AUTOMATED INFORMATION EXTRACTION FROM CONSTRUCTION-RELATED REGULATORY DOCUMENTS FOR AUTOMATED COMPLIANCE CHECKING

Jiansong Zhang, Graduate Student, jzhang70@illinois.edu
Nora El-Gohary, Assistant Professor, gohary@illinois.edu
University of Illinois at Urbana-Champaign, Urbana, Illinois, USA

ABSTRACT

Manual regulatory compliance checking is usually time-consuming, costly, and error-prone. Automating the process of compliance checking is expected to reduce the time and cost of the process, as well as reduce the probability of making compliance assessment errors. One aspect of automating the compliance checking process is automating the extraction of information (rules that the project needs to comply with) from construction-related regulatory documents (which are expressed in textual format). With the advancements in the artificial intelligence domain, natural language processing (NLP) techniques are being widely used in many fields for information extraction from unstructured text. There have been few research efforts to apply NLP techniques in the construction domain. However, none of these efforts attempted to automatically extract rules from textual regulatory documents.

In this paper, the authors propose an approach for semantic information extraction (using domain-specific meaning, in addition to syntax-related text features) to automatically extract information from construction-related regulatory documents and represent it in a computer-understandable, structured format. Preliminary experimental results are presented and discussed in the paper.

Keywords: Compliance Checking in Construction, Natural Language Processing, Information Extraction, Artificial Intelligence, Automation

1. INTRODUCTION

Construction projects are typically governed by various laws and regulations, such as the International Building Code (ICC 2006), the ADA Standards for Accessible Design (DOJ 1994), the International Fire Code (ICC 2006), the International Energy Conservation Code (ICC 2006), the OSHA's Cranes and Derricks in Construction (DOL-OSHA 2010), etc. Due to the variety and the large number of provisions in these regulations that are relevant to a typical construction project, the manual process of checking the conformance to these provisions is time-consuming, costly, and error-prone. With the development of information technology, there has been growing interests in automating this manual checking process (Tan, Hammad, and Fazio 2010; Eastman et al. 2009; Delis and Delis 1995; Fenves and Rasdorf 1985). Automating the manual checking process is expected to save time, reduce compliance checking errors (Han, Kunz, and Law 1997; Tan, Hammad, and Fazio 2010), and as a result reduce cost. Few tools for semi-automated code compliance checking have been developed, such as the Singapore's Automated Code Checking System CORENET (Khemlani 2005). However, these tools never achieved full automation of the checking process. They require manual extraction of rules from regulatory documents and manual encoding of these rules. Therefore, an automated method for extracting regulatory information from their original format and representing them in a computer-understandable format is needed to achieve full (or at least higher degree of) automation. Since typical construction-related regulations are represented in unstructured text format (e.g. *.doc, *.pdf, *.txt, etc.), a method for processing such unstructured text to extract needed information and represent that information into a computer-understandable format is essential. Information extraction (IE), a subfield of natural language processing (NLP), is a potential technique that can be applied to achieve this function.

In this paper the authors propose an approach utilizing semantic (using meaning/context-related features, in addition to syntax/grammar-related features) IE to extract information from construction-related regulatory documents and represent it in a computer-processable, semantic format to support automated compliance checking. Preliminary experiments were conducted to test the initial validity of the proposed approach. Section 2 introduces the background of automated compliance checking in the construction domain, the background of NLP and IE, and NLP efforts in the construction domain; Section 3 introduces our proposed approach; Section 4 presents the preliminary experimental results; Section 5 compares syntactic and semantic IE and discusses the potential effectiveness of the proposed approach; and Section 6 concludes the paper and discusses future work.

2. BACKGROUND

There has been significant research efforts to automate the code compliance checking process, such as code compliance checking of building envelope performance (Tan, Hammad, and Fazio 2007; Tan, Hammad, and Fazio 2010), fire code compliance checking (Delis and Delis 1995), and facility accessibility code compliance checking (Han, Kunz, and Law 1997). However, these approaches/tools require manual extraction of rules from regulatory documents and manual encoding of these rules.

