Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking

Jiansong Zhang[1] and Nora M. El-Gohary, A.M.ASCE[2]

Abstract: Automated regulatory compliance checking requires automated extraction of requirements from regulatory textual documents and their formalization in a computer-processable rule representation. Such information extraction (IE) is a challenging task that requires complex analysis and processing of text. Natural language processing (NLP) aims to enable computers to process natural language text in a human-like manner. This paper proposes a semantic, rule-based NLP approach for automated IE from construction regulatory documents. The proposed approach uses a set of pattern-matching-based IE rules and conflict resolution (CR) rules in IE. A variety of syntactic (syntax/grammar-related) and semantic (meaning/context-related) text features are used in the patterns of the IE and CR rules. Phrase structure grammar (PSG)-based phrasal tags and separation and sequencing of semantic information elements are proposed and used to reduce the number of needed patterns. An ontology is used to aid in the recognition of semantic text features (concepts and relations). The proposed IE algorithms were tested in extracting quantitative requirements from the 2009 International Building Code and achieved 0.969 and 0.944 precision and recall, respectively. **DOI: 10.1061/(ASCE)CP.1943-5487.0000346.**

Author keywords: Automated compliance checking; Automated information extraction; Semantic systems; Natural language processing; Automated construction management systems.

Introduction

Construction projects are governed by a multitude of federal, state, and local regulations, such as the International Building Code (IBC), the Americans with Disabilities Act (ADA) Standards for Accessible Design, the International Fire Code, the International Energy Conservation Code, the Occupational Safety and Health Administration's (OSHA) Cranes and Derricks in Construction, the Illinois Accessibility Code, the Illinois Energy Conservation Code, the Illinois Plumbing Code, and the Municipal Code of Chicago. Each regulation has a large set of provisions. For example, the IBC 2006 is composed of 329 sections, and each section includes several to tens of provisions that address a variety of requirements (e.g., safety, environmental).

Building codes are the primary sets of regulations governing the design, construction, alteration, and maintenance of building structures. Within the fifty states of the United States, different versions of the IBC and the International Residential Code (IRC) are adopted, such as IBC 2003, 2006, and 2009, and IRC 2003, 2006, and 2012. Federal and state laws further allow for the adoption of local jurisdiction to adapt these codes to various local conditions (e.g., weather conditions). Thus, in most states, the IBC/IRC is adapted and/or amended for local adoption. Further, some states, such as Mississippi, Missouri, and Delaware, do not enforce a statewide-adopted building code and require their local jurisdictions to adopt and enforce their own selected building code. The state of Massachusetts, further, drafted its own building code. As such, a large number of building codes exist, with each code usually having its own formatting and semantic structure. Moreover, the formatting and semantics of the provisions could vary from one chapter to another within a single code.

Given the large number of construction regulatory documents, the variability of their provisions in terms of formatting and semantics, and the large amount and complexity of the information they describe, the manual process of regulatory compliance checking is time consuming, costly, and error prone, similar to other manual processes (Boken and Callaghan 2009). For example, in the city of Mesa, Arizona, the turnaround time for a single commercial building plan review is 18 business days, with a fee assessed at a rate of $90 per hour (City of Mesa 2012). Failure to comply with regulations could further result in incurring much higher costs. For example, Wal-Mart Stores Inc. was fined $1 million for violating stormwater regulations (EPA 2004; Salama and El-Gohary 2011). Automated compliance checking (ACC) is expected to reduce the time, cost, and errors of compliance checking (Tan et al. 2010; Eastman et al. 2009). With the advancements in computing technology, many research efforts endeavored to automate the compliance checking process (e.g., Garrett and Fenves 1987; Delis and Delis 1995; Han et al. 1997; Lau and Law 2004; Eastman et al. 2009; Tan et al. 2010). Larger research and software development efforts for automated building code checking led by industry bodies/associations, software companies, and/or government organizations include Solibri Model Checker (Corke 2013), EPLAN/BIM led by FIATECH (Fiatech 2011), CORENET led by the Singapore Ministry of National Development (Singapore Building and Construction Authority 2006), REScheck and COMcheck led by the U.S. Department of Energy (DOE 2011), SMARTcodes led by the International Code Council (AEC3 2012), and Avolve Software (Avolve Software Corporation 2011).

[1]Graduate Student, Dept. of Civil and Environmental Engineering, Univ. of Illinois at Urbana-Champaign, 205 N. Mathews Ave., Urbana, IL 61801.

[2]Assistant Professor, Dept. of Civil and Environmental Engineering, Univ. of Illinois at Urbana-Champaign, 205 N. Mathews Ave., Urbana, IL 61801 (corresponding author). E-mail: gohary@illinois.edu

Note. This manuscript was submitted on February 28, 2013; approved on July 17, 2013; published online on July 25, 2013. Discussion period open until August 3, 2015; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Computing in Civil Engineering***, © ASCE, ISSN 0887-3801/04015014(14)/$25.00.

Undoubtedly, previous research and software development efforts paved the way for ACC in the architectural, engineering, and construction (AEC) industry. However, these efforts are limited in their automation and reasoning capabilities (Zhong et al. 2012; Zhang and El-Gohary 2013); existing ACC systems require manual effort to extract requirements from textual regulatory documents (e.g., codes) and encode these requirements in a computer-processable rule format. Rules are either hard-coded into the developed systems or hand-coded as a rule database or set of files. For example, in the most recent effort of the International Code Council's SMARTcodes, creating SMARTcodes rules requires manual extraction and encoding effort.

To address this gap, the authors propose a new approach for automated regulatory information extraction to support ACC in construction. The proposed approach utilizes semantic modeling and semantic natural language processing (NLP) techniques to facilitate automated textual regulatory document analysis (e.g., code analysis) and processing for extracting requirements from these documents and formalizing these requirements in a computer-processable format. NLP is a field that utilizes artificial intelligence to enable computers to process natural language text (or speech) in a human-like manner (Cherpas 1992). Information extraction (IE) is a subfield of NLP that aims to extract desired information from text sources to fill in predefined information templates. IE could be based on the syntactic (i.e., grammatical) and/or semantic (i.e., meaning descriptive) features of the text.

Proposed Approach for Automated Regulatory Information Extraction

NLP Approach

A semantic, rule-based NLP approach for automated IE from construction regulatory documents is proposed. The authors' analysis shows that domain-specific regulatory text is more suitable for automated NLP (i.e., allows for better interpretability and less ambiguity in automated processing) relative to general nontechnical text (e.g., news articles, general websites) because of three main text characteristics. First, construction text is likely to have less homonym conflicts than nontechnical text. For example, in news articles, the term "bridge" may refer to, for example, a structural bridge, the card game, a bridge of understanding, or a dental bridge. Second, developing an ontology that captures domain knowledge as opposed to one that captures general knowledge (or a wide variety of domains) is easier. A domain ontology may enhance automated interpretability and understandability of domain-specific text. Third, regulatory text is likely to exhibit less coreference resolution problems. For example, construction regulatory text tends to explicitly mention the subjects (e.g., door) for each provision rather than referring to the subjects using pronouns (e.g., "it").

Rule-Based Approach

The proposed approach is rule based. NLP takes two primary types of approaches: a rule-based approach and a machine learning (ML)-based approach. Rule-based NLP uses manually coded rules for text processing. These rules are iteratively constructed and refined to improve the accuracy of text processing. ML-based NLP uses ML algorithms to train text processing models based on the text features of a given training text (Tierney 2012). Rule-based NLP tends to show better text processing performance (in terms of precision and recall) but requires greater human effort. In this research, a rule-based approach is adopted because of its expected higher performance. The proposed approach uses IE rules that rely on pattern matching to identify the part(s) of the text to extract based on recognized text patterns. The approach relies on both the semantic and syntactic features of the text in defining these patterns. The syntactic features [e.g., part of speech (POS) tags] of the text are captured using various NLP techniques, including tokenization, sentence splitting, morphological analysis, POS tagging, and phrase structure analysis. The semantic features (concepts and relations) of the text are captured based on an ontology that represents the domain knowledge. Given the compositional and recursive nature of text, sentences could be long and complex, which may result in a large number of patterns. The proposed approach utilizes phrase structure grammar (PSG) in the syntactic analysis to reduce the number of patterns needed in IE rules (Zhang and El-Gohary 2012b). Reducing this number is essential for making IE rules more general and, thus, increasing their extraction power, resulting in requiring less IE rules for extraction and reducing the human effort needed to develop IE rules. The proposed approach also separates and sequences the extraction of different semantic information elements to further limit the number of needed IE patterns. In addition to IE rules, a set of rules for resolving conflicts in information extraction (CR rules) are used.

Semantic Approach

The semantic features of the text are captured using a domain ontology. An ontology models domain knowledge in the form of concept hierarchies, relationships (between concepts), and axioms (El-Gohary and El-Diraby 2010). Ontology-based semantic IE (i.e., using meaning/context-related features in addition to syntax/grammar-related features) is expected to achieve higher performance compared with syntactic IE (i.e., using only syntactic features) because domain knowledge (represented in an ontology) could assist in identifying or distinguishing domain-specific terms and meanings (Soysal et al. 2010). For example, Zhang and El-Gohary (2011) showed enhanced performance with semantic IE compared with syntactic-only IE (an increase in precision from 75 to 100% and in recall from 75 to 95%).

