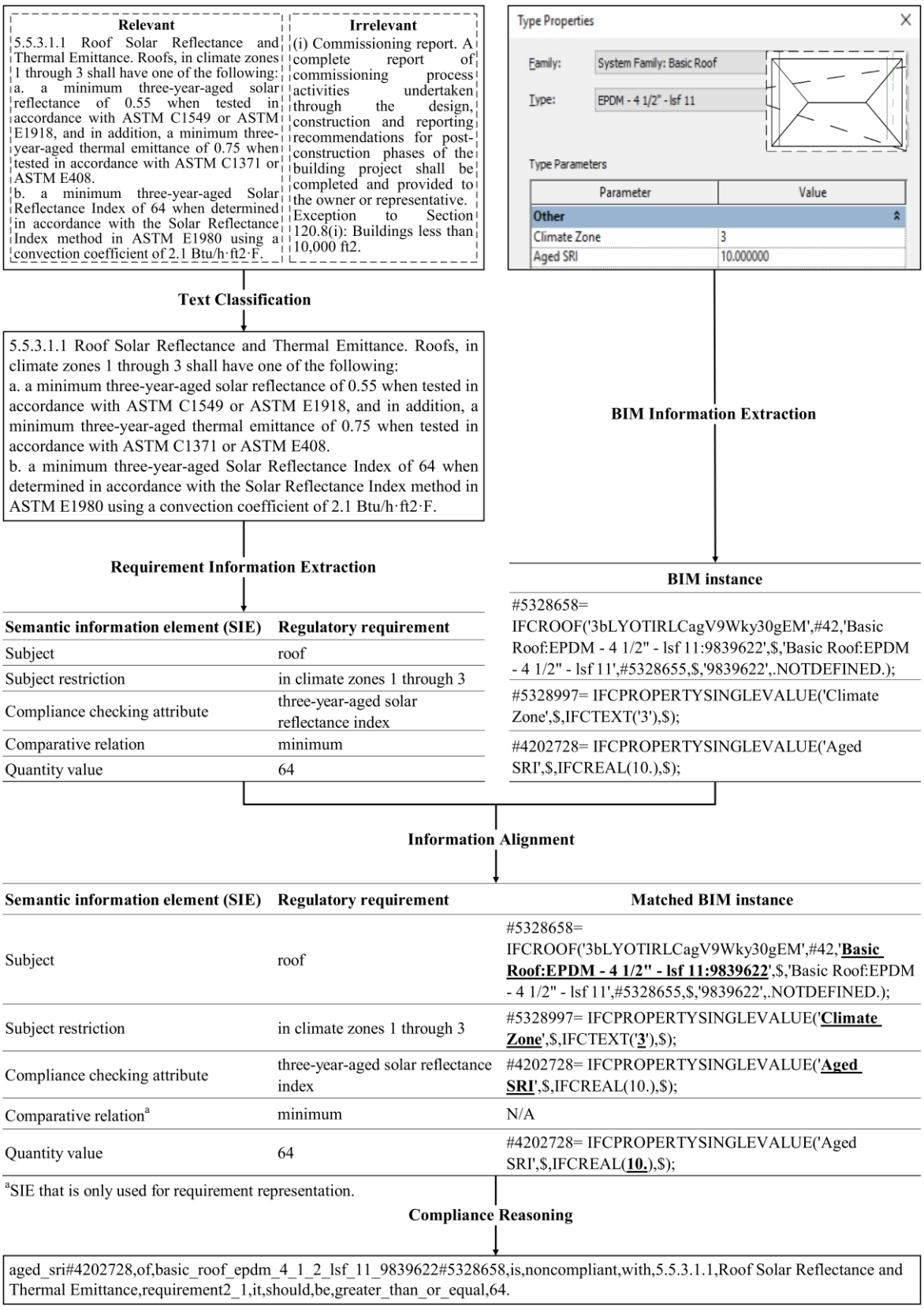


Figure 7.3. An example to illustrate energy compliance checking and reporting

The EnergyACC prototype was implemented in Java programming language using the Eclipse OXYGEN (Eclipse Foundation 2017). A number of softwares and application programming interfaces (APIs) were used to accomplish the specific tasks in each module. For text classification, the Stanford CoreNLP API (Manning et al. 2014) and the Snowball API (Richard 2018) were used to implement the text preprocessing techniques – tokenization and stemming. The deeplearning4j API (Deeplearning4j Development Team 2016) was used to learn and measure semantic similarities.

For requirement IE, the "a nearly new information extraction" (ANNIE) system of the General Architecture for Text Engineering (GATE) 8.0 (Cunningham et al. 2011) was used to implement the rule-based IE. Specifically, the following processing resources in ANNIE were used. The ANNIE English Tokeniser, ANNIE Sentence Splitter, and GATE Morphological Analyser were used to implement the text preprocessing techniques – tokenization, sentence splitting, and morphological analysis. The ANNIE POS Tagger, ANNIE Gazetteer, and OntoRootGazetteer were used to generate the syntactic and semantic features of the text. The extraction rules were developed in the Java Annotation Patterns Engine (JAPE) grammar, and were added to the JAPE Transducer for execution.

For BIM IE, the Java Standard Data Access Interface (JSDAI) API (LKSoftWare GmbH 2016) was used to implement the EXPRESS-based data extraction.

For information alignment, the buildingSMART Data Dictionary (bSDD) API (buildingSMART 2016b) and the Apache Jena Ontology API (Apache Jena 2016) were used to access the bSDD and the energy ontology for concept matching. The deeplearning4j API (Deeplearning4j Development Team 2016) was used to learn and measure semantic similarities.

For compliance reasoning, the logic rules and facts were represented in B-prolog language (Zhou 2014). The built-in interpreter of B-prolog 8.1 (Afany Software 2013) was used for automated compliance reasoning.

7.2 Case Study Experiment

7.2.1 Overall Design of the Experiments

The experiments aimed to identify the feasibility and challenges for fully-automated and generalized compliance checking across different types of documents – particularly energy codes versus contract specifications. A fully-automated energy compliance checking prototype, called EnergyACC (in Section 7.1), was used to check a BIM for compliance with energy requirements from energy codes and contract specifications. The experimental results were analyzed to answer three primary research questions. What are the performances of automated energy code checking and automated contract specification checking? What are the errors in both cases, and how do they compare? How do the errors propagate through the different prototype modules, in both cases? The first question aims to assess whether acceptable performance could be achieved across different types of documents (i.e., energy codes versus contract specifications) – and how would

the performance compare to the state of the art – to assess the feasibility of generalized automated approaches. The second question aims to study the errors to identify the challenges to automation and generalizability. The third question aims to study the error propagation features to identify the most critical errors to avoid. To limit the scope of the study, the experiments focused on two subtypes of energy requirements, thermal insulation and lighting power requirements. Two test cases were used in the experiments, which were developed based on a BIM of an educational building project in Illinois (called thereafter case study), three energy codes, and the contract specifications of the project (as described in Section 7.2.2).

7.2.2 Preparation of the Test Cases for Experiments 1 and 2

Two test cases were prepared: one for the energy code checking experiment (Experiment 1) and one for the contract specification checking experiment (Experiment 2). For Experiment 1, the preparation of the testing case included two primary steps: (1) selecting a set of testing requirements from the energy codes, and (2) adding compliant and noncompliant design information to the BIM of the case study for ensuring a variety of compliance and noncompliance cases are tested. For Experiment 2, similar steps were conducted, except that the requirements were selected from the contract specifications.

7.2.2.1 Selection of Testing Requirements from the Energy Codes for Experiment 1

For the energy codes, the following chapters/sections of energy codes were selected: Chapter 4 of the 2012 International Energy Conservation Code (ICC 2012), Subchapters 2-5 of the 2013

Building Energy Efficiency Standards (known as the California Energy Code) (California Energy Commission 2013), and Sections 5-10 of the Ontario Building Code Supplementary Standard SB-10 (Ontario Ministry of Municipal Affairs 2011). The three energy codes were selected because they represent energy codes developed by international councils, specific U.S. states, and other countries, respectively. The aforementioned chapters/sections were selected (which included 274 clauses) because of the scope of the experiments (focusing on thermal insulation requirements and lighting power requirements). The set of 274 clauses were then used for testing the text classification, in which 112 clauses are relevant. Sixteen of the 112 were randomly selected for testing the noncompliance detection. The 16 clauses include a total of 43 requirements and 10 exceptions – 33 requirements and 10 exceptions from text and 10 requirements from tables. The text and the tables (in the selected clauses) were prepared in different ways. The text was saved in .txt format prior to text classification. The tables were first saved in .htm format for retaining the table structure information, and a Hypertext Markup Language (HTML)-based parsing method was subsequently used to parse and transform the HTML tables into a number of sentences based on the table structure patterns. These sentences were saved in .txt format for information extraction.

7.2.2.2 Selection of Testing Requirements from the Contract Specifications for Experiment 2

For the contract specifications, the specifications of the case study was used, which is in MasterFormat. The sections in divisions 07 and 26 were further selected for the aforementioned scope reasons. A set of 300 articles (a comparable number to 274 clauses) were then randomly

selected for testing the text classification, in which 104 articles are relevant. Thirteen of the 104 articles were randomly selected for testing the noncompliance detection. The 13 articles include a total of 36 requirements from text, and zero requirement from tables. The text in the selected articles was prepared in a similar way to that for the energy codes.

7.2.2.3 Addition of BIM Design Information for Experiments 1 and 2

For the BIM, the Revit model of the case study project was used. The Revit model was saved as two .rvt files – one for conducting Experiment 1 and the other for Experiment 2. Design information was added to both files to include both compliant and noncompliant design information – for each requirement – to test different all possible compliance cases. So, design information was added for each of the 43 (with 10 exceptions) and 36 requirements for Experiments 1 and 2, respectively.

Four compliance cases were considered. First, if a provision (clause or article) contains one or multiple requirements, compliant and noncompliant design information for each requirement was included. Second, if there are conjunctive/disjunctive relationships among multiple requirements in a provision, compliant and noncompliant design information for the whole provision through all possible ways of conjunctive/disjunctive relationships was included. Third, if a provision has exception(s), compliant and noncompliant design information for each exception was included. Fourth, if a requirement has subject restriction(s), compliant and noncompliant design information that satisfies (or not) the subject restriction(s) was included. For example, the provision C1

contains three quantitative requirements, with a disjunctive relationship (i.e., one of) between alternative a and b, and a conjunctive relationship (i.e., and) within alternative a. Thus, eight design information sets were created, which correspond to eight scenarios: (1) scenario 1: only the attribute "three-year-aged solar reflectance index" is noncompliant (i.e., alternative a is compliant and b is noncompliant); (2) scenarios 2-4: one or both of the three-year-aged solar reflectance and the three-year-aged thermal emittance are noncompliant, while the three-year-aged solar reflectance index is compliant (i.e., alternative a is noncompliant and b is compliant); (3) scenarios 5-7: same as scenarios 2-4 except that the three-year-aged solar reflectance index is noncompliant (i.e., both alternative a and b are noncompliant); and (4) scenario 8: no attributes are noncompliant (i.e., both alternative a and b are compliant). In sum, a total of 222 and 93 design information sets were created for Experiment 1 and 2, which include 70 and 45 noncompliant instances, respectively.

- C1: "5.5.3.1.1 Roof Solar Reflectance and Thermal Emittance. Roofs, in climate zones 1 through 3 shall have one of the following:
  a. a minimum three-year-aged solar reflectance of 0.55 when tested in accordance with ASTM C1549 or ASTM E1918, and in addition, a minimum three-year-aged thermal emittance of 0.75 when tested in accordance with ASTM C1371 or ASTM E408.
  b. a minimum three-year-aged Solar Reflectance Index of 64 when determined in accordance with the Solar Reflectance Index method in ASTM E1980 using a convection coefficient of 2.1 Btu/h·ft^2·F."

The design information for Experiments 1 and 2 was created based on a few public BIM object libraries [e.g., NBS national BIM library (NBS 2018), SmartBIM (SmartBIM Technologies 2018), BIMobject (BIMobject Corporation 2018), ARCAT (ARCAT Inc. 2018), RevitCity (Pierced

Media LC 2018)], and added to the .rvt files using Revit 2016 (Autodesk 2016a). The .rvt files were then exported to .ifc data files, in IFC4 schema, using the Revit IFC exporter for BIM IE. The Revit IFC exporter (Autodesk 2016b) was extended for exporting energy-related design information (e.g., material thermal properties) in C# programming language using the Microsoft Visual Studio 2015 (Microsoft 2016).

7.2.3 Evaluation Metrics

Recall and precision of noncompliance detection were used to evaluate the checking performance. Recall refers to the percentage of the total number of correctly detected noncompliant instances out of the total number of noncompliant instances that should be detected. Precision refers to the percentage of the total number of correctly detected noncompliant instances out of the total number of noncompliant instances detected. Two gold standards – for Experiments 1 and 2 – were manually developed based on the prepared test cases (in Section 7.2.2) for performance evaluation. The gold standards include the ground truth of both compliant and noncompliance instances. The gold standards were developed by three annotators – the author and two other researchers. Initial inter-annotator agreements of 92% and 94% in F-measure were achieved for Experiments 1 and 2, respectively. These scores were considered sufficient: “an F-measure of 0.8 or above is generally considered sufficient inter-annotator agreement” (Pestian et al. 2012). Final full annotator agreement were achieved after the discrepancies were discussed and resolved to reach consensus.

7.3 Case Study Results and Analysis

The experimental results were analyzed to answer the aforementioned research questions (in Section 7.2.1), as described in the following subsections.

7.3.1 What are the Performances of Automated Energy Code Checking and Automated Contract Specification Checking?

This research question was further broken down into two questions. Is acceptable performance achieved for both energy code checking and contract specification checking? How does the performance compare to the state of the art?

7.3.1.1 Is Acceptable Performance Achieved for Both Energy Code Checking and Contract Specification Checking?

Acceptable – or even high – performance was achieved for both energy code checking and contract specification checking, as shown in Table 7.1. An acceptable level of performance could be defined as above 85% recall and precision, with recall given higher priority because recall errors mean noncompliance instances are missed while precision errors are easily addressed through human verification. Contract specification checking achieved a perfect recall (100%) and 86.5% precision in noncompliance detection, while energy code checking reached 95.7% recall and 85.9% precision. This level of performance indicates that it is feasible to develop a fully-automated and generalized ACC method across different types of regulations/documents. Table 7.2 also shows the performance of the different modules, for both energy code checking and contract specification checking.

Table 7.1. Experimental Results for Noncompliance Detection

Total number of noncompliant instances	Results	
	Energy codes	**Contract specifications**
In gold standard	70	45
Detected	78	52
Correctly detected	67	45
Precision	85.9%	86.5%
Recall	95.7%	100.0%

Table 7.2. Experimental Results for the Different Modules

Prototype module		Energy codes		Contract specifications	
		Precision	**Recall**	**Precision**	**Recall**
Text classification		95.6%	97.3%	94.2%	93.3%
Information extraction (IE)	Requirement IE	99.3%	95.8%	94.0%	100.0%
	BIM IE	100.0%	100.0%	100.0%	100.0%
Information alignment		95.5%	91.6%	96.7%	96.0%
Compliance reasoning		100.0%	100.0%	100.0%	100.0%

7.3.1.2 How does the Performance Compare to the State of the Art?

The performance was compared to the state of the art [i.e., Zhang and El-Gohary (2017)], which is a fully-automated ACC effort – except for BIM-requirement matching, which is semi-automated – that focuses on building code checking. As shown in Figure 7.4, energy code checking achieved a comparable, but lower, performance to Zhang and El-Gohary (2017), with 2.3% lower F-1 measure (90.5% % vs. 92.8%), 3% lower recall (95.7% vs. 98.7%), and 1.7% lower precision (85.9% vs. 87.6%). These minor performance drops are likely caused by the challenging text complexities in energy codes – long provisions, requirement exceptions, and hierarchically-complex sentence structures. This indicates the importance of addressing the text complexities in different types of documents for achieving high performance levels. It is noted, however, that EnergyACC is entirely

automated, while Zhang and El-Gohary (2017) involves minimal manual effort because its method for BIM-requirement matching is semi-automated – not fully-automated.

