Hierarchical representation and deep learning-based method for automatically transforming textual building codes into semantic computable requirements

Ruichuan Zhang[a]; and Nora El-Gohary[b]

[a] Graduate Student, Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, 205 N. Mathews Ave., Urbana, IL 61801, United States. E-mail: rzhang65@illinois.edu.

[b] Associate Professor, Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, 205 N. Mathews Ave., Urbana, IL 61801, United States (corresponding author). E-mail: gohary@illinois.edu; Tel: +1-217-333-6620.

Abstract

Most of the existing automated code compliance checking (ACC) systems are unable to fully automatically convert building-code requirements, especially requirements that have hierarchically complex semantic and syntactic structures, into computer-processable forms. The state-of-the-art rule-based ACC methods that are able to deal with complex requirements are based on information extraction and transformation rules, which are inflexible when applied to different types of regulatory documents. More research is, thus, needed to develop a flexible method to automatically process and understand requirements to support the downstream tasks in ACC systems such as information matching and compliance reasoning. To address this need, this paper proposes (1) a new representation of requirements, the requirement hierarchy, and (2) a deep learning-based method to automatically extract semantic relations between words from building-code sentences, which are used to transform the sentences into such hierarchies. The proposed method was evaluated using a corpus of sentences from multiple regulatory documents. It achieved high semantic relation and requirement hierarchy extraction performance.

Keywords: Requirement representation; Semantic relation extraction; Code checking; Deep learning; Semi-supervised learning.

Introduction

Building designs are governed by a variety of regulatory documents (e.g., building codes, standards, and specifications) in the architecture, engineering, and construction (AEC) domain. To reduce the time, cost, and error of the process of checking the compliance of building designs with these regulatory documents, various automated compliance checking (ACC) systems have been developed. Existing ACC systems require that the regulatory information (e.g., subjects, compliance checking attributes, and quantity values and units) is first extracted from

natural language regulatory-document sentences and converted into computer-processable forms. Although these ACC systems have achieved different levels of automation, representativeness, and accuracy, most of them are unable to extract and convert the regulatory information in a fully automated way (Nawari 2019; Bloch and Sacks 2020; Sacks et al. 2020). This can be attributed to two factors/challenges: (1) the complexity of the task at hand, in terms of text characteristics and level of analytics needed, and (2) the limitations of the approaches used.

Automatically converting natural-language requirements into computable rules is not an easy task for two reasons. First, different from other types of text (e.g., social media posts), sentences in the regulatory documents usually have hierarchically complex semantic and syntactic structures (e.g., nested clauses, conjunctive and alternative obligations, multiple restrictions or exceptions, etc., see Fig. 1) (Zhou and El-Gohary 2017), and thus are more challenging for computers to automatically process, represent, and understand. Second, different from other applications [e.g., sentiment analysis or shallow information extraction (e.g., Marzouk and Enaba 2019, Akanbi and Zhang 2021)], ACC requires full understanding of the text to correctly capture the meaning of the requirements including the alternatives, restrictions, and exceptions, etc. (Beach et al. 2020; Solihin et al. 2020; Xue and Zhang 2020), which adds to the challenge.

The other factor is the limitations of the approaches used. First, most of the existing ACC methods that can represent, process, or check regulatory requirements are not fully automated, requiring intensive human effort. For example, manual encoding-based ACC methods require professionals to read the regulatory text and then encode these sentences into computer-processable forms, such as the Building Environment Rule and Analysis language (Lee et al. 2015), regulatory knowledge query language (Dimyadi et al. 2016), and visual language for compliance checking (Preidel and Borrmann 2016). Annotation-based ACC methods require professionals to read the regulatory text and then annotate the sentences with a set of predefined semantic labels or markups that indicate the semantic roles or the semantic relations between these roles in the context of compliance checking. For example, Hjelseth and Hisbet (2011) defined the requirement, application, selection, and exception (RASE) markups to manually annotate the sentences from the International Building Code (IBC) via the Smartcode user interface for supporting compliance checking. Second, efforts that have achieved full automation, which are very limited in number, have scalability limitations. For example, rule-based methods have achieved the state-of-the-art performance in extracting regulatory information and transforming the extracted information into computer-processable forms, while only requiring

limited human effort (e.g., Zhang and El-Gohary 2013; Li et al. 2016; Zhou and El-Gohary 2017; Kim et al. 2019). However, rule-based methods usually lack flexibility and scalability, because the rules for extraction are developed manually by experts and may need to be updated or adapted when applied to a different type of regulatory document or when the document goes through major updates. In contrast, machine learning methods, instead of relying on hand-crafted rules, develop computational models that automatically capture the underlying semantic and syntactic patterns of the training text and have a good capability of generalizing to a variety of text. Particularly, deep learning approaches have shown increased capabilities in text analytics (LeCun et al. 2015). However, such deep learning methods have not been leveraged in analyzing the semantic and syntactic structures of AEC domain-specific regulatory text in the context of ACC.

To address the aforementioned challenges, this paper proposes a deep learning-based approach to automatically convert building-code sentences, especially the structurally complex ones, into smaller, easier-to-digest, and less-complex interconnected units of requirements for facilitating the succeeding steps of ACC such as compliance reasoning. The proposed approach is threefold. First, this paper proposes a new representation of requirements for supporting ACC-related building-code analytics – the requirement hierarchy – to decompose the sentences into the interconnected requirement units. Each requirement hierarchy consists of one or more requirement units and the dependencies between the units, with each unit representing a simple requirement or condition that can be processed by most of the existing ACC systems. Second, the paper proposes a set of semantic relations between words in sentences, which are used to segment and link the units and construct the requirement hierarchies. Third, the paper proposes a deep learning-based method to automatically extract these semantic relations and transform the building-code sentences with the relations into requirement hierarchies. The proposed method uses a semi-supervised learning strategy to train the deep learning models on both labeled and unlabeled building-code text data.

Background

Semantic Representations of Natural-langue Requirements

In the AEC domain, different types of semantic representations have been developed and used for representing, processing, and checking natural-language requirements by computers in ACC systems. For example, Hjelseth and Nisbet (2011) proposed a set of markups including the requirement, applicability, selection, and exception (RASE) for supporting the annotation of normative requirements. Solihin and Eastman (2016) used conceptual graphs to

represent building-code requirements, where nodes and edges represent the entities and relations contained in the requirements. Sydora and Stroulia (2020) developed a BIM-based rule language for describing interior design rules. Yurchyshyna and Zarli (2009) and Xu and Cai (2020) used modeling languages such as SPARQL protocol and Resource Description Framework (RDF) query language to represent requirements as queries to the BIM models. Häußler et al. (2021) adopted the Business Process Model and Notation (BPMN) and Decision Model and Notation (DMN), which are graphical representations for specifying business and management processes, for supporting rule representation. Despite the importance of these representations, they are limited in supporting fully automated (i.e., without manually crafted rules or human annotations) conversion of natural-language requirements into computable rules.