Natural language processing (NLP) is a field utilizing artificial intelligence technology to enable computers to understand and process natural language text (and speech) in a human-like manner (Cherpas 1992). It is theoretically-based and practice-driven. Examples of NLP tasks include part of speech (POS) tagging, handwriting recognition, speech recognition, semantic role labeling, information extraction (IE), etc. (Marquez 2000; McCallum, Bellare, and Pereira 2005). POS tagging aims at tagging each word with the part-of-speech role it plays in the context, such as noun, verb, adjective, etc. Semantic role labeling aims at labeling each component (a word, phrase, or a clause) with its meaningful context-dependent role. For example, in the sentence "Jack opens the door," "Jack" could be labeled as the subject, "opens' could be labeled as the action, and "door" could be labeled as the object. IE aims at extracting structured data/information from unstructured text automatically. Many effective models/algorithms have been developed for those tasks.

IE applications are being widely used in many domains (Liddy 2003). The need for automated IE is increasing as a result of the growing amount of data/information. It is required where manual IE would be too time-demanding and/or too complex for human processing (e.g. extracting information – such as product features and consumer opinion – from the enormous amount of online review data) (Popescu 2007). IE approaches could be generally categorized into two types – syntactic-based and semantic-based. The Little Oxford dictionary (1986) defines syntactic knowledge as "the grammatical arrangement of words/rules or analysis of it" and semantic knowledge as "the meaning in language." We could simply say that a syntactic-based task is syntax/grammar-related only (e.g. POS tagging), while a semantic-based task is also meaning/context-related (e.g. semantic role labeling). In this paper, we refer to those IE methods utilizing only syntactic features as syntactic IE and those IE methods utilizing both syntactic and semantic features as semantic IE.

Semantic IE can be achieved through the use of ontologies. Ontologies are utilized to represent domain knowledge. Ontology-based IE is expected to have better performance in comparison to syntactic IE, because domain knowledge (represented in an ontology) could help to identify or distinguish domain-specific terms and meanings (Saggion et al. 2007; Soysal, Cicekli, and Baykal 2010). The term ontology, meaning "the study of being or existence," originated in philosophy. Recently this term migrated to the computer and information science domains to refer to "an explicit specification of a conceptualization" (Gruber 1995). This definition establishes the features of an ontology: 1) an ontology is representing a conceptualization (i.e. an abstract, simplified view of a domain of interest); and 2) the representation of the conceptualization is explicit. An ontological model consists of concept hierarchies, relationships (between the concepts), and axioms (Noy and Hafner 1997). The axioms are used together with the concepts and relationships to define the semantic meaning of the conceptualization (El-Gohary and El-Diraby 2010). An ontology offers a computer-understandable, domain-specific representation of the knowledge in a domain of interest, in a reusable, extendable format.

Nowadays many off-the-shelf NLP tools or tool sets have been developed such as the Stanford Parser by the Stanford Natural Language Processing Group, the OpenNLP tools by the Apache

Software Foundation, the GATE (General Architecture for Text Engineering) by the University of Sheffield, and Sundance by University of Utah (Riloff and Phillips 2004). Similarly, many tools are available to create and edit ontologies, such as Protégé by the Stanford Center for Biomedical Informatics Research.

There have been few research efforts to leverage NLP techniques in the construction domain; for example in project knowledge retrieval (Scherer and Reul 2000), document structure extraction (Kim et al. 2010), and concept relation extraction from contracts (Al Qady and Kandil 2010). However, none of these research efforts attempted to automatically extract rules from construction-related regulatory documents. As such, in this paper, we attempt to address this research gap by proposing a semantic approach for automated IE from textual construction-related regulatory documents and representation of extracted information in a computer-understandable, structured format.

3. PROPOSED APPROACH FOR AUTOMATED INFORMATION EXTRACTION FROM TEXTUAL CONSTRUCTION-RELATED REGULATIONS

This section introduces an approach for automated IE from construction-related regulatory documents (e.g. the International Building Code). The approach includes six phases: preprocessing, feature generation, matching pattern identification and feature selection, development of information extraction rules, extraction, and evaluation. These phases are arranged in a streamlined fashion while phases 3-6 could be iterated until a satisfying result is achieved (Figure 1). The approach is semantic; an ontology is developed and ontological features are used, in addition to syntactic features, in the extraction of information. Semantic IE is expected to improve the extraction performance, because it utilizes domain knowledge, in the analysis of text, in addition to syntactic information.

