Automation in Construction 74 (2017) 103–117

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Ontology-based automated information extraction from building energy conservation codes

Peng Zhou, Nora El-Gohary *

Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, 205 N. Mathews Ave., Urbana, IL 61801, United States

ARTICLE INFO

Article history:
Received 2 January 2016
Received in revised form 22 August 2016
Accepted 19 September 2016
Available online 1 December 2016

Keywords:
Information extraction
Ontology
Natural language processing
Automated compliance checking
Energy conservation codes

ABSTRACT

An ontology-based information extraction algorithm for automatically extracting energy requirements from energy conservation codes is proposed. The proposed algorithm aims to support fully-automated energy compliance checking in the construction domain by allowing automated extraction of the requirements from the codes instead of the status quo which relies on manual extraction of requirements from codes and manual formalization of those requirements in a computer-processable format. Automated information extraction from energy conservation codes, compared to other building codes, is a far complex task because many code provisions are long, hierarchically-complex, and with exceptions. A combination of text classification methods, domain-specific preprocessing techniques, ontology-based pattern-matching extraction techniques, sequential dependency-based extraction methods, and cascaded extraction methods is proposed to deal with such complexity in extraction. The proposed algorithm was tested in extracting energy requirements from Chapter 4 of the 2012 International Energy Conservation Code, and the results showed 97.4% recall and 98.5% precision.

1. Introduction

Environmental compliance checking aims to help construction projects comply with environmental codes and regulations such as the International Energy Conservation Code (IECC). Because manual compliance checking is time-consuming and costly [10,39], a number of research efforts aimed to automate the compliance checking process. Examples of automated compliance checking (ACC) efforts in the past five years include checking of building envelope performance [39], building safety design [33], building structural design [30], construction quality [45], building safety design and planning [27], building water network design [25], building fire safety [9,23], building evacuation [6], building sustainability [5], and formwork constructability [17]. Despite the importance of these efforts, existing ACC systems and methods are not fully automated; they require (1) intensive manual effort in extracting requirements from regulatory documents and encoding these requirements in a computer-processable format (e.g., [17]), or (2) substantial manual effort in annotating regulatory documents (e.g., [5,14]).

To address this gap, Zhang and El-Gohary [43,44] proposed an information extraction (IE) methodology for automatically extracting information from building codes [43] and an information transformation methodology for automatically transforming the extracted information into a computer-processable rule format [44]. Compared to building codes, automatically extracting requirements from energy codes is more challenging because of (1) longer provisions: provisions in energy codes are longer, which indicates that requirements are more likely to be noisy and/or semantically complex; (2) hierarchically-complex sentence structure s: text in energy codes has more complex sentence structures, in which one provision may contain multiple levels of subprovisions, and one subprovision may contain multiple requirements; and (3) more exceptions: a requirement in energy codes may contain one or multiple exceptions for waiving the compliance with the requirement if one or all of a set of exception conditions are met.

In this paper, an ontology-based information extraction (OBIE) algorithm for automatically extracting regulatory requirements from energy conservation codes is proposed. The proposed algorithm advances existing IE methods in the construction domain in four main ways. First, it extracts regulatory requirements from pre-classified text rather than unclassified text, which aims to improve the efficiency (by avoiding unnecessary computational processing of irrelevant text) and performance (by avoiding potential noise and errors resulting from processing irrelevant text) of IE. Second, it uses a deeper (more detailed) ontology, which aims to better capture domain-specific meaning. Third, it applies conceptual dependency theory to build a conceptual dependency structure and proposes a sequential dependency-based extraction method, which aim to reduce text ambiguities. Fourth, it proposes domain-specific preprocessing techniques and cascaded extraction methods, which aim to deal with the complexity of the text (i.e., longer provisions, hierarchically-complex sentence structures, and more exceptions). The proposed algorithm was tested in extracting

* Corresponding author.
E-mail address: gohary@illinois.edu (N. El-Gohary).

http://dx.doi.org/10.1016/j.autcon.2016.09.004
0926-5805/

commercial building energy efficiency regulatory requirements from the 2012 IECC [15].

2. Background

2.1. Information extraction

Natural language processing (NLP) is a subdiscipline of artificial intelligence that aims to enable computers to understand human language [24]. IE applies NLP techniques [e.g., part-of-speech (POS) tagging, morphological analysis, etc.] to recognize information from unstructured data and formalize it into structured data [18]. According to the level of complexity, IE can be categorized into four types: (1) named entity recognition, which aims to identify particular entities [18]; (2) relation detection, which aims to discern the relationships among the identified entities [18]; (3) event extraction, which aims to identify events from text (each event has a trigger and a number of associated arguments, and each event may be composed of a number of entities and their relationships) [13,32]; and (4) full IE, which aims to extract all information expressed by a sentence based on a full analysis of the sentence [43]. Named entity recognition, relation detection, and event extraction can be classified as shallow IE because they aim to extract partial information from a sentence, whereas full IE could be classified as deep IE because it aims to extract all information from a sentence [43].

There are different approaches to IE [18,28,29], including rule-based and supervised machine learning (ML)-based approaches. A rule-based approach requires human effort to analyze the text features in a relatively small set of text corpus (sometimes called developing data, which is analogous to training data in the case of ML), define the text patterns in terms of the text features, and then develop extraction rules based on the defined patterns. Text features may include [28]: (1) syntactic features, which refer to syntax-related features that are determined based on grammatical analysis, such as POS tags (e.g., tag "IN" represents a preposition like "for"); and/or (2) semantic features, which refer to concepts that capture the meaning of the information (e.g., "mass wall" is a concept that represents a type of wall). The patterns may be defined in terms of combinations of different syntactic and/or semantic features via regular expressions. Regular expressions is a language that is implemented by computers for pattern matching to characterize possible sequences of text [18].

A supervised ML-based approach requires human effort to collect a relatively large set of training data and annotate them with the relevant text features and with the information that should be extracted. Then, an ML algorithm (e.g., using Support Vector Machines, Hidden Markov Model, or Conditional Random Fields) is used to automatically learn the extraction rules from the annotated training data. Compared with the rule-based approach, the ML-based approach (1) requires a much larger size of annotated training data: because the performance of an ML-based IE algorithm depends on the training data for learning, a sufficiently large size of training data is required to accurately learn the text patterns and the extraction rules; and (2) does not require manual effort in pattern definition and extraction rule development: an ML algorithm automatically learns the patterns of the text and the extraction rules.

Although an ML-based approach can save the manual effort in pattern definition and extraction rule development, a rule-based approach is adopted in this research for two main reasons. First, a rule-based approach tends to yield higher performance, because human expertise usually results in more accurate patterns and extraction rules [28]. The performance of ML in a complex task such as IE is usually inconsistent and insufficient [16]. In this specific application, the level of complexity in IE is even much higher, compared to the state-of-the-art IE, which makes a rule-based approach especially suitable in this case; deep IE is needed to extract all information that describes a regulatory requirement and high performance is needed to support high performance ACC – both making the IE problem quite challenging. Second, in this application, the manual effort in pattern definition and extraction rule development in the rule-based approach is expected to be less than that required for manually annotating a sufficiently large size of training data if taking an ML-based approach.

2.2. Ontology-based information extraction

OBIE is a subfield of IE. Comparing to non-ontology-based IE, which only depends on the lexical and/or syntactic information of the text, OBIE further relies on semantic information to extract information based on meaning. In many cases, OBIE is domain and application-oriented, when a domain and/or an application ontology is used to assist in extracting semantic information that is specific to a particular domain and/or application [19,42,43]. In this case, OBIE captures domain-specific semantic information as semantic features, which are then used in the patterns in the extraction rules. Compared with non-ontology-based IE, the domain-specific semantic information that is used in OBIE is promising in improving the IE performance for a specific domain [42,43].

OBIE has been explored in different domains such as biology (e.g., [29]), business (e.g., [3,40]), law (e.g., [28]), medicine (e.g., [38]), mechanical engineering (e.g., [22]), and civil engineering (e.g., [43]). OBIE has also been explored in different complexity levels of IE: named entity recognition (e.g., [29]), relation detection (e.g., [22,38,40]), event extraction (e.g., [3]), and full IE (e.g., [43]). The most complex level (i.e., full IE) is the most challenging and the least explored. In terms of approach, all these efforts used a rule-based approach to deal with the OBIE problem.

3. State of the art and knowledge gaps in automated information extraction in construction

Despite the large number of IE efforts outside the construction domain, the number of IE efforts, especially OBIE efforts, are limited in the construction domain. For non-ontology-based IE efforts, Al Qady and Kandil [2] used limited syntactic features [i.e., specific phrases like VP (i.e., verb phrase) segment and its role ACTIVE_VERB] to extract concepts and relations from contract documents, with the aim to improve construction document management. Abuzir and Abuzir [1] used document structure features (i.e., HTML tags) and simple lexico-syntactic features (e.g., "such as" is a lexico-syntactic feature that was used to extract the terms following it because it usually indicates a synonym relationship among these terms) to extract terms and their relations from web pages, with the aim to construct a thesaurus of civil engineering. For OBIE efforts, Zhang and El-Gohary [43] used a combination of syntactic and semantic features to extract regulatory requirements from building codes for supporting automated code compliance checking, where the semantic features were extracted using a building ontology. Despite the importance of these efforts, they are still limited in one or more of the following four main ways. First, existing efforts extract information from unclassified text, which may result in unnecessary processing effort and may increase extraction errors due to processing irrelevant text. None of these efforts explored the use of text classification techniques to filter out irrelevant text prior to IE to improve the efficiency and performance of IE. Second, existing efforts were not tested in deep IE from long provisions with multiple exceptions. For example, Abuzir and Abuzir [1] and Al Qady and Kandil [2] conducted shallow IE (extracting partial information from a sentence, whereas deep IE aims to extract all information expressed by a sentence based on a full analysis of the sentence). Zhang and El-Gohary [43], on the other hand, conducted deep IE, but tested their algorithms in extracting requirements from international building codes, which include relatively shorter provisions with fewer exceptions in comparison to energy conservation codes; energy conservation codes include relatively long provisions with several exceptions. Third, existing efforts are limited in automatically dealing with text that includes hierarchically-complex sentence structures. For example, Al Qady and Kandil [2] used a manual approach

to break down contract sentences that contain enumerations and lists into separate sentences, each containing only one single component of the enumeration/list. This manual approach is time-consuming, if there are a large number of sentences. Fourth, there is still a need for improving OBIE performance to support high performance ACC. For example, there are no IE efforts that explored building a conceptual dependency structure to capture the dependency information among the target information and using this dependency information when defining the patterns in the extraction rules, in order to reduce text ambiguities for enhancing the extraction performance. Similarly, it is important to explore the use of a deeper (more detailed) ontology (e.g., deeper compared to that used in Zhang and El-Gohary [43]) in improving the extraction performance in the environmental regulatory domain.