Deep Learning for Text Analytics

Deep learning methods use computational models that consist of multiple layers to capture different levels of information representations from large-scale data (LeCun et al. 2015). Deep learning methods have drastically improved the state-of-the-art performance in automatically processing and understanding different types of data, including image, video, and text, and meanwhile reduced or eliminated the manual effort in feature engineering compared to traditional machine learning methods. Recurrent neural networks (RNN) are deep learning models consisting of internal states specifically designed to process sequential data, such as text (sequences of words) and time series (sequences of quantity values). However, original RNN units suffer from vanishing gradient problems and are unable to capture long-term dependencies. Thus, two variants of the original RNN units, long short-term memories (LSTM) (Greff et al. 2016) and gated recurrent units (GRU) (Cho et al. 2014) have been proposed and adopted.

RNN have been widely used in natural language understanding tasks including machine translation [e.g., sequence-to-sequence RNN model for machine translation (Sutskever et al. 2014)], semantic analysis [e.g., bidirectional LSTM and multiplayer perceptron (MLP) for dependency parsing and part-of-speech tagging (Clark et al. 2018)], and information extraction [e.g., bidirectional LSTM and conditional random fields (CRF) for extracting named entities (Lample et al. 2016)]. However, only limited research efforts have been focused on RNN-based methods to solve text analysis problems in the AEC domain. Pan and Zhang (2020) developed RNN-based models to mine information from building information modeling (BIM) log data to support design decisions. Zhang and El-Gohary

(2020) developed encoder-decoder models that consist of LSTM layers to generate semantically enriched building-code sentences to facilitate automatic semantic analysis of regulatory documents. Zhang and El-Gohary (2021) used bidirectional LSTM and CRF with transfer learning strategies for extracting semantic and syntactic information elements (e.g., subjects, compliance checking attributes, and quantity values and units) from building-code sentences for supporting compliance checking.

Semi-supervised Learning

Semi-supervised learning is a machine learning approach that learns from both labeled and unlabeled data. Semi-supervised learning approaches combine the supervised and unsupervised learning approaches by leveraging unlabeled data to improve the performance, flexibility, and scalability of machine learning models trained using labeled data. Examples of recent semi-supervised learning approaches in the domains of computer vision and natural language understanding include (1) creating labels for unlabeled data using labeling functions (e.g., Ye and Ling 2019; Ratner et al. 2020; Sohn et al. 2020); (2) capturing the underlying structures from both labeled and unlabeled data using data augmentation methods (e.g., Miyato et al. 2018, Berthelot et al. 2019; Sohn et al. 2020); and (3) improving the representativeness of deep learning models by training the models simultaneously on both unlabeled and labeled data (e.g., Clark et al. 2018, Chen et al. 2020).

Semi-supervised learning-based methods have been proposed to solve various AEC domain-specific tasks. For example, Naganathan et al. (2016) used semi-supervised learning and clustering techniques to automate the identification of the loss factors contributing to the energy loss during transmission for facilitating building energy modeling. Yang et al. (2016) used a semi-supervised learning algorithm to automatically detect near-miss falls in ironwork based on worker's kinematic data captured from wearable inertial measurement units. Liu and El-Gohary (2017) used ontology-based semi-supervised CRF for automated information extraction from bridge inspection reports.

State of the Art and Knowledge Gaps in Semantic Relation Extraction

Semantic relation extraction aims to detect and classify the semantic relationships between words or phrases in natural-language text. Examples of such relations include entity relations (e.g., Ratner et al. 2020), event relations (e.g., Wang et al. 2020), and word dependencies (e.g., Liu and El-Gohary 2021). Rule-based semantic relation extraction methods use rules that are usually manually developed based on domain ontologies, dictionaries,

gazetteers, knowledge bases, and/or syntactics such as part-of-speech tags for the extraction (e.g., Ravikumar et al. 2017, Farahmand et al. 2020). Rule-based methods have been used in the AEC domain to extract domain-specific semantic relations for supporting various downstream tasks. For example, Al Qady and Kandil (2010) developed rules based on the shallow parsing of sentences to extract semantic relations from construction contract documents for improving document management. Zhang and El-Gohary (2013, 2015) developed a rule-based, semantic natural language processing approach to extract information from regulatory documents including building codes and transform them into logic clauses for supporting ACC. Despite the wide use of rule-based methods, in general, they do not scale well when the characteristics of the text change – the rules might need additions or modifications to deal with the different text characteristics (Zhou and El-Gohary 2017; Sacks et al. 2020).

Machine learning-based methods, rather than relying on experts for developing relation extraction rules, employ machine learning models to automatically learn the semantic and syntactic patterns from existing text data, based on which the relations are detected from new text data. Supervised learning-based relation extraction methods (e.g., Wang et al. 2020, Liu and El-Gohary 2021) require labeled data – text data annotated with semantic relations. Compared to rule-based methods, supervised learning-based methods have achieved a higher level of flexibility and scalability, but they often suffer from lack of training data due to the high cost of annotating the data. Semi-supervised learning-based methods (e.g., Ratner et al. 2020, Hu et al. 2020), on the other hand, allow for improving the models trained on small-scale labeled data using large-scale unlabeled data. For application in the ACC domain, there is a lack of research efforts to leverage semi-supervised learning in building-code analytics; and, most of the semantic relations studied in the aforementioned efforts (outside of the AEC domains) cannot capture the semantic and syntactic structural characteristics of AEC domain-specific regulatory text in the context of ACC.

Proposed Requirement Hierarchy and Semantic Relations for Supporting Automated Compliance Checking

Requirement Hierarchy

This paper proposes a new semantic representation – requirement hierarchy – to model building-code requirements, including hierarchically complex requirements with restrictions and/or exceptions for automated representation, processing, and understanding of these requirements for supporting ACC. A requirement hierarchy aims to represent a building-code requirement in a hierarchical structure that consists of several requirement units and the dependencies between these units.

A requirement unit, which consists of only essential semantic information elements (see Table 1), describes a requirement or a constraint on a subject or a compliance checking attribute for supporting compliance checking. Thus, each unit has one subject, one compliance checking attribute, or both, and may or may not have other types of information elements. Each unit does not have any secondary semantic information elements such as restrictions and exceptions, and thus is easily processable by most of the existing ACC methods and applications.

There are two types of dependencies between the requirement units: simple dependency and complex dependency. Simple dependencies include conjunctions (e.g., "and") and disjunctions (e.g., "or"). Complex dependencies include exceptions (e.g., "egress doors shall be side-hinged swinging type, except for doors serving a bathroom within a sleeping unit in Group R-1") and restrictions (e.g., "the door between a private garage and a dwelling unit"). The restrictions can be further classified as locative, attributive, and other restrictions (e.g., restrictions indicating the time, manner, or purpose) in terms of the content of the restriction or how the restriction is applied to the restricted requirement unit. The restrictions can also be classified as subject, attribute, and quantity restrictions in terms of which type of semantic information element in the restricted requirement unit is applied to.

Fig. 2 shows an example building-code sentence from IBC 2018 and the requirement hierarchy corresponding to the sentence. The example sentence is modeled as a requirement hierarchy consisting of three requirement units, where unit 1 is the main unit and units 2 and 3 are restrictions of the main unit.