Comparison to State of the Art

Many research efforts were conducted for IE in various domains (Soysal et al. 2010; Sapkota et al. 2012; Hogenboom et al. 2013). State-of-the-art semantic IE studies have four major focuses: named entity extraction, attribute extraction, relation extraction, and event extraction. Named entity extraction, attribute extraction, and relation extraction aim to extract instances of a single concept (e.g., named entity) or of two related concepts (Ling and Weld 2012; Pasca 2011; Wang et al. 2010). Event extraction aims to extract instances of multiple concepts (Patwardhan 2010). From this perspective, the proposed approach is more similar to event extraction because instances of multiple concepts in a provisional requirement are extracted. However, compared with event extraction, the approach is different in two primary ways. First, the information is extracted in a more flexible manner. In the proposed approach, two types of information elements are defined: "rigid information elements" and "flexible information elements." A rigid information element has a pre-defined, fixed number of concepts/relations (e.g., in a terrorist event case, it is predefined that "victim" is associated with only one concept). In contrast, a flexible information element has a varying number of concepts/relations depending on the instance at hand (e.g., in this approach, "subject restriction" has a varying number of multiple concepts/relations). Unlike event extraction, the proposed approach can extract the instances of flexible information elements. Second, because a method for extracting

information elements in a more flexible way is introduced, a deeper level of information extraction is performed (i.e., a deeper level toward full sentence interpretation). Shallow NLP conducts partial analysis of a sentence or analyzes a sentence from a specific angle of view (e.g., part-of-speech tagging, text chunking). Deep NLP aims at full sentence analysis with a more complex understanding of the text toward capturing the entire meaning of sentences (Zouaq 2011). Correspondingly, shallow IE extracts specific type(s) of information from a sentence, whereas deep IE aims to extract all information expressed by a sentence based on the full analysis of the sentence.

In terms of IE performance, for the four main types of information (entities, attributes, relations, and events), state-of-the-art performance results are within the range of 0.80 to 0.90 for both precision and recall (e.g., Li et al. 2012; Bing et al. 2013; Sun et al. 2011; Tang et al. 2012). One of the most recent IE studies that aimed to extract protected health information reported a best performance of 0.9668 and 0.9377 for precision and recall, respectively (Deleger et al. 2013).

In the construction domain, a number of important research efforts utilized NLP techniques [e.g., Caldas and Soibelman (2003) conducted ML-based text classification of construction documents]; however, only a few of these efforts conducted some type/level of information extraction [e.g., Abuzir and Abuzir (2002) and Al Qady and Kandil (2010)]. Al Qady and Kandil (2010) used shallow parsers to extract concepts and relations from construction contracts. In Al Qady and Kandil (2010), (1) the extraction is only based on syntactic features produced by shallow parsing; and (2) information recognition is based on specific types of phrases and their roles (produced by shallow parsing) [e.g., NP segment and its role SUBJ (i.e., subject)], which allows for extracting relations between concepts. In the authors' approach, (1) semantic features are used in addition to syntactic ones; and (2) patterns that consist of a variety of syntactic and semantic features are used in the IE and CR rules, which allows for a deeper level of information extraction (i.e., extracting all information of a requirement for further representation in a logic-based rule format). Abuzir and Abuzir (2002) used IE techniques to extract terms and relations from HTML documents for constructing a civil engineering thesaurus. In Abuzir and Abuzir (2002), (1) the extraction uses HTML-based document structure features (including title tags, heading tags, and URLs) and simple lexical syntactic features; and (2) because the main purpose of the extraction is thesaurus construction, their information extraction focuses on extracting terms. In the authors' approach, (1) document structure features are not used (because of dealing with unstructured text rather than HTML documents) and the extraction relies on the syntactic and semantic features of the text; and (2) because the ultimate purpose is automated reasoning about regulatory requirements, information extraction is conducted on a deeper level; not only terms/concepts need to be extracted, but also other information elements (e.g., restrictions) need to be extracted for extracting all information expressed in a sentence/requirement. As such, compared with these efforts, in this research, the authors are

1. Addressing a different application (i.e., ACC). NLP methods, algorithms, and results are highly application-dependent (Salama and El-Gohary 2013a);
2. Tackling a deeper NLP/IE task. The authors aim to automatically process the text to extract regulatory requirements/rules and represent them as logic sentences; and
3. Taking a deeper semantic approach for NLP (Zhang and El-Gohary 2012a). The authors utilize a domain ontology for identifying semantic text features. Using domain-specific semantics and "flexible information elements" to achieve relatively deep semantic NLP allows for
   a. Analyzing complex sentences that would otherwise be too complex for automated information extraction;
   b. Recognizing domain-specific text meaning; and
   c. In turn, improving the performance of IE.

Background—Phrase Structure Grammar

Phrase-structure grammar (PSG) was first introduced by Noam Chomsky (Chomsky 1956) to represent the structure of constituents (i.e., phrases, words) in sentences. PSG relies on constituency relations. According to Chomsky (1956), "a phrase-structure grammar is defined by a finite vocabulary (alphabet) Vp, a finite set Σ of initial strings in Vp, and a finite set F of rules of the form: $X \rightarrow Y$, where X and Y are strings in Vp." The key advantage of PSG is that it singles out and encodes the most important recursive structure and syntactic constituency of a sentence (Levine and Meurers 2006). Using PSG, a complex sequence of features on the right-hand side of a rule could be represented by a few or even just one simple symbol on the left-hand side of the rule. This advantage makes PSG a potentially powerful technique for encoding complex sentence structures. Context-free grammar (CFG) is a more restricted form of PSG. The restriction of CFG beyond general PSG is that the left-hand side of a generative rule must be a single nonterminal (i.e., a symbol that could be further broken down). This restriction simplifies the representation of patterns and, thus, reduces the number of patterns needed in IE rules. Fig. 1 shows an example sentence derivation based on a set of CFG rules. If the left-hand side of a CFG rule matches a node, then the node can be replaced by the right-hand side of the CFG rule. Derivation of all sentences starts from the single root node—the "Sentence" node in the example used. In the first step of the derivation, the root node "Sentence" is replaced by the nodes "NP" and "VP" according to the CFG rule "Sentence→NP VP." Then, the node "VP" could be replaced by the nodes "MD" and "VP" according to the CFG rule "VP→MD VP." This process continues until all nodes are terminals (i.e., words or numbers in the case of the example). The meanings of the nonterminals are explained in the upper right part of Fig. 1. They are either POS tags or phrasal tags (except for the root node "Sentence"). POS tags and phrasal tags are discussed in the following section.

Proposed Information Extraction Methodology

This section presents the proposed methodology for automatically extracting information from construction regulatory documents. The methodology is presented as a domain-specific, semantic IE methodology that can be adopted (as is or with adaptation) by other researchers in the construction domain. The methodology is composed of the following seven phases (as per Fig. 2): information representation, preprocessing, feature generation, target information analysis, development of information extraction rules (IE and CR rules), extraction execution, and evaluation. The approach is iterative to improve performance.

Phase I—Information Representation

This phase is proposed to define the representation format for the extracted information. In this methodology, the ultimate representation format is one or more logic sentences that could be directly used to automate compliance reasoning. For intermediate processing, a new ACC-tuple is proposed to represent the extracted information. The use of a tuple format for intermediate processing is

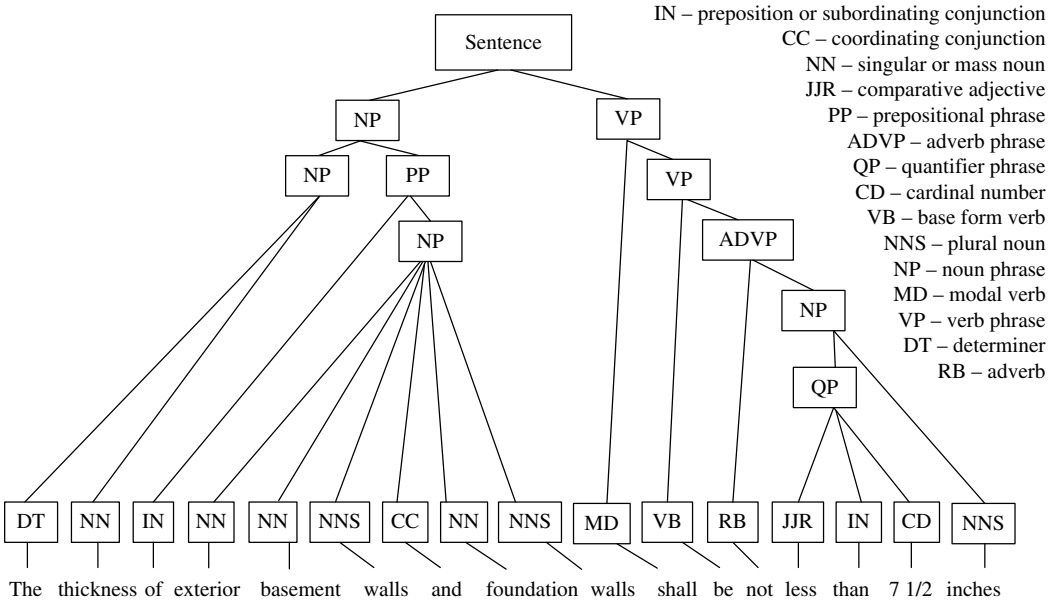

Fig. 1. Sample set of CFG rules (partial) and corresponding derivation of a sentence

proposed because it is easy for computer manipulation and evaluation (e.g., <Subject, Attribute, Value> is a three-tuple).

In the ACC-tuple representation, each element is called a "semantic information element," which is (1) an ontology concept; 2)) an ontology relation; (3) a "deontic operator indicator," which is a term indicating an obligation, permission, or prohibition following the semantic ACC model in Salama and El-Gohary (2013b); or (4) a "restriction", which is an element that places a constraint on the definition of another semantic information element, where the constraint is expressed in terms of ontology concepts and relations. The following types of semantic information elements are introduced: a "simple semantic information element" (SIE) versus a "complex SIE," and a "rigid SIE" versus a "flexible SIE." A simple SIE is associated with a single concept/relation/indicator, whereas a complex SIE is expressed in terms of a number of concepts and relations. The simple SIEs are rigid, whereas the complex SIEs are flexible. As previously discussed, a rigid SIE is an information element with a predefined fixed number of concepts/relations, whereas a flexible SIE has a varying number of concepts/relations depending on the instance at hand. Accordingly, in the ACC-tuple, an ontology concept, an ontology relation, and a deontic operator indicator are simple (and thus rigid) SIEs, whereas a restriction is a complex (and thus flexible) SIE. The use of flexible SIEs is key to providing the flexibility needed to facilitate full sentence analysis. A specific word, phrase, or chunk of text extracted and mapped according to a SIE is referred to as an "information element instance."