4. Proposed ontology-based information extraction algorithm

To address these knowledge gaps, in this paper, a rule-based OBIE algorithm for extracting building energy requirements from energy conservation codes in the construction domain is proposed. The proposed IE algorithm is composed of seven primary steps (see Fig. 1): text classification, preprocessing, feature selection, identification of target semantic information elements (SIEs) and their conceptual dependency structure, development of extraction rules for sequential dependency-based extraction and cascaded extraction, implementation of the extraction algorithm, and evaluation. An illustrative example of the inputs and outputs of the main processing steps (i.e., Steps 1–6) is shown in Fig. 2.

4.1. Step 1: Text classification

Prior to extracting requirements from documents, the documents were first classified to filter out the text that is not related to building energy requirements. For example, the following sentence shows an example of text that was filtered out, because it describes a document administration requirement rather than a building energy requirement: "The construction documents shall specify that the documents described in this section be provided to the building owner within 90 days of the date of receipt of the certificate of occupancy" [15]. Filtering out such text could avoid unnecessary processing effort and potential errors resulting from processing irrelevant text, which may improve both the efficiency and the performance of IE. To classify the text in energy conservation codes, a topic hierarchy was developed to identify the labels used in classification, and a subontology was built for each topic (label) in the hierarchy. After preprocessing the documents (including tokenization, stemming, and stopword removal), a document was assigned with zero, one, or multiple labels by measuring the semantic similarity between the document and each subontology, using a deep learning technique. The documents assigned with a zero label were filtered out. The classification methodology is outside the scope of this paper and is explained in Zhou and El-Gohary [46].

4.2. Step 2: Preprocessing

The raw classified text (from Step 1) was preprocessed for preparation for the following processing and analysis steps. Two primary types of preprocessing were conducted: (1) domain-specific text preprocessing: preprocessing techniques for addressing the specific complexity of the text in energy conservation codes were proposed and used; and (2) general text preprocessing: three commonly-used text preprocessing techniques were utilized, including tokenization, sentence splitting, and morphological analysis.

4.2.1. Proposed domain-specific preprocessing

Two main domain-specific preprocessing techniques were proposed and used: provision splitting and meaning-based stitching. In addition, parenthesis removal and quotation mark removal were proposed and used. Provision splitting and stitching were proposed and used to deal with the following types of text complexities in energy conservation codes: provisions with hierarchically-complex sentence structures and provisions with exceptions. Two levels of splitting were proposed and used: (1) if a provision has exception(s), then the exception(s) is(are) split from the provision; and (2) based on the splitting results, if the provision and/or exception(s) contain(s) a list of sublevel provisions/exceptions, then the provision and/or exceptions are further split to the lowest level in which each resulting provision/exception contains a component from the list. During stitching, (1) the heading of the provision gets extracted and stitched to each split provision/exception to form the complete provision/exception; and (2) the relationship indicators, which indicate the conjunctive/disjunctive relationships among those split provisions/exceptions, are recognized based on key words such as "and", "or", "one of the following", "all of the following", and are extracted and stitched to each split provision/exception to retain the overall meaning of the requirement – if these split provisions (i.e., subprovisions) are conjunctive obligations or alternative obligations. For example, "or" is a disjunctive relationship indicator meaning that each of the split provisions in one set is an alternative obligation. An alternative obligation is an obligation that allows the obligor to choose which of a number of things to follow [7], where the compliance with any of them would achieve compliance with the main provision. An illustration of provision splitting and stitching is shown in Fig. 3.

Parenthesis removal aims to remove all parentheses and the text inside of them for simplifying the extraction problem in terms of pattern definition, because in energy conservation codes the information in parenthesis is usually equivalent to that preceding the parenthesis. For example, as per Fig. 3, "(929 m^2)" is removed from the following text, because it represents the same quantitative information but in a different unit: "In an enclosed space greater than 10,000 square feet (929 m^2)…" [15]. Removing such information can both simplify the following IE steps (by avoiding extracting multiple semantically-repetitive information) and ensure the consistency of information (e.g., all quantitative information is in the same metric). Quotation mark removal aims to remove all quotation marks because they may interrupt the identification and extraction of specific domain concepts.

4.2.2. Tokenization

Tokenization splits the raw text into tokens (e.g., words, numbers, punctuations, symbols, whitespaces) [24,28]. For example, the text "1. Not less than 3 percent with a skylight VT of at least 0.40; or" is tokenized into "'1' '.' 'Not' 'less' 'than' '3' 'percent' 'with' 'a' 'skylight' 'VT' 'of' 'at' 'least' '0' '.' '40' ';' 'or'" (the whitespace tokens are not shown). This task aims to identify the boundary of sentences (e.g., periods) and prepare for the following POS tagging task [29].

4.2.3. Sentence splitting

This task aims to split the text into sentences for future processing by detecting sentence boundary indicators like question marks, exclamation points, and periods [18]. Unlike question marks and exclamation points, periods are ambiguous in delimiting sentences. For example, the period in "C402.4" is part of the name of a regulatory provision and is not a sentence boundary indicator. A set of domain-specific sentence splitting rules were, thus, developed for domain adaptation, because existing sentence splitters [e.g., the A Nearly-New Information Extraction (ANNIE) Sentence Splitter] are domain and application-independent [8]; and, thus, caused errors in splitting the text in energy conservation codes. For example, in partial provision PP1 (which is the result of provision splitting and stitching shown in Fig. 3), existing sentence splitters mistakenly recognized the period in the list number "1." as full stop of a sentence. The developed sentence splitting rules helped address this issue. This set of domain-specific sentence splitting rules is potentially reusable for similar IE applications and similar construction regulatory text.

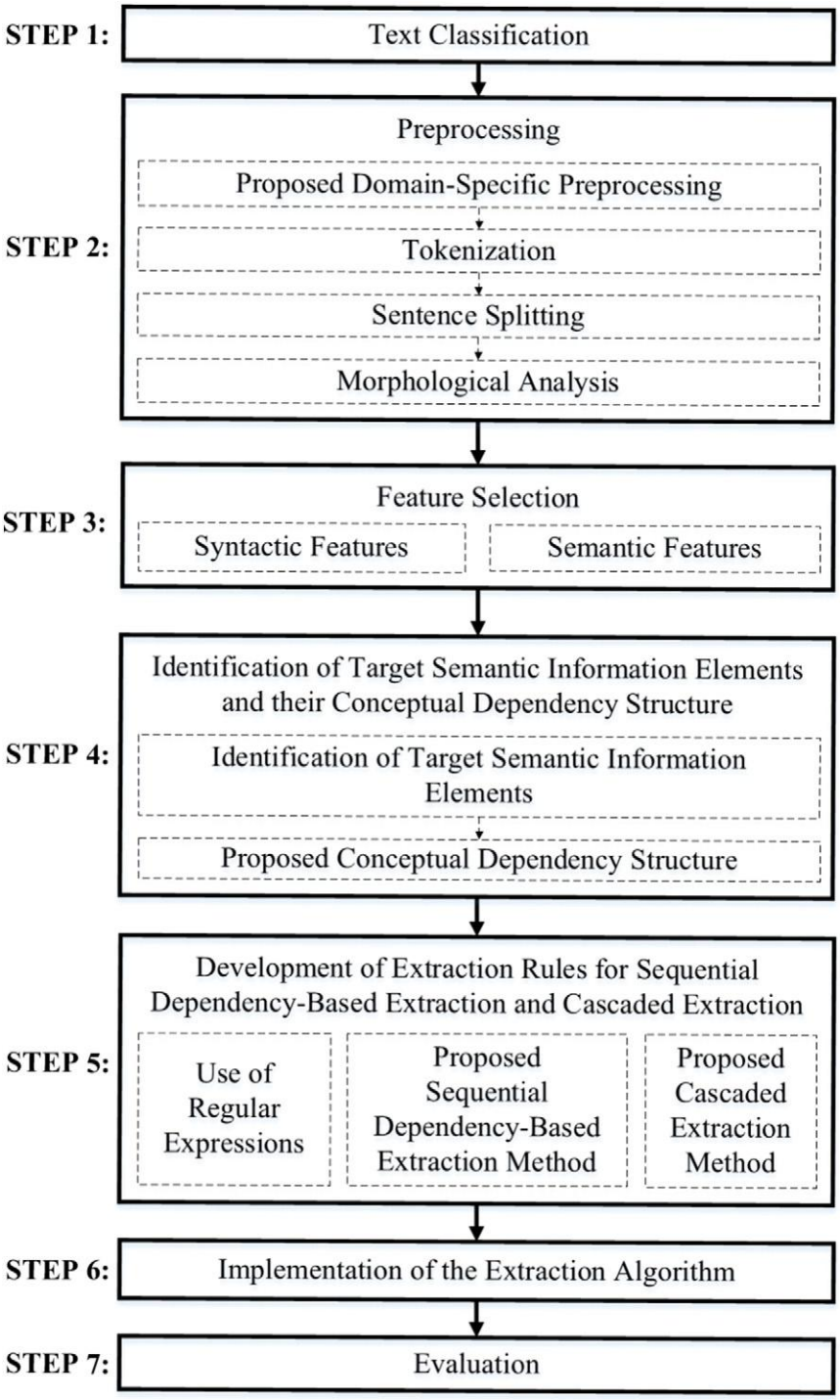

Fig. 1. Proposed ontology-based information extraction algorithm.

- PP1: "C402.3.2 Minimum skylight fenestration area. In an enclosed space greater than 10,000 square feet, directly under a roof with ceiling heights greater than 15 feet … the total daylight zone under skylights shall be not less than half the floor area and shall provide a minimum skylight area to daylight zone under skylights of either: 1. Not less than 3 percent with a skylight VT of at least 0.40; or" [15].

4.2.4. Morphological analysis

Morphological analysis collapses different derivational (e.g., affixes like "ly", "ion") and inflectional forms (e.g., plural, progressive) of a word to their base form [24]. For example, "balances", "balancing", "balance", and "balanced" are all mapped to "balance". This task aims to help recognize the semantic features of the text by mapping the morphologically-analyzed text to the ontology concepts. For example, through morphological analysis, "balancing valves" in the natural text is recognized and mapped to the concept "balance valve" in the ontology.