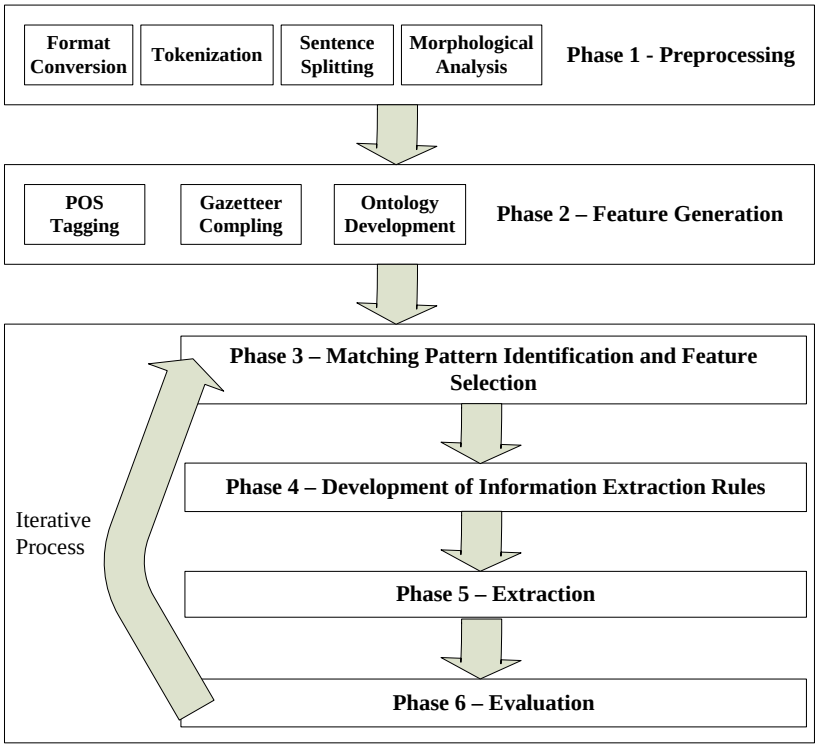

Figure 1: Proposed approach for information extraction.

3.1 Phase 1 - Preprocessing

This phase is intended to prepare the raw (i.e. unprocessed) text for further processing. Preprocessing generally consists of format conversion, tokenization, sentence splitting, and morphological analysis. Preprocessing may also include other case-specific processes, such as de-hyphenation (i.e. restoring the words split at boundaries of lines with hyphen to their original forms).

Format Conversion

All regulatory documents are in an unstructured (or semi-structured) textual format. However, they vary in the file format (e.g. *.pdf, *.txt, *.doc, etc.). This step aims at converting various file formats into a standardized *.txt format. This will facilitate easy manipulation and processing of the text.

Tokenization

Tokenization is the process of dividing the sequences of characters (pure strings) from raw 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 is typically conducted based on whitespaces and punctuations, because they are very good indicators of word boundaries. In the proposed approach, tokenization divides the sequences of characters into tokens. A token is a single word, a number, a punctuation, a white space, or a symbol (e.g. "&," "$").

Sentence Splitting

Sentence splitting aims at recognizing boundaries of sentences. It is very important in preparing the text for NLP tasks, such as syntactic parsing and IE (Mikheev 2000). Sentence splitting usually utilizes punctuations, such as periods, exclamation marks, and question marks (identified as a result of tokenization) to detect the boundaries. However, disambiguation is also needed because those punctuations not always indicate end of sentences (e.g. periods in abbreviations, periods as decimal points, etc.). In the proposed approach, the result of sentence splitting is a set of sentence segmentations (with recognized boundaries).

Morphological Analysis

Morphology refers to the study of composition and structure of words. Morphological analysis aims at recognizing the different forms of a word and mapping them to the standard form of that word in a dictionary (Kay 1973). Morphological analysis maps various nonstandard forms of a word (e.g. plural form of noun, past tense of verb) to its standard form (e.g. singular form of noun, infinitive form of verb). For example, "constructs," "constructed," and "constructing" are all mapped to "construct" in morphological analysis. In the proposed approach, the morphological analysis aids in the identification of concept terms and attribute/property values of the ontology.