Semantic Relations

This paper proposes a set of semantic relations to (1) link single words in a regulatory sentence/requirement into multi-terms concepts (i.e., the semantic information elements); (2) link these semantic information elements into requirement units; and (3) link these requirement units into a requirement hierarchy. Four types of relations were defined: inner information element, inner requirement unit, inter requirement unit, and root semantic relations. An inner information element relation is a relation between two words, where the words belong to a single semantic or syntactic information element in a requirement unit. An inner requirement unit relation is a relation between two words, where the words belong to a single requirement unit, but to two different information elements. An inter requirement unit relation is a relation between two words, where the words belong to two different requirement units. An inter requirement unit relation may also exist between words in different sentences that form a single requirement (i.e., a multi-sentence requirement); however, for this paper, the analysis of semantic relations and

191 requirement hierarchies is at the individual sentence-level. A root semantic relation is a relation that denotes the
192 starting word for transforming a sentence annotated with semantic relations into a requirement hierarchy. Thus, each
193 sentence must have one and only one word associated with the root relation. Fig. 3 shows the semantic relations
194 between the words in the example building-code sentence (same as the sentence used in Fig. 2).

195 **Proposed Deep Learning-based Method for Semantic Relation and Requirement Hierarchy Extraction**

196 The proposed deep learning-based method is composed of two primary components: (1) a deep learning model for
197 semantic relation extraction, and (2) an algorithm for transforming regulatory sentences with semantic relations into
198 requirement hierarchies. The research methodology included five main steps, as per Fig. 4: data preparation,
199 semantic relation extraction model development, model training using a semi-supervised learning strategy, semantic
200 relation-based requirement hierarchy construction, and evaluation. An example to illustrate how the requirement
201 hierarchies are extracted from building codes using the proposed method is shown in Fig. 5.

202 ***Data Preparation***

203 Labeled Data Preparation

204 The labeled data – regulatory sentences that are represented in the form of the requirement hierarchy and are labeled
205 with the four semantic relations – were prepared for training the semantic relation extraction model and evaluation.
206 The labeled data were prepared following three steps: corpus preparation, sentence selection, and annotation. First, a
207 small-scale corpus was constructed, which consists of text from multiple codes and standards in the AEC domain,
208 including the IBC, International Energy Conservation Code (IECC), Americans with Disabilities Act (ADA)
209 Standards for Accessible Design, and building-code amendments (e.g., the Champaign IBC Amendments). All
210 documents were converted to the text file format (i.e., .txt) and combined into a single corpus. The corpus was then
211 converted to sentences following two steps – sentence segmentation and sentence tokenization. Sentence
212 segmentation aims to detect the sentence boundaries (e.g., punctuations) and segment the text into sentences.
213 Sentence tokenization aims to further split the sentences into tokens (e.g., words). Second, a group of 600 sentences,
214 which consists of about 15,000 words, were randomly selected from the developed corpus. The selected sentences
215 have different levels of computability. Computability is defined as the ability of the building-code sentence to be
216 represented and processed by a computer in an effective manner (Zhang and El-Gohary 2021). Third, a group of
217 four experts – two from academia (faculty) and two from industry – annotated each word in the selected sentences

with the proposed semantic relations. A purposive sampling strategy, which pinpoints a specific type of participants according to predefined selection criteria (Clark and Creswell 2008), was adopted for selecting the experts. Three main selection criteria were defined: (1) expertise in the AEC domain; (2) familiarity with building codes and compliance checking processes; and (3) awareness of natural language processing and text analytics techniques. Each expert independently annotated the entire set of selected sentences, and the initial inter-annotator agreement was 80% in F1 measure, indicating good reliability of the annotations (Pestian et al. 2012). The discrepancies among the annotations were then resolved by the experts to reach full agreement on the final annotations. The labeled data were split into two sets using a 9:1 ratio: training and validation dataset and testing dataset. The first dataset was further split into a training set (for training the model) and a validation set (for tuning the hyperparameters of the model) for cross-validation. The testing dataset was used for evaluation only.

Unlabeled Data Preparation

The unlabeled data – a large corpus consisting of sentences and sentence fragments from a collection of multiple regulatory documents in the AEC domain – were prepared for training the semantic relation extraction model using a semi-supervised learning strategy. The corpus consists of a total of 20,000 sentences, or about 200,000 tokens, and the size of the entire corpus is 100 Megabytes. The collection used to build the corpus includes four main types of regulatory documents: (1) building codes published by the International Code Council (ICC) (e.g., IBC, IECC, International Residential Code, etc.), (2) local regulatory documents in the AEC domain (e.g., Champaign building code amendments), (3) standard specifications (e.g., MasterFormat specification), and (4) other regulatory documents (e.g., Occupational Safety and Health Standard 1910). The documents of different formats were converted to the TXT format and combined into a single file. The file was then cleaned based on heuristic rules that (1) filter out noises introduced during file format conversion; (2) filter out short-word sequences or fragments that are likely to have no semantic relations (e.g., section headings and standalone words and phrases); and (3) filter out non-textual sequences (e.g., sequences of symbols, numbers, and units converted from equations and tables in the documents). Examples of the rules used include: remove candidate sentences and sentence fragments that (1) have less than 20 characters or have less than five tokens, (2) do not have verbs of any form, or (3) start with section indices.

Semantic Relation Extraction Model Development

The deep-learning model (Dozat and Manning 2016) – the LSTM with MLP – was adopted as the base model for extracting the semantic relations from the regulatory sentences, where LSTM learns the representations of words (or tokens) in the sentences to extract the semantic relations, and MLP, on top of the LSTM, further classifies the extracted relations given the learnt representations. The model consists of three types of layers: the input layer, the encoding layer, and the output layer, as shown in Fig. 6.

Input Layer

The input layer aims to represent the semantics of each word in a vector representation for deep neural network computation purposes. Pretrained word embeddings were used to initialize the weights in the input layer. Pretrained word embeddings are vector representations of words learned on a large, cross-domain corpus by training a machine learning model on the corpus. The word embeddings that were learned by applying the Global Vectors for Word Representation (GloVe) algorithm (Pennington et al. 2014) on a corpus consisting of Wikipedia 2014 and Gigaword 5 were adopted. The adopted word representations consist of vector representations of 40,000 uncased English words, which have a dimension of 300.

Encoding Layer

The encoding layer aims to further learn the contextual vector representations of each word in the input sentence that is discriminative in terms of the semantic relation extraction task. Two strategies were adopted when developing the encoding layer for improved semantic relation extraction performance. First, to enhance the capability of the model to capture long-term semantic and syntactic dependencies that exist in hierarchically complex regulatory sentences, bidirectional LSTM layers were employed. Each such layer consists of a backward and a forward LSTM layer for using the representations of both forward and backward context words when learning the representation of the current word.

Second, to improve the capability of the model in reducing overfitting, dropout layers were added. The layers drop a random fraction of the LSTM units in the encoding layer during the training of the model, according to dropout probabilities. Two dropout probabilities were used, one for training using the unlabeled data and the other using the labeled data. Typically, the dropout probabilities are between 0 and 0.5, which means that less than half of the

LSTM units are dropped and the rest of the LSTM units are retained. The dropout layers are disabled during the evaluation and future use of the semantic relation extraction model (i.e., use in the ACC system).