To prepare for further information transformation into logic sentences, a semantic mapping step is used to match the extracted information element instances to their respective semantic concepts: (1) for ontology concepts and relations, their information element instances are mapped to the corresponding concepts and relations; for example, "courts" is mapped to "court," "net area" is mapped to "net_area," "not less than" is mapped to "greater_than_or_equal"; (2) for deontic operator indicators, their instances are mapped to the indicated deontic concepts; for example, "shall" is mapped to "obligation"; and (3) for restrictions, their instances are decomposed and mapped to one or more ontology concepts and relations; for example, "between the insulation and the roof sheathing" is mapped to "relation(between, insulation, roof_ sheathing)."

The extracted information element instances (in ACC-tuple format)—after conducting necessary semantic mapping—are

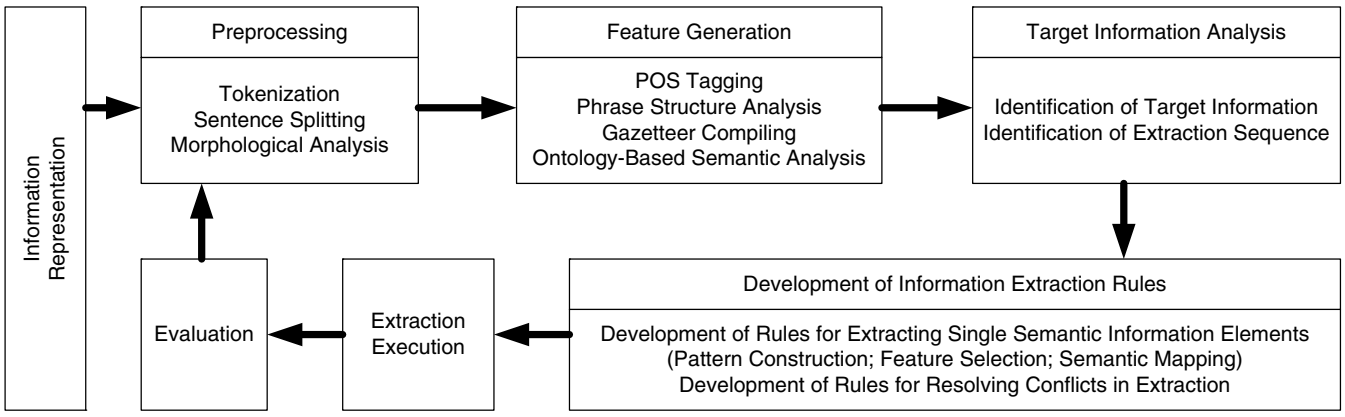

Fig. 2. Proposed information extraction methodology

J. Comput. Civ. Eng., 2016, 30(2): 04015014

Table 1. Example of Extracted Semantic Information Elements and Their Corresponding Logic Representation

Information tuple extracted from text sentences	Subject	airspace
	Subject restriction	relation (between, insulation, roof_sheathing)
	Compliance checking attribute	N/A
	Deontic operator indicator	obligation
	Quantitative relation	provide
	Comparative relation	greater_than_or_equal
	Quantity value	1
	Quantity unit/reference	inch
	Quantity restriction	N/A
Horn clause logic representation		∀ (a, i, r, s) [airspace(a) ∧ insulation(i) ∧ roof_sheathing(r) ∧ between(a, i, r) ∧ has(a, s)] - > *O* {greater_than_or_equal[s, quantity(1, in.)]}

Note: Universal quantifier ('∀' or 'for all') asserts that the sentence is true for all instances of a variable; conjunction '∧': 'A∧B' indicates that 'A' is true and 'B' is true; implication '->': 'A - > B' indicates that 'A' implies 'B' (if 'A' is true then 'B' is true); obligation operator (*O*): *O* A indicates that 'A' is obligated.

further transformed to Horn-Clause-type logic sentences (as shown in Table 1) for logic-based deduction and reasoning about compliance. The methodology/algorithms for information transformation will be presented in future work.

Phase II—Preprocessing

This phase is used to prepare the raw (i.e., unprocessed) text for further processing. In the proposed methodology, preprocessing consists of tokenization, sentence splitting, dehyphenation, and morphological analysis.

Tokenization

Tokenization is the process of dividing the sequences of characters (pure strings) in the text into units (sentences or words) (Grefenstette and Tapanainen 1994). This process aims to prepare the text for further unit-based processing, such as sentence splitting and POS tagging, and is conducted based on parsing the text according to common delimiters (i.e., white spaces and punctuations) with disambiguation consideration (e.g., "," as a delimiter in a number instead of punctuation). In the proposed methodology, tokenization divides the sequences of characters into tokens, where a token is a single word, a number, a punctuation mark, a white space, or a symbol (e.g., "&" and "$"). For example, as shown in Fig. 3, each word, number, and punctuation mark was recognized and labeled as a token.

Sentence Splitting

Sentence splitting is the process of recognizing each sentence of the text. Similar to tokenization, sentences are recognized based on typical sentence boundaries (i.e., periods, exclamation marks, and question marks) with disambiguation consideration (e.g., recognizing "." as a decimal point in a number instead of a period). In the proposed methodology, the result of sentence splitting is a set of sentence segmentations (with recognized boundaries). For example, as shown in Fig. 3, the boundaries of the sentence were recognized and labeled out using the "<sentence>" (i.e., starting of a sentence) or "</sentence>" (i.e., ending of a sentence) tags.

Morphological Analysis

Morphology refers to the study of composition and structure of words. Morphological analysis (MA) aims to recognize the different forms of a word and to map them to the lexical form of that word in a dictionary (Fautsch and Savoy 2009). MA maps various nonstandard forms of a word (e.g., plural form of noun, past tense of verb) to its lexical form (e.g., singular form of noun, infinitive form of verb). For example, "constructs," "constructed," and "constructing" are all mapped to "construct." Additionally, as shown in Fig. 3, "rooms" and "feet" were mapped to their lexical forms "room" and "foot," respectively. Whereas tokenization and sentence splitting are essential for IE because the text must be broken down into units for further processing, MA is not essential for IE but is used to improve the identification of words with the same lexical form. The proposed preprocessing methodology incorporates MA because it aids in the recognition of ontology concepts. For example, the plural form of a concept could be recognized although the ontology uses only the singular form.

Dehyphenation

Dehyphenation is used to remove hyphens that indicate continuations of words across two lines. Doing so prevents a word from not being recognized because of such a hyphen.

Phase III—Feature Generation

This phase generates a set of features that describe the text. The proposed methodology uses domain-specific ontology-based semantic features, in addition to syntactic features and proposes the use of PSG-based phrasal tags to reduce the number of needed patterns. The proposed feature generation methodology consists of POS tagging, phrase structure analysis (using PSG), gazetteer compiling, and ontology-based semantic analysis. Syntactic features, such as POS tags, are widely used for IE, as in Afrin (2001). Semantic features benefit IE tasks beyond solely using syntactic features because they express domain-specific meaning/knowledge, as in Soysal et al. (2010). In the proposed methodology, both syntactic (POS tags, PSG-based phrasal tags, gazetteer terms) and semantic features (concepts and relations) are generated; subsequently, these features are used to define patterns (text patterns in the proposed IE and CR rules that aid in the process of pattern matching for IE).

Part-of-Speech Tagging

Part-of-speech (POS) tags are the labels assigned to words of a sentence that indicate their lexical and functional categories showing the structure inherent in the language. POS tagging aims to tag each word with the POS of the word, such as NN (singular nouns), JJ (adjectives), VB (verb), and CC (coordinating conjunctions) (Galasso 2002). For example, as shown in Fig. 3, "floor," "Habitable," and "have" were tagged as NN, JJ, and VB, respectively. In the proposed methodology, the POS tagging process also tags other tokens, such as numbers, punctuations, and symbols.

