Automated Regulatory Information Extraction from Building Codes
Leveraging Syntactic and Semantic Information

Jiansong ZHANG¹, Nora EL-GOHARY²

¹ Graduate Student, Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, 205 North Mathews Avenue, Urbana, IL 61801; PH (614) 886-2868; FAX (217) 265-8039; email: jzhang70@illinois.edu
² Assistant Professor, Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, 205 North Mathews Avenue, Urbana, IL 61801; PH (217) 333-6620; FAX (217) 265-8039; email: gohary@illinois.edu

ABSTRACT

Manual regulatory compliance checking of construction projects is usually time-consuming and error-prone. There have been efforts both in academia and industry to automate this process. However, none of them achieved full automation. Specifically, the extraction of rules from regulatory text (e.g. building code) and its representation in a computer-processable format is still conducted manually or semi-automatically.

Natural language processing (NLP) aims at enabling computers to process natural language text in a human-like manner. It provides basic concepts and methods for text processing and analysis, such as part of speech (POS) tagging, tokenization, sentence splitting, named entity recognition, and semantic role labeling, etc.

This paper is intended to explore the effectiveness of utilizing syntactic (i.e. grammatical) and semantic (i.e. meaning descriptive) features of the text (using NLP tools and techniques) to automatically extract regulatory information from building codes. An automated information extraction (IE) approach – involving the use of IE rules – is proposed. Chapter 12 of the 2006 International Building Code was used to develop the IE rules, while Chapter 12 of the 2009 International Fire Code was used to test the approach. An overall F-measure of 0.94 shows the potential of the proposed approach. Based on the experimental results and their analysis, we conclude the paper by pinpointing possible ways for improving the proposed approach.

INTRODUCTION

There has been much research effort in the domain of automated compliance checking (ACC) in construction. Generally, ACC serves two purposes: 1) assisting authorities in decision-making (e.g. permit approval), and 2) assisting designers and/or contractors in verifying compliance to applicable laws and regulations. ACC efforts in the building domain included checking requirements of safety and reliability of structures (Garrett and Fenves 1987), accessibility (Lau and Law 2004), egress, environmental protection, and energy conservation (FIC 2007), etc. Previous ACC efforts have used several representation formats to encode regulatory requirements/rules, such as the SASE (Standards Analysis, Synthesis and Expression) representation developed by NBS (now NIST) (Garrett and Fenves 1987), and the semantically tagged/annotated documents (e.g. XML) representation as in (Yurchyshyna and Zarli 2009). However, the extraction of rules from regulatory text (e.g. building code) and its representation in a computer-processable format is still
conducted manually or semi-automatically. As such, in this paper, the authors are addressing this gap by proposing an automated approach for extracting regulatory requirements from building codes to support automatic regulatory compliance checking. Various tools and techniques from NLP and IE fields are utilized in our approach.

BACKGROUND

Natural Language Processing & Information Extraction

Natural Language Processing (NLP) aims at enabling the computer to understand and process natural language text (or speeches) in a human-like manner. Examples of NLP tasks are part-of-speech (POS) tagging, co-reference resolution, text summarization, and machine translation, etc. (Marquez 2000).

Information Extraction (IE) is a sub field of NLP that deals with extracting desired information from text sources to fill in predefined information templates. Pattern matching is the core technique in IE. A variety of text features could be utilized in pattern matching such as tokens, POS tags, text structural information, and semantic information. Semantic-based IE (i.e. IE utilizing semantic (i.e. meaning descriptive) features in addition to syntactic (i.e. grammatical) features) has been shown to have better performance in comparison to syntactic-only-based IE (i.e. IE utilizing only syntactic features) (Soysal et al. 2010).