Output Layer

For each pair of words in the regulatory sentence, the output layer aims to predict (1) whether there is a semantic relation linking the two words and (2) the type of the semantic relation if it exists. The output layer consists of MLP layers and a softmax function. For two words *d* and *h* in a sentence, given their encoded representations (denoted as z_d and z_h) generated by the encoding layers and a specific type of semantic relation *r*, the MLP layers computes a score [denoted as $s(z_d, z_h, r)$]. For each word in the sentence, the scores corresponding to the word and all the other words in the sentence, with all possible types of relations, are normalized into a probability distribution through the softmax function. Thus, the probability that a word *d* is linked to another word *h* by a specific type of semantic relation *r* is $p(d, h, r) = \frac{s(d,h,r)}{\sum_{h \in W, r \in R} s(d,h,r)}$, where *W* is the set of all words in the sentence in which *d* and *h* are located and *R* is the set of all possible types of semantic relations. Thus, for each word in a sentence of length $|W|$, the output layer evaluates all $|W||R|$ different semantic relations and the final semantic relation predicted by the output layer is the one that has the highest probability.

Model Training

To enable the training of the semantic relation extraction model on both labeled and unlabeled data, the semi-supervised training strategy proposed by Clark et al. (2018) was adopted. In each iteration during the training process, the model was first trained on a batch of labeled data [as shown in Fig. 7 (a)] and then trained on a batch of unlabeled data [as shown in Fig. 7 (b)], so that the model learns from the labeled and unlabeled data simultaneously.

When the model is trained on the labeled data, the objective function L_s that describes the difference between the semantic relation from the gold standard, denoted as y, and the semantic relation *c* predicted by the model θ for the word *x*, is minimized, as shown in Eq. (1) (Clark et al. 2018), where *D* is the batch of labeled data, *C* is the set of all possible semantic relations, and $p_\theta(c|x)$ is the probability of *c* given the input word *x* generated by the softmax layer in the output layer of the model.

$$L_s(\theta) = \frac{1}{|D|} \sum_{x,y \in D} \sum_{c \in C} 1_{y=c} \log p_\theta(c|x) \quad (1)$$

The model is trained on the unlabeled data following three steps (Clark et al. 2018). First, for each word in a sentence in the unlabeled data, modified versions of the original sentence were created by removing the preceding or following words in the sentence. Second, the original and modified sentences were provided to the semantic relation extraction model as input separately. Third, the objective function L_u that describes the difference between the probability distribution generated by the output layer for the original sentences, denoted as $p_\theta^o(c|x)$, and the probability distribution generated by the output layer m for the modified sentences, denoted as $p_\theta^m(c|x)$, is minimized, as shown in Eq. (2) (Clark et al. 2018), where D is the batch of unlabeled data, M is the set of all output layers for modified sentences for unsupervised learning, and KL is the Kullback–Leibler divergence. The parameters of the output layer for supervised learning are fixed and only the parameters of the other layers are updated during the training on the unlabeled data. By being trained to minimize the semantic relation extraction results attained using the original and modified sentences, the model improves its ability to capture contextual syntactic and semantic information that can support semantic relation extraction.

$$L_u(\theta) = \frac{1}{|D|}\sum_{x\in D}\sum_{m\in M} KL(p_\theta^o(c|x)||p_\theta^m(c|x)) \quad (2)$$

Semantic Relation-based Requirement Hierarchy Construction

A hybrid width and depth first traversal-based algorithm was used to convert the regulatory sentences annotated with semantic relations to the form of the requirement hierarchy, as shown in Fig. 8. For applying this algorithm, two types of data structures were maintained. First, the sentence (i.e., words) with the semantic relations is modeled as a directed graph, where the words are the vertices and the semantic relations are the edges, and for each edge, its direction is from the node corresponding to the dependent word to the node corresponding to the head word. Second, the requirement hierarchy was constructed by updating three lists: two lists of requirement units, unfinished and finished requirement units, and one list of the dependencies between requirement units.

The algorithm consists of three main stages. First, the algorithm starts at the node corresponding to the word annotated with the root semantic relation. Meanwhile, the unfinished requirement unit list is initialized with one empty requirement unit. Second, the algorithm traverses the entire graph along the edges. At each node, the algorithm traverses the edges that point to the node, according to the following order: inner information element semantic relations, inner requirement unit semantic relations, and inter requirement unit semantic relations (as

shown in Fig. 8). The words corresponding to the traversed nodes and the semantic relations corresponding the traversed edges are added to the lists. Third, the algorithm terminates when all the nodes are traversed. The finished requirement unit list and the dependency list together are the final requirement hierarchy.

Additionally, two rules are applied to index the requirement units and interpret the dependencies between the requirement units. First, the requirement units are indexed based on the order that they appear in the unfinished requirement unit list (e.g., the empty requirement unit contained in the unfinished requirement unit list when initialized is indexed as the first requirement unit), and the first requirement unit is the main requirement unit of the requirement hierarchy, which does not act as a restriction or an exception to any other requirement units. Second, the dependencies between two requirement units are interpreted based on the type of semantic relation that link the words in these two requirement units – one requirement unit is a restriction or an exception of the other requirement unit if there is a inter requirement unit semantic relation linking a word (i.e., the dependent word) in the first requirement unit to a word (i.e., the head word) in the second requirement unit.

Experimental Results and Analysis

A set of experiments were conducted to first optimize the hyperparameters of the proposed model for cross-validation and then to assess the performance of the proposed method in ablation studies to better understand the impact of the important components of the method. Three ablation experiments were conducted to (1) evaluate the impact of the semi-supervised learning strategy by comparing the proposed semi-supervised method to a fully supervised method; (2) evaluate the impact of unlabeled training data, in terms of type and scale; and (3) evaluate the performance of the proposed method on different types of sentences, in terms of types of building codes/standards and levels of computability.

Evaluation Metrics

Semantic Relation Extraction Evaluation

Two metrics were used to evaluate the semantic relation extraction performance: accuracy and unlabeled attachment score (UAS). Accuracy (Zhai and Massung 2016) was used to evaluate the extracted semantic relations in terms of relation types, as shown in Eq. (3), where *SRT* is the set of words that are labeled with the correct type of semantic relation and *N* is the total number of words in the regulatory sentence.

$$\text{Accuracy} = \frac{|SRT|}{N} \quad (3)$$

UAS (Buchholz and Marsi 2006) was used to evaluate the extracted semantic relations in terms of head word indices, as shown in Eq. (4), where *SRI* is the set of words that are linked to the correct head word and *N* is the total number of words in the regulatory sentence.

$$\text{UAS} = \frac{|SRI|}{N} \quad (4)$$

Requirement Hierarchy Extraction Evaluation

The labeled attachment score (LAS) (Buchholz and Marsi 2006) was used to evaluate the overall performance of the proposed method on extracting the requirement hierarchies from regulatory sentences, as shown in Eq. (5), where *SRI* is the set of words that are linked to the correct head word, *SRT* is the set of words that are labeled with the correct type of semantic relation, and *N* is the total number of words in the sentence.

$$\text{LAS} = \frac{|SRI \cap SRT|}{N} \quad (5)$$

Semantic Relation Extraction Model Hyperparameter Optimization

The semantic relation extraction models and semi-supervised learning strategies were implemented using TensorFlow built in Python 2 and run using the Tesla K80 GPU provided in Google Colaboratory. A five-fold cross validation was conducted for optimizing the main hyperparameters of the model. For the cross validation, the labeled training data were further split into two subsets – one set for model training and the other for model validation. The values of other hyperparameters were determined based on the characteristics of the sentences used in the experiments (e.g., the maximum sentence length is 100 and maximum token length is 20), or the practices adopted by Clark et al. (2018) (e.g., the base learning rate is 0.5). The values of the optimized main hyperparameters are shown in Table 2.

