The published version is found in the ASCE Library here: http://ascelibrary.org/doi/abs/10.1061/(ASCE)CP.1943-5487.0000346
Zhang, J. and El-Gohary, N. (2015). "Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000346, 04015014.

CP0948

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 at enabling 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. In our proposed approach, we use a set of pattern-matching-based IE rules and conflict resolution (CR) rules in IE. We use a variety of syntactic (syntax/grammar-related) and semantic (meaning/context-related) text features in the patterns of the IE and CR rules. We also propose and use phrase structure grammar (PSG)-based phrasal tags and separation and sequencing of semantic information elements to reduce number of needed patterns. We utilize an ontology to aid in the recognition of semantic text features (concepts and relations). We tested our proposed IE extraction algorithms in extracting quantitative requirements from the 2009 International Building Code and achieved 0.969 and 0.944 precision and recall, respectively.

CE Database subject headings: Project management; Construction management; Information management; Computer applications; Artificial intelligence.

[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; Tel: +1-217-333-6620; Fax: +1-217- 265-8039.

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, etc. Each regulation has a large set of provisions. For example, the IBC 2006 is composed of 329 sections, where each section includes several to tens of provisions that address a variety of requirements (e.g. safety, environmental, etc.).

Building codes are the primary sets of regulations governing the design, construction, alteration, and maintenance of building structures. Within the fifty states of the U.S., 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 adopting local jurisdiction in order to adapt these codes to various local conditions (e.g. weather conditions). Thus, in most of the 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. There is also the state of Massachusetts which 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. Even in one code, format and semantics of provisions could vary from one chapter to another.

Due to 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; like

other manual processes (Boken and Callaghan 2009), the manual process of regulatory compliance checking is time-consuming, costly, and error-prone. For example, in the city of Mesa, Arizona, the turn-around 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 due to violation of storm-water regulations (US EPA 2004; Salama and El-Gohary 2011). Automated compliance checking (ACC) is expected to reduce the cost, time, and errors of compliance checking (Tan *et al.* 2010; Eastman *et al.* 2009). With the advancement in computing technology, there have been many research efforts in automating 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 US Department of Energy (US DOE 2011), SMARTcodes led by the International Code Council (ICC 2011), and Avolve Software (Avolve Software Corporation 2011). Previous research and software development efforts have undoubtedly 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 for extracting requirements from textual regulatory documents (e.g. codes) and encoding 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, the creation of SMARTcodes rules requires manual extraction and encoding effort.

To address this gap, we are proposing a new approach for automated regulatory information extraction for supporting ACC in construction. Our 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 utilizing artificial intelligence to enable computers to understand and process natural language text (or speech) in a human-like manner (Cherpas 1992). Information extraction (IE) is a subfield of NLP. It aims at extracting 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

We propose a semantic, rule-based NLP approach for automated IE from construction regulatory documents. In our analysis, in comparison to general non-technical text (e.g. news articles, general websites, etc.), domain-specific regulatory text is more suitable for automated NLP (i.e. would allow for better interpretability and less ambiguity in automated processing) due to three main text characteristics. First, construction text is likely to have less homonym conflicts than non-technical text. For example, in news articles, the term “bridge” could refer to a structural bridge, the card game, a bridge of understanding, a dental bridge, etc. Second, it is easier to develop an ontology that captures domain knowledge as opposed to an ontology that captures general knowledge (or a wide variety of domains). A domain ontology may enhance automated interpretability and understandability of domain-specific text. Third, regulatory text is likely to exhibit less co-reference resolution problems. For example, construction regulatory text tends to mention the subjects (e.g. door) for each provision explicitly rather than referring to the subjects using pronouns (e.g. “it”).

Rule-Based Approach

Our approach is rule-based. There are two main types of approaches taken in NLP: rule-based approach, and 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 for training 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 more human effort. We are taking a rule-based approach, because of its expected higher performance. We are using IE rules that rely on pattern matching to identify the part(s) of text to extract based on recognized text patterns. Our approach relies on, both, the semantic and syntactic features of the text in defining these patterns. We capture the syntactic features (e.g. part of speech (POS) tags) of the text using various NLP techniques, including tokenization, sentence splitting, morphological analysis, POS tagging, and phrase structure analysis. We capture the semantic features (concepts and relations) of the text based on an ontology that represents the domain knowledge. Due to the compositional and recursive nature of text, sentences could be long and complex, which may result in a large number of patterns. We utilize phrase structure grammar (PSG) in our 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. This will result in requiring less IE rules for extraction and thus reducing human effort in developing IE rules. We also separate and sequence the extraction of different semantic information elements to further limit the number of needed IE patterns. In addition to IE rules, we use a set of rules for resolving conflicts in information extraction (CR rules).

Semantic Approach

We utilize a domain ontology to capture the semantic features of the text. An ontology models domain knowledge in 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 in comparison to syntactic IE (i.e. IE using syntactic features only), because domain knowledge (represented in an ontology) could help to identify or distinguish domain-specific terms and meanings (Soysal *et al.* 2010). For example, Zhang and El-Gohary (2011) have shown an enhanced performance with semantic IE, in comparison to syntactic-only IE (an increase of precision from 75% to 100% and of recall from 75% to 95%).