Context Free Grammar

Context Free Grammar (CFG) is a grammar that could be used to derive sentences of a language. A CFG rule takes the form \( X \rightarrow w \) where \( X \) is a non-terminal (i.e. symbols that can be further broken down) and \( w \) is a sequence of terminals (i.e. literal symbols that cannot be further broken down) and non-terminals (Afrin 2001). The idea of CFG is illustrated in Figure 1. In Figure 1, the left part shows a sample set of CFG rules. The right part shows a derivation tree for the sentence “Courts shall not be less than 3 feet in width” using CFG rules. For example, applying the first CFG rule: ‘Sentence’ \( \rightarrow \) NP VP, the non-terminals ‘noun phrase’ (NP) and ‘verb phrase’ (VP) could be derived from the root node (i.e. the non-terminal ‘Sentence’) at the top of the derivation tree. Since non-terminals could be derived from themselves (e.g. a ‘noun phrase’ (NP) followed by a ‘prepositional phrase’ (PP) could be derived from a ‘noun phrase’), CFG is able to represent complex sentence structures (constructed by recursively nesting clauses and phrases). Thus, when used together with phrasal and clausal tag features for IE, this expressive power of CFG could significantly reduce the needed enumerations of matching patterns (as further explained below).

NLP and IE Efforts in the Civil Engineering Domain

Few researches in the civil engineering domain have utilized tools and techniques from NLP and IE fields. For example, shallow parsers were used for extracting concepts and relations from construction contract documents (Al Qady and Kandil 2010); text file parsers were used for extracting heading symbols and headings from structural calculation documents (Kim et al. 2010); text analysis and statistical techniques were used to build meta-data models of Request for Information (RFI)
questions (Zhu et al. 2007); machine-learning-based text classification methods were used for classifying construction documents (Caldas and Soibelman 2003); IE techniques were used to extract terms and relations from HTML documents for constructing a civil engineering thesaurus (Abuzir and Abuzir 2002); and text analysis and knowledge extraction methods were proposed for use in developing a project knowledge retrieval system (Scherer and Reul 2000). In comparison to these efforts, in this research, we are dealing with a different application (automated compliance checking); we are addressing a different level of NLP/IE task (i.e. processing text to automatically extract requirements/rules and represent them in a semantic format); and we are taking a deeper semantic approach for NLP (utilizing a detailed ontology for identifying semantic text features).

Fig. 1. A sample set of CFG rules and derivation of a sentence

PROPOSED APPROACH FOR AUTOMATED REGULATORY INFORMATION EXTRACTION

The proposed approach is summarized in Figure 2. The last three phases of the approach are iterative. Our approach uses semantic and syntactic text features, text feature pattern matching, and sequential information extraction. Semantic text features are defined based on a domain ontology. CFG is used to reduce the number of possible text matching patterns by defining complex syntactic features (i.e. phrasal and clausal features) which are more expressive than simple syntactic features (e.g. POS features). Reducing the number of possible text matching patterns is essential for reducing human rule development effort.

Phase 1: Preprocessing

Tokenization

Tokenization is the process of dividing the text into tokens. A token is typically a single word, a number, a punctuation, a white space, or a symbol (e.g. “&”, “$”). This process is conducted based on parsing the text according to common delimiters (i.e. white spaces and punctuations) and some disambiguation distinctions (e.g. “,” as delimiter in a number instead of punctuation).
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 (e.g. periods and question marks) with disambiguation consideration (e.g. recognizing “.” as the decimal point in a number instead of recognizing it as a period).

Morphological Analysis

Morphological analysis (MA) finds the root form of a word. This process is essential when different forms of a word (e.g. different grammatical numbers, senses, etc.) need to be treated equally. By virtue of MA, different forms (e.g. plural form, single form, etc.) of a concept could be recognized based on one concept in the ontology.

Phase 2: Feature Generation

This phase aims at generating features that could be used in the matching patterns of extraction rules. POS tags and phrasal and clausal structure information are widely used syntactic features for IE, such as in (Afrin 2001) and (Al Qady and Kandil 2010). Semantic features benefit IE tasks because they express domain-specific meaning/knowledge, such as in (Soysal et al. 2010). In this approach, we generate both syntactic and semantic features.