The hyperparameters were optimized through search including grid search and random search. For example, for the dropout rate for labeled data and the dropout rate for unlabeled data, three values were tested: 0 (i.e., no dropout), 0.2, and 0.4. Thus, nine pairs of dropout rates were tested and compared in terms of all the evaluation metrics for both semantic relation extraction and requirement hierarchy extraction. The model achieved the highest performance when the dropout rates for unlabeled and labeled data were set as 0.4 and 0.2, respectively. The results indicate that

models with no or very little dropout for any of the two types of data might overfit to the training data, while models with large dropout for any of the two types of data might underfit to the training data.

Comparison of the Proposed Semi-supervised and Supervised Methods in Semantic Relation Extraction and Requirement Hierarchy Extraction

In addition to the proposed semi-supervised learning-based method, a supervised method was developed (to serve as a baseline) and tested for comparative evaluation. The supervised method uses the semantic parser proposed by Chen and Manning (2014) and extracts the requirement hierarchies based on the generated semantic parsing trees. The two methods were trained on the same labeled training data and were evaluated using the same testing data. As shown in Table 3, the proposed semi-supervised method achieved better semantic relation extraction performance: it outperformed the supervised method by 3.5% of accuracy, 3.0% of UAS, and 3.2% of LAS.

Performance of the Proposed Approach Using Semi-supervised Learning Strategy with Different Unlabeled Training Data

Different Types of Unlabeled Training Data

Two types of unlabeled data were tested for comparative evaluation: the domain-general text data – the one-billion-word language model benchmark data (Chelba et al. 2013), which consists of English news sentences – and the domain-specific text data. The one-billion-word language model benchmark data were selected for comparison because the dataset was used as a benchmark in many natural language understanding tasks, such as statistical language modeling (e.g., Chelba et al. 2013) and semantic parsing (e.g., Clark et al. 2018) and the dataset is not domain-specific. For this comparative evaluation, the model was trained using the same labeled data.

As shown in Table 4, the model achieved better performance when trained using domain-specific unlabeled data in terms of both semantic relation extraction and requirement hierarchy extraction – the model trained using domain-specific unlabeled data outperformed the one trained using general-domain unlabeled data by 2.2% of accuracy, 2.0% of UAS, and 2.1% of LAS. The results indicate that the semantic and syntactic patterns in the domain-specific unlabeled data increased the ability of the model to extract semantic relations defined in the same domain, especially predicting the semantic relation types. Although the level of increase is not large, it shows the potential of enhancing the performance of domain-specific semantic relation extraction models by leveraging domain-specific unlabeled data. Domain-specific text data prepared using more types of regulatory documents in the AEC domain and with

more tailored and elaborate data cleaning techniques could be tested in future research to further improve the benefit of using such data.

Different Scales of Unlabeled Training Data

Four different scales of unlabeled data were tested for comparative evaluation: large [100 Megabyte (MB), approximately 10,000,000 tokens], medium (10 MB, approximately 1,000,000 tokens), small (1 MB, approximately 100,000 tokens), and zero (0 MB, 0 tokens). For this comparative evaluation, the model was trained using the same labeled data.

As shown in Table 5, the model achieved the best performance when trained using the large-size data, in terms of both semantic relation extraction and requirement hierarchy extraction. For example, compared to the model trained without (zero) unlabeled data, the one trained using the large unlabeled data increased the accuracy by 3.2%, UAS by 4.0%, and LAS by 3.4%. The results indicate that as the scale of unlabeled training data increases, the model benefits from the semantic and syntactic patterns contained in the extra data and hence its performance increases. However, such increase (i.e., the increase of accuracy, UAS, and LAS per token) gradually diminishes due to both the constraints on the architecture of the model (e.g., depth of the encoding layer) and the diversity and the relativeness of the semantic and syntactic patterns contained in the data. Larger data sizes could be tested in future research to assess the upper bound of the performance increase and the amount of additional unlabeled data required to achieve such increase.

Performance of the Proposed Method on Semantic Relation Extraction and Hierarchy Extraction Across Different Types of Regulatory Sentences

Performance Across Different Types of Building Codes and Standards

The trained semantic relation extraction model was evaluated using sentences from three different types of building codes and standards: IBC, IECC, and ADA Standards. As shown in Table 6, the proposed method achieved high performance across all three types, in terms of both semantic relation extraction and requirement hierarchy extraction, indicating that the proposed method has high scalability and flexibility across different types of building codes and standards. A slightly lower performance was shown for IECC, compared to the other two types, which is likely due to unseen syntactic and semantic patterns in the training data (e.g., IECC-specific tokens such as “R-30”).

Performance Across Sentences with Different Levels of Computability

The performance of the trained semantic relation extraction model was evaluated across sentences with three different levels of computability: moderately high, moderately low, and low computability, which are the top three types of sentences in terms of computability that appear most frequently in building codes (e.g., they account for 22%, 39%, and 23% of a corpus of sentences from IBC and its amendments, respectively) (Zhang and El-Gohary 2021). Sentences with different levels of computability have different semantic and syntactic structures. For example, sentences of moderately high computability have relatively simple syntactic and semantic structures, sentences of moderately low computability have relatively complex syntactic and semantic structures, and sentences of low computability have very complex syntactic and semantic structures.

Table 7 shows the performance of the proposed method across sentences with different levels of computability, in terms of both semantic relation extraction and requirement hierarchy extraction. As expected, the performance decreased for low-computability sentences. However, high performance (i.e., over 90% of accuracy, and over 85% of both UAS and LAS) was still achieved for all three types, which indicates that the method has high scalability and flexibility. Given their complexity, the performance level achieved on moderately low- and low-computability sentences additionally indicates that the proposed method is able to extract semantic relations and requirement hierarchies from hierarchically complex sentences.

Error Analysis

Two types of errors were identified based on the experimental results. First, the proposed method showed some errors when dealing with multiword expressions or concepts that are domain-specific and include prepositions. For example, the semantic relation (i.e., inner information element relation) within the multiword expression “path of egress” (i.e., a subject in the requirement) was not correctly extracted. Second, some errors occurred in segmenting the requirement units, which was caused by errors in differentiating between the inter requirement unit relation and the inner requirement unit relation. For example, the semantic relation between “span” and “concrete” in the sentence fragment “the clear span of the gypsum concrete” is an inner requirement unit relation, because the entire sentence fragment forms a part of a requirement unit. However, the method misclassified it as an inter requirement unit, and thus segmented “the clear span” and “of the gypsum concrete” into two different requirement units. Such error also appeared for sentences with low computability. For example, the following sentence has four requirement

453 units (i.e., “the area of a Group H-2 aircraft paint hangar shall not be limited”, “a Group H-2 aircraft paint hangar no
454 more than one story above grade plane”, “such aircraft is surrounded and adjoined by public ways or yards”, and
455 “not less in width than one and one-half time the building height”): “the area of a Group H-2 aircraft paint hangar no
456 more than one story above grade plane shall not be limited where such aircraft is surrounded and adjoined by public
457 ways or yards not less in width than one and one-half time the building height”. However, the method failed to
458 segment these requirement units correctly due to the high syntactic and semantic complexity of the sentence.