Phrase Structure Analysis

The proposed phrase structure analysis builds on the POS tagging step and aims to assign type labels (phrasal tags) to phrases of a sentence. Examples of phrasal tags are NP (noun phrase), VP

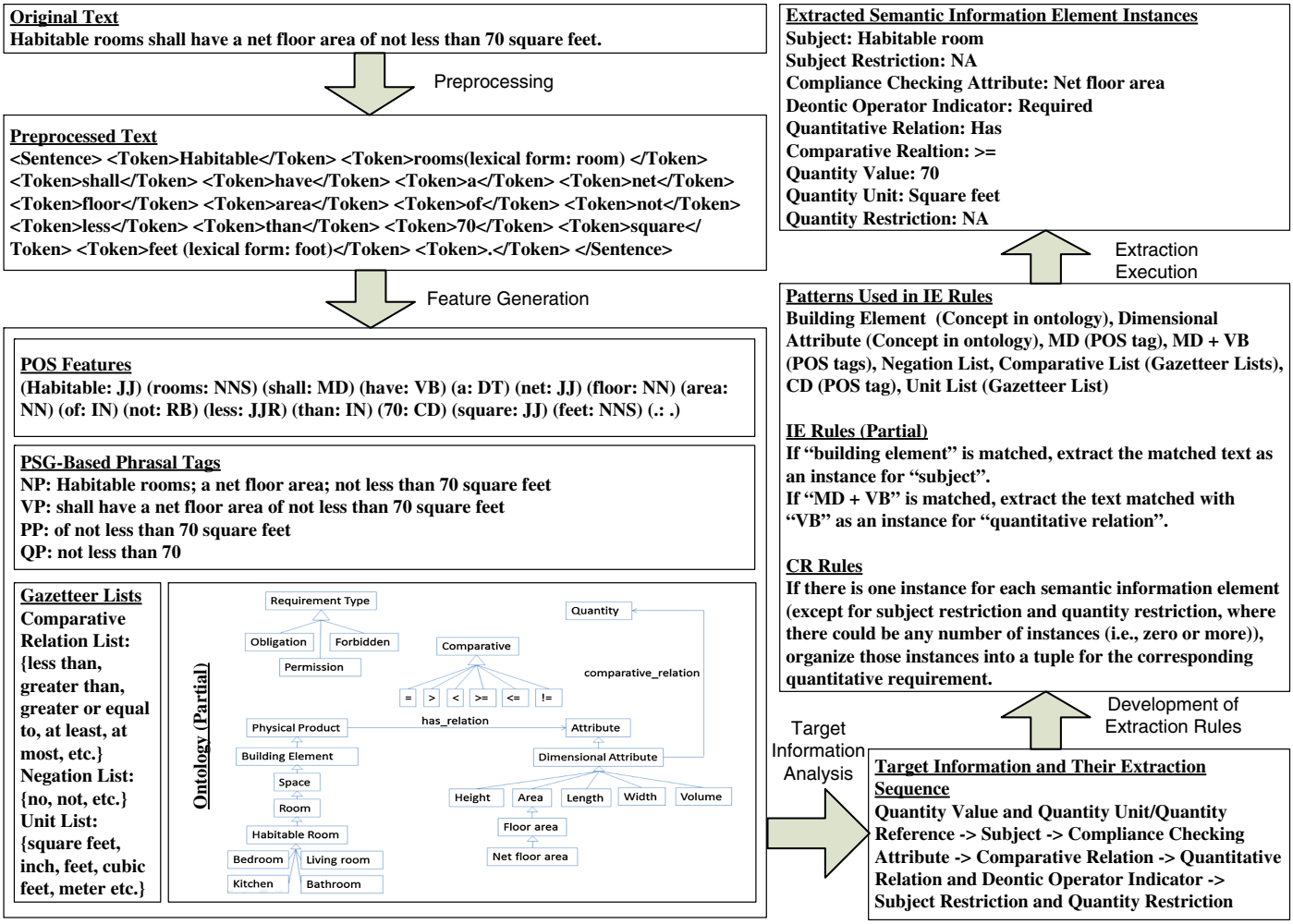

Fig. 3. Illustrative example applying proposed information extraction methodology

(verb phrase), and PP (prepositional phrase). For example, as shown in Fig. 3, "Habitable rooms," "shall have a net floor area of not less than 70 ft^2," and "of not less than 70 ft^2" were assigned NP, VP, and PP tags, respectively. In the proposed methodology, PSG is used to generate phrasal tags. In the methodology, application-specific PSG rules are derived based on a randomly selected sample of text (called, here, "development text," which is also used for text analysis and further development of IE and CR rules). By applying these PSG rules, phrasal tags are assigned when a certain combination of POS tags and/or phrasal tags are encountered. For example, the rule "QP→JJR IN CD" states that the phrasal tag "QP" (quantifier phrase) should be assigned when the sequence of POS tags "JJR IN CD" is encountered, as in the phrase, "less (JJR) than (IN) 0.07 (CD)." The use of phrasal tags together with PSG reduces the possible number of enumerations in patterns. For example, the three PSG rules NP→NP PP; NP→DT NN; and PP→IN NP together enable the phrasal tag feature NP to match many (actually an infinite number of) noun phrases expressed by recursively attaching prepositional phrases to a base noun, such as "the wall," "the wall of the room," "the wall of the room in the building," "the wall of the room in the building with a vent," and "the wall of the room in the building with a vent at the bottom." In this step, PSG is derived from previously POS-tagged source text and is subsequently used to assign PSG-based phrasal tags to sentences in the source text.

To empirically study the effect of utilizing PSG-based phrasal tags on the number of patterns, an experimental test was conducted for preliminary verification of the proposed methodology. The authors developed the patterns for extracting "subjects" two times: one time with PSG-based phrasal tags and one time without. Twenty-two (22) and 46 patterns were needed, with and without PSG-based phrasal tags, respectively, indicating that the use of PSG-based phrasal tags in pattern construction reduces the number of needed patterns in IE rules.

Gazetteer Compiling

A gazetteer is a set of lists containing names of specific entities (e.g., cities, organizations) (Cunningham et al. 2011). In general, a gazetteer list groups any set of terms based on any specific commonality possessed by these terms. In the proposed methodology, the information that a word or phrase belongs to a certain list in the gazetteer is used as a feature for IE tasks. Different gazetteer lists are available [e.g., lists for currency, data units, and cities in the ANNIE (A Nearly-New Information Extraction System) Gazetteer of the GATE (General Architecture for Text Engineering)]. The use of a gazetteer in automated IE aids in recognizing terms based on those commonalities (Maynard et al. 2004). In the proposed methodology, a gazetteer is used to provide a set of term lists, in which each list has a specific function. For example, terms such as "no" and "not" have a function "negation," and, as such, are included in the authors' "negation gazetteer list." In the proposed methodology, several types of gazetteer lists are compiled and used, such as the "comparative relation gazetteer list," which is composed of terms

J. Comput. Civ. Eng., 2016, 30(2): 04015014

Table 2. Comparative Testing of Syntactic-Only IE and Semantic IE: Experimental Results for Section 1203 of Chapter 12 of IBC 2006

Performance measure	Syntactic-only IE	Semantic IE
Precision	0.85	0.96
Recall	0.81	0.92
F-measure	0.83	0.94

indicating comparative relations, including "greater or equal," "less or equal," "at most," and "at least." For example, as shown in Fig. 3, "not," "less than," and "square feet" were in the "negation gazetteer list," "comparative relation gazetteer list," and "unit gazetteer list," respectively. The information presented in a gazetteer list could have been represented as part of an instantiated ontology (e.g., the list of countries could have been represented as instances of the concept "country"). However, for computational efficiency, such instances were separated from the ontology (in the form of gazetteer lists).

Ontology-Based Semantic Analysis

Ontologies are used to represent domain knowledge. A construction domain ontology offers a semantic representation of the knowledge in the construction domain and, thus, could aid in extracting relevant information based on domain-specific meaning. In the proposed methodology, the concepts and relations of an ontology help extract the semantic features of the text and, thus, in semantic IE. Fig. 3 shows a partial (and schematic) view of the used ontology, including its concepts (e.g., dimensional attribute) and subconcepts (e.g., floor area).

To verify the selection of a semantic approach, by comparing semantic IE results to that of syntactic-only IE, the authors conducted an experiment on extracting quantitative requirements from a randomly selected section of Chapter 12 of IBC 2006—Section 1203. Table 2 shows the comparative results in terms of precision, recall, and F-measure. The results show that semantic IE outperforms syntactic-only IE, with an increase in precision from 0.85 to 0.96 and an increase in recall from 0.81 to 0.92.

Phase IV—Target Information Analysis

This phase is proposed to manually analyze the text to identify the types of semantic information elements to be extracted and their interrelationships, and the sequence of their extraction. In the proposed methodology, an approach for separation and sequencing of semantic information elements (SSSIE) is proposed to reduce the number of needed IE patterns.

Identification of Target Information

In this step of the methodology, the development text is manually analyzed to identify the types of requirements that are expressed in the text (e.g., quantitative requirement). Based on domain knowledge (expressed in the ontology), the types of semantic information elements that are needed to represent the types of requirements are defined. For example, if the information to be extracted is related to terrorist attack events, then the types of semantic information elements could include "perpetrator individual," "perpetrator organization," "target," "victim," and "weapon." For the example in Fig. 3, the information to be extracted is related to quantitative requirements, so the authors identified the following types of semantic information elements: "subject," "compliance checking attribute," "deontic operator indicator," "quantitative relation," "comparative relation," "quantity value," "quantity unit," "quantity reference," "subject restriction," and "quantity restriction."

Identification of Extraction Sequence

This step identifies the sequence of extracting the semantic information elements. The experimental studies of this research showed that extracting all semantic information elements from a sentence using a single IE rule (i.e., extracting all instances at the same time) is not efficient because the amount of possible patterns increases largely as the number of semantic information elements increases. Because some independency exists (but not fully independent) among information elements, extracting information elements separately and sequentially is proposed. The decision regarding the sequence of extraction for different semantic information elements is based on manually analyzing the text and identifying (1) the level of difficulty for extraction: the easiest semantic information element should be extracted first and the level of difficulty is positively correlated to a combination of the amount of features, the amount of patterns, and the complexity of the patterns; and (2) the existing dependencies across the extractions of the different semantic information elements. For example, (1) if the extraction of "quantity value" only needs the POS tag "CD" as the feature for recognizing cardinal numbers (both appearances of digits and words) and the level of difficulty for its extraction is lowest, then it should be extracted first; and (2) if the extraction of "subject restriction" depends on the extraction of "subject," then "subject" should be extracted before "subject restriction." For the example in Fig. 3, the sequence of extraction of semantic information elements was "quantity value" and "quantity unit/quantity reference" > "subject" > "compliance checking attribute" > "comparative relation" > "quantitative relation" and "deontic operator indicator" > "subject restriction" and "quantity restriction."