Comparison to the State of the Art

Many research efforts have been 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 at extracting 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 at extracting instances of multiple concepts (Patwardhan 2010). From this perspective, our approach is more similar to event extraction, because we also extract instances of multiple concepts in a provisional requirement. In comparison to event extraction, however, our approach is different in two main ways. First, in our approach, we extract information in a more flexible way. We define two types of information elements: "rigid information elements" and "flexible information elements". A rigid information element is an information element that has a pre-defined, fixed number of concepts/relations (e.g. in a terrorist event case, it is pre-defined that "victim" is associated with only one concept). A flexible information element, in contrast, has a varying number of concepts/relations depending on the instance at hand (e.g. in our approach, "subject restriction" has a varying number of multiple concepts/relations). Unlike event extraction, in our approach we can extract instances of flexible information elements. Second, because we introduced a way to extract information elements in a more flexible way, we are able to perform a deeper level of information extraction (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, etc.). Deep NLP aims at full sentence analysis, with more complex

understanding of the text, towards capturing the entire meaning of sentences (Zouaq 2011). Correspondingly, shallow IE extracts specific type(s) of information from a sentence, while deep IE aims at extracting all information that is expressed by a sentence based on 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 around 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, which aimed at extracting protected health information, reported a best performance of 0.9668, 0.9377 precision and recall, respectively (Deleger *et al*. 2013).

In the construction domain, there has been a number of important research efforts that have utilized NLP techniques (e.g. Caldas and Soibelman (2003) have conducted machine-learning-based text classification of construction documents), but only a few of these efforts have conducted some type/level of information extraction such as Abuzir and Abuzir (2002) and Al Qady and Kandil (2010). For example, Al Qady and Kandil (2010) used shallow parsers to extract concepts and relations from construction contracts. In this work, 1) the extraction is based on syntactic features, produced by shallow parsing, only. In our approach, we use semantic features, in addition to syntactic ones; 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 our approach, in our IE and CR rules, we use patterns that consist of a variety of syntactic and semantic features, 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 this work, 1) the extraction uses HTML-based document structure features (including title tags, heading tags, and URLs) and simple lexical syntactic features. In our approach, we do not use document structure features (since we deal with unstructured text, rather than HTML documents), and we rely on the syntactic and semantic features of the text; and 2) since the main purpose of the extraction is thesaurus construction, their

information extraction focuses on extracting terms. In our approach, since our ultimate purpose is automated reasoning about regulatory requirements, our information extraction is conducted on a deeper level, because we do not only need to extract terms/concepts, but we also need to extract other information elements (e.g. restrictions) for extracting all information expressed in a sentence/requirement. So, in comparison to these efforts, in this research, we are 1) dealing with 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. We aim at automatically processing 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). We utilize a domain ontology for identifying semantic text features. Using domain-specific semantics and "flexible information elements" to achieve relatively deep semantic NLP will allow for: a) analyzing complex sentences that would otherwise be too complex for automated information extraction (IE), b) recognizing domain-specific text meaning, and c) in turn, improving accuracy 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. It 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 → Y, where X and Y are strings in Vp". The key advantage of a 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, complex sequence of features on the right-hand side of the rules could be represented by a few or even just one simple symbol on the left-hand side of the rules. 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 has to be a single non-terminal (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 – "Sentence" node in our example. 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 non-terminals are explained on 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

In this section, we present our proposed methodology for automatically extracting information from construction regulatory documents. We present it 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 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 for the purpose of achieving improved performance.

Phase I- Information Representation

We propose this phase to define the representation format for the extracted information. In our methodology, the ultimate representation format is one or more logic sentences which could be directly used for automated compliance reasoning. For intermediate processing, we propose our new ACC-tuple to represent the extracted information. We propose the use of a tuple format for intermediate processing, because it is easy for computer manipulation and evaluation (e.g. <Subject, Attribute, Value> is a 3-tuple).

In our ACC-tuple representation, we call each element a "semantic information element". A "semantic information element" is: 1) an ontology concept; 2) an ontology relation; 3) a deontic operator indicator: a

term indicating an obligation, permission, or prohibition – following our semantic ACC model in Salama and El-Gohary 2013b; or 4) a restriction: a restriction places a constraint on the definition of a semantic information element, where the constraint is expressed in terms of ontology concepts and relations. We introduce the following types of semantic information elements: "simple semantic information elements" versus "complex semantic information elements", and "rigid semantic information elements" versus "flexible semantic information elements". A simple semantic information element (SIE) is associated with a single concept/relation/indicator, while a complex SIE is expressed in terms of a number of concepts and relations. Our simple SIEs are rigid, while our complex SIEs are flexible. As discussed above, a rigid SIE is an information element that has a pre-defined, fixed number of concepts/relations, while a flexible SIE has a varying number of concepts/relations depending on the instance at hand. Accordingly, in our ACC-tuple, an ontology concept, an ontology relation, and a deontic operator indicator are simple (and thus rigid) SIEs, while a restriction is a complex (and thus flexible) SIE. The use of flexible SIEs is key in providing the flexibility that is needed for facilitating full sentence analysis. We refer to a specific word, phrase, or chunk of text extracted and mapped according to a SIE as an "information element instance".

To prepare for further information transformation into logic sentences, we use a semantic mapping step for matching 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 published version is found in the ASCE Library here: http://ascelibrary.org/doi/abs/10.1061/(ASCE)CP.1943-5487.0000346
Zhang, J. and El-Gohary, N. (2015). "Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000346, 04015014.

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

We use this phase to prepare the raw (i.e. unprocessed) text for further processing. In our methodology, preprocessing consists of tokenization, sentence splitting, de-hyphenation, and morphological analysis.

Tokenization

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

Sentence Splitting

Sentence splitting is the process of recognizing each sentence of the text. Similar to tokenization, the recognition of sentences is 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 our 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 at recognizing the different forms of a word and mapping 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". Also, as shown in Fig. 3, "rooms" and "feet" were mapped to their lexical forms "room" and "foot", respectively. While tokenization and sentence splitting are essential for IE, since the text must be broken down into units for further processing; MA is not essential for IE, but is used to improve identification of words with the same lexical form. We decided to incorporate MA in our pre-processing methodology, because it aids in the recognition of ontology concepts. For example, the plural form of a concept could be recognized although only the singular form is used in the ontology.