Contract specification checking, on the other hand, reached a comparable performance to Zhang and El-Gohary (2017), with same F1-measure (92.8% for both), 1.3% higher recall (100% vs 98.7%), and 1.1% lower precision (86.5% vs 87.6%). It is noted, however, that perfect recall is more desirable, as it ensures that no noncompliance instances are missed. From that perspective, contract specification checking could be seen as outperforming Zhang and El-Gohary (2017). The results also indicate that EnergyACC was successful in addressing the text complexities in contract specifications (i.e., incomplete sentence structures, hierarchically-complex text structures, and variety of LODs).

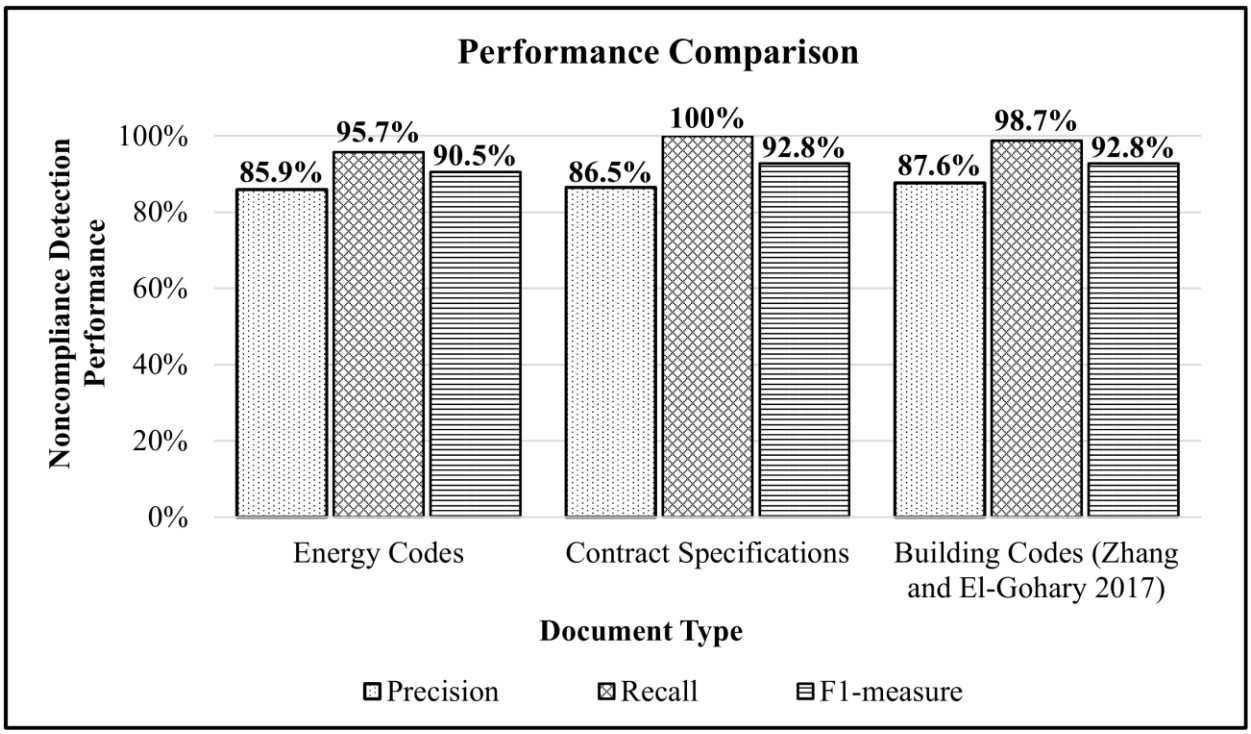


Figure 7.4. Performance comparison to the state of the art

7.3.2 What are the Errors in Energy Code Checking and Contract Specification Checking, and How do they Compare?

This research question was further broken down into four questions. What are the sources of errors that only belong to energy code checking? What are the sources of errors that only belong to contract specification checking? What are the common sources of errors between energy code checking and contract specification checking? What does the error analysis indicate?

7.3.2.1 What are the Sources of Errors that Only Belong to Energy Code Checking?

Errors occurred in text classification, requirement IE, and information alignment. No errors occurred in BIM IE or compliance reasoning.

7.3.2.1.1 What are the Sources of Errors in Text Classification?

Two sources of text classification errors were identified: topic semantic dominance and text preprocessing errors. First, semantic dominance of a primary topic (label) in a clause may result in missing secondary labels (i.e., the main meaning of a clause is too strong to make the semantic similarity measurement "believe" that the clause is also related to any secondary label). For example, a secondary label, "thermal insulation topic", was missed for the clause C2, because its primary topic, "air leakage topic", is semantically dominant. Second, hyphen-catenated concepts that carry discriminative information were split into individual terms during text preprocessing, which may result in missing discriminative information when measuring semantic similarities and consequently missing relevant labels. For example, a label "thermal insulation topic" was missed for the clause C3, because the semantically distinctive concept "R-value" was split into a stopword

“R” and a common term “value”; both terms are individually not semantically discriminative.

- C2: “C403.2.7.1.3 High-pressure duct systems. Ducts designed to operate at static pressures in excess of 3 inches water gauge (w.g.) (750 Pa) shall be insulated and sealed in accordance with Section C403.2.7. In addition, ducts and plenums shall be leak-tested in accordance with the SMACNA HVAC Air Duct Leakage Test Manual with the rate of air leakage (CL) less than or equal to 6.0 as determined in accordance with Equation 4-5. Documentation shall be furnished by the designer demonstrating that representative sections totaling at least 25 percent of the duct area have been tested and that all tested sections meet the requirements of this section.”
- C3: “C402.1.1 Insulation and fenestration criteria. The building thermal envelope shall meet the requirements of Tables C402.2 and C402.3 based on the climate zone specified in Chapter 3. Commercial buildings or portions of commercial buildings enclosing Group R occupancies shall use the R-values from the "Group R" column of Table C402.2. Commercial buildings or portions of commercial buildings enclosing occupancies other than Group R shall use the R-values from the "All other" column of Table C402.2. Buildings with a vertical fenestration area or skylight area that exceeds that allowed in Table C402.3 shall comply with the building envelope provisions of ANSI/ASHRAE/IESNA 90.1.”

7.3.2.1.2 What are the Sources of Errors in Information Extraction?

Two sources of information extraction errors were identified: extraction rule errors and cross reference errors. First, the extraction rules may fail to extract target information from long provisions that contain complex SIEs (e.g., “Subject Restriction”). For example, IE errors occurred in clause C4, which is a long provision that has a complex “Subject Restriction”. The “floor structures” and “incorporating radiant heating”, which were a subject and a subject restriction for a floor insulation requirement, were missed, because they were incorrectly extracted as part of the complex subject restriction (i.e., “designed for sensible heating of an indoor space through heat transfer from the thermally effective panel surfaces to the occupants or indoor space by thermal radiation and natural convection and the bottom surfaces of floor structures incorporating radiant

heating") for a radiant panel insulation requirement. Second, the information in text and tables is not linked, which may result in extraction errors when a table is cross referenced in a clause. For example, the requirements extracted from "Table C402.2.1.1" were missing the subject restriction "in Climate Zones 1, 2, and 3" (included in the cross-referenced clause C5).

- C4: "C402.2.8 Insulation of radiant heating systems. Radiant panels, and associated U-bends and headers, designed for sensible heating of an indoor space through heat transfer from the thermally effective panel surfaces to the occupants or indoor space by thermal radiation and natural convection and the bottom surfaces of floor structures incorporating radiant heating shall be insulated with a minimum of R-3.5 (0.62 m^2/K x W)."
- C5: "C402.2.1.1 Roof solar reflectance and thermal emittance. Low-sloped roofs, with a slope less than 2 units vertical in 12 horizontal, directly above cooled conditioned spaces in Climate Zones 1, 2, and 3 shall comply with one or more of the options in Table C402.2.1.1."

7.3.2.1.3 What are the Sources of Errors in Information Alignment?

Three sources of information alignment errors were identified: regulatory concept ambiguity errors, semantic similarity analysis errors, and relationship searching errors. First, it may be hard to automatically recognize matches for ambiguous concepts without human (domain expert) involvement. For example, both "inside diameter" and "outside diameter" were matched to the concept "diameter" in clause C6. Since no information is explicitly stated, a domain expert is needed to know that the correct match in this case is "inside diameter". Similarly, the concepts "building envelope assembly" and "unconditioned space" in clause C7 have particular definitions in energy codes, which makes it challenging for machine intelligence to capture such implicit domain knowledge to automatically find the matches. Second, semantic similarity analysis excludes rare terms (e.g., those with term frequency less than 5) and hyphen-catenated stopwords

in calculating the instance-concept semantic similarities to indicate the matching, which may result in failure to distinguish the matches. For example, “aged solar reflectance” and “solar reflectance index” showed equivalent instance-concept semantic similarities to the concept “three-year-aged solar reflectance”, because the rare terms “index” and “aged” and the hyphen-catenated stopwords “three-year” were excluded from assessing the semantic similarities. This indicates that rare terms and hyphen-catenated stopwords may also carry meaningful information. Third, the relationship searching does not consider the semantic types of the relationships when linking BIM instances to requirement concepts during alignment, which may result in incorrect matches. For example, an instance of indoor IfcDuctSegment (“Rectangular Duct 1"”) was incorrectly linked to the concept “supply and return air duct and plenum” in clause C7, which is incorrect because the former is a duct inside the building but the latter is a duct outside the building.

- C6: “Exceptions:
  5. Strainers, control valves, and balancing valves associated with piping 1 inch (25 mm) or less in diameter.”
- C7: “C403.2.7 Duct and plenum insulation and sealing. All supply and return air ducts and plenums shall be insulated with a minimum of R-6 insulation where located in unconditioned spaces and a minimum of R-8 insulation where located outside the building. Where located within a building envelope assembly, the duct or plenum shall be separated from the building exterior or unconditioned or exempt spaces by a minimum of R-8 insulation.”

7.3.2.2 What are the Sources of Errors that Only Belong to Contract Specification Checking?

Similar to energy code checking, errors occurred in text classification, requirement IE, and information alignment. No errors occurred in BIM IE or compliance reasoning.

7.3.2.2.1 What are the Sources of Errors in Text Classification?

Two text classification errors were identified: semantic similarity measurement errors and unbalance of relevant and irrelevant information. First, the section names of articles were used in measuring the semantic similarities because they may carry discriminative information. However, sometimes they may also be misleading, which led to false positive errors. For example, the section name “Variable Frequency Drives” introduced misleading information, which resulted in an incorrect label “lighting power topic” to the article A1. Second, sometimes irrelevant information overshadows relevant information, which may lead to false negative errors. For example, the label “thermal insulation topic” was missed for the article A2, because it included a lot of irrelevant information.

- A1: “262923 Variable Frequency Drives,
  2 – PRODUCTS,
  2.1 ACCEPTABLE VFD MANUFACTURERS,
  A. ABB - ACS 800 Series,
  B. Allen Bradley - PowerFlex 700 Configured Package Drives,
  C. Yaskawa - E7C Configured Series.”
- A2: “074213 Metal Wall Panels
  1 – GENERAL
  1.9 DELIVERY, STORAGE, AND HANDLING
  A. Deliver components, sheets, metal panels, and other manufactured items so as not to be damaged or, deformed. Package metal panels for protection during transportation and handling.
  B. Unload, store, and erect metal panels in a manner to prevent bending, warping, twisting, and surface damage.
  C. Stack metal panels horizontally on platforms or pallets, covered with suitable weather-tight and ventilated, covering. Store metal panels to ensure dryness, with positive slope for drainage of water. Do not store metal, panels in contact with other materials that might cause staining, denting, or other surface damage.
  D. Retain strippable protective covering on metal panel for period of metal panel installation.

E. Protect foam-plastic insulation as follows:

1. Do not expose to sunlight, except to extent necessary for period of installation and concealment.

2. Protect against ignition at all times. Do not deliver foam-plastic insulation materials to Project site before, installation time.

3. Complete installation and concealment of plastic materials as rapidly as possible in each area of, construction."

7.3.2.2.2 What are the Sources of Errors in Information Extraction?

The information extraction errors resulted from implicit context semantics errors – failures in capturing the implicit context semantics in conflict resolution. For example, in article A3, both "metal halide HID lamp" and "pulse start lamp" were candidates of "Subject" and the latter was extracted after conflict resolution, but the "Subject" should be "pulse start metal halide lamp".

- A3: "265100 Lighting
  2 – PRODUCTS
  2.6 HID LAMPS
  A. Metal Halide HID Lamps: Clear, suitable for all burning positions. Pulse start lamps shall have a lumen maintenance factor of .75 or greater and a CRI of 65 or greater with color stability of +/- 100°K. Correlated Color Temperature (CCT) to match fluorescent lamps, unless noted otherwise."

7.3.2.2.3 What are the Sources of Errors in Information Alignment?

Two semantic similarity analysis errors were identified: concept name semantics errors and BIM descriptor errors. First, the semantic similarity analysis may fail to capture the semantics behind concept names to distinguish the matches. For example, the meaning of "underground" in the concept "underground conduit" was not captured, which resulted in recognition of both underground and aboveground conduits as matches. Second, some types of BIM instance

descriptors may not be properly used in assessing the instance-concept semantic similarities to indicate the matching. For example, an elevator “Elevator-Electric[1]” was incorrectly matched to the concept “light emitting diode”, because the name “"LED Strip Light2:LED Strip Light:6068776” of a lighting fixture inside that elevator was used as the elevator descriptor, which resulted in a sufficient degree of instance-concept semantic similarity to (incorrectly) indicate the matching.

7.3.2.3 What are the Common Sources of Errors Between Energy Code Checking and Contract Specification Checking?

The common sources of errors between energy code checking and contract specification checking were identified. These include text classification, information extraction, and information alignment errors.

7.3.2.3.1 What are the Common Sources of Errors in Text Classification?

The common text classification errors resulted from semantic similarity overweighting. Common concepts between a clause/article and a subontology are semantically discriminative and were given higher semantic similarity values. However, those semantic similarity values are sometimes overweighted, which may result in false positive errors. For example, clause C8 and the subontology of “thermal insulation topic” shared many common concepts (e.g., “extruded polystyrene insulation board”, “foil-back polyisocyanurate insulation board”), which resulted in a sufficient degree of semantic similarity to indicate relevance to the “thermal insulation topic” – which was incorrect. Similarly, the “thermal insulation topic” was incorrectly assigned to article A4 because of their common concepts (e.g., “roof”, “roof curb”).

- C8: “C402.4.1.2.1 Materials.
  Materials with an air permeability no greater than 0.004 cfm/ft^2 (0.02 L/s m^2) under a pressure differential of 0.3 inches water gauge (w.g.) (75 Pa) when tested in accordance with ASTM E2178 shall comply with this section. Materials in Items 1 through 15 shall be deemed to comply with this section provided joints are sealed and materials are installed as air barriers in accordance with the manufacturer's instructions.
  1. Plywood with a thickness of not less than 3/8 inch.
  2. Oriented strand board having a thickness of not less than 3/8 inch.
  3. Extruded polystyrene insulation board having a thickness of not less than 1/2 inch.
  4. Foil-back polyisocyanurate insulation board having a thickness of not less than 1/2 inch.
  5. Closed cell spray foam a minimum density of 1.5 pcf having a thickness of not less than 11/2 inches.
  6. Open cell spray foam with a density between 0.4 and 1.5 pcf and having a thickness of not less than 4.5 inches.
  7. Exterior or interior gypsum board having a thickness of not less than 1/2 inch.”
- A4: “077200 Roof Accessories
  2 – PRODUCTS
  2.3 PRE-FABRICATED ROOF CURBS
  A. Refer to relevant contract sections for roof equipment and associated roof curbs.”

7.3.2.3.2 What are the Common Sources of Errors in Information Extraction?

Two common information extraction errors were identified: insufficient dependency information errors and conjunctive/disjunctive relationship errors. First, dependency information is sometimes insufficient to filter out information instances that match the feature patterns for extraction but despite that are incorrect instances of “Quantity Value” and “Quantity Unit/Reference”. For example, “one” was incorrectly extracted as a “Quantity Value” from clause C9, but it is instead part of a disjunctive relationship indicator (i.e., “one or more of”). Similarly, the “75” and “deg F” were incorrectly extracted from article A5 as “Quantity Value” and “Quantity Unit/Reference”, respectively. Second, conjunctive/disjunctive relationships are sometimes not correctly recognized

due to the sentence structure complexities. For example, the compound sentence in clause C10 contains three independent clauses joined by conjunction (and) and disjunction (or) words and a comma, which make it challenging to correctly recognize the conjunctive/disjunctive relationships among these clauses. Similarly, the incomplete sentence structures led to failures in correctly recognizing the disjunctive relationship in article A6.