459 ***Limitations***

460 Two limitations related to the proposed semantic relation and requirement hierarchy extraction method and the
461 conducted experiments are acknowledged. First, the proposed method was well-tested on individual building-code
462 sentences, but not on multi-sentence requirements. Additional experiments are needed to further evaluate the
463 performance of the proposed method on multi-sentence requirements – where both the proposed semantic relations
464 and requirement hierarchy and the proposed method would be directly applied, but additional efforts would be
465 needed for preparing both the labeled (e.g., expert annotation of semantic relations) and unlabeled domain-specific
466 data (e.g., heuristic rules for text cleaning). Second, although the proposed method and the proposed semantic
467 relations and requirement hierarchy have successfully addressed different text characteristics and different levels of
468 computability for different building codes and standards, they cannot address the semantic representation and
469 interpretation problems for all types of requirements, such as requirements that have hidden dependencies or
470 assumptions (Solihin and Eastman 2015, 2016) or requirements that require human judgment by nature. Further
471 research is needed to study the limit of the performance of such machine learning-based methods on these
472 challenging types of requirements, by (1) using a larger scale of labeled data or unlabeled data, or data with more
473 diversified semantic and syntactic patterns, (2) exploring more deep learning model structures (e.g., transformers-
474 based models), and (3) integrating additional external knowledge (e.g., legal practices) either in the current ACC
475 process (i.e., information extraction and transformation), or in downstream ACC processes (e.g., compliance
476 reasoning and information matching or disambiguation).

477 **Contribution to the Body of Knowledge**

478 This paper contributes to the body of knowledge in five main ways. First, the paper proposes a new representation,
479 the requirement hierarchy, to represent building-code requirements for supporting downstream ACC processes. The

requirement hierarchy consists of requirement units, each of which further consists of semantic information elements that define a simple requirement or a condition, and dependencies between the units. The requirement hierarchy could facilitate automated (or semi-automated if desired) compliance checking by enabling the decomposition of hierarchically complex requirements into units that are simple and readily processable by many of the existing ACC systems, and representing and visualizing the semantic and syntactic structures for better comprehension of complex requirements. Second, the paper proposes a set of semantic relations between words in regulatory-document sentences, which can be used not only for requirement hierarchy extraction, but also in other regulatory text analytics tasks such as requirement or sentence classification and knowledge graph extraction as features. Third, the paper offers a method to extract requirement hierarchies from building codes and standards. It is the first effort to automatically extract requirement hierarchies to facilitate the analysis of hierarchically complex regulatory sentences for supporting ACC. The proposed method showed high performance across different types of regulatory documents and across sentences with different levels of computability. Fourth, it is the first effort to use a deep learning model that consists of LSTM and MLP to extract semantic relations from building codes and train the model using a semi-supervised learning strategy. Fifth, this research leveraged both labeled data and large-scale unlabeled text data to enhance the performance of the proposed method, increase the scalability and flexibility of the method, and most importantly, significantly reduce the effort to prepare the labeled data for training the deep learning models.

Conclusions and Future Work

In this paper, the requirement hierarchy, a new representation for representing AEC regulatory requirements, was proposed to decompose sentences, especially the hierarchically complex ones, into much smaller, manageable requirement units that would be directly processable using most of the existing ACC methods. The requirement hierarchy consists of requirement units and dependencies between these units. To facilitate the extraction of such requirement hierarchies, a set of semantic relations between words in regulatory sentences was also proposed. A deep learning model, which consists of bi-directional LSTM and MLP, was used for extracting the hierarchies from regulatory-document sentences. Two types of data – labeled data, which consist of sentences annotated with semantic relations, and unlabeled data, which consists of AEC domain-specific regulatory text without any annotation – were prepared for training and evaluating the model. The model was trained using a semi-supervised learning strategy: it was alternatingly trained on the unlabeled and labeled training data, and the supervised and

unsupervised loss were minimized, so that the model can learn from the semantic and syntactic patterns contained in both types of training data. The extracted semantic relations and the original sentences were transformed into requirement hierarchies using a hybrid width and depth first traversal-based algorithm, with rules for indexing the requirement units and interpreting the dependencies between the units.

The proposed semantic relation model was trained on 100 MB of unlabeled text data from multiple types of regulatory documents of the AEC domain, and labeled data that consisted of 540 annotated sentences from the IBC, IECC, and ADA Standards. The trained model achieved an accuracy of 94.0%, a UAS of 90.1, and a LAS of 88.0, indicating high semantic relation and requirement hierarchy extraction performance. The model achieved high performance across different types of regulatory documents including IBC, IECC, and ADA Standards, and across regulatory sentences with different levels of computability.

The experimental results also showed that the semi-supervised learning strategy improves the performance of the relation extraction model by providing more semantic and syntactic patterns contained in the unlabeled training data. The results also showed that the model's performance increases when the amount of training data increases and domain-specific unlabeled training data is used, indicating the potential of the proposed method to achieve higher relation extraction performance if more unlabeled, domain-specific text data is used, which is significantly less costly to prepare than labeled, domain-specific text data.

In their future work, the authors plan to improve the proposed method and leverage the semantic relation extraction model in four directions. First, different deep learning model designs could be tested to enhance the semantic relation extraction performance. For example, different model architectures (e.g., adding more LSTM layers in the encoding layer) and recent deep neural network advances (e.g., transformers and attention mechanisms) could be explored. Second, different semi-supervised learning strategies could be explored for leveraging large-scale, pattern-rich unlabeled text data in the AEC domain. Third, the performance and flexibility of the model could be further improved by increasing the amount, quality, and the relativeness of the unlabeled training data. For example, text from more types of regulatory documents in the AEC domain such as local building-code amendments and contracts could be used. Fourth, and most importantly, the authors will further use the proposed semantic relation-based requirement hierarchy extraction method to improve automated processing and understanding of building codes, and to integrate it with downstream processes such as information matching and compliance reasoning. Our ultimate

goal is to leverage machine learning and other artificial intelligence (AI) approaches to reach a level where users can automatically process the entire building code and represent it in a computable manner for fully automated compliance checking, with minimum manual effort in developing the underlying models (e.g., for preparing the training and testing data and for adapting the models to different types of building codes). Fifth, beyond automated compliance checking applications, future research could explore how the proposed requirement representation could support further code- and contract-analytics processes such as detecting conflicting or onerous requirements. The computability of the representation, as well as its structured and semantic nature, would help leverage automated analytics and AI techniques to support further diagnostic analytics tasks to identify potential conflicts among different requirements within one document (e.g., a specific code) or across multiple documents (e.g., codes vs. specifications), detect onerous contractual requirements, or diagnose other unfair situations like one-sided contractual agreements.

Data Availability Statement

The trained model for semantic relation extraction and the labeled gold standard data generated and used during the study are available from this link: https://publish.illinois.edu/rzhang65-data-sharing/.

Acknowledgements

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

References

Akanbi, T. and Zhang, J., 2021. Design information extraction from construction specifications to support cost estimation. Automation in Construction, 131, p.103835.