De-hyphenation

We use de-hyphenation for removing hyphens that are used for indicating continuations of words across two lines. This avoids a word not being recognized because of such hyphen.

Phase III - Feature Generation

We use this phase to generate a set of features that describe the text. In our methodology, we 1) use domain-specific ontology-based semantic features, in addition to syntactic features and 2) propose the use of PSG-based phrasal tags to reduce the number of needed patterns. Our 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, such as in Afrin (2001). Semantic features benefit IE tasks beyond solely using syntactic features because they express domain-specific meaning/knowledge, such as in Soysal *et al.* (2010). In our methodology, we generate both syntactic (POS tags, PSG-based phrasal tags, gazetteer terms) and semantic features (concepts and

relations); and, subsequently, use these features in defining patterns (text patterns in our IE and CR rules that aid in the process of pattern matching for IE).

Part-of-Speech (POS) Tagging

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

Phrase Structural Analysis

Our phrase structural analysis builds on the POS tagging step, and aims at assigning type labels (phrasal tags) to phrases of a sentence. Examples of phrasal tags are NP (noun phrase), VP (verb phrase), and PP (prepositional phrase), etc. For example, as shown in Fig. 3, "Habitable rooms", "shall have a net floor area of not less than 70 square feet", and "of not less than 70 square feet" were assigned NP, VP, and PP tags, respectively. We use PSG to generate phrasal tags. In our methodology, we derive our application-specific PSG rules based on a randomly selected sample of text (we call it "development text", which we also use for text analysis and further development of IE and CR rules). 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)". Our 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 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", "the wall of the room

in the building with a vent at the bottom", etc. In this step, PSG is derived from previously POS-tagged source text; and is, subsequently, used in assigning PSG-based phrasal tags to sentences in the source text.

For empirically studying the effect of utilizing PSG-based phrasal tags on the number of patterns, for preliminary verification of our methodology, we conducted an experimental test. We 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. This shows 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 could group any set of terms based on any specific commonality possessed by these terms. We use the information that a word or phrase belongs to a certain list in the gazetteer 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 our methodology, a gazetteer is used to provide a set of term lists, where each list has a specific function. For example, terms like "no" and "not" have the function "negation", and as such are included in our "negation gazetteer list". In our methodology, we compiled and used several types of gazetteer lists, such as the "comparative relation gazetteer list", which is composed of terms indicating comparative relations, such as "greater or equal", "less or equal", "at most", "at least", etc. 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. We could have chosen to represent the information presented in a gazetteer list as part of an instantiated ontology (e.g. we could have represented the list of countries as instances of the concept "country").

However, for computational efficiency we chose to separate such instances from the ontology (in the form of gazetteer lists).

Ontology-Based Semantic Analysis

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

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

Phase IV – Target Information Analysis

We propose this phase for manually analyzing the text to identify the types of semantic information elements to be extracted and their interrelationships, and the sequence of their extraction. In our methodology, we propose an approach for separation and sequencing of semantic information elements (SSSIE) to reduce the number of needed IE patterns.

Identification of Target Information

In this step of our 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", etc. In the case of the example in Fig. 3, the information to be extracted is related to quantitative requirements, so we 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

We propose this step is to identify the sequence of extracting the semantic information elements. Based on our experimental studies, we found that extracting all semantic information elements from a sentence by a single IE rule (i.e. extracting all instances at the same time) is not efficient, because the amount of possible patterns increase largely as the number of semantic information elements increases. Since there is some independency (while not fully independent) between information elements, we propose to extract information elements separately and sequentially. The decision on 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. 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" is dependent on the extraction of "subject", then "subject" should be extracted prior to "subject restriction". In the case of 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 our proposed approach for separation and sequencing of semantic information elements (SSSIE), we conducted an experiment for comparing the performance results of two cases. In the first case, we developed and used IE rules that extract all semantic information elements from a sentence by a single IE rule (i.e. extracting all instances at the same time). For the second case, we used our proposed method for SSSIE in IE. For both cases, we developed the IE rules based on Chapter 12 and 23 of IBC 2006 and tested them on Chapter 19 of IBC 2009. Eighty-seven (87) and 50 patterns were needed for the first and second cases, respectively. This shows that the use of our SSSIE method reduces the number of needed patterns in IE rules. The comparative results in terms of precision, recall, and F-measure are shown in Table 3. The results show significantly higher performance using SSSIE (the second case). The lower performance in the first case could be partially attributed to: 1) the fact that it is difficult (if not impossible) to enumerate all possible patterns based on a limited development text, and 2) an error in recognizing a single semantic information element in a given IE rule would affect the extraction result of the whole IE rule (and thus all other information elements in that rule).

Phase V – Development of Information Extraction Rules

We use this phase for developing a set of rules to automatically execute the information extraction process. In our methodology, we propose 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, while 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) are based on pattern matching methods. The left-hand side of the rule defines the pattern to be matched, and the right-hand side defines which part of the matched pattern

should be extracted. We use, both, syntactic (POS tags, PSG-based phrasal tags, and gazetteer terms) and semantic (ontology concepts and relations) text features in the patterns of the IE rules. If a concept in the ontology is used in an IE rule, all its sub-concepts 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"". Applying this IE rule to the example in Fig. 3, "habitable rooms" will be extracted as an instance of "subject" because it matches a sub-concept of "building element" in the ontology – "Habitable_Room". A sample IE rule (in English) and its corresponding Java coding (using Java Annotation Patterns Engine (JAPE) rules in GATE) are shown in Fig. 4.