To verify the proposed approach for separation and sequencing of semantic information elements (SSSIE), an experiment was conducted to compare the performance results of two cases. In the first case, IE rules that extract all semantic information elements from a sentence using a single IE rule (i.e., extracting all instances at the same time) were developed and used. In the second case, the proposed method for SSSIE in IE was used. For both cases, the IE rules were developed based on Chapter 12 and 23 of IBC 2006 and were tested using Chapter 19 of IBC 2009. Eighty-seven (87) and 50 patterns were needed for the first and second cases, respectively, indicating that using the proposed SSSIE method reduces the number of needed patterns in IE rules. Table 3 shows the comparative results in terms of precision, recall, and F-measure. The results show significantly stronger performance using SSSIE (the second case). The weaker performance in the first case may be partially attributed to (1) the fact that enumerating all possible patterns based on a limited development text is difficult (if not impossible); and (2) an error in recognizing a single semantic information element in a given IE rule affects the extraction result of the entire IE rule (and, thus, all other information elements in that rule).

Phase V—Development of Information Extraction Rules

In this phase, a set of rules are developed to automatically execute the information extraction process. The proposed methodology includes the development and use of two types of rules: rules for extracting single semantic information elements (IE rules) and rules for resolving conflicts in extraction (CR rules). The IE rules recognize target information for extraction, whereas the CR rules define the strategy for handling conflicts in extraction.

Development of Rules for Extracting Single Semantic Information Elements (IE Rules)

The extraction rules (IE rules) utilize pattern matching methods. The left-hand side of the rule defines the pattern to be matched and the right-hand side defines the part of the matched pattern that

Table 3. Comparative Testing of IE Using or Not Using Separation and Sequencing of Semantic Information Elements (SSSIE): Experimental Results for Chapter 19 of IBC 2009

Number of instances	Subject	Compliance checking attribute	Comparative relation	Quantity value	Quantity unit/reference	Total
In gold standard	85	45	85	83	85	383
Extracted with SSSIE	85	46	79.5	83	83	376.5
Extracted without SSSIE	55	30	59.5	64	63.5	272
Correctly extracted with SSSIE	80	43	79.5	81	81	364.5
Correctly extracted without SSSIE	48	27	59.5	62	61.5	258
Precision with SSSIE	0.941	0.935	1.000	0.976	0.976	0.968
Precision without SSSIE	0.873	0.900	1.000	0.969	0.969	0.949
Recall with SSSIE	0.941	0.956	0.935	0.976	0.953	0.952
Recall without SSSIE	0.565	0.600	0.700	0.747	0.724	0.674
F-measure with SSSIE	0.941	0.945	0.967	0.976	0.964	0.960
F-measure without SSSIE	0.686	0.720	0.824	0.844	0.828	0.788

should be extracted. Both syntactic (POS tags, PSG-based phrasal tags, and gazetteer terms) and semantic (ontology concepts and relations) text features are used in the IE rules patterns. If a concept in the ontology is used in an IE rule, all of its subconcepts are included in the matching as well. For example, in the following IE rule, "building element" is a concept in the ontology: "If "building element" is matched, extract the matched text as an instance for "subject"." When applied to the example in Fig. 3, this IE rule extracts "habitable rooms" as an instance of "subject" because "habitable room" matches "Habitable_Room" (a subconcept of "building element" in the ontology)." Fig. 4 shows a sample IE rule (in English) and its corresponding Java coding (using Java Annotation Patterns Engine (JAPE) rules in GATE).

To develop these IE rules, the following three tasks are proposed: pattern construction, feature selection, and semantic mapping. For pattern construction, the patterns take the format of a sequential combination of features (e.g., the pattern "NP VP" matches a sentence, as in Fig. 1). The construction of such patterns is an iterative, empirical process (using initial manual text analysis, initial pattern construction, testing and results analysis, and testing-based improvement of constructed patterns). Feature selection aims to select all features present in the constructed patterns. In semantic mapping, the extracted information element instances are mapped to their semantic counterparts. For example, as shown in Fig. 3, the pattern "MD VB" (i.e., POS tags for "modal verb" "verb") was constructed for the extraction of "quantitative relation," POS tags were selected as features, "shall have" matched this pattern, "have" was semantically mapped to "has," and "has" was accordingly extracted as a "quantitative relation" instance.

Development of Rules for Resolving Conflicts in Extraction (CR Rules)

In the proposed methodology, the rules for resolving conflicts in extraction (conflict resolution (CR) rules) primarily address the following four types of conflict cases: (1) the number of information element instances of a semantic information element in a single sentence is more than the required, (2) the number of information element instances of a semantic information element in a single sentence is less than the required, (3) there is overlap of extraction results for different semantic information elements, and (4) no conflicts, the number of information element instances of a semantic information element in a single sentence is equal to the required. Each type of conflict case may be handled using one of a set of actions. For conflict case 1, one of the following two actions may be used: (1) keep all information element instances; or (2) set priority rules and select the information element instances with higher priority (e.g., set a higher priority for "not less than" comparing with "above" when encountering multiple comparative relation instances. For example, in the sentence part "nonabsorbent surface to a height not less than 70 in. above the drain inlet," the comparative relation instance extracted is only "not less than," although both "not less than" and "above" are recognized as candidate comparative relation instances). For conflict case 2, one of the following three actions may be used: (1) set a default information element

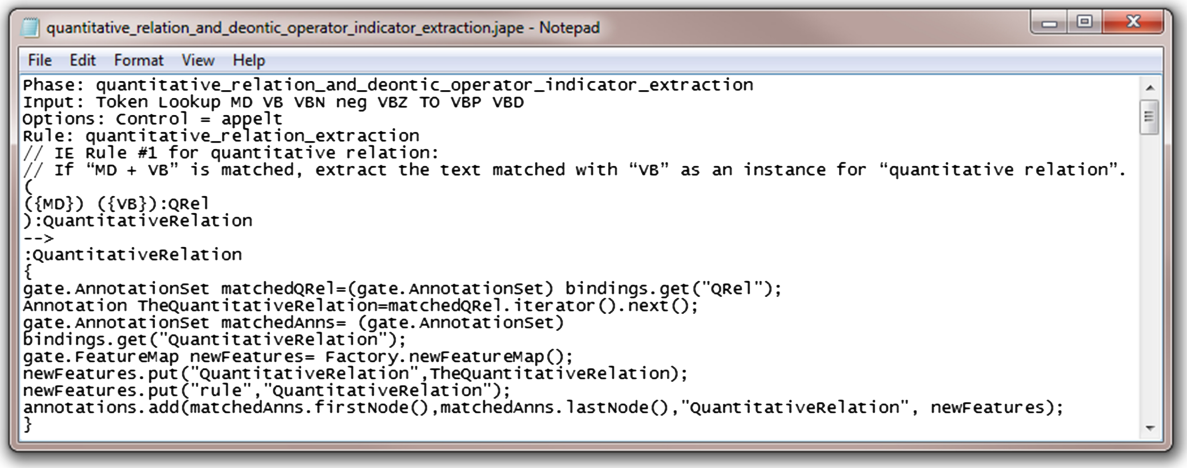

Fig. 4. Sample information extraction rule (in English and Java coding)

J. Comput. Civ. Eng., 2016, 30(2): 04015014

instance based on domain knowledge (e.g., the default comparative relation instance may be set to "greater_than_or_equal" when no information element instance is extracted. For example, in the sentence "The outside horizontal clear space measured perpendicular to the opening shall be one and one half times the depth of the opening," the default "greater_than_or_equal" is used as a comparative relation instance); (2) use the same instance from the nearest sentence/clause (left or right) if those sentences/clauses describe the same content (e.g., in the sentence "The openable area between the sunroom addition or patio cover and the interior room shall have an area of not less than 8 percent of the floor area of the interior room or space, but not less than 20 ft^2," the subject of the first quantitative relation should also be used for the second quantitative relation); or (3) drop this sentence. For conflict case 3, one of the following three actions may be used: (1) delete all overlapping information element instances and keep only the required number, (2) keep all information element instances, or (3) delete some overlapping information element instances and keep more than the required number. For conflict case 4, one action is used: organize all extracted information element instances into a tuple to describe the corresponding requirement. For example, as shown in Fig. 3, the following CR rule (a conflict case 4) was applied: if one instance exists for each semantic information element (except for subject restriction and quantity restriction, for which the number of instances could be zero or more), organize those instances into a tuple for the corresponding quantitative requirement. For each case, defining which one of the actions should be executed is determined based on the type of conflict pattern. For example, if the subject of a quantitative requirement is a "space," then the comparative relation is usually "greater_than_or_equal" when missing. The conflict patterns and corresponding actions are encoded as CR rules.

Phase VI—Extraction Execution

This phase aims to extract the target information element instances from the regulatory text using the rules developed in Phase V. For example, as shown in Fig. 3, "habitable room" and "net floor area" were extracted as instances of "subject" and "compliance checking attribute," respectively.

Phase VII—Evaluation

Evaluation is conducted by comparing the extracted information with a "gold standard." The "gold standard" includes all instances of the target information in the regulatory text source and is manually (or semiautomatically with the help of NLP tools) compiled by domain experts. Evaluation is conducted using the following measures: precision, recall, and F-measure. Precision is defined as the percentage of correctly extracted information element instances relative to the total number of information element instances extracted [Eq. (1)]. Recall is defined as the percentage of correctly extracted information element instances relative to the total number of information element instances existing in the source text [Eq. (2)]. A trade-off exists between precision and recall; using either indicator alone is not sufficient. Thus, F-measure is defined as a weighted combination (harmonic mean) of precision and recall (Makhoul et al. 1999) [Eq. (3)]. In the proposed methodology, α is set to 0.5 to give equal weights to recall and precision. If the evaluation results are satisfactory (e.g., the F-measure is greater than 0.9 or a specific value defined by the user), the process may be terminated and the rules (i.e., IE and CR rules) may be considered as final. If the evaluation results are not satisfactory, the phases may be iterated for performance improvement. Performance improvements in later iterations may be achieved by addressing extraction errors in earlier iterations.

$$P = \frac{\text{number of correct information element instances extracted}}{\text{total number of information element instances extracted}} \quad (1)$$

$$R = \frac{\text{number of correct information element instances extracted}}{\text{total number of information element instances existing}} \quad (2)$$

$$F = \frac{P \times R}{(1-\alpha) \times P + \alpha \times R}, \quad \text{where } 0 \le \alpha \le 1 \quad (3)$$

Validation: Experiments and Results

An experiment was conducted to validate the proposed algorithms. Evaluating the algorithms (in terms of precision and recall) and achieving satisfactory performance implies the validity of the proposed approach and methodology. Quantitative requirements were extracted from randomly selected chapters of IBC 2006 and 2009. The IE performance of the algorithms was evaluated by comparing the extraction results against a semiautomatically (using NLP tools) developed gold standard.