- C9: "(b) Alternate Lighting Sources.
  The sign shall comply if it is equipped only with one or more of the following light sources:
  3. Neon or cold cathode lamps with transformer or power supply efficiency greater than or equal to following:
  A. A minimum efficiency of 75 percent when the transformer or power supply rated output current is less than 50 mA; or"

- A5: "072100 Building Insulation
  2 – PRODUCTS
  2.3 MINERAL-WOOL BOARD INSULATION
  C. Form MW-1: Un-faced, Semi-Rigid Mineral-Wool Board Insulation of thickness indicated with width and length as required to suit job conditions: ASTM C612, Type 1A; with maximum flame-spread and smoke-developed indexes of 15 and zero, respectively, per ASTM E84; passing ASTM E136 for combustion characteristics.
  2. Nominal density of 4 lb/cu. ft. (64 kg/cu. m), Types IA and IB, thermal resistivity of 4 deg F x h x sq. ft./Btu x in. at 75 deg F (27.7 K x m/W at 24 deg C)."

- C10: "b. Steep-sloped roofs in Climate Zones 1 through 16 shall have a minimum aged solar reflectance of 0.20 and a minimum thermal emittance of 0.75, or a minimum SRI of 16."

- A6: "262500 Busway
  2 – PRODUCTS
  2.2 INDOOR BUSWAY
  A. Plug-In Busway: NEMA BU 1; 3 phase, 4 wire low impedance plug-in busway rated 208Y/120 or 480Y/277 volts, 60 Hertz. Provide non-ventilated housing with plug-in openings on 24 inch centers each side, with hinged doors to protect opening where plug-in unit is not installed. Ampere ratings as shown on the drawings."

7.3.2.3.3 What are the Common Sources of Errors in Information Alignment?

Three common information alignment errors were identified: BIM noise errors, concept matching errors, and semantic similarity overweighting errors. First, noise in BIM instances may result in incorrect matches. For example, a structural material was incorrectly matched to the concept "insulating material" in the clause C11, because the noisy terms "batt, insulation" in the structural material name (i.e., "Structure - Wood Joist/Rafter Layer, Batt Insulation [Structure]") resulted in a sufficient degree of semantic similarity to indicate the matching. Similarly, a glass fiber board was incorrectly matched to the concept "foil scrim kraft or foil scrim polyethylene vapor retarder", because of the noisy terms "foil, scrim, kraft" in the property value of the glass fiber board [i.e., "It is available unfaced or with a foil-scrim-kraft (FSK) or white kraft-scrim-foil (ASJ) facing adhered to the fiber glass board"].

- C11: "5.8.1.7.3 Insulation materials in ground contact shall have a water absorption rate greater than 0.3% when tested in accordance with ASTM C272."

Second, concept matching may fail to recognize the correct matches, considering the restrictions. For example, the subject restriction "in ground contact" in clause C11 was not considered when recognizing the matches of "insulation material"; i.e., the insulation materials that do not contact the ground were incorrectly recognized. Similarly, for article A7, the matches of "adhesive-coated HDPE sheet" in vertical applications (e.g., walls) were incorrectly recognized.

- A7: "071326 Self-Adhering Sheet Waterproofing
  2 – PRODUCTS
  2.2 ADHESIVE-COATED HDPE SHEET WATERPROOFING

C. Adhesive-Coated HDPE Sheet for Horizontal Applications: 46-mil- (1.2-mm-) thick, uniform, flexible sheets consisting of 30-mil- (0.76-mm-) thick, HDPE sheet coated with a pressure-sensitive rubber adhesive, a protective adhesive coating, a detackifying surface treatment, an uncoated self-adhering side lap strip, and a release liner with the following physical properties:
7. Water Absorption: 0.5 percent; ASTM D570."

Third, common adjacent terms between instances and concepts are sometimes overweighted in calculating the instance-concept semantic similarities, which may result in both false negative and false positive errors. For example, the instance "Amvic Insulated Radiant PEX Panels" was not matched to the regulatory concept "radiant panel" in clause C4, because its instance-concept semantic similarity was assessed as significantly insufficient to indicate the matching, compared to the "Plastic - Berko - Radiant Panels" which shares the common adjacent terms "radiant" and "panel". Similarly, several unfaced semi rigid mineral wool boards were incorrectly matched to the specification concept "foil faced semi rigid mineral wool board" in article A8, because the common adjacent terms "semi, rigid, mineral, word, board" resulted in sufficient instance-concept semantic similarities to indicate the matching.

- A8: "072100 Building Insulation
  2 – PRODUCTS
  2.3 MINERAL-WOOL BOARD INSULATION
  D. Form MW-2: Foil-Faced, Semi-Rigid Mineral-Wool Board Insulation of thickness indicated with width and length as required to suit job conditions: ASTM C612, Type 1A; faced on one side with foil-scrim or foil-scrim-polyethylene vapor retarder having maximum permeance of 0.10 perm (5.75×10^{-9} g/ Pa x s x m^2) when tested in accordance with ASTM E96; with maximum flame-spread and smoke-developed indexes of 25 and 5, respectively, per ASTM E84."

7.3.2.4 What does the Error Analysis Indicate?

The error analysis revealed a number of primary findings. The analysis of text classification errors

indicates that different types of documents may contain different densities of information distribution, which indicates the need for methods to estimate and normalize the information distribution densities across different types of documents for consistent performance across densities/documents. For example, clauses usually contain higher density of energy information than articles: one energy code clause (e.g., C2) may contain many energy requirements related to multiple topics; while a contract specification article (e.g., A2) usually contains only a few energy requirements related to a single topic, plus much irrelevant information.

The analysis of information extraction errors revealed two main findings. First, semantic entities in text are linked into an information network, and capturing this network is essential for extracting a deeper level of information. The information may be linked explicitly (e.g., the references between tables and text are explicitly stated in C5) or implicitly (e.g., the "pulse start lamp" implicitly refers to a "metal halide HID lamp" in A3). Capturing implicitly linked information may be more challenging than the explicit, because some level of knowledge reasoning is required. Second, different types of documents may have different sentence structure complexities, which may indicate the need of different methods for recognizing the conjunctive/disjunctive relationships.

The analysis of information alignment errors revealed two main findings. First, the semantic similarity weightings may not well capture the intrinsic characteristics of different types of documents that impact the degree of semantic similarity. For example, the same number of

common terms may not indicate the same degree of instance-concept semantic similarity; the intrinsic semantic power of each common term in the context of energy codes and contract specifications should be considered. This indicates the need for knowledge bases (e.g., ontology) to capture document-specific knowledge for semantic similarity weightings. Second, semantic ambiguities and implicit domain knowledge may commonly exist in different types of documents. Although energy codes may have deeper levels of ambiguities and implicitness (e.g., concepts may have particular definitions) than contract specifications, both suffer from ambiguities and implicit information. This indicates the existence of a common challenge that could be hard for machine intelligence to address; perhaps human experts are sometimes necessary to clarify ambiguities and capture implicit knowledge.

7.3.3 How do the Errors Propagate through Different Checking Modules, for Both Energy Code Checking and Contract Specification Checking?

This research question was further broken down into three questions. How do the errors propagate through the different checking modules for energy code checking? How do the errors propagate for contact specification checking? What does the error propagation analysis reveal?

7.3.3.1 How do the Errors Propagate for Energy Code Checking?

To analyze the propagation of errors, this research question was further broken down into two questions. Which errors propagate into noncompliance detection errors? Which errors do *not* propagate into noncompliance detection errors?

7.3.3.1.1 Which Errors Propagate into Noncompliance Detection Errors?

Three information extraction errors propagated into noncompliance detection errors: extraction rule errors, cross reference errors, and conjunctive/disjunctive relationship errors. First, missing the extraction of an essential SIE (e.g., “Subject”, “Compliance Checking Attribute”) leads to missing the whole requirement, which results in false negative errors. For example, the subject “floor structure” was not extracted from clause C4, which resulted in not detecting the floors with noncompliant R-values. Second, missing target information due to cross reference errors may result in false positive errors. For example, the subject restriction “in Climate Zones 1, 2, and 3” in clause C5 was not extracted, which resulted in false positive errors; the roofs that are not in climate zones 1, 2, or 3 were found noncompliant. Third, conjunctive/disjunctive relationship errors may result in false positive or false negative errors with a provision (clause). For example, a conjunctive relationship was recognized for the three requirements in clause C10, which resulted in false positive errors. The steep-sloped roofs having compliant aged solar reflectance and thermal emittance but noncompliant solar reflectance index (SRI), and those having compliant SRI but noncompliant aged solar reflectance and/or thermal emittance, were both found noncompliant.

Five information alignment errors propagated into noncompliance detection errors: semantic similarity analysis errors, semantic similarity overweighting errors, regulatory concept ambiguity errors, relationship searching errors, and concept matching errors. The first two errors resulted in false negative errors, while the other three resulted in false positive errors. The false negative errors

resulted from missing the matches for the concepts “three-year-aged solar reflectance” and “radiant panel”, which resulted in not detecting the roofs with noncompliant three-year-aged solar reflectance and the radiant panels with noncompliant R-value. The false positive errors resulted from similar reasons: the subject restrictions were not considered in the matching. For example, in clause C7, the subject restriction “where located within a building envelope assembly” was not considered when recognizing the matches, which resulted in that the ducts and plenums not within the building envelope assembly were found noncompliant.

7.3.3.1.2 Which Errors do not Propagate into Noncompliance Detection Errors?

All text classification errors did not propagate into noncompliance detection errors. This is due to two reasons. First, there were simply no relevant energy requirements in the false positive clauses. For example, clause C8 (a false positive) only contains air leakage requirements – no thermal insulation requirements. Second, the false negative clauses only included irrelevant or general requirements. For example, clause C3 (a false negative) contains only general table descriptive information.

One type of information extraction error did not propagate into noncompliance detection errors: insufficient dependency information errors. The incorrectly extracted instances were false positive extractions (i.e., were not part of any requirement), which did not cause noncompliance detection errors because extracted SIEs that are missing the essential elements of a requirement (at least one subject, one compliance checking attribute, one comparative relation, and one quantity value) are

ignored. For example, “one” was incorrectly extracted from clause C9 as “Quantity Value”, but was not part of a requirement.

One type of information alignment error did not propagate into noncompliance detection errors: BIM noise errors. The false positive BIM instances could not be linked to a whole requirement, which stopped the propagation of error. For example, the BIM instance “Structure - Wood Joist/Rafter Layer, Batt Insulation [Structure]” was incorrectly matched to the concept “insulating material” in clause C11, but it was not linked to the whole requirement because no matches to “water absorption rate” were found.

7.3.3.2 How do the Errors Propagate for Contract Specification Checking?

Similar to Section 7.3.3.1, this research question was further broken down into two questions.

7.3.3.2.1 Which Errors Propagate into Noncompliance Detection Errors?

Two information extraction errors resulted in noncompliance detection errors: implicit context semantics errors and conjunctive/disjunctive relationship errors. First, failures in capturing the implicit context semantics may result in extraction of inaccurate information, which may lead to false positive errors. For example, “pulse start lamp”, rather than “pulse start metal halide lamp”, was extracted as a subject from article A3, which resulted in incorrectly finding other types of pulse start lamps (e.g., pulse start high pressure sodium lamps) noncompliant. Second, conjunctive/disjunctive relationship errors may result in false positive errors. For example, a conjunctive relationship was incorrectly recognized for the requirements in article A6, which

resulted in two false positive errors; the plug-in busway that has a compliant voltage rating "208Y/120" was found noncompliant with the alternative "480Y/277", and vice versa.

Three information alignment errors propagated into noncompliance detection errors: concept name semantics errors, concept matching errors, and semantic similarity overweighting errors. The first two types of errors resulted in false positive errors for similar reasons; the subject restrictions were not considered in the matching. For example, aboveground conduits were incorrectly matched to the concept "underground conduit" in article A9 because the subject restriction "in underground conduit" was not considered when recognizing the matches, which resulted in false positive errors; the feeders and branch circuits in aboveground conduits were found noncompliant. The third type of errors, semantic similarity overweighting, resulted in false positive errors. The instances of unfaced semi rigid mineral wool board were found noncompliant with the thermal resistivity requirements for "foil-faced semi rigid mineral wool board".

- A9: "260513 Wire And Cable
  2 – PRODUCTS
  2.1 BUILDING WIRE
  D. Feeders and Branch Circuits in Underground Conduit: Copper, stranded conductor, 600 volt insulation, THWN."

7.3.3.2.2 Which Errors do not Propagate into Noncompliance Detection Errors?

Similar to energy code checking, all text classification errors did *not* propagate into noncompliance detection errors for the same two reasons pointed out in Section 7.3.3.1.2, and an additional reason: the false negative articles (e.g., A2) did not contain BIM design requirements and were out of the

research scope for testing noncompliance detection. Also, one type of information extraction error did not propagate into noncompliance detection errors, similar to energy code checking: insufficient dependency information errors. For information alignment, two types of errors did not propagate into noncompliance detection errors: BIM noise errors and BIM descriptor errors. The reasons are similar to those discussed in Section 7.3.3.1.2.

7.3.3.3 What does the Error Propagation Analysis Reveal?

The aforementioned error propagation studies were further analyzed. The analysis revealed three types of error: the most dangerous errors, the least dangerous errors, and the most "interesting" error propagation patterns.

Information extraction errors (i.e., requirement IE) were found to be the most dangerous for noncompliance detection. Extraction errors, especially for essential SIEs (e.g., "Subject") and conjunctive/disjunctive relationships, are likely to result in noncompliance detection errors. This indicates the importance of achieving high performance in information extraction. For example, missing the subject "floor structure" from clause C4 resulted in false negative errors, while the inaccurate subject "pulse start lamp" from article A3 resulted in false positive errors.

In contrast, text classification errors were found the least dangerous for noncompliance detection. For text classification, this is likely because clauses/articles that contain specific and relevant energy requirements tend to contain more detail and discriminative features, which make them naturally easier to classify.

Three interesting error propagation patterns were observed. First, the errors in an intermediate module may not necessarily propagate into noncompliance detection errors, which indicates that perfect performance may not be required for all intermediate steps to achieve perfect noncompliance detection performance. Second, the same type of intermediate error may lead to different noncompliance detection results, which indicates that there is no isomorphism between intermediate error types and noncompliance detection error types (i.e., false negative or false positive). Third, different types of intermediate errors may lead to similar extraction/alignment errors, which indicates that intermediate errors may converge and result in the same type of extraction/alignment error. For example, all information alignment errors – including regulatory concept ambiguity errors, concept matching errors, relationship searching errors, and concept name semantics errors – resulted in failures of considering the subject restrictions in the matching. That could be away to classify intermediate errors into priority groups. For example, if missing subjects is most dangerous, dealing with intermediate errors resulting in missing subjects should be given high priority.

CHAPTER 8 – CONCLUSIONS, CONTRIBUTIONS, LIMITATIONS, AND RECOMMENDATIONS FOR FUTURE RESEARCH

8.1 Conclusions

8.1.1 Conclusions for the Proposed Methods for Text Classification of Energy Codes and Contract Specifications

8.1.1.1 Conclusions for the Proposed Machine Learning-Based Text Classification Method

A domain-specific, ML-based hierarchical text classification (TC) method and algorithm for classifying clauses in environmental regulatory documents into a number of hierarchically-detailed topics to support automated energy compliance checking in construction was developed. The algorithm classifies clauses according to leaf topics at the fifth level of a semantic TC topic hierarchy. The algorithm, thus, addresses a relatively deep TC granularity level, and therefore a more challenging TC problem. As we go to a more specialized level of topics, TC becomes typically more challenging, because the levels of knowledge and terminology of the text become more specialized and more specific which make the text harder to discriminate. A flat approach was used to deal with the hierarchical TC problem. The multilabel classification problem was transformed into a multiclass classification problem. For preparing the training and testing data, around 1,200 clauses were collected from ten environmental regulatory documents, such as the 2012 International Energy Conservation Code, and were classified into ten leaf subtopics of the energy efficiency topic (a subclass of environmental topic in the semantic topic hierarchy).