3.2 Phase 2 - Feature Generation

This phase is intended to generate a feature repository to be used in the extraction process. It consists of POS tagging, gazetteer compiling, and ontology development.

POS Tagging

POS tagging aims at tagging each word with information explicitly indicating the structure inherent in the language, namely the part of speech of the word (also called word class, lexical class, or lexical category), such as noun, verb, adverb, etc. (Bellegarda 2010). In the proposed approach, the POS tagging process also tags other tokens, such as numbers, punctuations, and symbols.

Gazetteer Compiling

A gazetteer is a set of lists containing names of specific entities (i.e. 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. The use of a gazetteer is essential for automated IE, because it aids in recognizing terms based on those commonalities (Maynard, Bontcheva, and Cunningham 2004). For example, the "negation gazetteer list," utilized in the proposed approach, includes a list of words that indicate negation. Such gazetteer list is essential to distinguish between negated and non-negated segments. In the proposed approach, the gazetteer is used to provide a set of term lists - each list has a specific function. For example, terms like "no," "not," etc. have the function "negation," and as such, are included in the "negation gazetteer list." In the proposed approach, other types of gazetteer lists are also compiled and used, such as the "comparison gazetteer list," which is composed of terms indicating comparative relations, such as "greater or equal," "less or equal," "at most," "at least," etc.

Ontology Development

Ontologies are used to represent domain knowledge. Ontologies are composed of concept hierarchies, relationships between those concepts, and axioms defining the constraints and semantic meaning of concepts and relations (El-Gohary and El-Diraby 2010). In the proposed approach, an ontology will be developed and coded in OWL (Web Ontology Language), i.e. *.owl format. OWL has been selected for encoding the ontology, because it is the most widely-used semantic web standard. The ontology will offer a semantic representation of the knowledge in the construction domain; and thus will aid in extracting relevant information based on domain-specific meaning. This is expected to enhance the performance of the IE process. In developing the ontology, existing construction ontologies will be re-used as necessary (e.g. the IC-PRO-Onto (El-Gohary and El-Diraby 2010)).

3.3 Phase 3 – Matching Pattern Identification and Feature Selection

This phase aims at observing the target text segments (text segments where the information to be extracted exists) and manually identifying matching patterns based on the features generated in Phase 2. Matching patterns are the patterns expressed by sequences/structures of features. They match text segments to locate desired information for extraction. The process of matching pattern identification follows a heuristic approach. Features included in the identified matching patterns are selected together with the patterns. For example, in order to locate information of subject, action, and object in a sentence, the matching pattern "noun + verb + noun" might be identified. This would extract the following information from this sentence "Jack opens the door:" "Jack" is the subject, "opens" is the action, and "door" is the object. Further, if the IE task requires that the subject is a person, then some knowledge of the domain could be used and the matching pattern becomes "person + verb + noun" (e.g. the concept 'person' could be defined in an ontology). In general, this process is iterative, because it is difficult to achieve good extraction performance results from the first trial; the process requires fine-tuning and re-evaluation after each trial.

3.4 Phase 4 - Development of Information Extraction Rules

This phase aims at developing a set of information extraction rules using the patterns and features selected in Phase 3. Each rule is composed of two sides – left hand side and right hand side. The left hand side describes the pattern used to match the text to locate target information (the information to be extracted). The right hand side encodes the actions to be taken when the target information is located. A simple example left hand side is "noun + verb + noun." A corresponding example right hand side is "Put the first noun as the subject, put the verb as the action, and put the second noun as the object." This rule would extract all text segments that conform to the matching pattern "noun + verb + noun" and put the corresponding parts into the following classes: subjects, actions and objects.

3.5 Phase 5 - Extraction

This phase aims at applying the extraction rules developed in Phase 4 on the input text and extracting the desired information in a pre-defined, structured format. Ultimately, we aim at representing the extracted information in the form of an ontology. As an intermediate, experimental step, we will use tuple format (e.g. <Subject, Attribute, Comparison, Quantity>). In general, tuple format may be used as an intermediate processing step.