Al Qady, M. and Kandil, A., 2010. Concept relation extraction from construction documents using natural language processing. Journal of construction engineering and management, 136(3), pp.294-302.

Beach, T.H., Hippolyte, J.L. and Rezgui, Y., 2020. Towards the adoption of automated regulatory compliance checking in the built environment. Automation in Construction, 118, p.103285.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A. and Raffel, C., 2019. Mixmatch: A holistic approach to semi-supervised learning. arXiv preprint arXiv:1905.02249.

Bloch, T. and Sacks, R., 2020. Clustering Information Types for Semantic Enrichment of Building Information Models to Support Automated Code Compliance Checking. Journal of Computing in Civil Engineering, 34(6), p.04020040.

Buchholz, S. and Marsi, E., 2006, June. CoNLL-X shared task on multilingual dependency parsing. In Proceedings of the tenth conference on computational natural language learning (CoNLL-X) (pp. 149-164).

Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P. and Robinson, T., 2013. One billion word benchmark for measuring progress in statistical language modeling. arXiv preprint arXiv:1312.3005.

Chen, D. and Manning, C.D., 2014, October. A fast and accurate dependency parser using neural networks. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 740-750).

Chen, T., Kornblith, S., Swersky, K., Norouzi, M. and Hinton, G., 2020. Big self-supervised models are strong semi-supervised learners. arXiv preprint arXiv:2006.10029.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Clark, K., Luong, M.T., Manning, C.D. and Le, Q.V., 2018. Semi-supervised sequence modeling with cross-view training. arXiv preprint arXiv:1809.08370.

Dimyadi, J., Pauwels, P. and Amor, R., 2016. Modelling and accessing regulatory knowledge for computer-assisted compliance audit. Journal of Information Technology in Construction, 21, pp.317-336.

Dozat, T. and Manning, C.D., 2016. Deep biaffine attention for neural dependency parsing. arXiv preprint arXiv:1611.01734.

Farahmand, S., Riley, T. and Zarringhalam, K., 2020. ModEx: A text mining system for extracting mode of regulation of transcription factor-gene regulatory interaction. Journal of biomedical informatics, 102, p.103353.

Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R. and Schmidhuber, J., 2016. LSTM: A search space odyssey. IEEE transactions on neural networks and learning systems, 28(10), pp.2222-2232.

Häußler, M., Esser, S. and Borrmann, A., 2021. Code compliance checking of railway designs by integrating BIM, BPMN and DMN. Automation in Construction, 121, p.103427.

589 Hjelseth, E. and Nisbet, N. 2011. Exploring semantic based model checking. Proc. 2010 27th CIB W78 Int. Conf.,
590 http://itc.scix.net/data/works/att/w78-2010-54.pdf, accessed Feb 2020.
591 Hu, X., Ma, F., Liu, C., Zhang, C., Wen, L. and Yu, P.S., 2020. Semi-supervised relation extraction via incremental
592 meta self-training. arXiv preprint arXiv:2010.16410.
593 Kim, H., Lee, J.K., Shin, J. and Choi, J., 2019. Visual language approach to representing KBimCode-based Korea
594 building code sentences for automated rule checking. Journal of Computational Design and Engineering, 6(2),
595 pp.143-148.
596 Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K. and Dyer, C., 2016. Neural architectures for named
597 entity recognition. arXiv preprint arXiv:1603.01360.
598 LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning. Nature. 521(7553), pp. 436.
599 Lee, J.K., Eastman, C.M. and Lee, Y.C., 2015. Implementation of a BIM domain-specific language for the building
600 environment rule and analysis. Journal of Intelligent & Robotic Systems, 79(3-4), pp.507-522.
601 Li, S., Cai, H. and Kamat, V.R., 2016. Integrating natural language processing and spatial reasoning for utility
602 compliance checking. Journal of Construction Engineering and Management, 142(12), p.04016074.
603 Liu, K. and El-Gohary, N., 2017. Ontology-based semi-supervised conditional random fields for automated
604 information extraction from bridge inspection reports. Automation in Construction, 81, pp.313-327.
605 Liu, K. and El-Gohary, N., 2021. Semantic neural network ensemble for automated dependency relation extraction
606 from bridge inspection reports. Journal of Computing in Civil Engineering, 35(4), p.04021007.
607 Marzouk, M. and Enaba, M., 2019. Text analytics to analyze and monitor construction project contract and
608 correspondence. Automation in Construction, 98, pp.265-274.
609 Miyato, T., Maeda, S.I., Koyama, M. and Ishii, S., 2018. Virtual adversarial training: a regularization method for
610 supervised and semi-supervised learning. IEEE transactions on pattern analysis and machine intelligence, 41(8),
611 pp.1979-1993.
612 Naganathan, H., Chong, W.O. and Chen, X., 2016. Building energy modeling (BEM) using clustering algorithms
613 and semi-supervised machine learning approaches. Automation in Construction, 72, pp.187-194.
614 Nawari, N.O., 2019. A Generalized Adaptive Framework (GAF) for Automating Code Compliance Checking.
615 Buildings, 9(4), p.86.

Pan, Y. and Zhang, L. 2020. BIM log mining: Learning and predicting design commands. Automation in Construction, 112, p.103107.

Pennington, J., Socher, R. and Manning, C.D., 2014, October. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

Pestian, J.P., Deleger, L., Savova, G.K., Dexheimer, J.W. and Solti, I., 2012. Natural language processing–the basics. In Pediatric Biomedical Informatics (pp. 149-172). Springer, Dordrecht.

Preidel, C. and Borrmann, A. 2016. Towards code compliance checking on the basis of a visual programming language. ITcon. 21(25), pp.402–421.

Ratner, A., Bach, S.H., Ehrenberg, H., Fries, J., Wu, S. and Ré, C., 2020. Snorkel: Rapid training data creation with weak supervision. The VLDB Journal, 29(2), pp.709-730.

Ravikumar, K.E., Rastegar-Mojarad, M. and Liu, H., 2017. BELMiner: adapting a rule-based relation extraction system to extract biological expression language statements from bio-medical literature evidence sentences. Database, 2017.

Sacks, R., Girolami, M. and Brilakis, I., 2020. Building Information Modelling, Artificial Intelligence and Construction Tech. Developments in the Built Environment, p.100011.

Sohn, K., Berthelot, D., Li, C.L., Zhang, Z., Carlini, N., Cubuk, E.D., Kurakin, A., Zhang, H. and Raffel, C., 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. arXiv preprint arXiv:2001.07685.

Solihin, W. and Eastman, C., 2015. Classification of rules for automated BIM rule checking development. Automation in construction, 53, pp.69-82.

Solihin, W. and Eastman, C.M., 2016. A knowledge representation approach in BIM rule requirement analysis using the conceptual graph. ITcon, 21, pp.370-401.

Solihin, W., Dimyadi, J., Lee, Y.C., Eastman, C. and Amor, R., 2020. Simplified schema queries for supporting BIM-based rule-checking applications. Automation in Construction, 117, p.103248.

Sutskever, I., Vinyals, O. and Le, Q.V., 2014. Sequence to sequence learning with neural networks. In Advances in neural information processing systems (pp. 3104-3112).