In developing the TC algorithm, the following techniques were tested and evaluated in terms of average recall and precision and their standard deviation: (1) ten popular ML algorithms; (2) two

text representation methods (BOW model and Bigram model); and (3) three term weighting schemes – two supervised term weighting schemes ($TFRF_M$ and $TF_{max}RF_M$) that were extended to adapt them to multiclass classification, and one unsupervised term weighting scheme (TFIDF) that is commonly-used. The best performance was achieved using an SVM algorithm with linear kernel, BOW model, and TFIDF weighting.

For further performance enhancement, two performance improvement strategies were implemented: (1) feature selection: a number of methods were tested and, accordingly, K-best feature selection method and CHI feature scoring function were selected; and (2) domain-specific stopword removal: construction-domain-specific stopword lists were created and used to facilitate domain-adaptation. A number of primary conclusions were drawn during the performance improvement. First, both strategies were effective in enhancing recall. Second, during this process, the standard deviation of both recall and precision continued to decrease, which indicates enhanced performance consistency and robustness. Third, each environmental topic (or subtopic) may need a different stopword list. The final classifier achieved around 97% and 84% average recall and precision, respectively, on the testing data. Two characteristics of environmental regulatory text may have contributed to such performance. Compared to general text like that in news articles, environmental regulatory text is (1) more specialized in terms of topics; specialized text is usually characterized by a smaller number of features and, thus, more discriminative features; and (2) more standardized in terms of terminology, with less homonyms and synonyms; standardized

terminology results in higher term frequencies and, thus, more discriminative features.

8.1.1.2 Conclusions for the Proposed Ontology-Based Text Classification Method

An ontology-based, multilabel text classification method and algorithm for classifying environmental regulatory clauses for supporting ACC in construction was developed. A domain ontology was developed for representing the hierarchy of environmental topics and the concepts and relationships associated with each topic. An unsupervised deep learning technique was used to learn the similarities between each clause (based on the terms in the clause) and each topic (based on the ontological concepts related to this topic) for classifying each clause into zero or more topics according to two experimentally set similarity thresholds. Four types of multilabel classification evaluation metrics were used to measure the performance of the developed algorithm. Based on the testing data, across the four types of metrics, the developed algorithm achieved overall recall and precision values from 97.32% to 98.69% and from 86.51% to 92.70%, respectively.

The experiment results indicate a number of conclusions. First, topics may semantically overlap because of their interrelationships. For example, “air leakage topic” and “thermal insulation topic” overlap because an air leakage in the building envelope is likely to affect the performance of the building’s thermal insulation. Second, the semantic overlaps among topics may be manifested through their positions in the topic hierarchy. For example, the semantically overlapping “air leakage topic” and “thermal insulation topic” are located under the same branch in the hierarchy.

Third, threshold values may help identify semantic relationships among topics. For example, close threshold values of two topics may indicate that two topics are semantically overlapping or related to each other. Relatively small threshold values of one topic compared to other topics may indicate semantic distinctiveness of that topic (i.e., less overlaps with the other topic). Relatively large thresholds of a primary topic compared to the secondary topics may indicate semantic dominance of the primary topic compared to the secondary topics. Therefore, finding the "right" threshold values is key in achieving optimal classification performance. Fourth, different thresholds should be used for different topics. Since each topic is associated with different semantics, the thresholds values should be customized for each topic to determine the optimal assignment of clauses to that topic. The thresholds should be set experimentally for maximizing performance.

Compared with existing ontology-based text classification methodologies, the proposed methodology uses an unsupervised deep learning algorithm for capturing the semantics behind the words and addresses the multilabel classification problem in a direct way without transformation to multiple single-label ones. Compared with the non-ontology-based, supervised ML-based approach (used in Section 3.1), the proposed ontology-based approach outperforms based on four evaluation metrics, reduces the efforts of data preprocessing and classifier building, and is easier to adapt for classifying other types of documents.

The proposed ontology-based TC approach could be generalized to other domains such as safety regulatory documents. The same methodology could be employed and tested, but a different

ontology – one that is relevant to the domain of application (e.g., safety ontology) – would be needed. Like any other ontology-based method, the performance of the proposed methodology could vary depending on the quality of the ontology; and like any other ML-based algorithm, the performance of the proposed method could vary depending on the size of the training and testing datasets.

8.1.2 Conclusions for the Proposed Method for Automated Information Extraction from Building Energy Codes

An ontology-based information extraction method and algorithm for automatically extracting energy requirements from energy conservation codes to support automated energy compliance checking in construction was developed. Pattern-matching extraction rules that utilize both semantic features (ontology concepts) and syntactic features (POS tags, gazetteers, and auxiliary tags) were used. To reduce text ambiguities and enhance extraction performance, a sequential dependency-based extraction method was used, including building a conceptual dependency structure based on conceptual dependency theory and defining extraction sequence based on dependency relations. To deal with the complex text in energy conservation codes (long provisions, hierarchically-complex provisions, and provisions with exceptions), domain-specific preprocessing techniques and cascaded extraction methods were used.

The proposed information extraction method was tested in extracting building thermal insulation and lighting power requirements from Chapter 4 of the 2012 IECC (ICC 2012). A performance of 97.4% recall and 98.5% precision was achieved on the testing data. The experimental results

indicate a number of conclusions. First, the proposed information extraction method was effective in automatically extracting regulatory requirements from energy conservation codes. Second, the domain-specific preprocessing techniques were successful in simplifying hierarchically-complex sentences and separating exceptions from requirements using splitting and stitching. Third, the sequential dependency-based extraction method was effective in reducing text ambiguities and improving extraction performance, although in some cases dependency in extraction can result in failure to extract the depender which leads to recall errors. Fourth, the cascaded extraction methods were successful in handling hierarchically-complex and long provisions with multiple exceptions.

8.1.3 Conclusions for the Proposed Method for Automated Information Extraction from Contract Specifications

A semantic information extraction method and algorithm for automatically extracting energy requirements from contract specifications in MasterFormat to support automated energy compliance checking in construction was developed. Pattern-matching extraction rules that utilize both semantic features (ontology concepts) and syntactic features (POS tags, gazetteer features, and domain-specific tags) and semantic features (ontology concepts) were used to develop the extraction rules. To deal with the text complexities of contract specifications (hierarchically-complex text structures, incomplete sentence structures, variety of LODs), three submethods were proposed and used: a domain-specific text splitting and stitching method, an incompleteness-aware sequential dependency extraction method, and a detail-aware LOD extraction method.

The proposed information extraction method was tested in extracting building thermal insulation

and lighting power requirements from the MasterFormat specifications of an educational building project. A performance of 96.8% recall and 97.6% precision was achieved on the testing data. The experimental results indicate a number of conclusions. First, the proposed information extraction method is promising in extracting energy requirements from MasterFormat specifications. Extraction errors may arise from dependency information, uncommon patterns, conflict resolution errors, extraction tool errors, ambiguous concept representation in the text, and/or coreference ambiguity. Second, text with hierarchically-complex text structures can be successfully split and simplified in an automated way. Third, the use of incompleteness features, along with dependency information, is effective in reducing ambiguities. Fourth, the requirements related to a target LOD can be successfully filtered and extracted. A few errors occurred in tagging the action verbs of imperative sentences, which indicate LOD 400 or above, due to word-class ambiguity and incorrect exclusion verbs. These tagging errors, however, did not propagate into information extraction errors; no information extraction errors were caused by the detail-aware LOD extraction. Fifth, a semantic rule-based approach is potentially scalable across different types of regulatory documents in the building domain, if necessary adaptations are conducted and document-type-specific text complexities are properly addressed.

8.1.4 Conclusions for the Proposed Method for Automated Semantic Information Alignment

A fully-automated semantic information alignment method and algorithm for aligning BIM information to regulatory information to support automated energy compliance checking in

construction was developed. A first-level simple alignment method was proposed to align single BIM instances to single regulatory concepts, including concept interpretation and matching for interpreting the meaning of concepts to recognize the candidate matches, and semantic similarity analysis to select the matches. To recognize the instance groups that belong to one requirement/exception, a final complex alignment method was proposed, including supervised and unsupervised searching to identify the instance pairs, and network construction to group and link the instance pairs to the requirement/exception.

The proposed information alignment method was tested in aligning a set of BIM instances (extracted from an educational building model) to a number of commercial building energy efficiency requirements/exceptions (extracted from three energy regulatory documents). An overall performance of 93.4% recall and 94.7% precision was achieved on the testing data. The experimental results indicate a number of conclusions. First, the proposed semantic information alignment method is promising. Second, errors in first-level simple alignment arise from failure to recognize the matches due to the ambiguity of regulatory concepts, noise of BIM instances, and errors in semantic similarity analysis. Third, errors in final complex alignment arise from some failures in recognizing spatially-related instance pairs or errors in the relationships found by searching.

8.1.5 Conclusions for the Case Study of Fully-Automated Energy Compliance Checking

A study that was conducted to identify the feasibility and challenges for fully-automated and

generalized compliance checking across different types of documents – particularly energy codes versus contract specifications. Specifically, the study aimed to answer three primary research questions: What are the performances of automated energy code checking and automated contract specification checking? What are the errors in both cases, and how do they compare? How do the errors propagate through the different intermediate steps, in both cases? An experiment was set up to answer these questions. A fully-automated energy compliance checking prototype, EnergyACC, was used to conduct the experiment. A BIM of an educational project was checked for compliance with thermal insulation and lighting power requirements from three energy codes and the project's contract specifications.

The experimental results indicate a number of conclusions. First, the results indicate the feasibility of developing a fully-automated and generalized ACC method. The EnergyACC prototype showed high performance in noncompliance detection, for both energy code checking (95.7% recall, 85.9% precision) and contract specification checking (100% recall, 86.5 precision). Second, the error analysis indicates a number of potential challenges to automation and generalizability: varying information distribution densities across documents, capturing the information networks in the text, varying sentence structure complexities across documents, capturing the impact of document-specific characteristics on semantic similarity analysis, and semantic ambiguities and implicit meanings in text. Third, the error propagation analysis reveals a number of findings, including the most dangerous errors (i.e., information extraction errors), the least dangerous errors (i.e., text

classification errors), and the most "interesting" error propagation patterns.

8.2 Contributions to the Body of Knowledge

8.2.1 Contributions of the Proposed Methods for Text Classification of Energy Codes and Contract Specifications

8.2.1.1 Contributions of the Proposed Machine Learning-Based Text Classification Method

This research contributes to the body of knowledge in six main ways. First, this research offers a domain-specific, ML-based hierarchical text classification (TC) algorithm for classifying clauses in environmental regulatory documents according to a semantic TC topic hierarchy. This algorithm is key in enabling automated energy compliance checking in the construction domain by enhancing the efficiency of automated information extraction. In comparison to the previous text classification efforts for ACC in construction by Salama and El-Gohary (2013b), this algorithm addresses a more challenging TC problem – hierarchical TC as opposed to non-hierarchical TC. Hierarchical TC allows for a more granular classification of text according to detailed subtopics (e.g., "thermal insulation" as opposed to "environmental") and thus would result in further enhancement of automated information extraction efficiency. In addition, future research efforts could use this work as a benchmark and could adapt the algorithm to classify other types of environmental documents (e.g., EPA regulations) and using other types of topic hierarchies (e.g., hierarchy of environmental emergencies like chemical pollution). Second, it shows that high recall and precision results can be achieved for a relatively deep TC granularity level (classifying the text according to topics in the fifth level of the hierarchy) for environmental regulatory text, and

that the flat approach in dealing with hierarchical TC is effective. As we go to a more specialized level of topics, TC becomes typically more challenging (Khan et al. 2014) and performance could highly drop [e.g., in the construction domain, classification accuracy dropped from 95.88% at the first level to 86.37% at the third level of the hierarchy (Caldas and Soibelman 2003)]. Compared with the previous work in non-hierarchical TC by Salama and El-Gohary (2013b), this research shows a relatively small drop in recall, a drop from 100% at the first level to 97% at the fifth level. Third, the research shows the effectiveness of adopting a multiclass classification approach to deal with multilabel classification problems in avoiding a data imbalance problem. The use of a multiclass classification approach is, thus, especially helpful in cases where the document frequency is imbalanced across different topics. Fourth, this research offers two extended supervised term weighting schemes, which were adapted to the multiclass classification problem. The experimental results showed that they did not perform as well, in this application, compared to the commonly-used TFIDF weighting scheme. But, having those extended weighting schemes would allow other researchers to further evaluate them in classifying other types of documents (e.g., OSHA standards) and in other domains (e.g., safety). Fifth, this research shows the effectiveness of feature selection in enhancing recall, even if the original feature size is relatively small (around 4,200 features compared with the millions of features that are commonly seen in CS domain datasets). It shows that a small selected feature size (less than 2,900) is enough to achieve high performance. Sixth, the research shows that recursive, domain-specific stopword removal is very effective in improving recall. It shows that the use of a general stopword list is not sufficient

as the domain of text becomes highly specialized, and that the development and use of domain-specific stopword lists is highly effective in achieving increased performance at a low manual effort, especially that such lists are reusable.

8.2.1.2 Contributions of the Proposed Ontology-Based Text Classification Method

This research contributes to the body of knowledge on two main levels. First, a new ontology-based TC methodology for classifying environmental regulatory documents in the construction domain is proposed. This research offers a leading initiative; it is the first ontology-based TC effort in the construction domain. In comparison to the commonly-used non-ontology-based, supervised ML-based approach (e.g., the research in Section 3.1), the proposed ontology-based approach (1) utilizes the knowledge of the domain (in the form of an ontology) to capture the semantics of the text for enhanced classification: In non-ontology-based ML-based TC, some semantics are captured partially and indirectly using statistical and probabilistic methods. For example, the conditional probability of adjacent words in a sentence may indicate some useful word relationships for classification (e.g., using a Bigram model). However, such limited semantic information is not sufficient in adequately capturing the semantics of the text. In comparison, an ontology-based approach allows for capturing deeper semantic information by representing each category in terms of concepts and relationships; (2) outperforms in terms of recall and precision: The experimental results showed that the proposed algorithm achieved higher recall and precision in comparison to that in Section 3.1; (3) eliminates the need for labeling training data, since unsupervised deep learning is implemented for exploring similarities: This makes the proposed

approach more practical to use in real-life applications where most of the data are unlabeled; (4) reduces the efforts of data preprocessing: The proposed algorithm only requires data in .txt format. In contrast, non-ontology-based ML-based TC requires preprocessing of data into numeric vectors using text representation methods like the bag of words model; and (5) is easier to adapt for classifying other types of documents: Using the proposed methodology, the major effort in TC lies in developing an ontology (if one is not readily available) and labeling the testing dataset. In contrast, non-ontology-based ML-based TC requires experimental testing of different text representation methods, term weighting schemes, ML algorithms, etc.

Second, an improved method for ontology-based TC is proposed. In comparison to existing ontology-based TC methods, the proposed method: (1) uses unsupervised instead of supervised ML, which saves the manual effort needed in labeling the training data; (2) deals with the multilabel classification problem in a direct way instead of conducting problem transformation, which reduces the data preparation and classifier building effort; and (3) uses deep instead of shallow learning, which aims to better represent the complexity that exists in text semantics.

8.2.2 Contributions of the Proposed Method for Automated Information Extraction from Building Energy Codes

This research contributes to the body of knowledge in four main ways, in comparison to existing information extraction efforts in the construction domain. First, the proposed method integrates text classification with information extraction. Integrating text classification with information extraction allows for extracting information from pre-classified text, which avoids both errors and

computational effort resulting from processing irrelevant text. Second, domain-specific preprocessing techniques are proposed to handle hierarchically-complex sentence structures and exceptions using splitting and semantic-based stitching. This allows for, both, simplifying hierarchically-complex sentence structures while taking meaning and obligation type into account, and separating the processing of exceptions from requirements. Third, this research uses conceptual dependency theory to build a conceptual dependency structure for the target information and offers a sequential dependency-based extraction method. The developed conceptual dependency structure allows for capturing the dependency relations among the semantic information elements in a way that helps define the best sequence of extraction. The proposed dependency-based extraction method allows for taking such dependency relations into consideration during extraction, which leads to reduced text ambiguities and enhanced performance. The experimental results show that the use of dependency relations was effective in reducing semantic ambiguities. Fourth, this research proposes cascaded extraction methods to deal with text complexities in terms of long provisions, hierarchically-complex sentences, and exceptions. Cascaded extraction methods allow for handling a complex extraction task by breaking it down to a number of simple extraction tasks (i.e., a complex extraction task is cascaded on a number of simple extraction tasks). The experimental results show that the proposed cascaded information extraction methods are effective in handling these three types of text complexities.