In order to develop these IE rules, we propose to conduct the following three tasks: 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, testing-based improvement of constructed patterns, etc.). Feature selection aims at selecting all features that are 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 accordingly "has" was extracted as a "quantitative relation" instance.

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

In our methodology, the rules for resolving conflicts in extraction (Conflict Resolution (CR) rules) mainly address four types of conflict cases: 1) more than the required number of information element instances of a semantic information element in a single sentence, 2) less than the required number of information element instances of a semantic information element in a single sentence, 3) overlap of extraction results

for different semantic information elements, and 4) no conflicts – equal to the required number of information element instances of a semantic information element in a single sentence. Each type of conflict case may be handled using one of a set of actions. For conflict case 1, two actions may be used: a) keep all information element instances; or b) set priority rules and select the information element instances with higher priority (e.g. set a higher priority for "not less than" comparing to "above" when encountering multiple comparative relation instances. For example, in the part of sentence "nonabsorbent surface to a height not less than 70 inches above the drain inlet", the comparative relation instance extracted would be "not less than", only, although both "not less than" and "above" are recognized as candidate comparative relation instances). For conflict case 2, three actions may be used: a) set a default information element instance based on domain knowledge (e.g. the default comparative relation instance may be set to "greater_than_or_equal" when there is no information element instance 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" would be used as a comparative relation instance); b) use the same instance from the nearest sentence/clause (left or right) if those sentences/clauses are describing 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 square feet", the subject of the first quantitative relation should be used for the second quantitative relation as well); or c) drop this sentence. For conflict case 3, three actions may be used: a) delete all overlapping information element instances and keep the required number only, b) keep all information element instances, or c) 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 for describing the corresponding requirement. For example, as shown in Fig. 3, the following CR rule (a conflict case 4) was applied: If there is one instance 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. Defining which action should be

executed in which case is 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 at extracting 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. It is manually (or semi-automatically 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)). There is a trade-off 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, we set α to 0.5 to give equal weights to recall and precision. If the evaluation result is 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. On the other hand, if the evaluation results are not satisfactory, the phases can be re-iterated for performance improvement. Performance improvements in later iterations may be achieved by addressing extraction errors in earlier iterations.

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

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

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

Validation: Experiments and Results

We conducted an experiment for validating our proposed algorithms. Evaluating the algorithms (in terms of precision and recall) and achieving satisfactory performance would imply the validity of our proposed approach and methodology. We extracted quantitative requirements from randomly selected chapters of IBC 2006 and 2009. We evaluated the IE performance of our algorithms by comparing the extraction results against a semi-automatically (expert using NLP tools) developed gold standard.

Source Text Selection (International Building Code)

Our proposed methodology is intended for extracting information from a variety of construction-related regulatory documents (e.g. building codes, environmental regulations, safety regulations and standards, etc.). At this phase, we tested the proposed algorithms on building codes. We selected the IBC because it is the most widely-adopted building code in the U.S. We used IBC 2006 (ICC 2006) and IBC 2009 (ICC 2009). We randomly selected Chapters 12 and 23 of IBC 2006 for development, and Chapter 19 of IBC 2009 for testing. At this phase, we identified two main types of requirements in IBC: 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 feet 6 inches (2286 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 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 there should be a bathroom with water closet, lavatory, and bathtub or shower in an efficiency dwelling unit. We decided to experiment on 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 to be more complex than those describing existential requirements. This implies that they are more difficult to extract.

Ontology Development

We developed an application-oriented and domain-specific ontology for buildings. 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 re-used as necessary. We coded the ontology in OWL (Web Ontology Language), i.e. *.owl format, because OWL is the most widely-used semantic web language.

Information Representation

For building codes, we used a nine-tuple format 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 our semantic model of ACC, as presented in previous work (Salama and El-Gohary 2013b), we define the semantic information elements as follows (for further elaboration about 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, etc.) 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 foot in width and 2 feet in length for each additional story", the quantitative relation is "increase". It 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, etc., which is commonly-used for comparing quantitative values (i.e. comparing an existing value to a required minimum or maximum value). A "quantity value" is a value, or a range of values, which 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") – for example by defining the properties of the "subject" (or "quantity)".

In each extracted requirement: 1) there is one and only one instance of each of the following semantic information elements: subject, comparative relation, quantity value, and quantity unit/reference; 2) there is at most one instance of each of the following semantic information elements: compliance checking attribute, deontic operator indicator, and quantitative relation; and 3) there could be zero, one, or more instances of each of the following semantic information elements: subject restriction and quantity restriction. Table 4 shows some examples of the 9-tuple representation.

Development of Gold Standard

We developed the gold standard semi-automatically. First, we automatically extracted all sentences that include a number (both appearances of digits and words forms of a number; this way ensures 100% recall of sentences describing quantitative requirements). 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, we recognized 304 sentences containing quantitative requirements – which formed our gold standard.

Tool Selection (GATE)

Many off-the-shelf tools are available today for supporting various NLP tasks including IE, such as Stanford Parser by the Stanford NLP Group, and GATE by the University of Sheffield. We conducted the

experiment using GATE. We selected GATE to implement our IE algorithms, because: 1) It 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. We utilized the following built-in GATE tools: We used 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 our IE and CR rules.