Figure 2. Proposed approach for automated regulatory information extraction
Part-of-Speech (POS) Tagging

Part-of-speech tags are the labels assigned to each word of a sentence indicating their lexical and functional categories. Typical part of speech tags include NN (singular nouns), JJ (adjectives), VB (verb), CC (coordinating conjunctions), etc. (Galasso 2002). In our proposed approach, the POS tags (generated as a result of POS tagging) are further used as features in subsequent IE tasks.

Phrase and Clause Structure Analysis (Using CFG)

This step builds on the POS tagging step and aims at assigning type labels (phrasal and/or clausal tags) to phrases and clauses of a sentence. All tags generated could be used as features in IE tasks. Phrasal tags and clausal tags together with CFG reduce the possible number of enumerations in matching patterns of IE. For example, the three CFG 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 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. Examples of phrasal tags are NP (noun phrase), VP (verb phrase), and PP (prepositional phrase), etc. Examples of clausal tags are S (simple declarative sentence), SBAR (clause introduced by a subordinating conjunction), and SQ (clause led by wh-phrase) (Bies et al. 1995). In this step, CFG will be deterministically derived from readily tagged source text, and any substructure of the derived CFG could be used to assign phrasal and/or clausal tags to sentences in the text source.

Gazetteer Compiling

A gazetteer is a set of lists grouping words or phrases according to some specific categories. The information that a word or phrase belongs to a certain list in the gazetteer could be used as a feature when extraction tasks are conducted. 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 General Architecture for Text Engineering).

Ontology Development

In the domain of information/knowledge modeling, the term “Ontology” refers to “an explicit specification of a conceptualization” (Gruber 1995). An ontology models domain knowledge in the form of concept hierarchies, relationships (between concepts), and axioms (El-Gohary and El-Diraby 2010). In the proposed approach, an ontology for the building domain is developed and used for semantic-based information extraction. Concepts and relations modeled in the ontology are utilized as features in subsequent extraction tasks.

Phase 3: Target Information Analysis

Target Information Identification

Each information item extracted (e.g. a noun, verb, phrase, etc.) corresponds to a semantic information element (concept or relation). For the remainder of this paper, we call a concept or relation an “information element”. In this step, the types of concepts and relations to be extracted will be identified based on the type of requirement to be checked. For example, in the sentence “Courts shall not be less than
3 feet in width”, “courts”, “not less than”, “3 feet”, and “width” are extracted and correspond to ‘subject’, ‘comparative relation’, ‘quantity’, and ‘compliance checking attribute’, respectively.

Extraction Sequence Resolution

In experimental studies we found that extracting all information elements by a single pattern matching on a sentence is not quite efficient, because the amount of possible patterns increase drastically with the number of included information elements. Since there is some independency (while not fully independent) between the variations of information elements, we propose to extract information elements separately and sequentially. The decision on the ordering of extraction for different information elements is based on: 1) the level of difficulty for extraction: the easiest 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 dependency between extraction of the different information elements. For example, if the extraction of the ‘compliance checking attribute’ needs a NP (noun phrase) (a feature generated for extracting the ‘subject’), and the level of difficulty for extracting a ‘quantity’ is lower than that for extracting the ‘subject’, then the relative extraction sequence of these three information items should be: ‘quantity’– ‘subject’ – ‘compliance checking attribute’.

Phase 4 –Development of Extraction Rules

Development of Extraction Rules – for Extracting Single Information Elements

The extraction 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 to be extracted. Matching pattern construction and feature selection should be conducted for each information element identified as target information, following the sequence given in extraction sequence resolution.

- Matching Pattern Construction: The matching patterns take the format of a sequential combination of features (e.g. the pattern “NP VP” matches a sentence as in Figure 1). The construction of such matching patterns is an iterative, empirical process (using initial manual text analysis, initial matching pattern construction, testing and results analysis, testing-based improvement of constructed patterns, etc.).
- Feature Selection: This step aims at selecting all features that are represented in the constructed patterns, for use in further IE tasks.