8.2.3 Contributions of the Proposed Method for Automated Information Extraction from Contract Specifications

This research contributes to the body of knowledge in four main ways. First, a semantic IE method is proposed to automatically extract building energy requirements from contract specifications. This research offers a leading initiative; it is the first effort to use automated information extraction to extract compliance requirements from contract specifications. Second, a domain-specific text splitting and stitching method is proposed to deal with hierarchically-complex text structures. The proposed method uses a regular expressions-based pattern matching technique to automatically recognize the splitting signals, and split the text based on the text feature patterns defined in the PageFormat. This allows for simplifying the hierarchically-complex text structures and reducing the complexity of the text for further information extraction. Third, an incompleteness-aware sequential dependency extraction method is proposed to deal with incomplete sentence structures. The proposed method exploits incompleteness features, in addition to dependency information among SIEs, to reduce the text ambiguities. Fourth, a detail-aware LOD extraction method is proposed to deal with the variety of LODs. The LODs of the information are distinguished based on the grammatical moods of sentences. A domain-specific POS tagger is offered to support the recognition of domain-specific text features for use in analyzing the grammatical moods.

8.2.4 Contributions of the Proposed Method for Automated Semantic Information Alignment

This research contributes to the body of knowledge in three main ways. First, in comparison to existing information alignment efforts, this research offers a fully-automated approach for

alignment. Second, a novel method for first-level simple alignment is proposed to align single BIM instances to single regulatory concepts. The proposed method uses concept interpretation and matching to automatically recognize candidate matches to regulatory concepts. In the concept interpretation and matching, domain knowledge is captured in the form of ontology and bSDD and is used to support the interpretation of meaning of the regulatory concepts. The proposed method further uses semantic similarity analysis to select the matches to the regulatory concepts. In the semantic similarity analysis, a deep learning technique is used to explore the semantics behind words for enhanced assessment of semantic similarity, and an empirical way is used to analyze the patterns of semantic similarities for enhanced recognition of matches. Third, a novel method for final complex alignment method is proposed to recognize the associated regulatory concepts that belong to one requirement/exception, and group and link the matches (i.e., instance groups) to these associated regulatory concepts (i.e., concept group). The proposed method uses supervised and unsupervised searching to automatically search for the relationships that create the instance pairs, and uses network modeling to model a concept group and its associated instance groups as a network of linked concept pairs and instance pairs.

8.2.5 Contributions of the Case Study of Fully-Automated Energy Compliance Checking

This study contributes to the body of knowledge in four main ways. First, this study is the first to achieve a full level of automation in automated code checking. While the state of the art (Zhang and El-Gohary 2017) achieved a remarkable level of automation, some human effort was still

required for semi-automated BIM-code matching. Second, to the best of the author's knowledge, this study is the first that focuses on fully-automated checking of BIMs for compliance with contract specifications. Automated contract specification checking would reduce the cost and time of compliance verification, help ensure compliance with specification provisions, and promote the use of BIM for improved project delivery. Contract specifications have different challenging text complexities from energy codes, which naturally make them good subjects for studying generalizability of fully-automated ACC. Third, this study provides important insights on the generalizability of fully-automated energy compliance checking methods – across energy codes and contract specifications. Besides the three primary research questions, the results of the study additionally help answer important questions in this regard: Can full automation be achieved? Are similar levels of performance expected? Can methods be used as is, or is some level of adaptation necessary? Fourth, this study sheds important light on the sources of errors in automated compliance checking and how these errors propagate – or not propagate – from an intermediate step to the other. Such insights are very important – they are pointers to limitations, future research directions, and paths for improvement.

8.3 Limitations and Recommendations for Future Research

8.3.1 Limitations of the Proposed Methods for Text Classification of Energy Codes and Contract Specifications and Recommendations for Future Research

8.3.1.1 Limitations of the Proposed Machine Learning-Based Text Classification Method and Recommendations for Future Research

Two limitations of the research are acknowledged. First, not all ML algorithms in the general

computing domain have been tested. For example, some less commonly used ML algorithms including the labeled Latent Dirichlet Allocation (LDA) algorithm (Ramage et al. 2009), which showed outperforming performance compared to the SVM in the computer science domain, were not tested in this research. As such, in future research, more ML algorithms could be adapted and tested in the construction domain. Second, the stopwords found in this research may not be exhaustive. Future research may explore and add more construction-domain-specific stopwords to further improve the classification performance.

In addition, future research could go in three directions. First, explore the use (or adaptation) of the developed TC algorithm for classifying other types of regulatory documents (e.g., safety regulatory documents such as OSHA standards) on the basis of other types of topics (e.g., safety topics). Second, explore the reusability of the developed stopword lists in classifying other types of environmental documents and automate the process of constructing domain-specific stopword lists. Third, explore the use of other approaches for domain adaptation, in addition to domain-specific stopword lists, such as feature augmentation.

8.3.1.2 Limitations of the Proposed Ontology-Based Text Classification Method and Recommendations for Future Research

Two limitations of the research are acknowledged. First, the performance of the proposed approach – and of ontology-based approaches in general – depends on the quality of ontologies used. In future research, the proposed approach could be tested in classifying environmental regulatory documents using other ontologies. Second, the proposed methodology was tested on only six topics.

In future research, the methodology could be tested in classifying environmental regulatory clauses based on more environmental topics.

In addition, in future research, the proposed ontology-based methodology could be further refined. For example, a more complex threshold function could be used for assigning labels, considering that the TSD threshold value may relate to the length of each clause. Since a longer clause usually contains more concepts, its TS to a topic may be much larger than the TS of a shorter clause to the same topic. Therefore, the TSD of other topics for a longer clause may be much larger than that of a shorter clause.

Other researchers may also adapt the proposed methodology for classifying documents in other domains. Such adaptation research could focus on three main areas. First, testing the proposed methodology in classifying other types of documents based on other types of topics. Second, determining the threshold values of the different topics. Third, investigating how the characteristics (e.g., depth and breadth) of an ontology could affect the level of performance of the proposed methodology.

8.3.2 Limitations of the Proposed Method for Automated Information Extraction from Building Energy Codes and Recommendations for Future Research

Two limitations of the proposed information extraction method are acknowledged. First, although the proposed method has successfully addressed some semantic ambiguities (e.g., see the discussion in the Section 4.3.2), it cannot – at least at this point – address all semantic interpretation issues [e.g., those discussed in Solihin and Eastman (2015a, b) such as dependencies and hidden

assumptions] or deal with requirements that require human judgment by nature. Further research is needed to study the challenging types of semantic interpretation and ambiguity issues such as hidden assumptions, explore the limits of machine intelligence, and identify which types of requirements can be extracted in a fully-automated way and which would require some level of human involvement or verification. Even for the latter types of requirements, the proposed method could be very useful in acting as a first-level interpretation of the requirements in an automated, repeatable, and consistent manner – allowing a human user or expert to further verify the automatically extracted information, clarify any semantic ambiguities, capture any hidden assumptions or implicit domain knowledge, and ensure the alignment with the concept representations of the design information. Second, the use of dependency information may sometimes overconstrain the matching conditions, thereby resulting in failures to extract depender SIEs. It is, thus, essential to achieve high performance in extracting dependees, especially at the top of the conceptual dependency structure. In order to study possible ways for further performance improvement, further research could be conducted to explore different methods for avoiding such over-constraining cases (e.g., using different matching conditions).

In addition, three limitations that may manifest themselves in future applications, if the proposed algorithm is used for a different knowledge domain (e.g., construction safety) or to extract information from a different type of document (e.g., contract specifications), are acknowledged. First, like any other ontology-based method, (1) the performance of the proposed information

extraction method highly depends on the coverage of the ontology used, and (2) additional human effort may be required to build a new ontology or extend this ontology for applying this algorithm to a different knowledge domain (other than the domain of "commercial building energy efficiency", which is the scope of this ontology). However, ontologies are now more widely used in construction domain applications (Zhou et al. 2016), and are by nature easily reusable and extendable (El-Gohary and El-Diraby 2010). In future research, the ontology could be extended to cover other knowledge domains (e.g., fire safety) and the adapted algorithm could be tested in checking the compliance with related codes and regulations (e.g., the International Fire Code). The proposed methodology could also be tested in extracting information from the IECC using another ontology (but which also covers the domain of building energy efficiency) to test the impact of different ontologies (which could naturally vary in coverage, structure, semantics, etc.) on the performance of extraction. Second, because dependency relations may vary from one type of text to another, the developed conceptual dependency structure may need adaption for extracting requirements from different types of documents (e.g., OSHA standards). In future research, further studies could be conducted to see if the developed conceptual dependency structure will require adaptation for extracting requirements from other documents (e.g., OSHA standards). Third, like any other rule-based method, the developed extraction rules may require further adaptation when used for extracting different types of requirements or for extracting similar requirements but from a different type of text. However, these rules are potentially reusable in extracting building energy requirements from other types of energy regulatory documents/text. The rules could be reused as

is or adapted – through modification or extension – based on additional development text. Compared with the initial efforts, future efforts in adapting the extraction rules should be significantly lower. Once the rules are adapted, the process of information extraction is fully automated and requires no user manual effort.

Three limitations that are related to the scope of the work and the testing are also highlighted. First, the scope of the work is limited to natural text and excludes requirements in formulas and cross references. These could be addressed in future research, through separate but supporting linked algorithms. Second, the proposed method was only tested in extracting two types of requirements – thermal insulation and lighting power – because of the scope of the work. In future research, the proposed method could be tested in extracting other types of requirements (e.g., fenestration). Some extension effort might be needed in this case. For example, the commercial building energy ontology may need to be extended if the new energy subtopics are not already covered in the scope of the ontology. Third, due to the high amount of manual effort needed for developing a gold standard for testing and evaluation, the proposed method was tested only on one chapter. Thus, future research is needed to test the algorithm on more energy regulatory documents. The results are expected to show similar high performance because of the similarity in text across different energy codes. However, further testing is needed for verification.

8.3.3 Limitations of the Proposed Method for Automated Information Extraction from Contract Specifications and Recommendations for Future Research

Four limitations, related to the scope of the research and the testing, are acknowledged. First, the

proposed IE method is limited to the extraction of requirements from MasterFormat contract specifications. In future research, the proposed method could be tested on other specification standards. The UniFormat could specially be a good starting point, because both, the UniFormat and the MasterFormat, are developed by the same organization and thus are likely to share common characteristics.

Second, the proposed detail-aware LOD extraction method is limited to the extraction of information in LOD 350, because the research scope is currently limited to checking the compliance of BIMs in LOD 350. In future research, to extend the proposed information extraction method to multiple LODs, machine learning techniques could be used to develop a text classifier to automatically classify the sentences according to the required LOD, in which the grammatical moods, types of action verbs, and paragraph headings could be selected as features for the machine learning.

Third, the proposed method was only tested in extracting two types of requirements – thermal insulation and lighting power – because of the scope of the work. In future research, the proposed method could be tested in extracting other types of requirements (e.g., fenestration). Some extension effort would be needed in this case. For example, in addition to extending the ontology to cover the additional topics, the lexicon and the ruleset of the domain-specific tagger may need to be extended to cover new domain-specific words and conversion rules. The extraction rules and tagging rules (as mentioned in Section 5.2.2.5) may also need some modification/extension. A

similar level of performance is expected after such necessary extension is conducted.

Fourth, due to the manual effort involved in its development, the gold standard only included 148 requirements, from the contract specifications of one project. In future research, further testing of the proposed method could be conducted using a larger set of requirements, from the contract specifications of multiple projects. A comparable level of performance is expected.

In addition, future research could go in three directions. First, one direction is additional testing of the proposed semantic IE method – testing using more contract specifications. The testing could also cover specifications with different characteristics (e.g., levels of details in content, formatting quality) for evaluating the impacts of those characteristics on the extraction performance. Second, the proposed information extraction method could be extended to expand the scope of the extraction. The extension could go in several directions: covering more energy topics (e.g., fenestration) to expand the scope of energy compliance checking; covering additional types of topics (e.g., safety) from other types of documents (e.g., OSHA standards) to go into other ACC areas; and/or covering more LODs (e.g., LOD 400 or above, which refer to installation and verification requirements) to expand the scope to field verification. The extension efforts could also cover specifications in other formatting standards (e.g., UniFormat). Third, future research could further explore the use of ML approaches for the extraction of requirements from codes and specifications. This could provide more insights, as well as empirical evidence, on the comparison of both approaches – ML and rule-based – in such deep information extraction task. As we

compare both approaches, it would be important to pay attention to a number of different – and potentially conflicting – aspects: performance, development effort, scalability, computational efficiency, etc. A hybrid approach could also be explored. It may prove to be a good way of leveraging the best of both worlds.

8.3.4 Limitations of the Proposed Method for Automated Semantic Information Alignment and Recommendations for Future Research

Four limitations of the proposed semantic information alignment method are acknowledged. First, the proposed concept interpretation and matching method may not be able to correctly interpret the meaning of all single regulatory concepts automatically. Typically, some regulatory concepts are ambiguously expressed in the regulations, which requires human experts to clarify such ambiguities and specify the mapping of the BIM instances to those ambiguous concepts (Dimyadi et al. 2016b; Solihin and Eastman 2015a). Future research may further study such cases and explore the limit of artificial intelligence techniques in dealing with such ambiguities. Second, the proposed semantic similarity scoring method excluded the rare terms (i.e., terms with a frequency less than five) in assessing the term-to-term semantic similarity, because the deep learning technique was not able to learn accurate vector representations for rare terms. The experimental results showed that some rare terms may still carry important information in the energy domain. To assess the semantic similarity between rare terms, knowledge-based semantic similarity measures (which rely on ontology or WordNet, in contrast to the corpus-based measures used in this research) may be explored in future research. Third, the term position weighting function

weights terms by only considering their positions in concepts, which may not be accurate. In future research, more factors (e.g., part-of-speech tags and term frequencies) may be considered, and unsupervised machine learning techniques may be explored to automatically learn the term position weighting function. Fourth, the unsupervised searching method made an oversimplified assumption that the shortest among all found paths determines if two BIM instances are linked as an instance pair. The experimental results showed that the relationships occurred on the shortest path may not correctly match the relationships between concepts. Therefore, further investigation may focus on how to select the one, among all found paths, that matches the relationship in a concept pair to create instance pairs.

Two limitations related to the implementation and testing are acknowledged. First, the implementation of the proposed information alignment method may be computationally expensive, especially in the post-processing of extracted information, and in the case of recognizing a large number of instance groups that are linked to a highly-complex concept group (e.g., the "subject restriction", a complex SIE, may contain a large number of associated concepts, which may result in a complex concept group). In future research, techniques like parallel computing could be explored to improve the computational efficiency. Second, the proposed method was only tested on a limited number of energy regulatory requirements, because significant manual effort is needed for developing a gold standard. In future research, the proposed method could be tested on more requirements in other energy topics (e.g., fenestration topic) from energy regulatory documents,

and more energy requirements from other types of documents (e.g., EPA regulations). A similar level of performance is expected, after some necessary adaptation. For example, the commercial building energy ontology may need extension to cover new concepts in other energy topics. The values of the proposed threshold types may also need adjustment for different types of documents. Such adaptation efforts should be significantly lower compared to the initial efforts.

In addition, in future research, the proposed method could also be used to support the development of computer-interpretable regulations. Developing such computable regulations requires the development and use of standardized regulatory concepts and relationships – for example in the form of a regulatory ontology. The proposed method could be used – along with other evaluation methods – to evaluate candidate regulatory concepts in the ontology in terms of their degree of match to the IFC concepts.