Applying our IE Methodology

We developed our IE and CR rules based on Chapters 12 and 23 of IBC 2006, and then subsequently tested these rules on Chapter 19 of IBC 2009. We used the ANNIE Hepple POS Tagger to generate POS tag features (a sample is shown in Table 5). There was a total of 53 POS tag symbols in the set of Hepple POS Tags we used. For phrase structure analysis, we used the Penn Treebank phrasal tag labels. We compiled three gazetteer lists: comparative relation list, unit list, and negation list. In addition, we utilized the GATE built-in gazetteer lists of numbers and ordinal. The number of patterns, features, and CR rules for Chapters 12 and 23 of IBC 2006 are shown in Table 6 below. Our IE and CR rules (that we 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. We applied the rules to Chapter 19 of IBC 2009 for testing and evaluation.

Our IE and CR rules are also potentially reusable for extracting quantitative requirements from other types of documents/text. They can be reused – as is or with adaptation/extension based on additional development text. For testing the potential reusability of our IE and CR rules, we applied the rules (as is, without any modification) to a different type of text. We randomly selected the following document from the Web, with the only criterion being that the document contains a quantitative requirement: “Procedures (Section 700.4) in traffic cabinet ground rod specifications”. We used our rules in extracting quantitative requirements from the randomly-selected text, and evaluated the performance against a manually-developed gold standard. The results in terms of precision, recall, and F-measure are shown in Table 7.

As per Table 7, the overall F-measure is greater than 0.90, which indicates potential reusability of the rules.

Results and Discussion

The information extraction results are summarized in Table 8. For Chapter 19 of IBC 2009, on average, we achieved 0.969, 0.944, and 0.956 precision, recall, and F-measure, respectively. 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 (since each restriction instance may include multiple concepts and relations). When calculating the precision and recall for "comparative relation" instances, we consider partial extraction correctness for the following comparative relations: "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, while the gold standard included "at or above" (greater_than_or_equal).

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

Through error analysis, we find that: (1) The reasons for the relative low recall of "subject restriction" are: (a) The patterns are more complex. For example, one pattern for "subject restriction" typically involves several phrases, while one pattern for other elements such as "subject" could be as simple as corresponding to just one concept in the ontology; (b) 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; (2) The errors in the extraction of "subject" are due to 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 can achieve 100% performance, even for relatively simple NLP tasks such as POS tagging. Any error in POS tagging, for example, may further cause an error in information extraction,

since our IE rules include POS-features in its patterns; (3) The errors in extraction of "compliance checking attribute" are due to inner errors of the tools used and the limitations of CR rules. For example, one CR rule states if there is no "compliance checking attribute" extracted and there are extra "subject" candidates extracted, then put the "subject" candidate closest to the "quantity value" as the attribute. This rule lead 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 inch to the exterior steel shell"; (4) The errors in the extraction of "deontic operator indicator" and "quantitative relation" are due to missing patterns in IE rules (which were missed because the patterns are not common) and limitations of CR rules; and (5) The errors in the extraction of "comparative relation", "subject restriction", "quantity restriction", "quantity value", and "quantity unit/reference" are due to missing patterns in IE rules.

In future work, we will further explore how to improve our IE and CR rules to avoid/reduce these errors, and consequently improve the IE results. So far, we believe that one possible solution for solving the problem of missing patterns and limitations of CR rules is through the development of IE and CR rules based on more corpuses. But, we need to further explore how much 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 of development corpuses would result in significant improvement in precision and recall.

Limitations and Future Work

Our experimental results show that our proposed approach is promising for automatically extracting information from construction regulatory documents. Despite the high performance we achieved (0.969, 0.944, and 0.956 precision, recall, and F-measure, respectively), we acknowledge three limitations of the work, which we plan to address as part of our future/ongoing research. First, we only tested our methodology/algorithms on 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, we will test our methodology/algorithms on other types of requirements such as existential requirements. Second, we only tested our methodology/algorithms on one chapter, mainly because the development of the gold standard for testing is highly time-intensive. As part of future/ongoing research work, we will test our methodology/algorithms on more chapters of building codes. We expect that the results will show similar high performance since the chapter used in testing contains large amount of text (about 7000 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, we might see variation in the results due to the possible variability in the syntactic and semantic text features across different chapters and/or codes. In that case, our IE and CR rules can be adapted/extended based on additional development text. Third, we only tested our methodology/algorithms on building codes. In future work, we will extend our methodology/algorithms to extract information from other types of regulatory documents (e.g. environmental regulations), as well as 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 main ways. First, we offer domain-specific, semantic NLP methods that can help capture domain-specific meaning, and we show that ontology-based semantic IE outperforms syntactic-only IE (in terms of precision and recall). Domain-specific semantics allow for analyzing complex sentences that would otherwise be too complex for automated IE, recognizing domain-specific text meaning, and in turn improving performance of IE. Second, we offer relatively-efficient-to-develop rule-based NLP methods that can benefit from expert NLP knowledge which is encoded in the form of IE and CR rules. We show that the efficiency of algorithm development for rule-based methods can be enhanced through 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 our SSSIE method reduce the number of patterns needed in IE rules which result in requiring less IE rules for extraction and thus reducing human effort in developing IE rules. Third, we show 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 one platform. We show 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). Our IE rules, CR rules, and algorithms are potentially reusable (as we show in our experimental results). In comparison to our initial efforts, future efforts in adapting the rules and/or algorithms should be significantly less. 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 error of the checking process. This could speed up the regulatory process, enhance cost and time project efficiency, and lead to less violation 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 for the detection of inconsistencies, analysis of project documents and records for supporting claim analysis, analysis of daily site reports for supporting progress monitoring and project control, etc.