Development of Rules for Resolving Conflicts in Extraction

The rules for resolving conflicts in extraction – mainly – solve three types of conflicts: 1) multiple instances of an information element in a single sentence, 2) no instances of an information element in a single sentence, and 3) overlap of extraction results for different information elements. Each type of conflict may be resolved using one of a set of actions. For conflict 1, two actions may be used: a) keep all instances; or b) set priority rules and select the instances with higher priority (e.g. set a higher priority for “not less than” comparing to “above” when used as a comparative relation instance. For example, in the part of sentence “nonabsorbent
surface to a height not less than 70 inches above the drain inlet”, the comparative relation extracted would be “not less than” only). For conflict 2, three actions may be used: a) set a default instance based on domain knowledge (e.g. the default comparative relation may be set to “not less than” when there is no 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 “not less than” would be used as comparative relation instance); b) use the same instance from the nearest sentence/clause (left or right) if those sentences/clauses are describing the same contents (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 relations should be used for the second quantitative relations as well); or c) drop this sentence. For conflict 3, three actions may be used: a) delete any overlapping instance except for one; b) keep all instances; or c) select some instances to keep. Defining which action to execute is based on the type of conflict pattern. The conflict patterns and corresponding actions are encoded as rules.

**Phase 5: Extraction Execution**

This phase aims at extracting the target information instances from the regulatory text using the rules developed in Phase 4. In future work, the extracted information will be used to populate our semantic model for automatic compliance checking (ACC) for further compliance reasoning (Salama and El-Gohary 2011). For preliminary evaluation, we represented the extracted information in tuple-format.

**Phase 6: 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 will be manually (or semi-automatically with help of NLP tools) compiled by domain experts. Evaluation is conducted using the following measures: recall, precision, and F-measure. Recall is defined as the percentage of correctly extracted instances relative to the total number of instances existing in the text. Precision is defined as the percentage of correctly extracted instances relative to the total number of instances extracted. F-measure is a weighted combination of recall and precision (Makhoul et al. 1999).

**PRELIMINARY EXPERIMENT RESULTS AND ANALYSIS**

Chapter 12 of the 2006 International Building Code (ICC 2009) was used for developing the extraction rules, while Chapter 12 of the 2009 International Fire Code (ICC 2009) was used for testing the IE approach. The testing focused on extracting quantitative requirements. We used GATE (General Architecture for Text Engineering) for implementing and testing our approach. In this paper, we define a quantitative requirement as a rule that defines a relationship between a quantitative attribute of a subject and a specific quantity. We started testing our approach on quantitative requirements, because their embedded information is relatively straightforward and easier for computers to process in comparison to non-quantitative
requirements (e.g., procedural requirements). As such, this offers a good starting point for testing. On the other hand, GATE was selected, because it has a variety of built-in tools that could be used for a variety of NLP functions (e.g., tokenization, sentence splitting, POS tagging, gazetteer compiling, MA, ontology editing). For experimental purposes, we have developed and used a small-size ontology. The ontology is based on the IC-PRO-Onto (El-Gohary and El-Diraby 2010). ANNE in GATE was used for all preprocessing tasks (except MA), POS tagging, and gazetteer compiling. The built-in morphological analyzer in GATE was used for MA; the built-in ontology editor in GATE was used for ontology development; and JAPE (Java Annotation Patterns Engine) transducer was used to write IE rules. Some CFG rules that were derived are shown in Figure 3. The preliminary experimental results on quantitative requirements are shown in Table 1.