8.3.5 Limitations of the Case Study of Fully-Automated Energy Compliance Checking and Recommendations for Future Research

Five limitations of this study are acknowledged. First, the results and findings of this study (i.e., the answers to the research questions) are limited to the experimental setup of the study (i.e., using the EnergyACC and the two test cases). More testing and experimental studies are needed in future research to further understand if/how these findings change as the methods, implementations, or test cases change. Although the author believes that the key findings are likely to remain unchanged because the generalizability challenges are mostly originating from the nature and characteristics of the text (e.g., implicitness and ambiguities), more verification is needed. Second,

this study was limited to requirements in text and table formats – requirements expressed in equations, drawings, and images were excluded. Cross references were also excluded from the scope of the research. To achieve full automation for the entire domain of application, future research is needed to further cover these scope exclusions. Third, this study focused on quantitative design requirements – non-quantitative requirements, as well as requirements related to installation and verification (i.e., corresponding to LODs 400-500) were excluded from the scope. Fourth, this study focused on contract specifications in the MasterFormat, which is the most prevalent specifications-writing standard in North America – other types of specifications like UniFormat were excluded. Fifth, the scope of testing was limited to testing the compliance of a building information model – which included 115 noncompliant instances – with 79 thermal insulation and lighting power requirements from three energy codes and the project's contract specifications.

A number of future research directions can be pursued to extend or improve this research. First, the proposed methods could be tested in other AEC subdomains and applications, such as checking the compliance of safety requirements with OSHA standards. The proposed methods are expected to scale well, although some adaptation/extension effort would be needed – for example to develop and/or use a safety ontology. Similarly, the adaptation, implementation, and testing could be extended to cover requirements in higher LODs – such as installation and verification requirements in contract specifications. Second, more research is needed to deal with cross references – not only cross-referenced text, but also equations, drawings, and images. Third, more investigation is

needed to better understand how the characteristics of the BIM – in terms of completeness of information, quality of modeling, size, etc. – would impact the performance of information alignment and noncompliance detection. Fourth, more research studies could be conducted to better understand the computability of requirements in codes and specifications. Some requirements are by nature relatively easy to interpret automatically, while others are hard to automatically interpret without human involvement because of their semantic ambiguities and implicit meanings. The latter type is bound to result in errors. More studies are needed to better differentiate, characterize, and understand the different types and levels of computability and their impact on fully-automated noncompliance detection.

REFERENCES

Abuzir, Y., and Abuzir, M. O. (2002). "Constructing the civil engineering thesaurus (CET) using ThesWB." *Proc., 2002 Int. Workshop on Information Technology in Civil Engineering*, ASCE, Reston, VA, 400–412.

Afany Software. (2013). "B-prolog." <http://www.picat-lang.org/bprolog/download.html> (Apr. 23, 2018).

Aggarwal, C., and Zhai, C. (2012). "A survey of text classification algorithms. " *Mining text data*, Springer, New York, 163–222.

Al Qady, M. A., and Kandil, A. (2010). "Concept relation extraction from construction documents using natural language processing." *J. Constr. Eng. Manage.*, 10.1061/(ASCE)CO.1943-7862.0000131, 294–302.

Aly, M. (2005). "Survey on multiclass classification methods." Caltech, Pasadena, CA. <http://www.vision.caltech.edu/malaa/publications/aly05multiclass.pdf> (Feb. 25, 2015).

Apache Jena. (2016). "Jena ontology API." <https://jena.apache.org/documentation/ontology/> (Nov. 7, 2016).

ARCAT Inc. (2018). "ARCAT." <https://www.arcat.com/> (Apr. 24, 2018).

Arendarenko, E., and Kakkonen, T. (2012). "Ontology-based information and event extraction for business intelligence." *Proc., 15th Int. Conf. on Artificial Intelligence: Methodology, Systems, and Applications*, Springer, Berlin, 89–102.

ASHRAE (American Society of Heating, Refrigerating, and Air-Conditioning Engineers).

(2010). “ANSI/ASHRAE/IES standard 90.1-2010 energy standard for buildings except low-rise residential buildings.” <ftp://law.resource.org/pub/us/code/ibr/ashrae.90.1.ip.2010.pdf> (Jul. 30, 2015).

Autodesk. (2016a). “Revit 2016.” <https://www.autodesk.com/education/free-software/revit> (Apr. 24, 2018).

Autodesk. (2016b). “Revit API developers guide.” <https://knowledge.autodesk.com/search-result/caas/CloudHelp/cloudhelp/2016/ENU/Revit-API/files/GUID-F0A122E0-E556-4D0D-9D0F-7E72A9315A42-htm.html> (Aug. 3, 2016).

Avolve Software Corporation (2011). “ProjectDox electronic plan review.” <http://www.avolvesoftware.com/products/electronic-plan-review/> (Oct. 8, 2015).

Baumgärtel, K., Kadolsky, M., and Scherer, R. J. (2015). “An ontology framework for improving building energy performance by utilizing energy saving regulations.” *Proc., 10th European Conf. on Product & Process Modelling (ECPPM 2014)*, Taylor & Francis Group, London, 519–526.

Beach, T. H., Kasim, T., Li, H., Nisbet, N., and Rezgui, Y. (2013). “Towards automated compliance checking in the construction industry.” *Lecture Notes in Computer Science (LNCS)*, Springer, Berlin, 366–380.

Beach, T. H., Rezgui, Y., Li, H., and Kasim, T. (2015). “A rule-based semantic approach for automated regulatory compliance in the construction sector.” *Expert Syst. Appl.*, 42(12), 5219–5231.

Belsky, M., Sacks, R., and Brilakis, I. (2016). “Semantic enrichment for building information modeling.” *Comput-Aided Civil Infrastruct. Eng.*, 31(4), 261–274.

Bengio, Y. (2009). “Learning deep architectures for AI.” *Found. Trends® Mach. Learn.*, 2(1), 1–127.

Bengio, Y., Courville, A., and Vincent, P. (2013). “Representation learning: A review and new perspectives.” *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8), 1798–1828.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). “A neural probabilistic language model.” *J. Mach. Learn. Res.*, 3, 1137–1155.

Bengio, Y., and LeCun, Y. (2007). “Scaling learning algorithms towards AI.” *Large-Scale Kernel Mach.*, 34(5), 1–41.

BIMForum. (2017). “Level of development specification 2017.” <https://bimforum.org/wp-content/uploads/2017/11/LOD-Spec-2017-Part-II_2017-11-07.xlsx> (Aug. 19, 2017).

BIMobject Corporation. (2018). “BIMobject.” <https://bimobject.com/en-us> (Apr. 24, 2018).

Blitzer, J., Dredze, M., and Pereira, F. (2007). “Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification.” *Proc., 42nd Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, Stroudsburg, PA, 440–447.

Boulis, C., and Ostendorf, M. (2005). “Text classification by augmenting the bag-of-words representation with redundancy-compensated bigrams.” *Proc., Int. Workshop on Feature Selection in Data Mining*, International Computer Science Institute, Berkeley, CA, 9–16.

Breiman, L. (2001). “Random forests.” *Mach. Learn.*, 45(1), 5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and regression trees*, Chapman & Hall/CRC, Boca Raton, FL.

Brinker, K., and Hüllermeier, E. (2007). “Case-based multilabel ranking.” *Proc., 20th Int. Joint Conf. on Artificial Intelligence (IJCAI’07)*, Morgan Kaufmann, San Francisco, CA, 702–707.

buildingSMART. (2016a). “buildingSMART - international home of openBIM.” <http://www.buildingsmart-tech.org/> (Jul. 30, 2016).

buildingSMART. (2016b). “buildingSMART Data Dictionary.” <http://bsdd.buildingsmart.org/#peregrine/about> (Nov. 8, 2016).

buildingSMART. (2017). “bSDD content guidelines.” <https://docs.google.com/document/d/1YUiR07A27lK0UB8ImYoaoLKCUvh1QFG1FfcvvLOYdP0/edit> (May 19, 2017).

Caldas, C. H., and Soibelman, L. (2003). “Automating hierarchical document classification for construction management information systems.” *Automat. Constr.*, 12(4), 395–406.

Caldas, C. H., Soibelman, L., and Han, J. (2002). “Automated classification of construction project documents.” *J. Comput. Civ. Eng.*, 10.1061/(ASCE)0887-3801(2002)16:4(234), 234–243.

California Energy Commission. (2013). “2013 building energy efficiency standards for residential and nonresidential buildings.”

<http://www.energy.ca.gov/2012publications/CEC-400-2012-004/CEC-400-2012-004-CMF-REV2.pdf> (May 17, 2017).

California Energy Commission. (2016). "2016 building energy efficiency standards approved computer compliance programs." <http://www.energy.ca.gov/title24/2016standards/2016_computer_prog_list.html> (Jun. 19, 2018).

Cemesova, A., Hopfe, C. J., and McLeod, R. S. (2015). "PassivBIM: Enhancing interoperability between BIM and low energy design software." *Automat. Constr.*, 57, 17–32.

Chang, C., and Lin, C. (2011). "LIBSVM: A library for support vector machines." *ACM Trans. Intell. Syst. Technol.*, 2(3), 1–27.

Charte, F., Rivera, A., Jesus, M. J., and Herrera, F. (2013). "A first approach to deal with imbalance in multi-label datasets." *Proc., Int. Conf. on Hybrid Artificial Intelligence Systems*, Springer, Berlin, 150–160.

Chawla, N. V., Japkowicz, N., and Kotcz, A. (2004). "Editorial: Special issue on learning from imbalanced data sets." *J. SIGKDD Explor. Newsl.*, 6(1), 1–6.

Cheng, J. C. P., and Das, M. (2014). "A BIM-based web service framework for green building energy simulation and code checking." *J. Inf. Technol. Constr.*, 19, 150–168.

Chi, N. W., Lin, K. Y., and Hsieh, S. H. (2014). "Using ontology-based text classification to assist job hazard analysis." *Adv. Eng. Inform.*, 28(4), 381–394.

Choi, J., Choi, J., and Kim, I. (2014). "Development of BIM-based evacuation regulation

checking system for high-rise and complex buildings." *Automat. Constr.*, 46, 38–49.

Civil Law Dictionary. (2015). "Obligations." <http://civillawdictionary.pbworks.com/w/page/15934864/O%20Civil%20Law> (Dec. 15, 2015).

Corke, G. (2013). "Solibri model checker v8". *AECMagazine, Building Information Modelling (BIM) Technology for Architecture, Engineering and Construction.* <http://aecmag.com/software-mainmenu-32/527-solibri-model-checker-v8> (Oct. 3, 2015).

CSI (Construction Specifications Institute). (2004). *The project resource manual: CSI manual of practice (5th Edition)*. McGraw-Hill, New York, NY.

CSI (Construction Specifications Institute), and CSC (Construction Specifications Canada). (2009). "SectionFormat/PageFormat." <http://icee.usm.edu/dkemp/download/ACT380/Presentations/Supporting Docs-Samples/PDPG Docs/SectionFormat_PageFormat_2009.pdf> (Jun. 19, 2017).

CSI (Construction Specifications Institute), and CSC (Construction Specifications Canada). (2014). "MasterFormat – Master list of numbers and titles for the construction industry." <http://lor.rrc.mb.ca/file/9c8519dc-c193-4edc-a4b7-9084929d902a/1/MasterFormat_2014.pdf> (Jun. 19, 2017).

CSI (Construction Specifications Institute), and CSC (Construction Specifications Canada). (2017). "SectionFormat/PageFormat."

<https://www.csiresources.org/practice/standards/sectionformat-pageformat> (Jun. 19, 2017).

CSI (Construction Specifications Institute), and CSC (Construction Specifications Canada). (2018). “MasterFormat.” <https://www.csiresources.org/practice/standards/masterformat> (Oct. 24, 2018).

Cunningham, H., Maynard, D., and Bontcheva, K. (2011). *Text processing with GATE (Version 6)*, University of Sheffield Department of Computer Science, Sheffield, U.K.

Deeplearning4j Development Team. (2016). “Deeplearning4j: Open-source distributed deep learning for the JVM.” <http://deeplearning4j.org> (Nov. 8, 2016).

Delis, E. A., and Delis, A. (1995). “Automatic fire-code checking using expert-system technology.” *J. Comput. Civ. Eng.*, 10.1061/(ASCE)0887-3801(1995)9:2(141), 141–156.

Deng, Y., Cheng, J. C. P., and Anumba, C. (2016). “Mapping between BIM and 3D GIS in different levels of detail using schema mediation and instance comparison.” *Automat. Constr.*, 67, 1–21.

Dhillon, R. K., Jethwa, M., and Rai, H. S. (2014). “Extracting building data from BIM with IFC.” *Int. J. Recent Trends Eng. Technol.*, 11(1), 202–210.

Dijkstra, E. W. (1959). “A note on two problems in connexion with graphs.” *Numer. Math.*, 1(1), 269–271.

Dimyadi, J., Clifton, C., Spearpoint, M., and Amor, R. (2014). “Regulatory knowledge encoding guidelines for automated compliance audit of building engineering design.” *Proc.*, 2014

Int. Conf. on Computing in Civil and Building Engineering (ICCCBE), ASCE, Reston, VA, 536–543.

Dimyadi, J., Clifton, C., Spearpoint, M., and Amor, R. (2016a). "Computerizing regulatory knowledge for building engineering design." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000572, C4016001.

Dimyadi, J., Pauwels, P., and Amor, R. (2016b). "Modelling and accessing regulatory knowledge for computer-assisted compliance audit." *J. Inf. Technol. Constr.*, 21(Special Issue CIB W78 2015 Special Track on Compliance Checking), 317–336.

Dimyadi, J., Solihin, W., and Marchant, D. (2016c). "The design brief: Requirements and compliance." *J. Inf. Technol. Constr.*, 21(Special Issue CIB W78 2015 Special Track on Compliance Checking), 337–353.

Ding, L., Drogemuller, R., Rosenman, M., Marchant, D., and Gero, J. (2006). "Automating code checking for building designs-DesignCheck." *Proc., CRC for Construction*, University of Wollongong, Wollongong, Australia, 1–16.

Ding, L. Y., Zhong, B. T., Wu, S., and Luo, H. B. (2016). "Construction risk knowledge management in BIM using ontology and semantic web technology." *Saf. Sci.*, 87, 202–213.

Eastman, C., Jeong, Y., Sacks, R., and Kaner, I. (2010). "Exchange model and exchange object concepts for implementation of national BIM standards." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)0887-3801(2010)24:1(25), 25–34.

Eastman, C., Lee, J., Jeong, Y., and Lee, J. (2009). “Automatic rule-based checking of building designs.” *Automat. Constr.*, 18(8), 1011–1033.

Eastman, C., Teicholz, P., Sacks, R., and Liston, K. (2011). *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors*. John Wiley and Sons, Hoboken, NJ.

Eclipse Foundation. (2017). “Eclipse oxygen.” <http://www.eclipse.org/oxygen/> (Apr. 23, 2018).

El-Gohary, N., and El-Diraby, T. (2010). “Domain ontology for processes in infrastructure and construction.” *J. Constr. Eng. Manage.*, 10.1061/(ASCE)CO.1943-7862.0000178, 730–744.

Fagni, T., and Sebastiani, F. (2010). “Selecting negative examples for hierarchical text classification: An experimental comparison.” *J. Am. Soc. Inform. Sci. Technol.*, 61(11), 2256–2265.

Fang, J., Guo, L., Wang, X. D., and Yang, N. (2007). “Ontology-based automatic classification and ranking for web documents.” *Proc., 4th Int. Conf. on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*, IEEE, Washington, DC, 627–631.

Fiatech. (2014). “AutoCodes project.”, <http://www.fiatech.org/project-management/projects/593-automated-code-plan-checking-tool-proof-of-concept> (Jan. 21, 2015).

Friedman, J. H. (2001). “Greedy function approximation: A gradient boosting machine.” *Ann.*

Stat., 29(5), 1189–1232.

Ganu, G., Marian, A., and Elhadad, N. (2010). *URSA-user review structure analysis: Understanding online reviewing trends*. University of Rutgers, Piscataway, NJ.

Garrett, J. H., Palmer, M. E., and Demir, S. (2014). "Delivering the infrastructure for digital building regulations." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000369, 167–169.

Gerrish, T., Ruikar, K., Cook, M., Johnson, M., Phillip, M., and Lowry, C. (2017). "BIM application to building energy performance visualisation and management: Challenges and potential." *Energy Build.*, 144(1), 218–228.

Ghamrawi, N., and McCallum, A. (2005). "Collective multilabel classification." *Proc., 14th ACM Int. Conf. on Information and Knowledge Management*, Association for Computing Machinery (ACM), New York, 195–200.

Giorgetti, D., and Sebastiani, F. (2003). "Multiclass text categorization for automated survey coding." *Proc., 2003 ACM Symposium on Applied Computing*, Association for Computing Machinery (ACM), New York, 798–802.