Conclusions

This paper presented a semantic, rule-based NLP methodology/algorithm for automated information extraction (IE) from construction regulatory documents for supporting automated compliance checking. A set of pattern-matching-based IE rules and conflict resolution (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. Phrase structure grammar-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. Our information extraction is relatively deep; it aims at achieving full sentence analysis for extracting all information of a requirement for further representation in a logic-based rule format. We tested our algorithms on extracting quantitative requirements from the 2009 International Building Code. Comparing the extracted information element instances with those in our semi-automatically-developed gold standard, we achieved an average precision and recall of 0.969 and 0.944, respectively. These high performance results indicate that our proposed IE approach is promising. Through analysis, we also pinpointed the sources of errors in our experimental results and identified potential solutions for the possibility of further performance enhancement. As part of our future/ongoing work, we will test our 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). We expect that the results will show similar high performance. However, we might see variation in the results due to the possible variability in the syntactic and semantic text features across different requirements, chapters, codes, or documents.

Acknowledgement

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., Intl. Workshop on Info. Tech. in Civ. Eng. 2002*, ASCE, Reston, VA, 400-412.

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

Al Qady, M.A., and Kandil, A. (2010). "Concept relation extraction from construction documents using natural language processing." *J. Constr. Eng. Manage.*, 136(3), 294-302.

Avolve Software Corporation. (2011). *Electronic plan review for building and planning departments*. <http://www.avolvesoftware.com/index.php/solutions/building-departments/> (July 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 Intl. Conf. Web Search and Data Mining (WSDM '13)*. ACM, New York, NY, 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(2003), 395-406.

Cherpas, C. (1992). "Natural language processing, pragmatics, and verbal behavior." *Analysis of Verbal Behavior*, 10,135-147.

Chomsky, N. (1956). "Three models for the description of language." *Information Theory, IEEE Transactions*, 2(3), 113–124.

City of Mesa. (2012). "Construction plan review." *The 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)." The University of Sheffield, Department of Computer Science, Sheffield, U.K.

Deleger, L., Molnar, K., Savova, G., Xia, F., Lingren, T., Li, Q., Marsolo, K., Jegga, A., Kaiser, M., Stoutenborough, L., and Solti, I. (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.*, 9(2), 141-156.

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.*, 136(7), 730–744.

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> (July 15, 2011).

Galasso, J. (2002). "Analyzing English grammar: an introduction to feature theory: a companion handbook." California State University Northridge, Northridge, CA.

Garrett, Jr., J.H., and Fenves, S.J. (1987). "A knowledge-based standard processor for structural component design." *Eng. with 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. Comput. 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." *J. Manage.*, 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> (July 15, 2011)

International Code Council (ICC). (2011). <http://www.iccsafe.org/> (July 15, 2011).

International Code Council (ICC). (2009). "2009 International Building Code." *2009 Intl. Codes*, < http://publicecodes.cyberregs.com/icod/ibc/2009/index.htm > (Feb. 05, 2011).

International Code Council (ICC). (2006). "2006 International Building Code." *2006 Intl. Codes*, <http://publicecodes.citation.com/icod/ibc/2006f2/index.htm > (Feb. 05, 2011).

Lau, G.T., and Law, K. (2004). "An information infrastructure for comparing accessibility regulations and related information from multiple sources." *Proc., 10th Intl. Conf. on Comput. Civ. and Bldg. Eng. (ICCCBE)*, Intl. Society for Comput. in Civ. and Bldg. Eng. (ISCCBE).

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

Li, C., Weng, J., He, Q., Yao, Y., Datta, A., Sun, A., and Lee, B. (2012). "TwiNER: named entity recognition in targeted twitter stream." *Proc., 35th Intl. ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '12)*. ACM, New York, NY, 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, California, 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, California.

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, France.

Pasca, M. (2011). "Attribute extraction from synthetic web search queries." *Proc., 5th Intl. Joint Conf. Natural Language Processing*, Chiang Mai, Thailand, 401-409.

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

Salama, D.M., and El-Gohary, N.M. (2013a). "Semantic text classification for supporting automated compliance checking in construction." *J. Comput. Civ. Eng.*, accepted.

Salama, D.M., and El-Gohary, N.M. (2013b). "Towards automated compliance checking of construction operation plans using a deontology for the construction domain." *J. Comput. Civ. Eng.*, accepted.

Salama, D.M., and El-Gohary, N.M. (2011). "Semantic modeling for automated compliance checking." *Proc., 2011 ASCE Intl. Workshop on Comput. Civ. Eng.*, ASCE, Reston, VA, 641-648.

Sapkota, K., Aldea, A., younas, M., Duce, D.A., and Banares-Alcantara, R. (2012). "Extracting meaningful entities from regulatory text." *2012 Fifth IEEE Intl. Workshop on Requirements Eng. 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/> (July 15, 2011).

Soysal, E., Cicekli, I., and Baykal, N. (2010). "Design and evaluation of an ontology based information extraction system for radiological reports." *Comput. in Biology and 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 Assoc. for Comput. Linguistics: Human Language Technologies - Volume 1 (HLT '11)*, Vol. 1. Assoc. for Comput. 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.*, 24(2), 203-211.

Tang, K., Li, F., and Daphne, K. (2012). "Learning latent temporal structure for complex event detection." *Proc., CVPR. 2012*, IEEE, New York, NY, 1250-1257.

Tierney, P.J. (2012). "A qualitative analysis framework using natural language processing and graph theory." *The Intl. Review of Research in Open and Distance Learning*, 13(5).

US Department of Energy (US DOE). (2011). *Building energy codes program: Software and tools*.<http://www.energycodes.gov/software.stm> (July 15, 2011).

US Environmental Protection Agency (US 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).

Wang, C., Han, J., Jia, Y., Tang, J., Zhang, D., Yu, Y., and Guo, J. (2010). "Mining advisor-advisee relationships from research publication networks." *Proc., 16th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining* (KDD '10). ACM, New York, NY, 203-212.