3.6 Phase 6 - Evaluation

The evaluation is conducted using the following indicators: Recall (R), Precision (P), and F-measure (F) (Equations (1) ~ (3)).

$$R = \frac{\textit{correct information items extracted}}{\textit{total information items existing}}, \quad (1)$$

$$P = \frac{\textit{correct information items extracted}}{\textit{total information items extracted}}, \quad (2)$$

$$F = \frac{PR}{(1-\alpha)P+\alpha R} \ , \ \textit{where } 0 \leq \alpha \leq 1 \quad (3)$$

An information item is defined, in this paper, as a component of an instance of the target information. For example, if we want to extract construction accident information from text, then each specific accident event is one instance of the target information, and each detail of the event such as time, location, witness, victim, etc., is a component of that accident event. Recall is defined as the percentage of correctly extracted information items relative to the total number of information items existing in the text. Precision is defined as the percentage of correctly extracted information items relative to the total number of information items extracted (Maynard, Peters, and Li 2006). There is a trade-off between recall and precision. As such, using either indicator alone is not sufficient. Thus, F-measure is defined as a weighted combination (harmonic mean) of recall and precision (Makhoul et al. 1999). In the proposed approach, we set α to 0.5 to give equal weights to recall and precision.

4. PRELIMINARY EXPERIMENTAL RESULTS

The proposed approach is intended for extracting information from a variety of construction-related regulatory documents (e.g. building codes, environmental regulations, safety regulations and standards, etc.). The authors started with testing the proposed approach on building codes. In the future, the approach/methodology will be tested on different regulatory documents. The International Building Code was selected because it is the building code adopted in most parts of the U.S. The authors used the International Building Code 2006. Section 1208 (Interior Space Dimensions) was randomly selected as a sample text. We may classify the requirements in this section into two types: 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 or equal than 7'6"; and 2) "Existential Requirement" which requires the existence of certain building elements/parts. 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. The authors decided to experiment on the extraction of quantitative requirements, because: 1) most of the requirements identified in the section are quantitative requirements; and 2) the sentences describing quantitative requirements appear to be more complex than those describing existential requirements which implies that they are more difficult to extract. The ultimate goal of the proposed extraction approach is to represent the extracted information into a semantic format. For initial testing purposes, we will use a tuple format to represent the extracted information (instead of an ontology), because it is simple for computer manipulation. Ultimately, the output information will be represented in an ontological format. The following is an example of a four-tuple format that we used: <Subject, Attribute, Comparison, Quantity>." A subject is a 'thing' (e.g. building object, space, etc.) that is subject to a particular regulation or norm – for this tuple format the 'subject' is subject to a quantitative requirement. Attribute is the property of the subject that is relevant to the quantitative requirement. Comparison is the comparative relation such as greater, less or equal, etc. Quantity is the value or range for the attribute of the subject.

The experiment was conducted using the GATE tool because: 1) It has been widely and successfully used in IE, such as in Soysal, Cicekli, and Baykal (2010) and Saggion et al. (2007); 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: 1) ANNIE (a Nearly-New Information Extraction System) system was used for tokenization, sentence splitting, POS tagging, and gazetteer compiling, 2) built-in morphological analyzer was used for morphological analysis, 3) built-in ontology editor was used for ontology development, and 4) JAPE (Java Annotation Patterns Engine) transducer was used to write information extraction rules. The format conversion, on the other hand, was simply conducted by manually saving a *.pdf file as a *.txt file. For experimental purposes, the authors developed and used (as a starting point) a small-size, non-detailed ontology.

The 'gold standard' for evaluation was constructed manually. The authors read through the text and identified all quantitative requirements and manually wrote them down in tuple format. In this experiment, the recall and precision were evaluated on information component level (i.e. counting all information components – namely subject, attribute, comparison, and quantity – as information items).

As such, for recall and precision evaluation, a tuple correctly extracted would add four to the number of correctly extracted information items (since it has four information items).