643 Sydora, C. and Stroulia, E., 2020. Rule-based compliance checking and generative design for building interiors
644 using BIM. Automation in Construction, 120, p.103368.
645 Wang, H., Chen, M., Zhang, H. and Roth, D., 2020. Joint constrained learning for event-event relation extraction.
646 arXiv preprint arXiv:2010.06727.
647 Xu, X. and Cai, H., 2020. Semantic approach to compliance checking of underground utilities. Automation in
648 Construction, 109, p.103006.
649 Xue, X. and Zhang, J., 2020. Building Codes Part-of-Speech Tagging Performance Improvement by Error-Driven
650 Transformational Rules. Journal of Computing in Civil Engineering, 34(5), p.04020035.
651 Yang, K., Ahn, C.R., Vuran, M.C. and Aria, S.S., 2016. Semi-supervised near-miss fall detection for ironworkers
652 with a wearable inertial measurement unit. Automation in Construction, 68, pp.194-202.
653 Ye, Z.X. and Ling, Z.H., 2019. Distant supervision relation extraction with intra-bag and inter-bag attentions. arXiv
654 preprint arXiv:1904.00143.
655 Yurchyshyna, A. and Zarli, A., 2009. An ontology-based approach for formalisation and semantic organisation of
656 conformance requirements in construction. Automation in Construction, 18(8), pp.1084-1098.
657 Zhai, C., and Massung, S. 2016, Text data management and analysis: a practical introduction to information retrieval
658 and text mining, ACM, New York, USA.
659 Zhang, J. and El-Gohary, N., 2015. Automated information transformation for automated regulatory compliance
660 checking in construction. Journal of Computing in Civil Engineering, 29(4), p.B4015001.
661 Zhang, J., and El-Gohary, N. 2013. “Semantic NLP-based information extraction from construction regulatory
662 documents for automated compliance checking.” Journal of Computing in Civil Engineering,
663 10.1061/(ASCE)CP.1943-5487.0000346, 04015014.
664 Zhang, R. and El-Gohary, N., 2020. A Deep-Learning Method for Evaluating Semantically-Rich Building Code
665 Annotations. In EG-ICE 2020 Workshop.
666 Zhang, R. and El-Gohary, N., 2021. A deep neural network-based method for deep information extraction using
667 transfer learning strategies to support automated compliance checking. Automation in Construction.
668 Zhang, R., and El-Gohary, N., 2021. Clustering-based Approach for Building Code Computability Analysis. Journal
669 of Computing in Civil Engineering. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000967.

Zhou, P., and El-Gohary, N. 2017. “Ontology-based automated information extraction from building energy conservation codes.” Automation in Construction, 74, 103-117.

673 **List of Figure Captions**

674 **Fig. 1.** Example building-code sentences with different levels of semantic and syntactic complexities.

675 **Fig. 2.** Example building-code sentence with the corresponding requirement hierarchy.

676 **Fig. 3.** Example building-code sentence annotated with semantic relations.

677 **Fig. 4.** Research methodology.

678 **Fig. 5.** Semantic relation and requirement hierarchy extraction using the proposed method.

679 **Fig. 6.** Deep learning model for extracting semantic relations from regulatory sentences: architecture and example.

680 **Fig. 7.** Semi-supervised learning strategy for adapting and training the semantic relation extraction model using (a)
681 labeled data and (b) unlabeled data.

682 **Fig. 8.** Hybrid width and depth first traversal-based algorithm to convert regulatory sentences with semantic
683 relations to requirement hierarchies.

Tables

Table 1. Semantic Information Elements for Representing Requirements for Compliance Checking (Zhang and El-Gohary 2013, Zhang and El-Gohary 2021)

Semantic information element	Definition
Subject	An ontology concept representing a thing (e.g., building element) that is subject to a particular requirement
Subject relation	A term or phrase that defines the type of relation between two subjects, a subject and an attribute, or a subject or an attribute and a quantity
Compliance checking attribute	An ontology concept representing a specific characteristic of a "subject" that is checked for compliance
Deontic operator indicator	A term/phrase that indicates the deontic type of the requirement (i.e., obligation, permission, or prohibition)
Quantitative relation	A term/phrase that defines the type of relation for the quantity (e.g., extend)
Comparative relation	A term/phrase for comparing quantitative values, including "greater than or equal to," "greater than," "less than or equal to," "less than," and "equal to"
Quantity value	A numerical value that defines the quantity
Quantity unit	The unit of measure for a "quantity value"
Reference	A term or phrase that denotes the mentioning or reference to a chapter, section, document, table, or equation in a regulatory document

Table 2. Optimized Main Hyperparameters for the Semantic Relation Extraction Models

Hyperparameter	Value
Batch size for labeled training data	60
Batch size for unlabeled training data	60
Dimension of the input layer	50
Dimension of the bidirectional long short-term memories (LSTM) layers in the encoding layer	512
Dimension of the output layer	512
Dropout rate for labeled data	0.4
Dropout rate for unlabeled data	0.2

Table 3. Performance of Proposed Semi-supervised and Supervised Methods

Requirement hierarchy extraction method	Semantic relation extraction		Requirement hierarchy extraction
	Accuracy	Unlabeled attachment score (UAS)	Labeled attachment score (LAS)
Semi-supervised learning-based method	**94.0%**	**90.0%**	**87.2%**
Supervised learning-based method	90.5%	87.0%	84.0%

[1]Bolded font indicates highest performance.

Table 4. Performance of Proposed Method Across Different Types of Unlabeled Training Data

Type of unlabeled training data	Semantic relation extraction		Requirement hierarchy extraction
	Accuracy	Unlabeled attachment score (UAS)	Labeled attachment score (LAS)
Domain-specific text	**94.0%**	**90.0%**	**87.2%**
Domain-general text	91.8%	88.0%	85.1%

[1]Bolded font indicates highest performance.

Table 5. Performance of Proposed Method with Different Scales of Unlabeled Training Data

Scale of unlabeled training data in Megabyte (MB) and approximate number of tokens	Semantic relation extraction		Requirement hierarchy extraction
	Accuracy	Unlabeled attachment score (UAS)	Labeled attachment score (LAS)
Large (100 MB, 10,000,000 tokens)	**94.0%**	**90.0%**	**87.2%**
Medium (10 MB, 1,000,000 tokens)	92.3%	87.9%	85.3%
Small (1 MB, 100,000 tokens)	91.3%	86.5%	84.5%
Zero (0 MB, 0 tokens)	90.8%	86.0%	83.8%

[1]Bolded font indicates highest performance.

Table 6. Performance of Proposed Method Across Different Types of Building Codes and Standards

Type of regulatory document	Semantic relation extraction		Requirement hierarchy extraction
	Accuracy	Unlabeled attachment score (UAS)	Labeled attachment score (LAS)
International Building Code	95.0%	91.8%	88.9%
International Energy Conservation Code	93.0%	86.9%	83.4%
ADA Standards	95.0%	92.0%	89.3%

Table 7. Performance of Proposed Method Across Sentences with Different Levels of Computability

Computability level of regulatory sentences	Semantic relation extraction		Requirement hierarchy extraction
	Accuracy	Unlabeled attachment score (UAS)	Labeled attachment score (LAS)
Moderately high	95.0%	92.1%	89.2%
Moderately low	96.0%	91.2%	89.5%
Low	93.0%	89.4%	85.9%