Zhang, J., and El-Gohary, N.M. (2013). "Information transformation and automated reasoning for automated compliance checking in construction". *Proc., 2013 ASCE Intl. Workshop Comput. in Civ. Eng.*, ASCE, Reston, VA. Accepted.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 Intl. Conf. on Comput. Civ. Eng.*, ASCE, Reston, VA, 453-460.

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, The Netherlands.

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.

The published version is found in the ASCE Library here: http://ascelibrary.org/doi/abs/10.1061/(ASCE)CP.1943-5487.0000346
Zhang, J. and El-Gohary, N. (2015). "Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000346, 04015014.

Tables

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, inch)))

* a) universal quantifier (‘∀’ or ‘for all’) asserts that the sentence is true for all instances of a variable, b) conjunction ‘^’ : ‘A ^ B’ means ‘A’ is true and ‘B’ is true, c) implication ‘->’: ‘A -> B’ means ‘A’ implies ‘B’ (if ‘A’ is true then ‘B’ is true), and d) obligation operator (*O*): *O* A means ‘A’ is obligated.

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

The published version is found in the ASCE Library here: http://ascelibrary.org/doi/abs/10.1061/(ASCE)CP.1943-5487.0000346
Zhang, J. and El-Gohary, N. (2015). "Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000346, 04015014.

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	61.5	270
Correctly Extracted with SSSIE	80	43	79.5	81	81	364.5
Correctly Extracted without SSSIE	48	27	59.5	62	63.5	260
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	0.963
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	0.679
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	0.796

The published version is found in the ASCE Library here: http://ascelibrary.org/doi/abs/10.1061/(ASCE)CP.1943-5487.0000346
Zhang, J. and El-Gohary, N. (2015). "Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000346, 04015014.

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 inch 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 square foot for each 150 square feet of crawl space area.	Courts shall not be less than 3 feet 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

The published version is found in the ASCE Library here: http://ascelibrary.org/doi/abs/10.1061/(ASCE)CP.1943-5487.0000346
Zhang, J. and El-Gohary, N. (2015). "Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000346, 04015014.

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

Number of	Subject	Subject Restriction	Compliance Checking Attribute	Deontic Operator Indicator	Quantitative Relation	Comparative Relation	Quantity Value	Quantity Unit/ Reference	Quantity Restriction
Extraction Patterns	NA	29	NA	10	9	2	24	24	48
Features Selected	10(304)*	47	1(99)	8	7	5	28	31	60
CR rules	2	2	5	0	0	4	8	8	9

*The 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	NA	0.944	1.000	0.647	0.765	1.000	1.000	0.857	0.899
Recall	0.875	NA	0.944	1.000	0.688	1.000	0.960	0.960	1.000	0.924
F-Measure	0.875	NA	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

The published version is found in the ASCE Library here: http://ascelibrary.org/doi/abs/10.1061/(ASCE)CP.1943-5487.0000346
Zhang, J. and El-Gohary, N. (2015). "Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000346, 04015014.

Figure Captions List

Figure 1: A sample set of CFG rules (partial) and corresponding derivation of a sentence