4.3. Step 3: Feature selection

After preprocessing the text, the syntactic and semantic features were selected for further extraction rule development (Step 5). In this research, POS tags, gazetteers, and auxiliary tags were used as syntactic

Original Text

C402.3.2 Minimum skylight fenestration area. In an enclosed space greater than 10,000 square feet (929 m2), directly under a roof with ceiling heights greater than 15 feet (4572 mm), and used as an office, lobby, atrium, concourse, corridor, storage, gymnasium/exercise center, convention center, automotive service, manufacturing, nonrefrigerated warehouse, retail store, distribution/sorting area: transportation, or workshop, the total daylight zone under skylights shall be not less than half the floor area and shall provide a minimum skylight area to daylight zone under skylights of either:
1. Not less than 3 percent with a skylight VT of at least 0.40; or
2. Provide a minimum skylight effective aperture of at least 1 percent determined in accordance with Equation 4-1.
Exception: Skylights above daylight zones of enclosed spaces are not required in:
1. Buildings in climate zones 6 through 8.
2. Spaces where the designed general lighting power densities are less than 0.5 W/ft2 (5.4 W/m2).
3. Areas where it is documented that existing structures or natural objects block direct beam sunlight on at least half of the roof over the enclosed area for more than 1,500 daytime hours per year between 8 am and 4 pm.
4. Spaces where the daylight zone under rooftop monitors is greater than 50 percent of the enclosed space floor area.

.....

C408.2.5 Documentation requirements. The construction documents shall specify that the documents described in this section be provided to the building owner within 90 days of the date of receipt of the certificate of occupancy.

Step 1: Text Classification

Original Text (Relevant)

C402.3.2 Minimum skylight fenestration area. In an enclosed space greater than 10,000 square feet (929 m2), directly under a roof with ceiling heights greater than 15 feet (4572 mm), and used as an office, lobby, atrium, concourse, corridor, storage, gymnasium/exercise center, convention center, automotive service, manufacturing, nonrefrigerated warehouse, retail store, distribution/sorting area: transportation, or workshop, the total daylight zone under skylights shall be not less than half the floor area and shall provide a minimum skylight area to daylight zone under skylights of either:
1. Not less than 3 percent with a skylight VT of at least 0.40; or
2. Provide a minimum skylight effective aperture of at least 1 percent determined in accordance with Equation 4-1.
Exception: Skylights above daylight zones of enclosed spaces are not required in:
1. Buildings in climate zones 6 through 8.
2. Spaces where the designed general lighting power densities are less than 0.5 W/ft2 (5.4 W/m2).
3. Areas where it is documented that existing structures or natural objects block direct beam sunlight on at least half of the roof over the enclosed area for more than 1,500 daytime hours per year between 8 am and 4 pm.
4. Spaces where the daylight zone under rooftop monitors is greater than 50 percent of the enclosed space floor area.

Step 2: Preprocessing

Preprocessed Text (Partial preprocessed text shown, whitespace tokens not shown)

<Sentence>·····</Sentence><Sentence>·····
<Token>1</Token><Token>.</Token><Token>Not</Token><Token>less</Token><Token>than</Token><Token>3</Token><Token>percent</Token><Token>with</Token><Token>a</Token><Token>skylight</Token><Token>VT</Token><Token>of</Token><Token>at</Token><Token>least</Token>
<Token>0</Token><Token>.</Token><Token>40</Token><Token>;</Token><Token>or</Token></Sentence>

Step 3: Feature Selection

Syntactic Features

- **POS Tags (Shown in parenthesis as examples)**

e.g., 1(**LS**) .(**.**) Not(**RB**) less(**JJR**) than(**IN**) 3(**CD**) percent(**NN**) with(**IN**) a(**DT**) skylight(**NN**) VT(**NNP**) of(**IN**) at(**IN**) least(**JJS**) 0(**CD**).(**.**)40(**CD**);(**:**) or(**CC**)

- **Gazetteers**

Negation gazetteer (e.g., "not", "no")
Measurement unit gazetteer (e.g., "square feet" and "cfm/ft2")

- **Auxiliary Tags (Partial)**

ListNumber (e.g., "1."), CD (e.g., "3 percent", "0.40")

Semantic Features

- **Ontological Concepts** [e.g., "skylight", "visible transmittance (VT)"]

Step 4: Identification of Target Semantic Information Elements and their Conceptual Dependency Structure

Target Semantic Information Elements

"Subject", "Subject Restriction", "Compliance Checking Attribute", "Deontic Operator Indicator", "Quantitative Relation", "Comparative Relation", "Quantity Value", "Quantity Unit/Reference", and "Quantity Restriction"

Conceptual Dependency Structure

- Extraction sequence
- Dependency information

Step 5: Development of Extraction Rules for Sequential Dependency-Based Extraction and Cascaded Extraction

Extraction Rules

e.g., "**(IN JJS):cr + QuantityValue → cr.ComparativeRelation**": This rule states that if information (e.g., "at least") matches the pattern "**IN JJS**" and the information is followed by an instance of "**Quantity Value**" (e.g., "0.40"), the information is extracted as an instance of "**Comparative Relation**" (i.e., "at least" is an instance of "Comparative Relation")

Step 6: Implementation of the Extraction Algorithm

Extracted Instances of the Target Semantic Information Elements

Subject	skylight
Subject Restriction	in an enclosed space greater than 10,000 square feet, under a roof with ceiling heights greater than 15 feet, used as an office, lobby, atrium, concourse, corridor, storage, gymnasium/exercise center, convention center, automotive service, manufacturing, nonrefrigerated warehouse, retail store, distribution/sorting area: transportation, or workshop
Compliance Checking Attribute	VT
Deontic Operator Indicator	shall (obligation)
Quantitative Relation	N/A
Comparative Relation	at least
Quantity Value	0.4
Quantity Unit/Reference	N/A
Quantity Restriction	N/A

Fig. 2. An illustrative example of the inputs and outputs of the main algorithm steps.

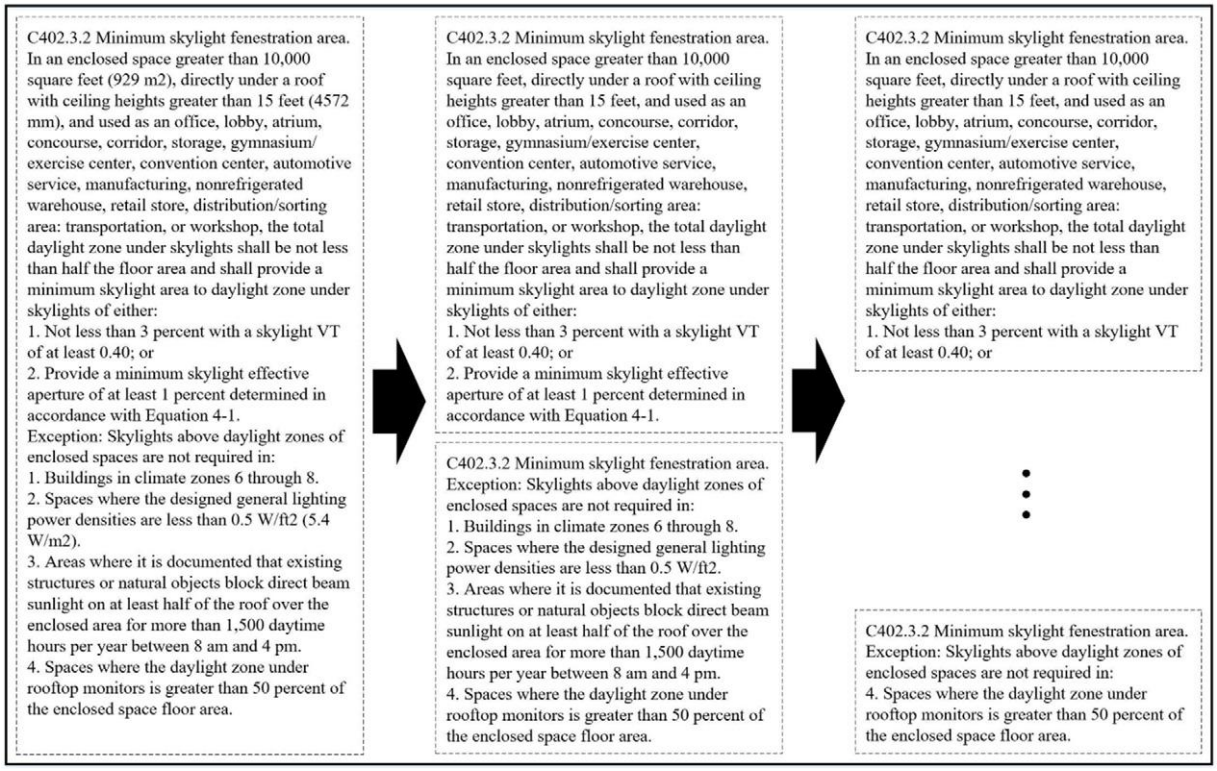

Fig. 3. An illustration of provision splitting and stitching.

features, while concepts from the ontology were used as semantic features. Semantic features were used to facilitate the extraction of domain-specific semantic information, which would be hard to extract using syntactic features only; semantic features are essential to recognize domain-specific meaning. For example, "building thermal insulation" and "lacking initial test" have exactly the same syntactic features in terms of POS tags (i.e., "VBG JJ NN", representing gerund, adjective, and singular noun), but the former is recognized to be an instance of "Subject" (an SIE) based on the concepts in the ontology. Both syntactic and semantic features were used in the patterns in the extraction rules.

4.3.1. Syntactic features

Three main syntactic features were used: POS tags, gazetteers, and auxiliary tags. POS tagging assigns a tag to each word based on its syntactic word class (e.g., noun, verb, adjective) [28]. For example, the tags "VBG", "JJ", and "NN" were assigned to the gerund, adjective, and singular noun in a sentence, respectively [18].

A gazetteer refers to a list of words that share a common category (e.g., list of countries) [42]. In this research, a number of words/symbols that represent similar meanings were collected as gazetteers. Each gazetteer was assigned with a tag and each tag was used as a syntactic feature. Accordingly, two gazetteers were manually developed and used: (1) a negation gazetteer, which includes negation words like "no" and "not"; and (2) a measurement unit gazetteer, which includes unit words/symbols like "square feet" and "cfm/ft^2". Words/symbols belonging to the first and second gazetteers were assigned "neg" and "unit" tags, respectively.