For comparative reasons, we started with conducting syntactic IE from the sample text (Section 1208), followed by semantic IE. The aim of this comparative experiment was to verify that semantic IE will result in better performance, in comparison to syntactic, as initially expected. The results of the experiment on the sample text are shown in Table 1: 1) 75% and 95% recall for syntactic and semantic IE, respectively, 2) 75% and 100% precision for syntactic and semantic IE, respectively, and 3) 0.750 and 0.974 F-measure for syntactic and semantic IE, respectively. Table 2 shows some examples of the features used in syntactic IE. For example, the feature "Cardinal Number (CD)" are POS tags representing cardinal number, which are used to detect numbers in the text. On the other hand, Figure 2 shows a partial view of the ontology that was used in semantic IE; and Table 3 shows some examples of the results of semantic IE extraction from the sample text.

Table 1: Initial experimental results - IE from sample text (Section 1208).

IE Method	Recall	Precision	F-measure
Syntactic IE	75%	75%	0.750
Semantic IE	95%	100%	0.974

Table 2: Examples of syntactic features used in IE.

Feature	Category	Feature Description	Function
Comparison	Gazetteer Feature	one of the following words or phrases: "above," "at least," "at most," "below," "equal to," "greater than," "greater or equal," "less or equal," "less than," "maximum," "minimum," "more than"	Detect quantitative relationship
Modal Auxiliary Verb (MD)	POS Feature	modal auxiliary verb such as "can," "could," "dare," "may," "might," "must," "ought," "shall," "should," "will," "would"	Detect normative requirement
Negation	Gazetteer Feature	the string "no" or "not"	Detect negation
Cardinal Number (CD)	POS Feature	cardinal number	Detect number
Unit	Gazetteer Feature	strings representing units such as "feet," "foot," "square feet," "sf," "sq.ft.," "inch," "inches," "in.,"	Detect unit of quantity
Quantity	Compound Feature	Cardinal + Unit	Detect quantity

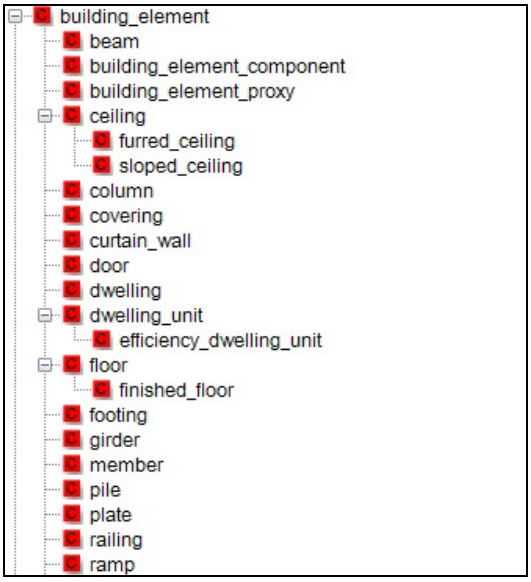

Figure 2: A partial view of ontology concepts (represented in GATE).

Table 3: Examples of semantic IE extraction results from sample text (Section 1208).

Example Number	Subject	Attribute	Comparison	Quantity
1	occupiable space	ceiling height	not less than	7 feet 6 inches
	habitable space			
	corridor			
2	furred ceiling	height	not less than	7 feet
3	room	net floor area	not less than	13.9 m2

5. DISCUSSION

Initial, preliminary experimental results indicate that semantic IE has significantly better recall and precision (95% and 100%, respectively) than syntactic IE (75% for both). As shown in Table 4, semantic IE performs better than syntactic IE, because syntactic IE may miss parts (or all) of the information items and/or extract concepts and/or attributes incorrectly. For example, Table 4 shows that syntactic IE incorrectly extracted "space" instead of "occupiable space," "ceiling" instead of "ceiling height," and "floor" instead of "net floor area." It is very difficult (if not impossible) to use syntactic features to extract those concept and attribute information perfectly, because that needs exhaustive enumeration of all possible matching patterns based on those syntactic features. It is also very difficult (if not impossible) to distinguish an attribute from its constituent terms (e.g. "net floor area" from "floor") based on their syntactic features like POS tags, while this is relatively easy based on domain knowledge (represented in an ontology). As such, the initial experimental results show that the use of domain knowledge in the IE task is vital.