Source Text Selection (International Building Code)

The proposed methodology is intended to extract information from a variety of construction-related regulatory documents (e.g., building codes, environmental regulations, safety regulations and standards). At this phase, the authors tested the proposed algorithms on building codes. IBC was selected because it is the most widely adopted building code in the United States. IBC 2006 (ICC 2006) and IBC 2009 (ICC 2009) were used, Chapters 12 and 23 of IBC 2006 were randomly selected for development and Chapter 19 of IBC 2009 was randomly selected for testing. The following two main types of requirements in IBC were identified: (1) "quantitative requirement," which defines the relationship between an attribute of a certain building element/part and a specific quantity value (or quantity range); for example, "Occupiable spaces, habitable spaces and corridors shall have a ceiling height of not less than 7 ft 6 in. (2,286 mm)" states that the "ceiling height" attribute of these spaces should be greater than or equal to 7'6"; and (2) "existential requirement," which requires the existence of a certain building element/part; for example, "The unit (efficiency dwelling unit) shall be provided with a separate bathroom containing a water closet, lavatory and bathtub or shower" states that an efficiency dwelling unit should have a bathroom with water closet, lavatory, and bathtub or shower. The decision was made to experiment with the extraction of quantitative requirements because (1) most of the requirements identified in these chapters are quantitative requirements; and (2) the sentences describing quantitative requirements appear more complex than those describing existential requirements, implying that they are more difficult to extract.

Ontology Development

An application-oriented and domain-specific ontology for buildings was developed. In developing the ontology, existing construction ontologies [e.g., the IC-PRO-Onto (El-Gohary and El-Diraby 2010)] and IFC (Industry Foundation Classes) (IAI 2007) concepts were reused as necessary. The ontology was coded in OWL (Web Ontology Language), i.e., *.owl format, because OWL is the most widely used semantic Web language.

Information Representation

For building codes, a nine-tuple format was used for intermediate information representation: <Subject, Subject Restriction, Compliance Checking Attribute, Deontic Operator Indicator, Quantitative Relation, Comparative Relation, Quantity Value, Quantity Unit/Reference, Quantity Restriction>. Following the semantic model of ACC as presented in the authors' previous work (Salama and El-Gohary 2013b), the semantic information elements are defined as follows [for further elaboration on the semantic model, including these concepts, the reader is referred to Salama and El-Gohary (2013b)]. A "subject" is an ontology concept; it is a "thing" (e.g., building object, space) that is subject to a particular regulation or norm. A "compliance-checking attribute" is an ontology concept; it is a specific characteristic of a "subject" by which its compliance is assessed. A "deontic operator indicator" is an indicator; it matches to (or indicates) the type of deontic modal operator (i.e., obligation represented by *O*, permission represented by *P*, and prohibition represented by *F*) applicable to the current requirement. A "quantitative relation" defines the type of relation for the quantity. For example, in the sentence "The court shall be increased 1 ft in width and 2 ft in length for each additional story," the quantitative relation is "increase," which semantically describes that the relation between "width of the court" and "1 foot" is "increased for each additional story." A "comparative relation" is a relation, such as greater_than_or_equal, less_than_or_equal, or equal, commonly used to compare quantitative values (i.e., comparing an existing value with a required minimum or maximum value). A "quantity value" is a value or a range of values that defines the quantified requirement. A "quantity unit" is the unit of measure for the "quantity value." A "quantity reference" is a reference to another quantity (which presumably includes a value and a unit). For example, in the sentence "The bearing area of headed anchors shall be not less than one and one-half times the shank area," "shank_area" is the "quantity reference." A "subject restriction" (and, similarly, "quantity restriction") places a constraint on the definition of a "subject" (or "quantity"), such as by defining the properties of the "subject" (or "quantity)."

Each extracted requirement (1) has one and only one instance of each of the following semantic information elements: subject, comparative relation, quantity value, and quantity unit/reference; (2) has at most one instance of each of the following semantic information elements: compliance checking attribute, deontic operator indicator, and quantitative relation; and (3) has zero, one, or more instances of each of the following semantic information elements: subject restriction and quantity restriction. Table 4 shows examples of the nine-tuple representation.

Development of Gold Standard

The gold standard was developed semiautomatically. First, all sentences that include a number (the appearance of both digit form and word form of a number to ensure 100% recall of sentences describing quantitative requirements) were extracted automatically. Subsequently, one of the authors manually deleted false positive sentences and identified all semantic information element instances for each sentence. The gold standard was reviewed by two other researchers and adjusted, if needed. In Chapters 12 and 23 of IBC 2006, 304 sentences containing quantitative requirements were recognized, forming the gold standard.

Tool Selection (GATE)

Many off-the-shelf tools are available today to support various NLP tasks including IE, such as the Stanford Parser by the Stanford NLP Group and GATE by the University of Sheffield. GATE was selected to implement the IE algorithms because (1) GATE has been widely and successfully used in IE, such as in Soysal et al. (2010); and (2) it embeds many other NLP tools in the form of plug-ins, such as the Stanford Parser and OpenNLP tools. The following built-in GATE tools were utilized in the experiments: (1) ANNIE system for tokenization, sentence splitting, POS tagging, and gazetteer compiling; (2) the built-in morphological analyzer for morphological analysis; (3) the built-in ontology editor for ontology building and editing; and (4) JAPE transducer for writing the IE and CR rules.

Applying the IE Methodology

The IE and CR rules were developed based on Chapters 12 and 23 of IBC 2006 and were subsequently tested on Chapter 19 of IBC 2009. The ANNIE Hepple POS Tagger was used to generate POS tag features (Table 5 provides a sample). A total of 53 POS tag symbols exist in the set of Hepple POS tags used. The Penn Treebank phrasal tag labels were used for phrase structure analysis. The following three gazetteer lists were compiled: comparative relation list, unit list, and negation list. In addition, the GATE built-in gazetteer lists of numbers and ordinals were used. Table 6 shows the number of patterns, features, and CR rules for Chapters 12 and 23 of IBC 2006. The IE and CR rules (developed based on Chapters 12 and 23 of IBC 2006) are intended to support automated extraction of quantitative requirements from any construction regulatory documents/text. The rules were applied to Chapter 19 of IBC 2009 for testing and evaluation.

Table 4. Examples of Semantic Information Elements and Information Element Instances

Semantic information element	Extracts of example sentence 1	Extracts of example sentence 2	Extracts of example sentence 3
Requirement	A minimum of 1 in. of airspace shall be provided between the insulation and the roof sheathing	The minimum net area of ventilation openings shall not be less than 1 ft^2 for each 150 ft^2 of crawl space area	Courts shall not be less than 3 ft in width
Subject	airspace	ventilation_opening	court
Subject restriction	relation(between, insulation, roof_sheathing)	N/A	N/A
Compliance checking attribute	N/A	net_area	width
Deontic operator indicator	obligation	obligation	obligation
Quantitative relation	provide	N/A	N/A
Comparative relation	greater_than_or_equal	greater_than_or_equal	greater_than_or_equal
Quantity value	1	1	3
Quantity unit/reference	inch	square_foot	feet
Quantity restriction	N/A	relation(for_each, 150, square_feet, crawl_space_area)	N/A

Table 5. Sample POS Tags and Phrasal Tags

Part of speech tag/phrasal tag	Meaning
ADVP	Adverb phrase
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
IN	Prepositional or subordinating conjunction
JJR	Comparative adjective
MD	Modal verb
NN	Singular or mass noun
NNS	Plural noun
NP	Noun phrase
PP	Prepositional phrase
QP	Quantifier phrase
RB	Adverb
VB	Base form verb
VP	Verb phrase

Additionally, the IE and CR rules are potentially reusable in extracting quantitative requirements from other types of documents/text. They may be reused as is or adapted/extended based on additional development text. To test the potential reusability of the IE and CR rules developed, they were applied (as is, without any modification) to a different type of text. The following document was randomly selected from the Web, with the only criterion being that the document contains quantitative requirements: "Procedures (Section 700.4) in traffic cabinet ground rod specifications." The rules were used to extract quantitative requirements from the randomly selected text, and performance was evaluated against a manually developed gold standard. Table 7 shows the results in terms of precision, recall, and F-measure. As per Table 7, the overall F-measure is greater than 0.90, indicating potential reusability of the rules.

Results and Discussion

Table 8 summarizes the information extraction results. For Chapter 19 of IBC 2009, on average, 0.969, 0.944, and 0.956 precision, recall, and F-measure, respectively, were achieved. When calculating the precision and recall for "subject restriction" and "quantity restriction" instances, the correctness of extracting one restriction instance is calculated as a ratio of the number of correctly extracted concepts and relations to the total number of concepts and relations in that restriction (because each restriction instance may include multiple concepts and relations). When calculating the precision and recall for "comparative relation" instances, partial extraction correctness for the following comparative relations was considered: "greater than or equal" and "less than or equal." For example, in the following case, the instance was calculated as "half-correctly extracted," i.e., 0.5: "above" (greater_than) was extracted, whereas the gold standard included "at or above" (greater_than_or_equal).