A total of 15 auxiliary tags were also defined and used. Tagging with auxiliary tags (because they are newly-defined tags) was conducted using a set of tagging rules (as explained in Step 6). Examples of auxiliary tags that appeared frequently when tagging energy conservation codes include: (1) "ListNumber": assigned to the serial number of each split provision/exception such as "1." and "2.1", and (2) "CD" (short for cardinal number): assigned to the numbers in the text that are potential quantity values of a regulatory requirement. For example, in partial provision PP1, both "3 percent" and "0.40" are potential quantity values of the requirement. However, not all numbers should be annotated with the tag "CD". For instance, in PP1, the numbers "402", "3", and "2" in "C402.3.2" are part of the provision designation number, which are not potential quantity values, and thus should not be tagged with "CD". Since this ambiguity may result in potential errors in extracting quantity values, a number of CD tagging rules were developed to reduce such ambiguity based on the patterns of adjacent syntactic features for a cardinal number. For instance, in PP1, if a number has a preceding capital letter "C" and is followed by one or more repetitive patterns (period + number), then all these numbers should not be annotated with the tag "CD".

4.3.2. Semantic features

An ontology was developed to help recognize the semantic features of the text by capturing the concepts related to commercial building energy conservation. The ontology was developed into the ninth level, including 335 concepts in total. A partial view of the ontology is shown in Fig. 4. The ontology was built/edited using the Web Ontology Language In-Memory (OWLIM) Ontology Editor in the General Architecture for Text Engineering (GATE) [8]. The ontology was then inputted into the OntoRoot Gazetteer module to: (1) create a gazetteer of all concepts for using each concept as a semantic feature; and (2) parse the hierarchical "is-a" relationship among concepts to facilitate pattern definition for extraction rule development (as discussed in Step 5).

For developing the ontology, the ontology development methodology by El-Gohary and El-Diraby [11] was benchmarked. Accordingly, the methodology for developing the ontology included four main steps: (1) purpose and scope definition, (2) taxonomy building, (3) relation modeling, and (4) ontology coding. The purpose of the ontology is to support OBIE. The scope of the ontology is limited to the commercial building energy efficiency domain. For taxonomy building, the main concepts in the domain of interest were identified based on a review of the main relevant environmental regulatory documents, and then the identified concepts were organized into a hierarchy of concepts

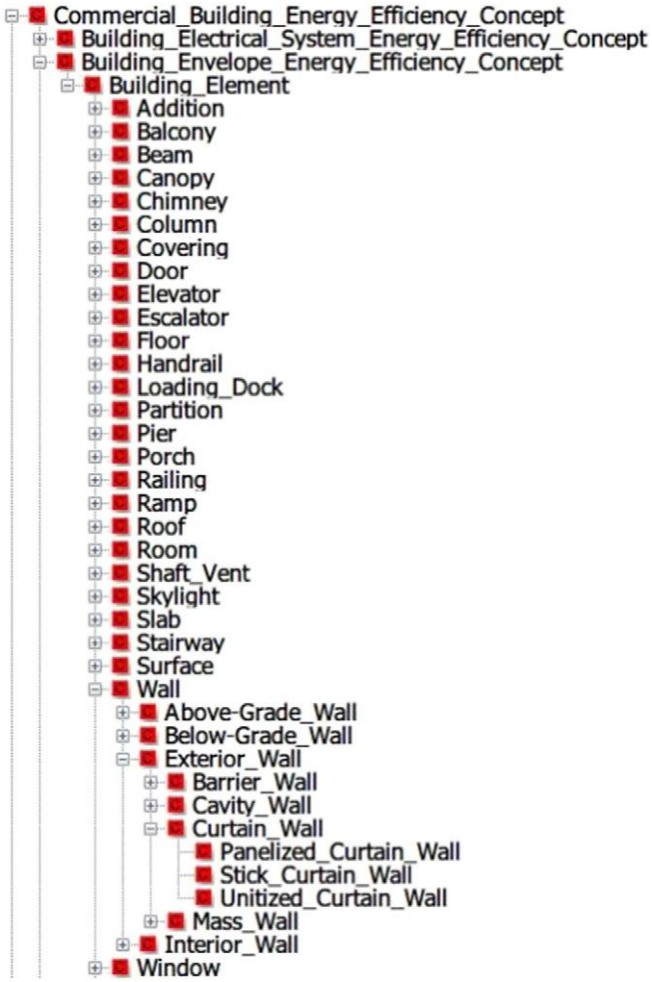

Fig. 4. Partial view of the ontology.

using a combination of a top-down (starting by defining the most abstract concepts) and a bottom-up approach (starting by defining the most specific concepts). For example, concepts related to the "building mechanical system energy efficiency" concept (a subconcept of "commercial building energy efficiency" concept) such as "HVAC system", "air economizer system", and "water economizer system" were identified; and then, for this specific example, "air economizer system" and "water economizer system" were modeled as subconcepts of "HVAC system". For relation modeling, the nonhierarchical relationships between concepts were identified and modeled to describe the semantic links between the concepts. For example, "is_controlled_by" is a nonhierarchical relationship that links "lamp" with "occupant sensor". As mentioned above, the concepts and relations of the ontology were coded in OWLIM.

4.4. Step 4: Identification of target semantic information elements and their conceptual dependency structure

4.4.1. Identification of target semantic information elements

Before developing the extraction rules, the target information that needs to be extracted should be identified based on the specific requirements of the application and the domain. Nine types of target SIEs for representing quantitative regulatory requirements were identified (following Zhang and El-Gohary [43]) and used: including "Subject", "Subject Restriction", "Compliance Checking Attribute", "Deontic Operator Indicator", "Quantitative Relation", "Comparative Relation", "Quantity Value", "Quantity Unit/Reference", and "Quantity Restriction".

"Subject" refers to the primary entity that is regulated in a requirement, and corresponds to a concept in the ontology. For example, the concept "skylight" from the ontology could be an instance of "Subject". "Compliance Checking Attribute" refers to a specific property of a "Subject" that is checked for compliance, and corresponds to a concept in the ontology. For example, the concept "minimum skylight area" in the ontology could be an instance of "Compliance Checking Attribute". "Deontic Operator Indicator" is a word or phrase that indicates the deontic type of the requirement [35,43]: obligation, permission, or prohibition. For example, in the following sentence "shall" indicates obligation: "The minimum thermal resistance of the insulating material installed in, or continuously on, the below-grade walls shall be as specified in Table C402.2, and shall extend to a depth of 10 feet below the outside finished ground level, or to the level of the floor, whichever is less." [15]. "Quantitative Relation" refers to the type of semantic relationship between the "Compliance Checking Attribute" and the "Quantity Value". In the example above, "extend" is an instance of "Quantitative Relation". "Comparative Relation" refers to a relationship, such as "less than" or "equal to", for stating a quantitative range of a quantity value. "Quantity Value" refers to the quantitative measure of the requirement, while "Quantity Unit/Reference" refers to an explicit measurement unit (e.g., "10 feet") or an implicit reference unit for the "Quantity Value" (e.g., "35 percent of its rated power"). "Subject Restriction" and "Quantity Restriction" refer to constraints that are placed on the "Subject" and "Quantity Value", respectively, where a restriction may consist of multiple ontology concepts and/or relationships. In this paper, for one quantitative requirement: (1) there must be only one "Subject", only one "Comparative Relation", and only one "Quantity Value". For "Comparative Relation", a default "greater than or equal" is used if the relation in a requirement is implicit (e.g., "…shall extend to a depth of 10 feet…"); (2) there could be at most one "Compliance Checking Attribute", at most one "Deontic Operator Indicator", at most one "Quantitative Relation", and at most one "Quantity Unit/Reference"; and (3) there could be zero, one, or multiple "Subject Restrictions" and "Quantity Restrictions". An illustrative example showing the SIEs for a requirement, after splitting and stitching (Fig. 3), is shown in Fig. 5.

4.4.2. Proposed conceptual dependency structure

The conceptual dependency structure of the SIEs was developed based on conceptual dependency theory. According to conceptual dependency theory, any two linguistic structures of identical meaning should have the same conceptual dependency structure [28]. In this research, the proposed IE algorithm is used to extract information describing quantitative requirements in energy conservation codes. Since all instances of quantitative requirements express the same meaning in terms of requirements expressed in numerical values on an entity, they could be represented by the same conceptual dependency structure. The conceptual dependency structure is composed of interdependent primary concepts and relations [28]. Since a sentence is usually composed of multiple concepts and relations, the sentences were analyzed to identify those primary concepts and relations that correspond to the target SIEs.

After analyzing the dependencies among the target SIEs, the conceptual dependency structure of the SIEs was built, as per Fig. 6. The conceptual dependency structure indicates that: (1) there exists an extraction sequence that dictates that an SIE should be extracted only after all its preceding SIEs are extracted, which is called – in this paper – the sequential dependency extraction method; and (2) the extraction rules to extract an SIE may use the preceding SIEs to reduce ambiguities/errors. Comparing to extracting an SIE isolatedly (i.e., extraction rules are developed without using dependency information), the use of dependency information in developing extraction rules imposes more stringent conditions on matching information, which rules out information that does not match the conditions.