As such, based on the initial experimental results, our proposed automated IE approach seems potentially effective in extracting information from construction-related regulatory documents. However, it is premature to generalize the results beyond this specific experiment. Further testing and experimentation is needed on different types of regulatory documents, and on larger samples of text.

Table 4: Examples of subjects and attributes extracted using syntactic and semantic IE methods.

Example No.	Text	IE Method	Subject Extracted	Attribute Extracted
1	Occupiable spaces, habitable spaces and corridors shall have a ceiling height of not less than 7 feet 6 inches (2286 mm).	Syntactic	space	ceiling
			space	
			corridor	
		Semantic	occupiable space	ceiling height
			habitable space	
			corridor	
2	in no case shall the height of the furred ceiling be less than 7 feet (2134 mm).	Syntactic	ceiling	height
		Semantic	furred ceiling	height
3	The unit shall have a living room of not less than 220 square feet (20.4 m2) of floor area.	Syntactic	room	floor
		Semantic	living room	net floor area

6. CONCLUSION & FUTURE WORK

In this paper, the authors propose an approach for automatically extracting information (rules) from construction-related regulatory documents to support the process of automated regulatory compliance checking. The approach is semantic, i.e. it utilizes domain-specific meaning (in the form of a semantic, ontological model), in addition to syntax-related text features, for automatically extracting information from unstructured text and representing such information in a semantic format (an ontological format). Preliminary experimental results are presented and discussed in the paper. Initial experimental results indicate that semantic IE has better performance than syntactic IE (utilizing syntax-related features only). A semantic approach is potentially capable of extracting information "more intelligently" based on an understanding of domain-specific terms and contexts. As such, based on the initial experimental

results, our proposed approach seems potentially effective in extracting information from construction-related regulatory documents. However, prior to the generalization of results, further experimentation is needed on different types of regulatory documents, and on larger samples of text. In the future, the authors will explore the extraction of information (rules) from other construction documents - which could be different in content and structure from regulatory documents - such as construction contract agreements, special conditions, specifications, etc. Our future work on automated compliance checking will also involve checking of designs (e.g. Building Information Modelling (BIM)-based design models) and construction operational plans (e.g. construction environmental plans, safety plans, etc.) for compliance to extracted rules.

REFERENCES

Al Qady, M.A., and Kandil, A. (2010) "Concept relation extraction from construction documents using natural language processing." *Journal of Construction Engineering and Management*, 136(3): 294-302.

Bellegarda, J.R. (2010) "Part-of-Speech tagging by latent analogy." *IEEE Journal of Selected Topics in Signal Processing*, 4(6): 985-993.

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

Cunningham, H. et al. (2011) "Developing language processing components with gate version 6 (a user guide)."

Delis, E.A., and Delis, A. (1995) "Automatic fire-code checking using expert-system technology." *Journal of Computing in Civil Engineering*, 9(2):141-156.

Department of Justice (DOJ). (1994) "ADA standards for accessible design." *In 28 CFR Part 36.*

Department of Labor - Occupational Safety and Health Administration (DOL-OSHA). (2010) "Cranes and derricks in construction; final rule." *In 29 CFR Part 1926.*

Eastman, C., Lee, J., Jeong, Y., and Lee, J. (2009) "Automatic rule-based checking of building designs." *Automation in Construction*, 18(8):1011-1033.

El-Gohary, N.M., and El-Diraby, T.E. (2010) "Domain ontology for processes in infrastructure and construction." *Journal of Construction Engineering and Management*, 136(7):730–744.

Fenves, J.S., and Rasdorf, W.J. (1985) "Treatment of engineering design constraints in a relational database." *Engineering with Computers*, 1(1):27-37.

Grefenstette, G., and Tapanainen, P. (1994) "What is a word, what is a sentence? Problems of tokenization." *In Proceedings of the 3rd Conference on Computational Lexicography and Text Research (COMPLEX'94)*, 1-11.

Gruber, T.R. (1995) "Toward principles for the design of ontologies used for knowledge sharing." *International Journal of Human-Computer Studies*, 43: 907–928.