Although only "subject restriction," "comparative relation," and "quantity restriction" showed a perfect performance value (1.00 for precision), all precision and recall values were greater than or equal to 0.90 except for the recall of "subject restriction."

Error analysis resulted in five findings. First, the reasons for the relatively low recall of "subject restriction" are as follows: (1) the patterns are more complex; for example, one pattern for "subject restriction" typically involves several phrases, whereas one pattern for other elements such as "subject" could be as simple as corresponding to just one concept in the ontology; and (2) the number of instances for "subject restriction" used in rule development is significantly less (at least 30% less) than that for other types of semantic information elements. Second, errors in the extraction of "subject" are the result of inner errors of the tools used; for example, GATE failed to recognize the term "connection" although it exists in the ontology. No existing NLP tool achieves 100% performance, even for relatively simple NLP tasks such as POS tagging, and any error in POS tagging, for example, may further cause an error in information extraction because the IE rules include POS-features in its patterns. Third, the errors in the extraction of "compliance checking attribute" are the result of inner errors of the tools used and the limitations of CR rules. For example, one CR rule states that if no "compliance checking attribute" was extracted and extra "subject" candidates were extracted, then place the "subject" candidate that is closest to the "quantity value" as the attribute. This rule led to an incorrect extraction of "clearance" as the compliance checking attribute instance in the sentence "The steel reinforcement shall be in the form of rods, structural shapes or pipe embedded in the concrete core with sufficient clearance to ensure the composite action of the section, but not nearer than 1 in. to the exterior steel shell." Fourth, the errors in the extraction of

Table 6. Number of Patterns, Features, and CR Rules for Chapters 12 and 23 of IBC 2006

Number	Subject	Subject restriction	Compliance checking attribute	Deontic operator indicator	Quantitative relation	Comparative relation	Quantity value	Quantity unit/ reference	Quantity restriction
Extraction patterns	N/A	29	N/A	10	9	2	24	24	48
Features selected	10 (304)[a]	47	1 (99)[a]	8	7	5	28	31	60
CR rules	2	2	5	0	0	4	8	8	9

[a]Number in parenthesis represents sub-concepts.

Table 7. Testing Reusability of IE Rules and CR Rules

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	24	0	18	17	16	13	25	25	6	144
Extracted	24	0	18	17	17	17	24	24	7	148
Correctly extracted	21	0	17	17	11	13	24	24	6	133
Precision	0.875	N/A	0.944	1.000	0.647	0.765	1.000	1.000	0.857	0.899
Recall	0.875	N/A	0.944	1.000	0.688	1.000	0.960	0.960	1.000	0.924
F-measure	0.875	N/A	0.944	1.000	0.667	0.867	0.980	0.980	0.923	0.911

Table 8. Experimental Results for Chapter 19 of IBC 2009

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	85	18	45	48	58	85	83	85	15	522
Extracted	85	15	46	47	57	79.5	83	83	13.5	509
Correctly extracted	80	15	43	46	54	79.5	81	81	13.5	493
Precision	0.941	1.000	0.935	0.979	0.947	1.000	0.976	0.976	1.000	0.969
Recall	0.941	0.833	0.956	0.958	0.931	0.935	0.976	0.953	0.900	0.944
F-measure	0.941	0.909	0.945	0.968	0.939	0.967	0.976	0.964	0.947	0.956

"deontic operator indicator" and "quantitative relation" are the result of missing patterns in IE rules (which were missed because the patterns are not common) and limitations of CR rules. Fifth, the errors in the extraction of "comparative relation," "subject restriction," "quantity restriction," "quantity value," and "quantity unit/reference" are the result of missing patterns in IE rules.

In their future work, the authors will further explore how to improve the proposed IE and CR rules to avoid/reduce these errors and, consequently, improve the IE results. The problems of missing patterns and limitations of CR rules could be solved through the development/adjustment of IE and CR rules based on more corpuses. However, further exploration is required to find out how many more corpuses could be sufficient to produce enough patterns for IE rules and to avoid the current limitations of the CR rules—and whether the increase in development corpuses would result in significant improvement in precision and recall.

Limitations and Future Work

The experimental results show that the proposed approach is promising for automatically extracting information from construction regulatory documents. Despite the high performance achieved (0.969, 0.944, and 0.956 precision, recall, and F-measure, respectively), three limitations of the work are acknowledged, which the authors plan to address in their future/ongoing research. First, the proposed methodology/algorithms were only tested in extracting quantitative requirements. The types of patterns and extraction conflicts in other types of requirements (e.g., existential requirements) may vary and, as a result, IE performance may vary. In future work, the methodology/algorithms will be tested on other types of requirements such as existential requirements. Second, the proposed methodology/algorithms were only tested on one chapter, primarily because the development of the gold standard for testing is highly time intensive. As part of future/ongoing research work, the methodology/algorithms will be tested on more building code chapters. The results are expected to show similar high performance because the chapter used in testing contains large amounts of text (approximately 7,000 words) and because of the similarity in text across different chapters of building codes and across different types of building codes (e.g., "Building Code and Related Excerpts of the Municipal Code of Chicago" versus IBC 2006). However, the results may vary because of the possible variability in the syntactic and semantic text features across different chapters and/or codes. In that case, the proposed IE and CR rules may be adapted/extended based on additional development text. Third, the proposed methodology/algorithms were tested only on building codes. In future work, the proposed methodology/algorithms will be extended to extract information from other types of regulatory documents (e.g., environmental regulations) and contractual documents (e.g., contract specifications).

Contributions to the Body of Knowledge

This research is important from both intellectual and application perspectives. From an intellectual perspective, this research contributes to the body of knowledge in four primary ways. First, this research offers domain-specific, semantic NLP methods that can assist in capturing domain-specific meaning and shows that ontology-based semantic IE outperforms syntactic-only IE (in terms of precision and recall). Domain-specific semantics allow for the analysis of complex sentences that would otherwise be too complex for automated IE, the recognition of domain-specific text meaning, and in turn the improvement of IE performance. Second, this research offers relatively efficient-to-develop rule-based NLP methods that can benefit from expert NLP knowledge encoded in the form of IE and CR rules. This research shows that the efficiency of algorithm development for rule-based methods can be enhanced through the following two main techniques: (1) use of PSG-based phrasal tags, and (2) separation and sequencing of semantic information elements (SSSIE) during extraction. Both PSG-based phrasal tags and the SSSIE method reduce the number of patterns needed in IE rules, resulting in fewer IE rules for extraction being required and, thus, reduced human effort to develop IE rules. Third, this research shows that deep NLP can be successfully achieved if both domain knowledge (represented in the form of a domain ontology) and expert NLP knowledge (represented in the form of IE and CR rules) are captured and integrated in a single platform. The research shows that semantic, rule-based deep NLP can provide high IE performance results (0.969 and 0.944 precision and recall, respectively). Fourth, and most importantly, this study is the first in the AEC domain that addresses automated IE using a semantically deep NLP approach. It offers baseline semantic IE methods/algorithms for extracting information from textual construction documents. Future research could use these methods/algorithms as a benchmark and build on this work by adapting the developed algorithms to extract information from other types of documents (e.g., contract documents) or for different purposes (e.g., contract analysis). The IE rules, CR rules, and algorithms developed in this study are potentially reusable (as the experimental results showed). Compared with the authors' initial efforts, future efforts in adapting the rules and/or algorithms should be significantly lower. Once the rules/algorithms are adapted (if needed), the process of information extraction is fully automated.

The impact of applying this work in the AEC domain could be far-reaching. First, this work brings automated construction regulatory compliance checking one step closer to reality. Automated regulatory compliance checking would reduce the time, cost, and errors of the checking process, which could speed-up the regulatory process, enhance cost and time project efficiency, and lead to fewer violations of regulations. Second, the application of this work could be extended to support automated information extraction and analysis for many other applications and purposes, such as analysis of contract documents to detect inconsistencies, analysis of project

documents and records to support claim analysis, and analysis of daily site reports to support progress monitoring and project control.

Conclusions

This paper presented a semantic, rule-based NLP methodology/algorithm for automated IE from construction regulatory documents to support automated compliance checking. A set of pattern-matching-based IE rules and CR rules are used in IE. The patterns are represented in terms of syntactic and semantic text features. NLP techniques are utilized to capture the syntactic features of the text, and a domain ontology is used to capture the semantic ones. PSG-based phrasal tags are used in syntactic analysis to reduce the number of needed patterns. Information elements are extracted separately and sequentially to further limit the number of needed patterns. The information extraction is relatively deep; it aims to achieve full sentence analysis to extract all information of a requirement for further representation in a logic-based rule format. The proposed algorithms were tested in extracting quantitative requirements from IBC 2009. A comparison of the extracted information element instances with those in a semiautomatically developed gold standard showed an average precision and recall of 0.969 and 0.944, respectively. These high performance results indicate that the proposed IE approach is promising. An error analysis also pinpointed the sources of errors in the experimental results and identified potential solutions for the possibility of further performance enhancement. As part of their future/ongoing research, the authors will test the proposed methodology/algorithms on other types of requirements (e.g., existential requirements), other types of building codes (e.g., Municipal Code of Chicago), other types of construction regulatory documents (e.g., EPA regulations), and other types of construction domain documents (e.g., contractual documents such as contract specifications). The results are expected to show similar high performance. However, variations in the results may occur as a result of the possible variability in the syntactic and semantic text features across different requirements, chapters, codes, or documents.

Acknowledgments

The authors would like to thank the National Science Foundation. This material is based upon work supported by the National Science Foundation under Grant No. 1201170. 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 National Science Foundation.