In Fig. 6, the arrow represents the dependency relationship and the serial number of each SIE indicates the extraction sequence. For

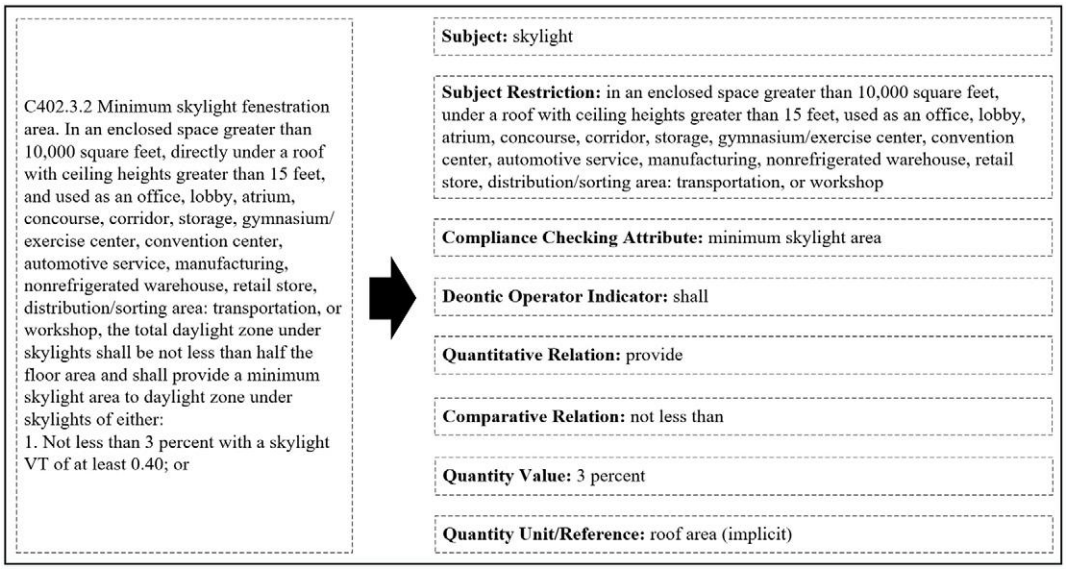

Fig. 5. Example semantic information element instances.

example, "Subject" depends on both "Deontic Operator Indicator" and "Comparative Relation", and "Deontic Operator Indicator" also depends on "Comparative Relation". Therefore, "Comparative Relation" should be extracted first, and "Subject" should be extracted only after its two preceding SIEs (i.e., "Deontic Operator Indicator" and "Comparative Relation") have been extracted. For the interdependent SIEs, they should be extracted together after all their preceding SIEs have been extracted. For example, the SIEs "Deontic Operator Indicator" and "Quantitative Relation" are interdependent and thus should be extracted together after their preceding "Comparative Relation" SIE has been extracted.

4.5. Step 5: Development of extraction rules for sequential dependency-based extraction and cascaded extraction

After identifying the target SIEs and their conceptual dependency structure, the extraction rules were manually developed to help extract the instances of the target SIEs. The extraction rules were developed after reviewing a number of energy regulatory documents (e.g., ANSI/ASHRAE/IES Standard 90.1-2010 [4]) – excluding the IECC which was used for testing – and manually analyzing the text features and patterns in these documents. These documents are the developing data, which – as mentioned above – are analogous to the training data in the case of ML. The left side of an extraction rule models the pattern of the text in terms of syntactic features (i.e., POS tags, gazetteers, and/or auxiliary tags) and/or semantic features (i.e., concepts from the ontology), while the right side defines the information that should be extracted when this pattern is matched. In developing the rules, regular expressions were used. Methods for sequential dependency-based IE and cascaded IE were proposed and considered when developing the rules.

4.5.1. Use of regular expressions

In defining those patterns, regular expressions were used to define the most simplified (but generalized) patterns so that a rule can deal with a variety of text sharing similar regularities in terms of syntactic and semantic features, regardless of the length and content of the text. As such, regular expressions may facilitate the extraction of complex SIEs (e.g., "Subject Restriction" and "Quantity Restriction") because they usually contain such feature regularities; they are usually composed of a number of repetitive semantic and/or syntactic features in certain patterns. For example, "designed for sensible heating of an indoor space through heat transfer from the thermally effective panel surfaces" is an instance of "Subject Restriction". The semantic features (i.e., concepts) and syntactic features (i.e., POS tags, in this example) of this instance can be analyzed as follows: "sensible heating", "indoor space", "heat transfer", and "thermally effective panel surface" are four ontology concepts, "VBN" is the POS tag for the past participle "designed", "DT" is for the determiners "an" and "the", and "IN" is for the prepositions "for", "of", "through", and "from". Thus, this instance is just a repetition of four prepositional phrases starting with a past participle (i.e., "designed"), and each prepositional phrase may contain an optional determiner. Accordingly, with the help of regular expressions, the pattern for extracting such similar instances could be defined as: "VBN (IN (DT)? commercial_building_energy_efficiency_concept)+", where the "?" indicates that there is at most one determiner in a

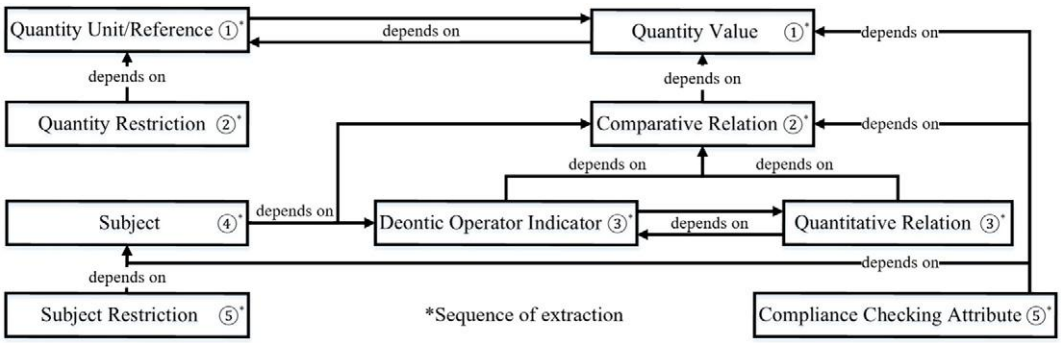

Fig. 6. Conceptual dependency structure.

prepositional phrase, the "commercial building energy efficiency" concept is a semantic feature representing a concept, the subpattern "(IN (DT)? commercial_building_energy_efficiency_concept)" represents a prepositional phrase, and the "+" indicates that there is at least one such kind of prepositional phrase.

For the use of semantic features in pattern definition, only the top concept of all the possibly matched subconcepts was used as the semantic feature. The top concept is the highest-level relevant concept in the ontology which subsumes all possibly matched subconcepts. For example, in the above instance, the semantic feature "commercial building energy efficiency" concept is the top concept of all possibly matched subconcepts such as "sensible heating", "indoor space", "heat transfer", and "thermally effective panel surface". For more details on the use of regular expressions in pattern definition, the readers are referred to Cunningham et al. [8].

4.5.2. Proposed sequential dependency-based extraction method

In developing the rules, the dependency information among the SIEs assisted in defining the patterns. For example, the following rule was developed to extract the instances of "Comparative Relation" (a target SIE): "(JJR IN):cr + QuantityValue ➔ cr.ComparativeRelation". "JJR" and "IN" are POS tags for comparative adjective and preposition. "JJR IN" is a pattern that matches information like "less than". When "JJR IN" is followed by an instance of "Quantity Value", which is the dependency information, the information matching "JJR IN" should probably be an instance of "Comparative Relation". Therefore, a pointer "cr" was set to pattern "JJR IN", and the information (which the pointer refers to) matching this pattern was extracted as an instance of "Comparative Relation". Similarly, Rule 7 ["(commercial_building_energy_efficiency_concept):sj + VBZ + ComparativeRelation ➔ sj.Subject"] was used to extract "radiant panel" as an instance of "Subject", where "radiant panel" is a subconcept of "commercial building energy efficiency" concept.

4.5.3. Proposed cascaded extraction method

In developing the rules for extracting complex SIEs (e.g., "Subject Restriction" and "Quantity Restriction"), which usually appear in longer provisions, a cascaded IE method was proposed and used to break down a complex extraction task into a number of simple extraction tasks (i.e., a complex extraction task is cascaded on a number of simple extraction tasks). As such, simple SIEs (or simpler SIEs, e.g., "Quantity Restriction" is simpler than "Subject Restriction") are used as features in the rules that extract complex SIEs. The extraction of such complex SIEs can, thus, be broken down into two steps: (1) extracting the simple SIEs; and (2) extracting the complex SIEs based on the simple SIEs. For example, in partial provision PP1, "…under a roof with ceiling heights greater than 15 feet..." [15] is an instance of "Subject Restriction" that contains quantitative information corresponding to five SIEs. Accordingly, the instances of the five SIEs were first extracted: "roof", "ceiling height", "greater than", "15", and "feet" were extracted as instances of "Subject", "Compliance Checking Attribute", "Comparative Relation", "Quantity Value", and "Quantity Unit/Reference", respectively. Then, these extracted instances were used to extract the instance of the "Subject Restriction" (i.e., these five simple SIEs were used as features in the rule that extracted the "Subject Restriction").

4.6. Step 6: Implementation of the extraction algorithm

Step 1 was implemented as a separate software program. Steps 2–5 were implemented in the ANNIE system of GATE [8], including: the ANNIE English Tokeniser, ANNIE Sentence Splitter, GATE Morphological Analyser, ANNIE POS Tagger, ANNIE Gazetteer, OntoRoot Gazetteer, and Java Annotation Patterns Engine (JAPE) Transducer. Each of these modules may have initialization parameters. For example, "caseSensitive" is a parameter having either "true" or "false" values, which indicates whether matching should be conducted in a case-sensitive manner or not. For the details of the parameters for all these modules in the ANNIE system, the readers are referred to Cunningham et al. [8].

The two gazetteers (see Step 3) were added to the ANNIE Gazetteer module in GATE, along with other existing gazetteers (e.g., location, currency, etc.). The 15 auxiliary tags (see Step 3) were added to the JAPE Transducer, where the tagging was conducted using a set of tagging rules.

All extraction rules (see Step 5) were developed in the grammar of JAPE [8] using a JAPE editor – Vim (Vi IMproved) [34]. The JAPE grammar has five control styles to assist the extraction in terms of rule matching. The most commonly used is the "applet" control style [8]. For a region of text starting from a fixed location of a sentence, under the "applet" control style, only the rule that matches the longest text starting from the fixed location will be fired. For instance, in extracting the instance of "Quantity Unit/Reference" from the following sentence, Rule 12 can match both the text "Btu" and "Btu per inch/h × ft^2 × °F": "For automatic-circulating hot water and heat-traced systems, piping shall be insulated with not less than 1 inch of insulation having a conductivity not exceeding 0.27 Btu per inch/h × ft^2 × °F." [15]. According to the matching mechanism of the "applet" control style (i.e., longest matching) used in Rule 12, it is the "Btu per inch/h × ft^2 × °F" that was extracted as an instance of "Quantity Unit/Reference". For further details on the other control styles and the JAPE grammar, the readers are referred to Cunningham et al. [8]. All the developed extraction rules were inputted into the JAPE Transducer for executing the extraction. The outputs of Step 6, as illustrated in Fig. 2, are the extracted instances of the target SIEs, which were used for performance evaluation in Step 7.

4.7. Step 7: Evaluation

Step 1 was tested and evaluated separately, since it was implemented in a separate software program (as mentioned in Section 4.6). The text classification testing and evaluation results are outside the scope of this paper and are explained in Zhou and El-Gohary [46]. The OBIE algorithm (Steps 2–5) was tested in extracting commercial building energy efficiency regulatory requirements from Chapter 4 of the 2012 IECC [15]. IECC was selected because it is the most widely-adopted building energy conservation code in the U.S. The performance was evaluated by comparing the extraction results to a gold standard.