Han, C.S., Kunz, J., and Law, K.H. (1997) "Making automated building code checking a reality." *Management Journal*, IFMA September/October: 1-7.

Hjelseth, E., and Nisbet, N. (2010) "Exploring semantic based model checking." *In Proceedings of the 2010 27th CIB W78 International Conference, No.54.*

International Code Council (ICC). (2006) "2006 International Building Code."

International Code Council (ICC). (2006) "2006 International Energy Conservation Code."

International Code Council (ICC). (2006) "2006 International Fire Code."

Kay, M. (1973) "Morphological analysis." *In Proceedings of the International Conference on Computational Linguistics*, 205-223.

Khemlani, K. (2005) "CORENET e-PlanCheck: Singapore's automated code checking system." *AECBytes, http://www.aecbytes.com/buildingthefuture/2005/CORENETePlanCheck.html.*

Kim, B., Park, S., Kim, H., and Lee, S. (2010) "Automatic extraction of apparent semantic structure from text contents of a structural calculation document." *Journal of Computing in Civil Engineering*, 24(3): 313-324.

Liddy, E.D. (2003) "Natural language processing," In Encyclopedia of Library and Information Science, 2nd Ed., Marcel Decker, Inc. NY.

Makhoul, J., Kubala, F., Schwartz, R., and Weischedel, R. (1999) "Performance measures for information extraction." *In Proceedings of the DARPA Broadcast News Workshop*, Hemdon, VA.

Marquez, L. (2000) “Machine learning and natural language processing.” *In Proceedings of the Conference “Aprendizaje automatico aplicado al procesamiento del lenguaje natural”*.

Maynard, D., Bontcheva, K., and Cunningham, H. (2004) “Automatic language-independent induction of gazetteer lists.” *In Proceedings of the 4th Language Resources and Evaluation Conference (LREC'04).*

Maynard, D., Peters, W., and Li, Y. (2006) “Metrics for evaluation of ontology-based information extraction.” *In Proceedings of the WWW 2006 Workshop on “Evaluation of Ontologies for the Web”*.

McCallum, A., Bellare, K., and Pereira, F. (2005) “A conditional random field for discriminatively-trained finite-state string edit distance.” *In Proceedings of the Uncertainty in AI Conference.*

Mikheev, A. (2000) “Tagging sentence boundaries.” *In Proceedings of the 1st Meeting of the North American Chapter of the Association of Computational Linguistics (NAACL 2000)*, 264–271.

Noy, N.F., and Hafner, C.D. (1997) “The state of the art in ontology design: A survey and comparative review.” *AI Magazine*, 18(3): 53–74.

Riloff, E., and Phillips,W. (2004) “An introduction to the Sundance and AutoSlog systems.” School of Computing, University of Utah; Technical Report UUCS-04-015.

Saggion, H., Funk, A., Maynard, D., and Bontcheva, K. (2007) “Ontology-based information extraction for business intelligence.” *In Proceedings of the ISWC/ASWC 2007 Conference, LNCS 4825*, 843–856.

Santos, I.A., and Farinha, F. (2005) “Code checking automation in building design: new trends for cognition.” *In Proceedings of the 2005 ASCE International Conference on Computing in Civil Engineering*, 143-150

Scherer, R.J., and Reul, S. (2000) “Retrieval of project knowledge from heterogeneous AEC documents.” *In Proceedings of the Computing in Civil and Building Engineering Conference*, 812-819.

Soysal, E., Cicekli, I., and Baykal, N. (2010) “Design and evaluation of an ontology based information extraction system for radiological reports.” *Computers in Biology and Medicine*. 40: 900-911.

Tan, X., Hammad, A., and Fazio, P. (2010) “Automated code compliance checking for building envelope design.” *Journal of Computing in Civil Engineering*, 24(2):203-211.

Tan, X., Hammad, A., and Fazio, P. (2007) “Automated code compliance checking of building envelope performance.” *Journal of Computing in Civil Engineering*, 256-263.

Popescu, A. (2007) “Information extraction from unstructured web text.” PhD thesis, University of Washington.