References

Abuzir, Y., and Abuzir, M. O. (2002). "Constructing the civil engineering thesaurus (CET) using ThesWB." *Proc., Int. Workshop on Information Technology in Civil Engineering 2002*, ASCE, Reston, VA, 400–412.

AEC3. (2012). "International Code Council." ⟨http://www.aec3.com/en/5/5_013_ICC.htm⟩ (Feb. 12, 2014).

Afrin, T. (2001). "Extraction of basic noun phrases from natural language using statistical context-free grammar." M.Sc. thesis, Virginia Polytechnic Institute and State Univ., Blacksburg, VA.

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.

Avolve Software Corporation. (2011). "Electronic plan review for building and planning departments." ⟨http://www.avolvesoftware.com/index.php/solutions/building-departments/⟩ (Jul. 15, 2011).

Bing, L., Lam, W., and Wong, T. (2013). "Wikipedia entity expansion and attribute extraction from the web using semi-supervised learning." *Proc., 6th ACM Int. Conf. Web Search and Data Mining (WSDM '13)*, Association for Computing Machinery, New York, 567–576.

Boken, P., and Callaghan, G. (2009). *Confronting the challenges of manual journal entries*, Protiviti, Alexandria, VA, 1–4.

Caldas, C. H., and Soibelman, L. (2003). "Automating hierarchical document classification for construction management information systems." *Autom. Constr.*, 12(4), 395–406.

Cherpas, C. (1992). "Natural language processing, pragmatics, and verbal behavior." *Anal. Verbal Behav.*, 10(1992), 135–147.

Chomsky, N. (1956). "Three models for the description of language." *IEEE Trans. Inform. Theor.*, 2(3), 113–124.

City of Mesa. (2012). "Construction plan review." *Official website of the City of Mesa, Arizona*, ⟨http://www.mesaaz.gov/devsustain/PlanReview.aspx⟩ (Nov. 25, 2012).

Corke, G. (2013). "Solibri model checker V8." *AECMagazine: Building information modelling (BIM) for architecture, engineering and construction*, ⟨http://aecmag.com/index.php?option=content&task=view&id=527⟩ (May 19, 2013).

Cunningham, H., et al. (2011). *Developing language processing components with gate version 6 (a user guide)*, Univ. of Sheffield, Dept. of Computer Science, Sheffield, U.K.

Deleger, L., et al. (2013). "Large-scale evaluation of automated clinical note de-identification and its impact on information extraction." *J. Am. Med. Inform. Assoc.*, 20(1), 84–94.

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.

DOE. (2011). "Building energy codes program: Software and tools." ⟨http://www.energycodes.gov/software.stm⟩ (Jul. 15, 2011).

Eastman, C., Lee, J., Jeong, Y., and Lee, J. (2009). "Automatic rule-based checking of building designs." *Autom. Constr.*, 18(8), 1011–1033.

El-Gohary, N. M., and El-Diraby, T. E. (2010). "Domain ontology for processes in infrastructure and construction." *J. Constr. Eng. Manage.*, 10.1061/(ASCE)CO.1943-7862.0000178, 730–744.

EPA. (2004). "Wal-Mart II storm water settlement." *Civil enforcement cases and settlements*, ⟨http://www.epa.gov/compliance/resources/cases/civil/cwa/walmart2.html⟩ (Nov. 25, 2012).

Fautsch, C., and Savoy, J. (2009). "Algorithmic stemmers or morphological analysis?" *J. Am. Soc. Inform. Sci. and Tech.*, 60(8), 1616–1624.

Fiatech. (2011). "Automated code plan checking tool." ⟨http://fiatech.org/active-projects/593-smartcodes%20.html⟩ (Jul. 15, 2011).

Galasso, J. (2002). *Analyzing English grammar: An introduction to feature theory: A companion handbook*, California State Univ. Northridge, Northridge, CA.

Garrett, J. H., Jr., and Fenves, S. J. (1987). "A knowledge-based standard processor for structural component design." *Eng. Comput.*, 2(4), 219–238.

Grefenstette, G., and Tapanainen, P. (1994). "What is a word, what is a sentence? problems of tokenization." *Proc., 3rd Conf. Computational Lexicography and Text Research (COMPLEX'94)*, Research Institute for Linguistics Hungarian Academy of Sciences, Budapest, Hungary, 79–87.

Han, C. S., Kunz, J. C., and Law, K. H. (1997). "Making automated building code checking a reality." *Facility Manage. J.*, 1997(September/October), 22–28.

Hogenboom, A., Hogenboom, F., Frasincar, F., Schouten, K., and Meer, O. V. D. (2013). "Semantics-based information extraction for detecting economic events." *Multimed Tools*, 2013(64), 27–52.

International Alliance for Interoperability (IAI). (2007). "IFC2x edition 3 technical corrigendum 1." *Industry foundation classes*, ⟨http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/index.htm⟩ (Jul. 15, 2011).

International Code Council (ICC). (2006). "2006 international building code." *2006 Int. Codes*, ⟨http://publicecodes.citation.com/icod/ibc/2006f2/index.htm⟩ (Feb. 5, 2011).

International Code Council (ICC). (2009). "2009 international building code." *2009 Int. Codes*, ⟨http://publicecodes.cyberregs.com/icod/ibc/2009/index.htm⟩ (Feb. 5, 2011).

Lau, G. T., and Law, K. (2004). "An information infrastructure for comparing accessibility regulations and related information from multiple sources." *Proc., 10th Int. Conf. on Computational Civil and Building Engineering (ICCCBE)*, ISCCBE, Hong Kong, China.

Levine, R., Meurers, W. (2006). "Head-driven phrase structure grammar linguistic approach, formal foundations, and computational realization." *Encyclopedia of language and linguistics*, 2nd Ed., K. Brown, ed., Elsevier, Oxford, U.K.

Li, C., et al. (2012). "TwiNER: Named entity recognition in targeted twitter stream." *Proc., 35th Int. ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '12)*, Association for Computing Machinery, New York, 721–730.

Ling, X., and Weld, D. S. (2012). "Fine-grained entity recognition." *Proc., 26th AAAI Conf. Artificial Intelligence*, Association for the Advancement of Artificial Intelligence, Palo Alto, CA, 94–100.

Makhoul, J., Kubala, F., Schwartz, R., and Weischedel, R. (1999). "Performance measures for information extraction." *Proc., DARPA Broadcast News Workshop*, Morgan Kaufmann, San Francisco.

Maynard, D., Bontcheva, K., and Cunningham, H. (2004). "Automatic language-independent induction of gazetteer lists." *Proc., 4th Conf. Language Resources and Evaluation (LREC'04)*, European Language Resources Association, Paris.

Pasca, M. (2011). "Attribute extraction from synthetic web search queries." *Proc., 5th Int. Joint Conf. Natural Language Processing*, Asian Federation of Natural Language Processing, Singapore, 401–409.

Patwardhan, S. (2010). "Widening the field of view of information extraction through sentential event recognition." Ph.D. thesis, Univ. of Utah, Salt Lake City.

Salama, D. M., and El-Gohary, N. M. (2011). "Semantic modeling for automated compliance checking." *Proc., 2011 ASCE Int. Workshop on Computational Civil Engineering*, ASCE, Reston, VA, 641–648.

Salama, D., and El-Gohary, N. (2013a). "Semantic text classification for supporting automated compliance checking in construction." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000301, 04014106.

Salama, D., and El-Gohary, N. (2013b). "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.

Sapkota, K., Aldea, A., Younas, M., Duce, D. A., and Banares-Alcantara, R. (2012). "Extracting meaningful entities from regulatory text." *2012 5th IEEE Int. Workshop on Requirements Engineering and Law (RELAW)*, IEEE, Piscataway, NJ, 29–32.

Singapore Building and Construction Authority. (2006). "Construction and real estate network: Corenet systems." ⟨http://www.corenet.gov.sg/⟩ (Jul. 15, 2011).

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.

Sun, A., Grishman, R., and Sekine, S. (2011). "Semi-supervised relation extraction with large-scale word clustering." *Proc., 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies—Volume 1 (HLT '11)*, Association for Computational Linguistics, Stroudsburg, PA, 521–529.

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.

Tang, K., Li, F., and Daphne, K. (2012). "Learning latent temporal structure for complex event detection." *Proc., CVPR. 2012*, IEEE, Piscataway, NJ, 1250–1257.

Tierney, P. J. (2012). "A qualitative analysis framework using natural language processing and graph theory." *Int. Rev. Res. Open Dist. Learn.*, 13(5), 173–189.

Wang, C., et al. (2010). "Mining advisor-advisee relationships from research publication networks." *Proc., 16th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD '10)*, Association for Computing Machinery, New York, 203–212.

Zhang, J., and El-Gohary, N. M. (2011). "Automatic information extraction from construction-related regulatory documents for automated compliance checking." *Proc., CIB W78 2011*, Conseil International du Bâtiment (CIB), Rotterdam, Netherlands.

Zhang, J., and El-Gohary, N. M. (2012a). "Automated regulatory information extraction from building codes leveraging syntactic and semantic information." *Proc., 2012 ASCE Construction Research Congress (CRC)*, ASCE, Reston, VA, 622–632.

Zhang, J., and El-Gohary, N. M. (2012b). "Extraction of construction regulatory requirements from textual documents using natural language processing techniques." *Proc., 2012 ASCE Int. Conf. on Computational Civil Engineering*, ASCE, Reston, VA, 453–460.

Zhang, J., and El-Gohary, N. M. (2013). "Information transformation and automated reasoning for automated compliance checking in construction." *Proc., 2013 ASCE Int. Workshop Computational in Civil Engineering*, ASCE, Reston, VA.

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." *Autom. Constr.*, 28(2012), 58–70.

Zouaq, A. (2011). "Ontology learning and knowledge discovery using the web." *An overview of shallow and deep natural language processing for ontology learning*, IGI Global, Hershey, PA, 16–38.