![Figure 3. Part of the derived CFG](image)

Table 1. Preliminary Experimental Results

<table>
<thead>
<tr>
<th></th>
<th>Subject</th>
<th>Compliance Checking Attribute</th>
<th>Comparative Relation</th>
<th>Quantity</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of instances in gold standard</td>
<td>21</td>
<td>17</td>
<td>21</td>
<td>21</td>
<td>80</td>
</tr>
<tr>
<td>Total number of instances extracted</td>
<td>21</td>
<td>17</td>
<td>21</td>
<td>21</td>
<td>79</td>
</tr>
<tr>
<td>Number of instances correctly extracted</td>
<td>20</td>
<td>16</td>
<td>21</td>
<td>18</td>
<td>75</td>
</tr>
<tr>
<td>Precision</td>
<td>0.95</td>
<td>1.00</td>
<td>1.00</td>
<td>0.86</td>
<td>0.95</td>
</tr>
<tr>
<td>Recall</td>
<td>0.95</td>
<td>0.94</td>
<td>1.00</td>
<td>0.86</td>
<td>0.94</td>
</tr>
<tr>
<td>F-measure</td>
<td>0.95</td>
<td>0.96</td>
<td>1.00</td>
<td>0.86</td>
<td>0.94</td>
</tr>
</tbody>
</table>

Four information elements were recognized which are ‘subject’, ‘compliance checking attribute’, ‘comparative relation’, and ‘quantity’. A ‘subject’ is a ‘thing’ that is subject to a particular regulation (Salama and El-Gohary 2011). In the context of compliance checking with the building code, it is typically a building element, facility, room, or space, etc. A ‘compliance checking attribute’, in this context, is the characteristic of the subject that could be defined by a specific quantity, such as length, width, etc. We define an ‘information set’ as a combination of instances for all identified information elements, one instance from each of the identified information elements. Our gold standard for Chapter 12 of the 2009 International Fire Code includes 21 information sets. Twenty-six (26), 17, 2, and 8 matching patterns were constructed for ‘quantity’, ‘subject’, ‘comparative relation’, and ‘compliance checking attribute’, respectively. Twenty-two (22) POS tags, 3 phrasal tags, 2 gazetteer lists, 21 tokens, and 106 concepts from the ontology were selected as features. Six rules for resolving conflicts in extraction were developed. The precision,
recall, and F-measure are 0.95, 0.94, and 0.94, respectively. Based on these preliminary results, our proposed IE approach seems potentially effective in automatically extracting information from building codes. However, further experimentation on different types of requirements, other building codes, and larger samples of text is required prior to the generalization of results.

Through the analysis of the preliminary experiment results, the causes for the IE errors are found to be: 1) inner flaw of the tools used (e.g. “beams” was POS-tagged as a verb while it should be a noun); 2) demand of inter-sentence co-reference resolution which is not dealt with in the current version of our approach (e.g. in “Any room with a furred ceiling shall be required to have the minimum ceiling height in two thirds of the area thereof” the term “thereof” refers to the subject “room with a furred ceiling” semantically); and 3) term ambiguity (e.g. “above” could be used to define a comparative relation while also could be used to indicate a location).

CONCLUSION AND FUTURE WORK

This paper presented an approach for automated IE from building codes for supporting automated compliance checking in construction. The approach utilizes both syntactic and semantic features of the text. Various NLP techniques were used for generating text features, including tokenization, sentence splitting, morphological analysis, POS tagging, phrase and clause structural analysis, and gazetteer lists. An ontology was developed and used for generating and extracting the semantic features of the text. CFG was used to handle the complexity of sentence structures. The phrasal and clausal features supported by CFG significantly reduce the possible number of enumerations of matching patterns in IE. Different information elements were extracted sequentially using both information extraction rules and rules for resolving conflicts in extraction. Chapter 12 of the 2009 International Fire Code (ICC 2009) was used for preliminary testing of the IE approach. The experiment focused on extracting quantitative requirements from the subject chapter. In comparison to a manually-developed gold standard, 95% precision and 94% recall were achieved.

Several causes of extraction errors were recognized through the analysis of the experiment results. In future work, the authors plan to address these errors by extending the approach, including incorporating co-referencing ability into the approach, and utilizing dependency information (obtained through dependency parsing) to solve term ambiguity, etc. In future work, the authors also plan to conduct further experimentation on different types of requirements, other building codes, and larger samples of text.

REFERENCES


formalization and semantic organization of conformance requirements in

Zhu, Y., Mao, W., and Ahmad, I. (2007). “Capturing implicit structures in
220-227.