Goel, S. K., and Fenves, S. J. (1969). "Computer-aided processing of structural design specifications." *University of Illinois at Urbana-Champaign*, Champaign, IL.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*, MIT Press. <http://www.deeplearningbook.org/> (Feb. 6, 2016).

Goutte, C., and Gaussier, E. (2005). "A probabilistic interpretation of precision, recall and f-

score, with implication for evaluation." *Proc., European Conf. on Information Retrieval (ECIR'05),* Springer, Berlin, 345–359.

Grishman, R. (2012). "Information extraction: Capabilities and challenges." *Notes Prepared for the 2012 International Winter School in Language and Speech Technologies*. <http://www.cs.nyu.edu/grishman/tarragona.pdf> (Jul. 12, 2016).

Harispe, S., Ranwez, S., Stefan, J., and Montmain, J. (2013). "Semantic measures for the comparison of units of language, concepts or instances from text and knowledge representation analysis." <http://arxiv.org/pdf/1310.1285.pdf> (Jan. 21, 2015).

He, Q., Qui, L., Zhao, G., and Wang, S. (2004). "Text categorization based on domain ontology." *Proc., 5th Int. Conf. on Web Information Systems Engineering (WISE 2004),* Springer, Berlin, 319–324.

Hijazi, I. H., Ehlers, M., and Zlatanova, S. (2012). "NIBU: a new approach to representing and analysing interior utility networks within 3D geo-information systems." *Int. J. Digit. Earth.,* 5(1), 22–42.

Hjelseth, E., and Nisbet, N. (2011). "Capturing normative constraints by use of the semantic mark-up RASE methodology." *Proc., 28th Int. Conf. CIB W78,* International Council for Research and Innovation in Building and Construction, Delft, Netherlands, 1–10.

Houston City Council. (2011). "2011 Houston commercial energy conservation code." < http://edocs.publicworks.houstontx.gov/documents/divisions/planning/enforcement/2009_iecc.pdf > (Mar. 24, 2015).

ICC (International Code Council). (2009a). “2009 international building code.” <http://publicecodes.cyberregs.com/icod/ibc/2009/index.htm> (Oct. 26, 2015).

ICC (International Code Council). (2009b). “2009 Virginia energy conservation code.” <http://www.ecodes.biz/ecodes_support/free_resources/virginia2009/09energy/09energy_main.html> (Mar. 23, 2015).

ICC (International Code Council). (2012). “2012 international energy conservation code.” <http://publicecodes.cyberregs.com/icod/iecc/2012/> (Mar. 21, 2015).

ICC (International Code Council). (2018). “International energy convervation code adoption.” <https://cdn-web.iccsafe.org/wp-content/uploads/Code_Adoption_Maps.pdf> (Jun. 1, 2018).

İlal, S. M., and Günaydın, H. M. (2017). “Computer representation of building codes for automated compliance checking.” *Automat. Constr.*, 82, 43–58.

Ireson, N., Ciravegna, F., Califf, M. E., Freitag, D., Kushmerick, N., and Lavelli, A. (2005). “Evaluating machine learning for information extraction.” *Proc., 22nd Int. Conf. on Machine Learning*, Association for Computing Machinery (ACM), New York, NY, 345–352.

ISO (International Organization for Standardization). (1994). *ISO 10303-11:1994, Industrial automation systems and integration – product data representation and exchange – part 11: description methods: the EXPRESS language reference manual*, Geneva, Switzerland.

ISO (International Organization for Standardization). (2013). *ISO 16739:2013, Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries*, Geneva, Switzerland.

Jiang, L., and Leicht, R. (2015). "Automated rule-based constructability checking: Case study of formwork." *J. Manage. Eng.*, 10.1061/(ASCE)ME.1943-5479.0000304, A4014004.

Jurafsky, D., and Martin, J. H. (2009). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*, Prentice Hall, Upper Saddle River, NJ.

Kalin, M., Weygant, R. S., Rosen, H. J., and Regener, J. R. (2010). *Construction specifications writing : principles and procedures*, John Wiley & Sons, Hoboken, NJ.

Karkaletsis, V., Fragkou, P., Petasis, G., and Iosif, E. (2011). "Ontology based information extraction from text." *Knowledge-Driven Multimedia Information Extraction and Ontology Evolution*, Springer, Berlin, 89–109.

Kasim, T., Li, H., Rezgui, Y., and Beach, T. (2013). "Automated sustainability compliance checking process: Proof of concept." *Proc., 13th Int. Conf. on Construction Applications of Virtual Reality*, Teesside University, Middlesbrough, UK, 11–21.

Khan, S., Baig, A. R., and Shahzad, W. (2014). "A novel ant colony optimization based single path hierarchical classification algorithm for predicting gene ontology." *Appl. Soft Comput.*, 16, 34–49.

Khemlani, L. (2005). "CORENET e-PlanCheck: Singapore's automated code checking system."

AECBytes. <http://www.aecbytes.com/buildingthefuture/CORENETePlanCheck.htm> (May 10, 2018).

Kilicoglu, H., Rosemblat, G., Fiszman, M., and Rindflesch, T. C. (2011). "Constructing a semantic predication gold standard from the biomedical literature." *BMC Bioinf.*, 12(1), 1–17.

Kim, H., and Anderson, K. (2013). "Energy modeling system using building information modeling open standards." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000215, 203–211.

Kim, H., Anderson, K., Lee, S., and Hildreth, J. (2013a). "Generating construction schedules through automatic data extraction using open BIM (building information modeling) technology." *Automat. Constr.*, 35, 285–295.

Kim, K., Kim, G., Yoo, D., and Yu, J. (2013b). "Semantic material name matching system for building energy analysis." *Automat. Constr.*, 30, 242–255.

Kim, H., Shen, Z., Kim, I., Kim, K., Stumpf, A., and Yu, J. (2016a). "BIM IFC information mapping to building energy analysis (BEA) model with manually extended material information." *Automat. Constr.*, 68, 183–193.

Kim, M. K., Wang, Q., Park, J. W., Cheng, J. C. P., Sohn, H., and Chang, C. C. (2016b). "Automated dimensional quality assurance of full-scale precast concrete elements using laser scanning and BIM." *Automat. Constr.*, 72, 102–114.

Kovacevic, M., Nie, J. Y., and Davidson, C. (2008). "Providing answers to questions from

automatically collected web pages for intelligent decision making in the construction sector.” *J. Comput. Civ. Eng.*, 10.1061/(ASCE)0887-3801(2008)22:1(3), 3–13.

Lee, Y. C., Eastman, C. M., and Lee, J. K. (2015). “Automated rule-based checking for the validation of accessibility and visibility of a building information model.” *Proc., 2015 Int. Workshop on Computing in Civil Engineering (IWCCE)*, ASCE, Reston, VA, 572–579.

Lee, S. K., Kim, K. R., and Yu, J. H. (2014). “BIM and ontology-based approach for building cost estimation.” *Automat. Constr.*, 41, 96–105.

Lee, H., Lee, J. K., Park, S., and Kim, I. (2016). “Translating building legislation into a computer-executable format for evaluating building permit requirements.” *Automat. Constr.*, 71(Part 1), 49–61.

Lee, Y. H., Tsao, W. J., and Chu, T. H. (2009). “Use of ontology to support concept-based text categorization.” *Proc., 7th Int. Workshop on e-Business (WEB 2008)*, Springer, Berlin, 201–213.

Li, T. S., Good, B. M., and Su, A. I. (2015). “Exposing ambiguities in a relation-extraction gold standard with crowdsourcing.” *Proc., 23rd Int. Conf. on Intelligent Systems for Molecular Biology* (ISMB), International Society for Computational Biology (ISCB), Bethesda, MY, 1–4.

Li, Z., and Ramani, K. (2007). “Ontology-based design information extraction and retrieval.” *J. Artif. Intell. Eng. Des. Anal. Manuf.*, 21(2), 137–154.

Liao, C., Alpha, S., and Dixon, P. (2003). “Feature preparation in text categorization.” *Proc., Autralasian Data Mining: Workshop,* Springer, Berlin.

Liebich, T. (2013). “IFC4 – the new buildingSMART standard.” <http://www.buildingsmart-tech.org/specifications/ifc-releases/ifc4-release/buildingSMART_IFC4_Whatisnew.pdf> (May 17, 2017).

Liebich, T., Adachi, Y., Forester, J., Hyvarinen, J., Richter, S., Chipman, T., Weise, M., and Wix, J. (2016). “Industry foundation classes version 4 – Addendum 2.” <http://www.buildingsmart-tech.org/ifc/IFC4/Add2/html/> (May 20, 2017).

Liebich, T., Wix, J., Forester, J., and Qi, Z. (2002). “Speeding-up the building plan approval–the Singapore e-plan checking project offers automatic plan checking based on IFC.” *Proc., 4th European Conf. on Product and Process Modelling (ECPPM),* Swets & Zeitlinger, Lisse, 467–471.

Lilis, G. N., Giannakis, G. I., and Rovas, D. V. (2016). “Automatic generation of second-level space boundary topology from IFC geometry inputs.” *Automat. Constr.*, 76, 108–124.

Lin, Y. H., Liu, Y. S., Gao, G., Han, X. G., Lai, C. Y., and Gu, M. (2013). “The IFC-based path planning for 3D indoor spaces.” *Adv. Eng. Informatics*, 27(2), 189–205.

Liu, K., and El-Gohary, N. (2017). “Ontology-based semi-supervised conditional random fields for automated information extraction from bridge inspection reports.” *Automat. Constr.*, 81, 313–327.

Liu, H., Lu, M., and Al-Hussein, M. (2016). “Ontology-based semantic approach for

construction-oriented quantity take-off from BIM models in the light-frame building industry." *Adv. Eng. Informatics*, 30(2), 190–207.

Livingston, O. V, Cole, P. C., Elliott, D. B., and Bartlett, R. (2014). "Building energy codes program: National benefits assessment 1992–2040." <https://www.energycodes.gov/sites/default/files/documents/BenefitsReport_Final_March20142.pdf> (Jun. 29, 2018).

LKSoftWare GmbH. (2016). "Java standard data access interface (JSDAI)." <http://www.jsdai.net/> (Nov. 11, 2016).

Luo, H., and Gong, P. (2015). "A BIM-based code compliance checking process of deep foundation construction plans." *J. Intell. & Robotic Syst.*, 79(3-4), 549–576.

Madjarov, G., Kocev, D., Gjorgjevikj, D., and Džeroski, S. (2012). "An extensive experimental comparison of methods for multi-label learning." *Pattern Recognit.*, 45(9), 3084–3104.

Mahfouz, T. (2011). "Unstructured construction document classification model through support vector machine (SVM)." *Proc., 2011 ASCE Int. Workshop on Computing in Civil Engineering (IWCCE)*, ASCE, Reston, VA, 126–133.

Makrehchi, M., and Kamel, M. S. (2008). "Automatic extraction of domain-specific stopwords from labelled documents." *Proc., 30th European Conf. on Advances in Information Retrieval*, Springer, Berlin, 222–233.

Malkani, Z., and Gillie, Z. (2012). "Supervised multiclass classication of tweets." Stanford University, Stanford, CA. <http://cs229.stanford.edu/proj2012/GillieMalkani-

SupervisedMulticlassClassificationOfTweets.pdf> (Mar. 4, 2015).

Malsane, S., Matthews, J., Lockley, S., Love, P. E. D., and Greenwood, D. (2015). "Development of an object model for automated compliance checking." *Automat. Constr.*, 49, Part A, 51–58.

Man, L., Tan, C. L., Jian, S., and Yue, L. (2009). "Supervised and traditional term weighting methods for automatic text categorization." *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(4), 721–735.

Manning, C. D., Raghawan, P., and Schütze, H. (2009). *An introduction to information retrieval*, Cambridge University Press, Cambridge, U.K.

Manning, C. D., and Schütze, H. (1999). *Foundations of statistical natural language processing*, MIT Press, Cambridge, MA.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). "The stanford CoreNLP natural language processing toolkit." *Proc., 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Association for Computational Linguistics, Stroudsburg, PA, 55–60.

Mark, J. C., Jack, C. T., Bahman, L. Y., Carroll, L. P., Wei, Y., James, H., and Joe, E. (2017). "SMARTreview." <https://smartreview.biz/home> (Apr. 20, 2018).

Martins, J. P., and Monteiro, A. (2013). "LicA: A BIM based automated code-checking application for water distribution systems." *Automat. Constr.*, 29, 12–23.

Maynard, D., Peters, W., and Li, Y. (2006). "Metrics for evaluation of ontology-based

information extraction." *Proc., WWW 2006 Workshop on Evaluation of Ontologies for the Web (EON),* Association for Computing Machinery (ACM), New York, NY.

Melzner, J., Zhang, S., Teizer, J., and Bargstädt, H. J. (2013). "A case study on automated safety compliance checking to assist fall protection design and planning in building information models." *Constr. Manage. Econ.,* 31(6), 661–674.

Microsoft. (2016). "Visual studio IDE." <https://www.visualstudio.com/> (May 10, 2017).

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). "Efficient estimation of word representations in vector space." *Proc., Workshop at Int. Conf. on Learning Reprentations (ICLR),* NEC Laboratories America, Princeton, NJ, 1–12.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., and Khudanpur, S. (2010). "Recurrent neural network based language model." *Proc., 11th Annual Conf. of the Int. Speech Communication Assocication (INTERSPEECH 2010),* International Speech Communication Association (ISCA), Baixas, France, 1045–1048.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). "Distributed representations of words and phrases and their compositionality." *Proc., Neural Information Processing Systems 2013 (NIPS 2013),* Curran Associates, Red Hook, NY, 3111–3119.

Mikolov, T., Yih, W., and Zweig, G. (2013c). "Linguistic regularities in continuous space word representations." *Proc., 2013 Conf. of the North American Chapter of the Association for Computatational Linguistics — Human Langugage Technologies (NAACL-HLT-2013),*

Association for Computational Linguistics, Stroudsburg, PA, 746–751.

Moens, M. F. (2006). *Information extraction: Algorithms and prospects in a retrieval context*, Springer, New York, Secaucus, NJ.

Moreno, A., Isern, D., and López Fuentes, A. C. (2013). "Ontology-based information extraction of regulatory networks from scientific articles with case studies for Escherichia coli." *Expert Syst. Appl.*, 40(8), 3266–3281.

Moschitti, A., and Basili, R. (2004), "Complex linguistic features for text classification: A comprehensive study." *Proc., 26th Annual European Conf. on Information Retrieval (ECIR 2004)*, Springer, Berlin, 181–196.

Musen, M. A. (2015). "The Protégé project: A look back and a look forward." *AI Matters*, 1(4), 4–12.

Nawari, N. (2012). "Automating codes conformance." *J. Archit. Eng.*, 10.1061/(ASCE)AE.1943-5568.0000049, 315–323.

NBS. (2018). "NBS national BIM library." <https://www.nationalbimlibrary.com/> (Apr. 24, 2018).

Nepal, M. P., Staub-French, S., Pottinger, R., and Zhang, J. (2013). "Ontology-based feature modeling for construction information extraction from a building information model." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000230, 555–569.

Nevada Governor's Office of Energy. (2018). "International energy conservation code significant change comparison study."

<http://energy.nv.gov/uploadedFiles/energynvgov/content/Programs/IECC%20Code%20Comparison%20Study.pdf> (Jun. 1, 2018).

Newcombe, R. G. (1998). "Two-sided confidence intervals for the single proportion: comparison of seven methods." *Stat. Med.*, 17(8), 857–872.

Nguyen, T. H., and Kim, J. L. (2011). "Building code compliance checking using BIM technology." *Proc., 2011 Winter Simulation Conf (WSC)*, IEEE, Washington, DC, 3395–3400.

Ontario Ministry of Municipal Affairs. (2011). "Ontario building code supplementary standard SB-10." <http://www.mah.gov.on.ca/AssetFactory.aspx?did=9227> (May 17, 2017).

Pauwels, P., Van Deursen, D., Verstraeten, R., De Roo, J., De Meyer, R., Van De Walle, R., and Van Campenhout, J. (2011). "A semantic rule checking environment for building performance checking." *Automat. Constr.*, 20(5), 506–518.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, É. (2011). "Scikit-learn: Machine learning in python." *J. Mach. Learn. Res.*, 12(10), 2825–2830.