The gold standard for Chapter 4 of the 2012 IECC [15] was manually developed. It was developed by three researchers – the first author and two other researchers. Although it is a good strategy to use domain experts to develop a gold standard to ensure its validity and reliability, in many cases this is not feasible [20], because domain experts are usually not easily available to participate in such time-intensive activities and their time is highly expensive [21]. It is, therefore, a common practice to have researchers with domain knowledge develop the gold standard [12,41]. In this research, the first author was involved in developing the gold standard because of his familiarity with all of the following three areas, which helps ensure the correctness – and thus the validity – of the gold standard annotations: energy conservation codes, civil engineering domain, and the NLP domain. Two other civil engineering researchers participated in developing the gold standard, in order to (1) avoid confirmation bias for validity, and (2) have multiple annotators annotate the same sentence and measure inter-annotator agreement to evaluate the reliability of the gold standard. Typically, two or three annotators annotate the same text for IE work [21], which indicates that using three annotators is sufficient.

The annotation was conducted in three main steps: (1) a short 15-min presentation was given to the annotators to explain the annotation objective, the target SIEs, and illustrate the instances of all SIEs using examples from the development text (e.g., ANSI/ASHRAE/IES Standard 90.1-2010 [4]); (2) another warm-up and question and answer (Q&A) session was conducted for training the annotators and clearing any confusion using example sentences from the development text; and (3) the

annotators conducted the annotation independently. The inter-annotator agreement was calculated. The initial inter-annotator agreement was 86% in F-measure, which indicates the reliability of the gold standard. "An F-measure of 0.80 or above is generally considered sufficient inter-annotator agreement" [31]. The discrepancies were then discussed and resolved to reach consensus, thereby achieving final full annotator agreement. So, overall, the use of this team of three annotators and the annotation process aimed to balance reliability and validity.

An illustrative example of three provisions and their corresponding target SIEs is shown in Table 1. The performance was evaluated by comparing the extraction results with the gold standard. The performance was measured in terms of recall and precision [26,28]. Recall is the percentage of correctly extracted instances out of the total number of instances that should be extracted. Precision is the percentage of correctly extracted instances out of the total number of extracted instances.

5. Experimental results and analysis

5.1. Performance results

The experimental results are summarized in Table 2. The number of patterns used to extract the "Subject", "Subject Restriction", "Compliance Checking Attribute", "Deontic Operator Indicator", "Quantitative Relation", "Comparative Relation", "Quantity Value", "Quantity Unit/Reference", and "Quantity Restriction" instances are 25, 15, 8, 11, 11, 9, 14, 14, and 9, respectively. In addition, ten patterns were defined for the cascaded extraction of "Subject Restriction" instances, while one pattern was used for the "Quantity Restriction" instances. The gold standard includes 127, 87, 53, 56, 52, 87, 87, 87, and 23 instances of "Subject", "Subject Restriction", "Compliance Checking Attribute", "Deontic Operator Indicator", "Quantitative Relation", "Comparative Relation", "Quantity Value", "Quantity Unit/Reference", and "Quantity Restriction", respectively, at a total of 659 instances. A performance of 97.4% recall and 98.5% precision was achieved, which indicates that the proposed IE algorithm is potentially effective in extracting regulatory requirements from energy conservation codes.

5.2. Effects of sequential dependency-based extraction

The results show that the use of dependency information was effective in reducing semantic ambiguities. This can be illustrated by the extraction results for the instances of "Quantity Unit/Reference" and "Comparative Relation"; both achieved 100% precision. For example, "above", which is a potential instance of "Comparative Relation", is a word that can create semantic ambiguity. For example, "above" could either be followed by a "Quantity Value" (e.g., "the pavement temperature is above 50 °F" [15]) or a location (e.g., "skylights are installed above daylight zone" [15]), but only in the former case "above" would mean a "Comparative Relation". The proposed algorithm was able to avoid such semantic ambiguities because of utilizing dependency information. For example, after extracting the SIE "Quantity Value", it is used as dependency information to assist in the extraction of other SIEs. Thus, to correctly extract "above" as an instance of "Comparative Relation", the pattern can be defined as "IN + QuantityValue", where "IN" is the POS tag of preposition (i.e., above) and "QuantityValue" is the dependency information, indicating that "above" should be extracted as an instance of "Comparative Relation" only in the case when it is followed by a quantity value. However, sometimes there is no dependency information to help resolve ambiguities, especially when extracting a target information at the top of the conceptual dependency structure. This could be illustrated by the errors in the extraction of "Quantity Value" instances, which is one of the top SIEs in the conceptual dependency structure. For example, in the following sentence, the number "15" was incorrectly extracted as an instance of "Quantity Value", because there is no dependency information: "…Materials in Items 1 through

Table 1
Examples of provisions and their corresponding semantic information elements in the gold standard.

Semantic information element	Partial provision (requirement)		
Provision	In an enclosed space greater than 10,000 square feet, directly under a roof with ceiling heights greater than 15 feet, and used as an office, lobby, atrium, concourse, corridor, storage, gymnasium/exercise center, convention center, automotive service, manufacturing, nonrefrigerated warehouse, retail store, distribution/sorting area: transportation, or workshop, the total daylight zone under skylights shall be not less than half the floor area and shall provide a minimum skylight area to daylight zone under skylights of either: 1. Not less than 3 percent with a skylight VT of at least 0.40; or [15]		
Requirement	R1	R2	R3
Subject	Total daylight zone	Skylight	Skylight
Subject restriction	Under skylights, in an enclosed space greater than 10,000 square feet, under a roof with ceiling heights greater than 15 feet, used as an office, lobby, atrium, concourse, corridor, storage, gymnasium/exercise center, convention center, automotive service, manufacturing, nonrefrigerated warehouse, retail store, distribution/sorting area: transportation, or workshop	In an enclosed space greater than 10,000 square feet, under a roof with ceiling heights greater than 15 feet, used as an office, lobby, atrium, concourse, corridor, storage, gymnasium/exercise center, convention center, automotive service, manufacturing, nonrefrigerated warehouse, retail store, distribution/sorting area: transportation, or workshop	In an enclosed space greater than 10,000 square feet, under a roof with ceiling heights greater than 15 feet, used as an office, lobby, atrium, concourse, corridor, storage, gymnasium/exercise center, convention center, automotive service, manufacturing, nonrefrigerated warehouse, retail store, distribution/sorting area: transportation, or workshop
Compliance checking attribute	Area (implicit)[1]	Minimum skylight area	VT[2]
Deontic operator indicator	Shall (obligation)	Shall (obligation)	Shall (obligation)
Quantitative relation	N/A	Provide	N/A
Comparative relation	Not less than	Not less than	At least
Quantity value	Half	3 percent	0.4
Quantity unit/reference	Floor area	Roof area (implicit)[1]	N/A
Quantity restriction	N/A	N/A	N/A

1. "Implicit" means the instance is not explicitly stated in the text.
2. VT = visible transmittance.

Table 2
Experimental results of extracting requirements from energy conservation codes.

Total 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	127	87	53	56	52	87	87	87	23	659
Extracted	124	85	53	55	51	87	88	87	22	652
Correctly extracted	115	85	53	55	51	87	87	87	22	642
Precision	92.7%	100.0%	100.0%	100.0%	100.0%	100.0%	98.9%	100.0%	100.0%	98.5%
Recall	90.6%	97.7%	100.0%	98.2%	98.1%	100.0%	100.0%	100.0%	95.7%	97.4%

15 shall be deemed to comply with this section provided joints are sealed and materials are installed as air barriers in accordance with the manufacturer's instructions…" [15].

"Compliance Checking Attribute", "Deontic Operator Indicator", and "Quantitative Relation" all showed 100% precision. This could be partially attributed to their unique semantic and/or syntactic features. For example, the concepts corresponding to "Compliance Checking Attribute" all semantically represent some properties like "air leakage rate" and "U-factor". The syntactic features corresponding to "Deontic Operator Indicator" all include the POS tag "MD" for a modal verb (e.g., shall, must), while the syntactic features for "Quantitative Relation" all correspond to verbs in different tenses. But, the perfect precision for "Deontic Operator Indicator" and "Quantitative Relation" may also be partially attributed to the use of dependency information: utilizing dependency information helped avoid errors that result from independent extraction.

One recall error, however, occurred due to sequential extraction. Failure to extract the "Subject" led to failure in extracting its related "Subject Restriction". This indicates that the use of dependency information may sometimes overconstrain the matching conditions, which rules out instances that should be extracted.

5.3. Effects of cascaded extraction

The results show that the proposed domain-specific preprocessing techniques and cascaded IE methods are effective in dealing with long provisions, hierarchically-complex sentence structures, and exceptions. This can be illustrated by the extraction results for the instances of "Subject Restriction". Only two out of the 87 subject restrictions showed recall errors, and only one of them was due to errors in cascaded extraction. The following instance (which is an exception consisting of a complex restriction) showed a recall error because of missing uncommon patterns: "Exception: Economizers are not required for the systems listed below. 2. Where more than 25 percent of the air designed to be supplied by the system is to spaces that are designed to be humidified above 35 F dew-point temperature to satisfy process needs." [15]. In extracting the information for this complex restriction, in a cascaded way, only partial information ("more than 25 percent of the air" and "spaces that are designed to be humidified above 35 F dew-point temperature") is correctly extracted, whereas the complex relationship "designed to be supplied by the system is to" was not extracted.

5.4. Sources of extraction errors

Three sources of errors were identified: missing uncommon patterns, conflict resolution errors, and NLP tool errors. Extraction of "Subject" both achieved the lowest recall (90.6%) and lowest precision (92.7%) among the nine SIEs because of the following two reasons: missing uncommon patterns and conflict resolution errors. There are two interesting cases of missing uncommon patterns. First, the subject is prescribed in the provision heading, not the provision itself. For example, in the following provision, although the "fan" was extracted as an instance of "Subject", it is the "heat rejection equipment fan" that should have been extracted as the subject: "C403.4.4 Heat rejection equipment fan speed control. Each fan powered by a motor of 7.5 hp or larger shall have the capability to operate that fan at two-thirds of full speed or less…" [15]. Second, the subject is implicitly prescribed. For example, in the following sentence, the subject that corresponds to "minimum skylight area" (i.e., the "Compliance Checking Attribute") is "skylight", which is implicitly prescribed: "In an enclosed space greater than 10,000 square feet, …the total daylight zone under skylights shall be not less than half the floor area and shall provide a minimum skylight area to daylight zone under skylights of either: 1. Not less than 3 percent with a skylight VT of at least 0.40;" [15].