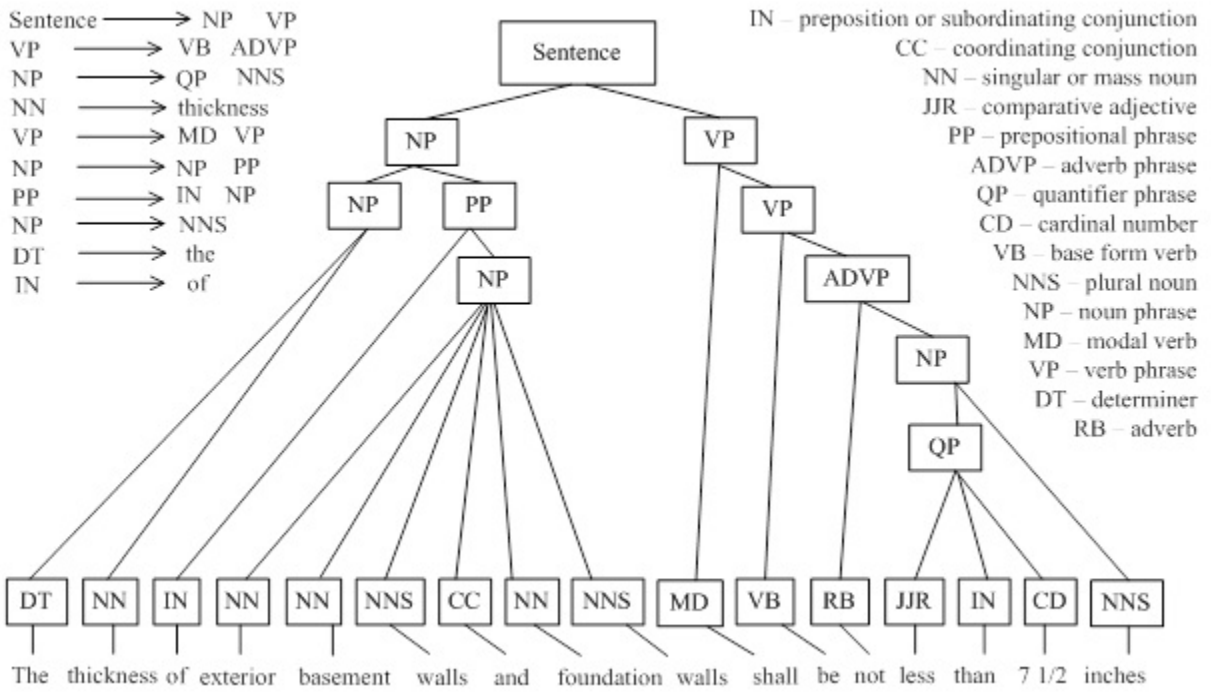

Figure 2: Proposed information extraction methodology

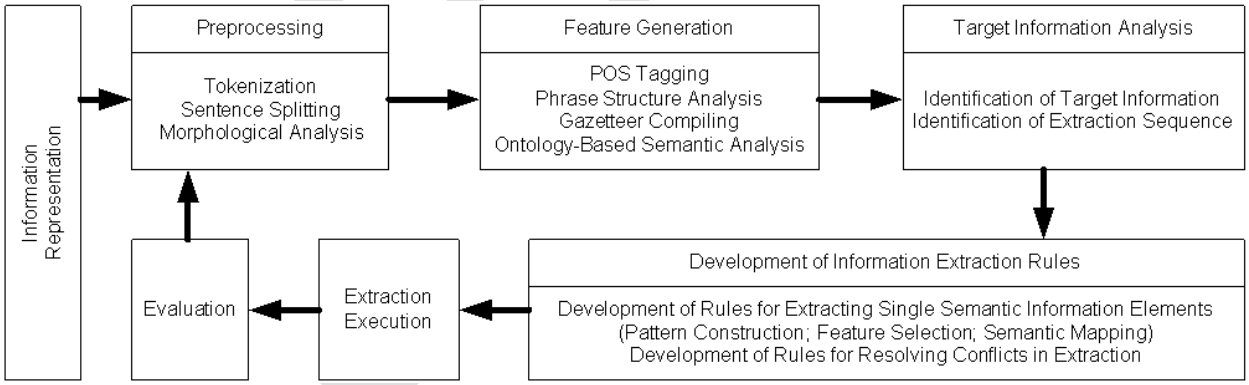

Figure 3: An illustrative example applying the proposed information extraction methodology

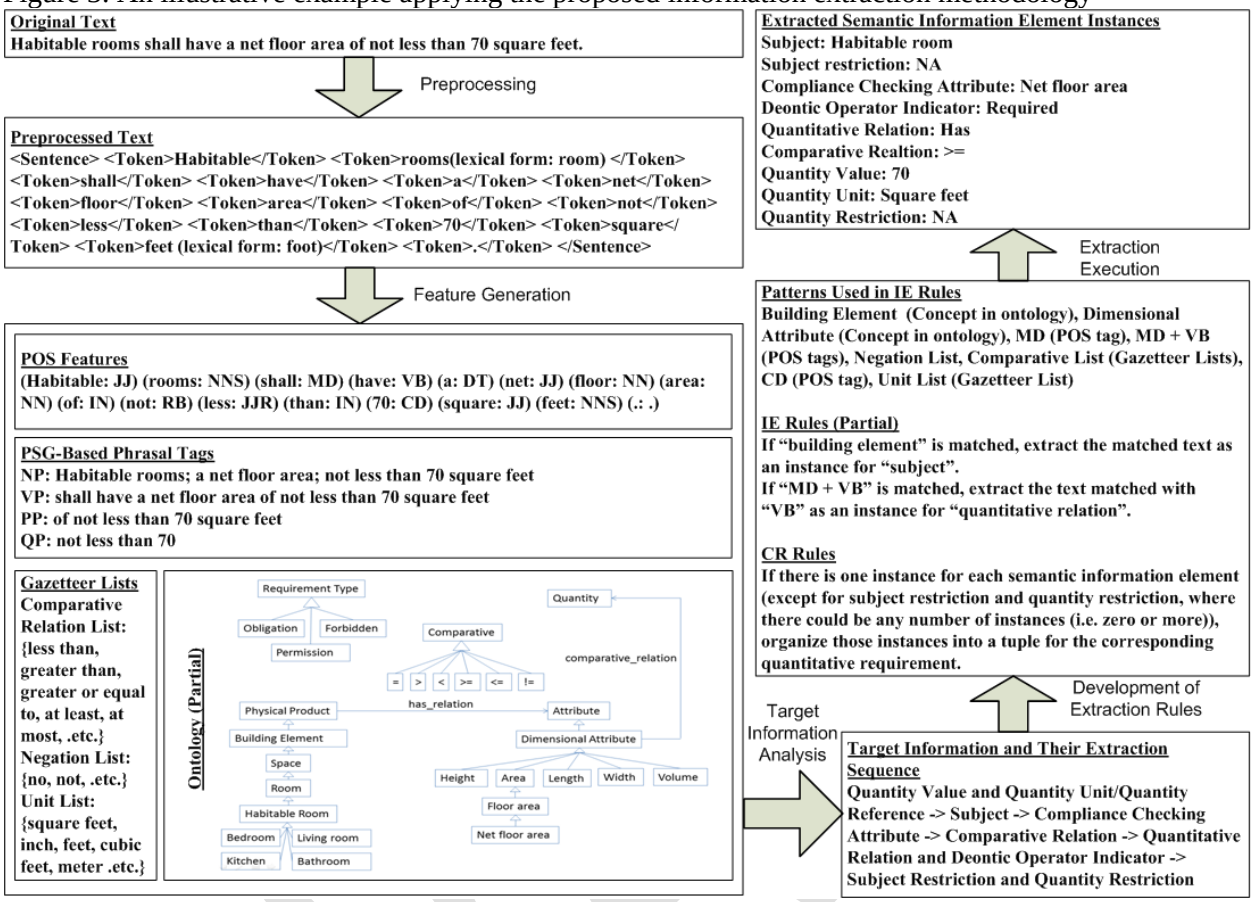

Figure 4: A sample information extraction rule (in English and Java coding)