Pestian, J. P., Deleger, L., Savova, G. K., Dexheimer, J. W., and Solti, I. (2012). "Natural language processing – the basics." *Pediatric Biomedical Informatics*, Springer, Dordrecht, 149–172.

Pierced Media LC. (2018). "RevitCity." <https://www.revitcity.com/index.php> (Apr. 24, 2018).

Piskorski, J., and Yangarber, R. (2013). “Information extraction: past, present and future.” *Multi-source, Multilingual Information Extraction and Summarization,* Springer, Berlin, 23–49.

Porter, M. F. (2006). “The English (Porter2) stemming algorithm.” <http://snowball.tartarus.org/algorithms/english/stemmer.html> (Dec. 17, 2014).

Preidel, C., and Borrmann, A. (2016). “Towards code compliance checking on the basis of a visual programming language.” *J. Inf. Technol. Constr.*, 21(Special Issue CIB W78 2015 Special Track on Compliance Checking), 402–421.

Qi, J., Issa, R., Hinze, J., and Olbina, S. (2011). “Integration of safety in design through the use of building information modeling.” *Proc., 2011 ASCE Int. Workshop on Computing in Civil Engineering (IWCCE),* ASCE, Reston, VA, 698–705.

Qi, J., Issa, R., Olbina, S., and Hinze, J. (2014). “Use of building information modeling in design to prevent construction worker falls.” *J. Comput. Civ. Eng.*, 10.1061/%28ASCE%29CP.1943-5487.0000365, 1–10.

Ramage, D., Hall, D., Nallapati, R., and Manning, C. D. (2009). “Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora.” *Proc., 2009 Conf. on Empirical Methods in Natural Language Processing,* Association for Computational Linguistics, Stroudsburg, PA, 248–256.

Rehurek, R., and Sojka, P. (2010). “Software framework for topic modelling with large corpora.” *Proc., LREC 2010 Workshop on New Challenges for NLP Frameworks*, European Language Resources Association (ELRA), Paris, France, 45–50.

Richard, B. (2018). “Snowball.” <http://snowballstem.org/> (May 9, 2018).

Van Rijsbergen, C. J. (1979). *Information retrieval*, Butterworth-Heinemann, Newton, MA.

Robbins, A., Lamb, L., and Hannah, E. (2008). *Learning the Vi and Vim Editors,* 7th Ed., O'Reilly, Sebastopol, CA.

Salama, D., and El-Gohary, N. (2013a). “Automated compliance checking of construction operation plans using a deontology for the construction domain.” *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000298, 681–698.

Salama, D., and El-Gohary, N. (2013b). “Semantic text classification for supporting automated compliance checking in construction.” *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000301, 04014106.

Sebastiani, F. (2002). “Machine learning in automated text categorization.” *J. ACM Comput. Surv.,* 34(1), 1–47.

See, R. (2008). “SMARTcodes: Enabling BIM based automated code compliance checking.” <http://projects.buildingsmartalliance.org/files/?artifact_id=1535> (May 17, 2017).

Silla, C., and Freitas, A. (2011). “A survey of hierarchical classification across different application domains.” *Data Min. Knowl. Discovery*, 22(1-2), 31–72.

Sinha, S., Sawhney, A., Borrmann, A., and Ritter, F. (2013). “Extracting information from building information models for energy code compliance of building envelope.” *Proc., RICS COBRA Conf. 2013*, CIB, Rotterdam, Netherlands, 1–16.

SmartBIM Technologies. (2018). “SmartBIM library.” <http://library.smartbim.com/> (Apr. 24,

2018).

SMC. (2009). “Development of Solibri Model Checker (SMC) accessibility rules set for checking building information models.” <http://www.universell-utforming.miljo.no/file_upload/statsbygg_smc_accessibility.pdf> (May 17, 2017).

Solihin, W., and Eastman, C. (2015a). “Classification of rules for automated BIM rule checking development.” *Automat. Constr.*, 53, 69–82.

Solihin, W., and Eastman, C. (2015b). “A knowledge representation approach to capturing BIM based rule checking requirements using conceptual graph.” *Proc., 32nd CIB W78 Conf. 2015*, Eindhoven University of Technology, Eindhoven, Netherlands, 686–695.

Solihin, W., and Eastman, C. (2016). “A knowledge representation approach in BIM rule requirement analysis using the conceptual graph.” *J. Inf. Technol. Constr.*, 21(Special Issue CIB W78 2015 Special Track on Compliance Checking), 370–401.

Song, M. H., Lim, S. Y., Kang, D. J., and Lee, S. J. (2005). “Automatic classification of web pages based on the concept of domain ontology.” *Proc., 12th Asia-Pacific Software Engineering Conf.*, IEEE, Washington, DC, 15–17.

Sorower, M. S. (2010). *A literature survey on algorithms for multi-label learning*, Oregon State University, Corvallis, OR.

Southern Nevada Building Officials. (2013). “Southern nevada amendments to the 2012 international energy conservation code.” <http://www.clarkcountynv.gov/Depts/development_services/plan_review/Building%20

Codes/2012_IECC%20Amendments.pdf> (Oct. 13, 2015).

Soysal, E., Cicekli, I., and Baykal, N. (2010). "Design and evaluation of an ontology based information extraction system for radiological reports." *Comput. Biol. Med.*, 40(11-12), 900–911.

Spyromitros, E., Tsoumakas, G., and Vlahavas, I. (2008). "An empirical study of lazy multilabel classification algorithms." *Proc., 5th Hellenic Conference on Artificial Intelligence*, Springer, Berlin, 401–406.

Sun, A., and Lim, E. (2001). "Hierarchical text classification and evaluation." *Proc., 2001 IEEE Int. Conf. on Data Mining*, IEEE Computer Society, Washington, DC, 521–528.

Sun, A., Lim, E., and Ng, W. (2003). "Performance measurement framework for hierarchical text classification." *J. Am. Soc. Inform. Sci. Technol.*, 54(11), 1014–1028.

Sun, Y. M., Kamel, M. S., Wong, A. K. C., and Wang, Y. (2007). "Cost-sensitive boosting for classification of imbalanced data." *Pattern Recogn.*, 40(12), 3358–3378.

Tan, X., Hammad, A., and Fazio, P. (2010). "Automated code compliance checking for building envelope design." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)0887-3801(2010)24:2(203), 203–211.

Tao, J., Deokar, A. V., and El-Gayar, O. F. (2014). "An ontology-based information extraction (OBIE) framework for analyzing initial public offering (IPO) prospectus." *Proc., 2014 47th Hawaii Int. Conf. on System Sciences*, IEEE, Washington, DC, 769–778.

Tateisi, Y., Shidahara, Y., Miyao, Y., and Aizawa, A. (2014). "Annotation of computer science

papers for semantic relation extraction." *Proc., 9th Int. Conf. on Language Resources and Evaluation (LREC'14),* European Language Resources Association (ELRA), Paris, France, 1423–1429.

Teo, T. A., and Cho, K. H. (2016). "BIM-oriented indoor network model for indoor and outdoor combined route planning." *Adv. Eng. Informatics,* 30(3), 268–282.

Teo, T. A., and Yu, S. C. (2017). "The extraction of indoor building information from BIM to OGC IndoorGML." *Proc., Int. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* Copernicus Publications, Göttingen, Germany, 167–170.

Tsoumakas, G., and Katakis, I. (2007). "Multi-label classification: An overview." *Int. J. Data Warehouse. Min.,* 3(3), 1–13.

Tsoumakas, G., Katakis, I., and Vlahavas, I. (2010). "Mining multi-label data." *Data mining knowledge discovery handbook,* Springer, Berlin, 667–685.

United States Depart of Energy (U.S. DOE). (2018a). "Status of state energy code adoption." <https://www.energycodes.gov/status-state-energy-code-adoption> (Jun. 1, 2018).

United States Depart of Energy (U.S. DOE). (2018b). "Software and web tools." <https://www.energycodes.gov/compliance/tools> (May 29, 2018).

United States Energy Information Administration (U.S. EIA). (2015). "Monthly energy review." <http://www.eia.gov/totalenergy/data/monthly/pdf/mer.pdf> (May 11, 2015).

Vogrinčič, S., and Bosnić, Z. (2011). "Ontology-based multi-label classification of economic articles." *Comput. Sci. Inf. Syst.,* 8(1), 101–119.

Waraporn, P., Meesad, P., and Clayton, G. (2010). “Ontology-supported processing of clinical text using medical knowledge integration for multi-label classification of diagnosis coding.” *Int. J. Comput. Sci. Inf. Security,* 7(3), 30–35.

Wei, G. Y., Yu, J., Ling, Y., and Liu, J. (2006). “Design and implementation of an ontology algorithm for web documents classification.” *Proc., 2006 Int. Conf. on Computational Science and Its Applications (2006),* Springer, Berlin, 649–658.

Wijewickrema, C. M., and Gamage, R. (2013). “An ontology based fully automatic document classification system using an existing semi-automatic system.” *Proc., IFLA 2013 World Library and Information Congress,* International Federation of Library Associations and Institutions (IFLA), Netherlands, 1–13.

Wilson, E. B. (1927). “Probable inference, the law of succession, and statistical inference.” *J. Am. Stat. Assoc.,* 22(158), 209–212.

Wimalasuriya, D. C., and Dou, D. J. (2010). “Ontology-based information extraction: an introduction and a survey of current approaches.” *J. Inf. Sci.*, 36(3), 306–323.

Won, J., Lee, G., and Cho, C. (2013). “No-schema algorithm for extracting a partial model from an IFC instance model.” *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000320, 585–592.

Wu, K., Lu, B. L., Uchiyama, M., and Isahara, H. (2007). “A probabilistic approach to feature selection for multiclass text categorization.” *Proc., 4th Int. Symposium on Neural Networks (ISNN 2007),* Springer, Berlin, 1310–1317.

Xuan, N., and Quang, H. (2014). “A new improved term weighting scheme for text categorization.” *Knowledge and systems engineering*, Springer, Cham, Switzerland, 261–270.

Yang, X. Q., Sun, N., Zhang, Y., and Kong, D. R. (2008). “General framework for text classification based on domain ontology.” *Proc., 3rd Int. Workshop on Semantic Media Adaptation and Personalization (SMAP)*, IEEE, Washinton, DC, 147–152.

Yoon, Y., Lee, C., and Lee, G. G. (2006). “An effective procedure for constructing a hierarchical text classification system.” *J. Am. Soc. Inform. Sci. Technol.*, 57(3), 431–442.

Yu, F., Zheng, D. Q., Zhao, T. J., Li, S., and Yu, H. (2006). “Text classification based on a combination of ontology with statistical method.” *Proc., 5th Int. Conf. on Machine Learning and Cybernetics*, IEEE, Washinton, DC, 13–16.

Yurchyshyna, A., Faron-Zucker, C., Thanh, N. L., and Zarli, A. (2008). “Towards an ontology-enabled approach for modeling the process of conformity checking in construction.” *Proc., CAiSE'08 Forum*, RWTH Aachen University, Aachen, Germany, 21–24.

Yurchyshyna, A., and Zarli, A. (2009). “An ontology-based approach for formalisation and semantic organisation of conformance requirements in construction.” *Automat. Constr.*, 18, 1084–1098.

Zhang, J., and El-Gohary, N. (2013). “Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking.” *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000346, 04015014.

Zhang, J., and El-Gohary, N. (2014a). “Extending building information models semi-automatically using semantic natural language processing techniques.” *Proc., 2014 Int. Conf. on Computing in Civil and Building Engineering (ICCCBE)*, ASCE, Reston, VA, 2246–2253.

Zhang, J., and El-Gohary, N. (2014b). “Automated reasoning for regulatory compliance checking in the construction domain.” *Proc., 2014 Construction Research Congress (CRC)*, ASCE, Reston, VA, 907–916.

Zhang, J., and El-Gohary, N. (2015a). “Automated extraction of information from building information models into a semantic logic-based representation.” *Proc., 2015 Int. Workshop on Computing in Civil Engineering (IWCCE)*, ASCE, Reston, VA, 173–180.

Zhang, J., and El-Gohary, N. (2015b). “Automated information transformation for automated regulatory compliance checking in construction.” *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000427, B4015001.

Zhang, J., and El-Gohary, N. (2016). “Semantic-based logic representation and reasoning for automated regulatory compliance checking.” *J. Comput. Civ. Eng.*, 10.1016/j.autcon.2016.08.027, 04016037.

Zhang, J., and El-Gohary, N. (2017). “Integrating semantic NLP and logic reasoning into a unified system for fully-automated code checking.” *Automat. Constr.*, 73, 45–57.

Zhang, L., and Issa, R. (2013). “Ontology-based partial building information model extraction.” *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000277, 576–584.

Zhang, S., Teizer, J., Lee, J. K., Eastman, C. M., and Venugopal, M. (2013). "Building information modeling (BIM) and safety: Automatic safety checking of construction models and schedules." *Automat. Constr.*, 29, 183–195.

Zhang, L., Wu, X., Ding, L., Skibniewski, M. J., and Lu, Y. (2016). "BIM-based risk identification system in tunnel construction." *J. Civ. Eng. Manage.*, 22(4), 529–539.

Zhang, M. L., and Zhou, Z. H. (2007). "ML-KNN: A lazy learning approach to multi-label learning." *Pattern Recognit.*, 40(7), 2038–2048.

Zhong, B. T., Ding, L. Y., Love, P. E. D., and Luo, H. B. (2015). "An ontological approach for technical plan definition and verification in construction." *Automat. Constr.*, 55, 47–57.

Zhong, B. T., Ding, L. Y., Luo, H. B., Zhou, Y., Hu, Y. Z., and Hu, H. M. (2012). "Ontology-based semantic modeling of regulation constraint for automated construction quality compliance checking." *Automat. Constr.*, 28, 58–70.

Zhou, N. F. (2014). "B-Prolog user's manual." <http://www.picat-lang.org/bprolog/download/manual.pdf> (May 10, 2018).

Zhou, P., and El-Gohary, N. (2014). "Semantic-based text classification of environmental regulatory documents for supporting automated environmental compliance checking in construction." *Proc., 2014 Construction Research Congress (CRC),* ASCE, Reston, VA, 897–906.

Zhou, P., and El-Gohary, N. (2014). "Ontology-based, multi-label text classification for enhanced information retrieval for supporting automated environmental compliance

checking." *Proc., 2014 Int. Conf. on Computing in Civil and Building Engineering (ICCCBE)*, ASCE, Reston, VA, 2238–2245.

Zhou, P., and El-Gohary, N. (2015). "Domain-specific hierarchical text classification for supporting automated environmental compliance checking." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000513, 04015057.

Zhou, P., and El-Gohary, N. (2015). "Ontology-based information extraction from environmental regulations for supporting environmental compliance checking." *Proc., 2015 Int. Workshop on Computing in Civil Engineering (IWCCE)*, ASCE, Reston, VA, 190–198.

Zhou, P., and El-Gohary, N. (2016a). "Automated extraction of environmental requirements from contract specifications." *Proc., 16th Int. Conf. on Computing in Civil and Building Engineering (ICCCBE)*, International Society for Computing in Civil and Building Engineering (ISCCBE), Osaka, Japan, 1669–1676.

Zhou, P., and El-Gohary, N. (2016b). "Ontology-based multilabel text classification of construction regulatory documents." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000530, 04015058.

Zhou, P., and El-Gohary, N. (2017). "Ontology-based automated information extraction from building energy conservation codes." *Automat. Constr.*, 74, 103–117.

Zhou, P., and El-Gohary, N. (2018). "Automated matching of design information in BIM to regulatory information in energy codes." *Proc., 2018 Construction Research Congress (CRC)*, ASCE, Reston, VA, 75–85.

Zhou, P., and El-Gohary, N. (2018). “Text and information analytics for fully automated energy code checking.” *Proc., 2nd GeoMEast Int. Congress and Exhibition on Sustainable Civil Infrastructures,* Springer International Publishing, Cham, Switzerland, 196–208.

Zhou, Z., Goh, Y., and Shen, L. (2016). “Overview and analysis of ontology studies supporting development of the construction industry.” *J. Comput. Civ. Eng.,* 10.1061/(ASCE)CP.1943-5487.0000594, 4016026.

Zhou, H., Lee, S., and Ying, H. (2018). “VPL-based code translation for automated compliance checking of building envelope energy efficiency.” *Proc., 2018 Construction Research Congress (CRC),* ASCE, Reston, VA, 1–12.