For conflict resolution errors, few conflict resolution rules caused errors in extraction as a result of resolving conflicts (e.g., a conflict occurs when multiple instances of a "Subject" are extracted) incorrectly. For example, in the following sentence, both "supply air systems" and "VAV systems" were initially extracted as instances of "Subject", and after conflict resolution "VAV systems" was finally extracted, which is incorrect: "Supply air systems serving multiple zones shall be VAV systems which, during periods of occupancy, are designed and capable of being controlled to reduce primary air supply to each zone to one of the following before reheating, recooling or mixing takes place: 1. Thirty percent of the maximum supply air to each zone." [15].

Both "Deontic Operator Indicator" and "Quantitative Relation" showed recall errors (98.2% and 98.1% recall, respectively) resulting from missing uncommon patterns and NLP tool errors. For "Deontic Operator Indicator", the recall error comes from missing uncommon patterns. For example, one extraction rule for "Deontic Operator Indicator" states that besides matching the syntactic feature tag "MD", the instance should also be preceded with a semantic feature (i.e., a concept). However, in the following sentence, the time adverbial "when initiated" is a very uncommon pattern which led to failure in extracting "shall" as an instance of "Deontic Operator Indicator": "…4. The override switch, when initiated, shall permit the controlled lighting to remain on for a maximum of 2 hours;" [15]. For "Quantitative Relation", the recall error comes from tokenization errors in the inner tool, which mistakenly assigned adjective tag "JJ" to a past participle verb; and, thus, the verb was not extracted as an instance of "Quantitative Relation".

The recall error for "Quantity Restriction" (95.7% recall) comes from missing uncommon patterns. The precision error for "Quantity Value" (98.9% precision) occurred due to conflict resolution errors. Other than that, as discussed above, the two recall errors for "Subject Restriction" (97.7% recall) occurred due to errors in cascaded extraction and sequential extraction.

5.5. Application

In terms of future applications, the proposed OBIE algorithm will be used to extract building energy requirements from energy conservation codes for supporting ACC. An application example is illustrated in Fig. 7. As shown in Fig. 7, the application includes six primary processes: (1) code text classification (input: raw unclassified text; output: raw classified text): automatically classifying the text in the energy conservation codes to filter out irrelevant sentences; (2) code information extraction (input: raw classified text; output: instances of SIEs): automatically extracting the building energy requirements from an energy conservation code, using the proposed OBIE algorithm, where each extracted requirement is represented in terms of instances of SIEs (e.g., "internally illuminated exit sign" is an instance of the SIE "Subject"); (3) code information transformation (input: instances of SIEs; output: logic rules): automatically transforming the extracted instances of SIEs into logic

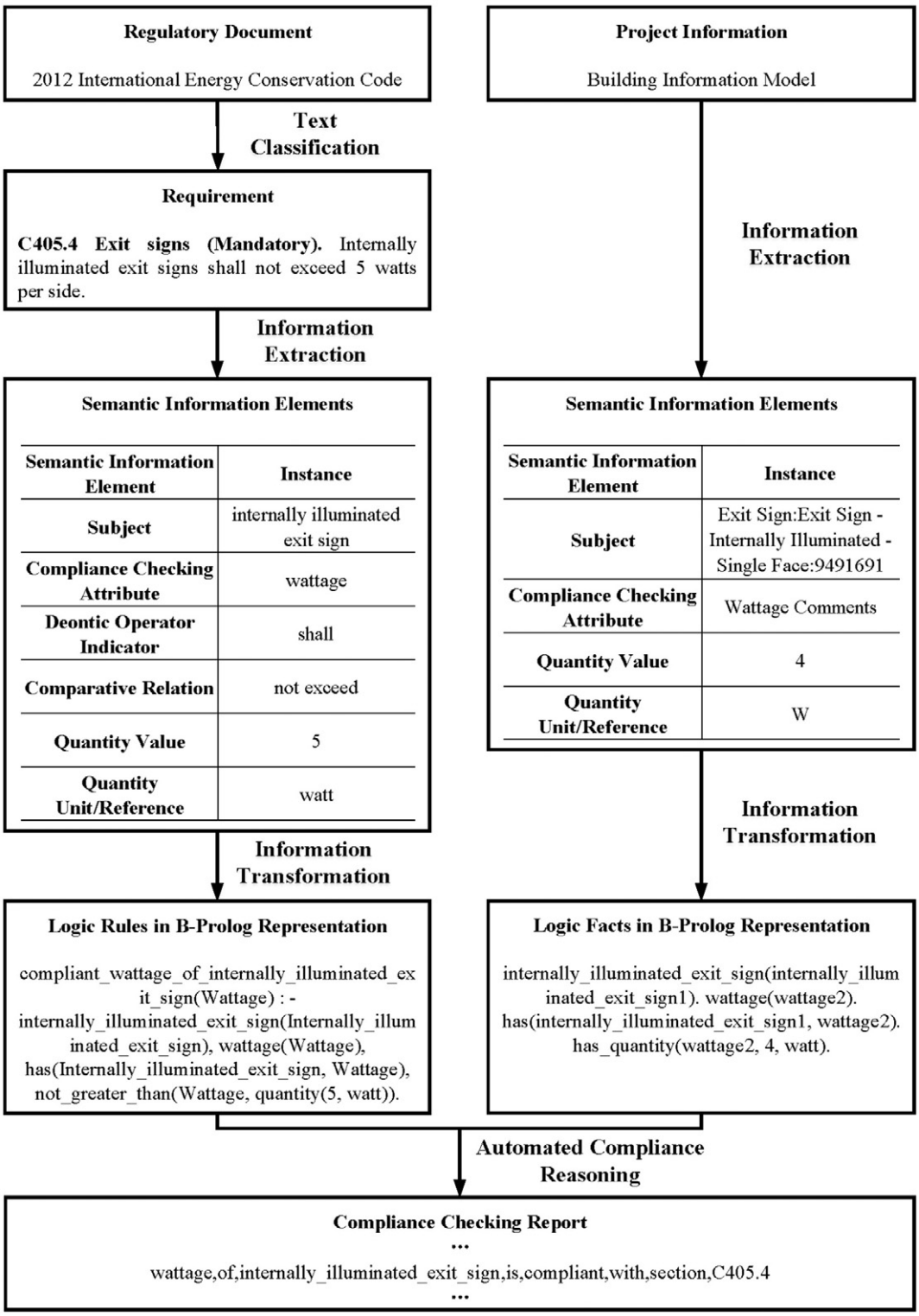

Fig. 7. Application example: use of the proposed OBIE algorithm for supporting energy compliance checking.

rules; (4) design information extraction (input: .ifc file; output: instances of SIEs): automatically extracting the design information from a building information model (.ifc file); (5) design information transformation (input: instances of SIEs; output: logic facts): automatically transforming the extracted information into logic facts that are aligned with the logic rules; and (6) compliance reasoning (input: logic facts and rules; output: compliance checking report): automatically reasoning about the logic facts and the logic rules and generating a compliance checking report. For the example, as illustrated in Fig. 7, the wattage of the internally illuminated exit sign is compliant with the rule because its value "4" is less than "5", which is the maximum required wattage. One major issue that needs further investigation, in future work, is aligning the concept representations of the logic facts with those of the logic rules, so that both representations can be interpreted together in one system. For instance, in the example shown in Fig. 7, it is important to match the "internally illuminated exit sign" (which is an instance of

the SIE "Subject") to the "Exit Sign:Exit Sign - Internally Illuminated - Single Face:9491691" (which is a design information instance) and align both concept representations (e.g., both are represented as "internally illuminated exit sign", as shown in Fig. 7). Similarly, as per Fig. 7, "wattage" and "watt" (which are instances of the SIEs "Compliance Checking Attribute" and "Quantity Unit/Reference", respectively) will be matched to "Wattage Comments" and "W"(which are design information instances).

6. Limitations and future work

Two main limitations of the OBIE algorithm are acknowledged. First, although the proposed OBIE algorithm has successfully addressed some semantic ambiguities (e.g., see the discussion in Section 5.2), it cannot – at least at this point – address all semantic interpretation issues (e.g., those discussed in Solihin and Eastman [36,37] such as dependencies and hidden assumptions) or deal with requirements that require human judgment by nature. Further research is needed to study the challenging types of semantic interpretation and ambiguity issues such as hidden assumptions, explore the limits of machine intelligence, and identify which types of requirements can be extracted in a fully-automated way and which would require some level of human involvement or verification. Even for the latter types of requirements, the proposed OBIE algorithm could be very useful in acting as a first-level interpretation of the requirements in an automated, repeatable, and consistent manner – allowing a human user or expert to further verify the automatically extracted information, clarify any semantic ambiguities, capture any hidden assumptions or implicit domain knowledge, and ensure the alignment with the concept representations of the design information. Second, the use of dependency information may sometimes overconstrain the matching conditions, thereby resulting in failures to extract depender SIEs. It is, thus, essential to achieve high performance in extracting dependees, especially at the top of the conceptual dependency structure. In future work, in order to study possible ways for further performance improvement, further research could be conducted to explore different methods for avoiding such overconstraining cases (e.g., using different matching conditions).

In addition, three limitations that may manifest themselves in future applications, if the proposed algorithm is used for a different knowledge domain (e.g., construction safety) or to extract information from a different type of document (e.g., contract specifications), are acknowledged. First, like any other ontology-based method, (1) the performance of the proposed OBIE algorithm highly depends on the coverage of the ontology used, and (2) additional human effort may be required to build a new ontology or extend this ontology for applying this algorithm to a different knowledge domain (other than the domain of "commercial building energy efficiency", which is the scope of this ontology). However, ontologies are now more widely used in construction domain applications [47], and are by nature easily reusable and extendable [11]. In future work, the ontology could be extended to cover other knowledge domains (e.g., fire safety) and the adapted algorithm could be tested in checking the compliance with related codes and regulations (e.g., the International Fire Code). The proposed methodology could also be tested in extracting information from the IECC using another ontology (but which also covers the domain of building energy efficiency) to test the impact of different ontologies (which could naturally vary in coverage, structure, semantics, etc.) on the performance of extraction. Second, because dependency relations may vary from one type of text to another, the developed conceptual dependency structure may need adaptation for extracting requirements from different types of documents (e.g., contract specifications). In future work, the authors will do further studies to see if the proposed conceptual dependency structure will require adaptation for extracting requirements from contract specifications. Third, like any other rule-based method, the developed extraction rules may require further adaptation when used for extracting different types of requirements or for extracting similar requirements but from a different type of text. However, these rules are potentially reusable in extracting building energy requirements from other types of energy regulatory documents/text. The rules could be reused as is or adapted – through modification or extension – based on additional development text. Compared with the authors' initial efforts, future efforts in adapting the extraction rules should be significantly lower. Once the rules are adapted, the IE process is fully automated and requires no user manual effort.

Two limitations that are related to the scope of the work and the testing are also highlighted. First, the scope of the OBIE algorithm is limited to natural text and excludes requirements in tables, formulas, and cross references. These will be addressed in future work, through separate but supporting linked algorithms. For example, for tables, a set of extraction rules will be developed to parse the tables (e.g., HTML format) and extract the requirements. It is expected that tables are relatively easier to process because in comparison to text – which is unstructured – requirements in tables are structured. Second, due to the substantial manual effort needed for developing a gold standard for testing and evaluation, the proposed OBIE algorithm was tested only on one chapter. Thus, future work is needed to test the algorithm on more energy regulatory documents, such as the ANSI/ASHRAE/IES/USGBC Standard 189.1-2014 Standard for the Design of High-Performance Green Buildings, the 2013 Building Energy Efficiency Standards, known as the California Energy Code (for representing energy codes developed by specific states), and the Ontario Building Code Supplementary Standard SB-10 (for representing energy codes developed by other countries). The results are expected to show similar high performance because of the similarity in text across different energy codes. However, further testing is needed for verification. Also, in future work, the authors will further evaluate the performance of the IE algorithm (Steps 2 to 5) when combined with both the text classification algorithm (Step 1) and the design information extraction and transformation algorithms, in one system for ACC.

7. Contribution to the body of knowledge

In comparison to existing IE efforts in the construction domain, this work contributes to the body of knowledge in four main ways. First, the proposed method integrates text classification with IE. Integrating text classification with IE allows for extracting information from pre-classified text, which avoids both errors and computational effort resulting from processing irrelevant text. Second, domain-specific preprocessing techniques are proposed to handle hierarchically-complex sentence structures and exceptions using splitting and semantic-based stitching. This allows for, both, simplifying hierarchically-complex sentence structures while taking meaning and obligation type into account, and separating the processing of exceptions from requirements. Third, this work proposes to use conceptual dependency theory to build a conceptual dependency structure for the target information and proposes a sequential dependency-based extraction method. The proposed conceptual dependency structure allows for capturing the dependency relations among the SIEs in a way that helps define the best sequence of extraction. The proposed dependency-based extraction method allows for taking such dependency relations into consideration during extraction, which leads to reduced text ambiguities and enhanced performance. The experimental results show that the use of dependency relations was effective in reducing semantic ambiguities. Fourth, this work proposes cascaded extraction methods to deal with text complexities in terms of long provisions, hierarchically-complex sentences, and exceptions. Cascaded extraction methods allow for handling a complex extraction task by breaking it down to a number of simple extraction tasks (i.e., a complex extraction task is cascaded on a number of simple extraction tasks). The experimental results show that the proposed cascaded IE methods are effective in handling these three types of text complexities.

8. Conclusions

This paper presented an OBIE algorithm for automatically extracting energy requirements from energy conservation codes to support automated energy compliance checking in construction. Ontology-based pattern-matching extraction rules that utilize both semantic features (ontology concepts) and syntactic features (POS tags, gazetteers, and/or auxiliary tags) were used. To reduce text ambiguities and enhance extraction performance, a sequential dependency-based extraction method was proposed and used, including building a conceptual dependency structure based on conceptual dependency theory and defining extraction sequence based on dependency relations. To deal with the complex text in energy conservation codes (long provisions, hierarchically-complex provisions, and provisions with exceptions), domain-specific preprocessing techniques and cascaded extraction methods were proposed and used.

The proposed algorithm was implemented in the ANNIE system in GATE, and was tested in extracting commercial building energy efficiency requirements from Chapter 4 of the 2012 IECC [15]. A performance of 97.4% recall and 98.5% precision was achieved. The experimental results indicate a number of conclusions. First, the proposed IE algorithm was effective in automatically extracting regulatory requirements from energy conservation codes. Second, the proposed domain-specific preprocessing techniques were successful in simplifying hierarchically-complex sentences and separating exceptions from requirements using splitting and stitching. Third, the proposed sequential dependency-based extraction method was effective in reducing text ambiguities and improving extraction performance, although in some cases dependency in extraction can result in failure to extract the depender which leads to recall errors. Fourth, the proposed cascaded extraction methods were successful in handling hierarchically-complex and long provisions with multiple exceptions.

Acknowledgement

The authors would like to thank the National Science Foundation (NSF). This material is based upon work supported by NSF 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 NSF.

References

[1] Y. Abuzir, M.O. Abuzir, Constructing the Civil Engineering Thesaurus (CET) using ThesWB, Proc. 2002 Intl. Workshop on Information Technology in Civil Engineering, ASCE, Reston, VA 2002, pp. 400–412, http://dx.doi.org/10.1061/40652(2003)34. ISBN 978-0-7844-0652-6.

[2] M.A. Al Qady, A. Kandil, Concept relation extraction from construction documents using natural language processing, J. Constr. Eng. Manag. 136 (3) (2010) 294–302, http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0000131. ISSN 0733-9364.

[3] E. Arendarenko, T. Kakkonen, Ontology-based information and event extraction for business intelligence, Proc. 15th Intl. Conf. on Artificial Intelligence: Methodology, Systems, and Applications, Springer, Berlin 2012, pp. 89–102, http://dx.doi.org/10.1007/978-3-642-33185-5_10. ISBN 978-3-642-33184-8.

[4] ASHRAE (American Society of Heating, Refrigerating, and Air-Conditioning Engineers), ANSI/ASHRAE/IES Standard 90.1-2010 Energy Standard for Buildings except Low-Rise Residential Buildings, 2010, ftp://law.resource.org/pub/us/code/ibr/ashrae.90.1.ip.2010.pdf (Jul. 30, 2015). ISSN 1041-2336.

[5] T.H. Beach, Y. Rezgui, H. Li, T. Kasim, A rule-based semantic approach for automated regulatory compliance in the construction sector, Expert Syst. Appl. 42 (12) (2015) 5219–5231, http://dx.doi.org/10.1016/j.eswa.2015.02.029. ISSN 0957-4174.

[6] J. Choi, J. Choi, I. Kim, Development of BIM-based evacuation regulation checking system for high-rise and complex buildings, Autom. Constr. 46 (2014) 38–49, http://dx.doi.org/10.1016/j.autcon.2013.12.005 ISSN 0926-5805.

[7] Civil Law Dictionary, Obligations, 2015, http://civillawdictionary.pbworks.com/w/page/15934864/O%20Civil%20Law (Dec. 15, 2015).

[8] H. Cunningham, D. Maynard, K. Bontcheva, Text Processing with GATE (Version 6), Univ. of Sheffield Dept. of Computer Science, 2011. ISBN 9780956599315.

[9] J. Dimyadi, C. Clifton, M. Spearpoint, R. Amor, Regulatory knowledge encoding guidelines for automated compliance audit of building engineering design, Proc. 2014 Intl. Conf. on Computing in Civil and Building Engineering, ASCE, Reston, VA 2014, pp. 536–543, http://dx.doi.org/10.1061/9780784413616.067. ISBN 978-0-7844-1361-6.

[10] C. Eastman, J. Lee, Y. Jeong, J. Lee, Automatic rule-based checking of building designs, Autom. Constr. 18 (8) (2009) 1011–1033, http://dx.doi.org/10.1016/j.autcon.2009.07.002. ISSN 0926-5805.

[11] N. El-Gohary, T. El-Diraby, Domain ontology for processes in infrastructure and construction, J. Constr. Eng. Manag. 136 (7) (2010) 730–744, http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0000178. ISSN 0733-9364.

[12] G. Ganu, A. Marian, N. Elhadad, URSA-User Review Structure Analysis: Understanding Online Reviewing Trends, Rutgers Univ., New Brunswick, NJ, 2010, http://itc.scix.net/data/works/att/w78-2010-54.pdf (July 14, 2016).

[13] R. Grishman, Information extraction: capabilities and challenges, Notes for the 2012 Intl. Winter School in Language and Speech Technologies, 2012, http://www.cs.nyu.edu/grishman/tarragona.pdf (July 12, 2016).

[14] E. Hjelseth, N. Nisbet, Exploring semantic based model checking, Proc. CIB W78 2010: 27th Intl. Conf., Virginia Tech., Blacksburg, VA, 2010, http://itc.scix.net/data/works/att/w78-2010-54.pdf (Dec. 20, 2015). ISBN 0 534-94965-7.

[15] ICC (International Code Council), 2012 International Energy Conservation Code, 2012, http://publicecodes.cyberregs.com/icod/iecc/2012/ (Mar. 21, 2015). ISBN 978-1-60983-058-8.

[16] N. Ireson, F. Ciravegna, M.E. Califf, D. Freitag, N. Kushmerick, A. Lavelli, Evaluating machine learning for information extraction, Proc. 22nd Intl. Conf. on Machine Learning, ACM, New York, NY 2005, pp. 345–352, http://dx.doi.org/10.1145/1102351.1102395. ISBN 1-59593-180-5.

[17] L. Jiang, R. Leicht, Automated rule-based constructability checking: case study of formwork, J. Manag. Eng. (2014), http://dx.doi.org/10.1061/(ASCE)ME.1943-5479.0000304 A4014004. ISSN 0742-597X.

[18] D. Jurafsky, J.H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Prentice Hall, Upper Saddle River, NJ, 2009. ISBN 978-0131873216.

[19] V. Karkaletsis, P. Fragkou, G. Petasis, E. Iosif, Ontology based information extraction from text, Lecture Notes in Computer Science, Springer, Berlin 2011, pp. 89–109, http://dx.doi.org/10.1007/978-3-642-20795-2_4. ISBN 978-3-642-20794-5.

[20] H. Kilicoglu, G. Rosemblat, M. Fiszman, T.C. Rindflesch, Constructing a semantic predication gold standard from the biomedical literature, BMC Bioinf. 12 (1) (2011) 486, http://dx.doi.org/10.1186/1471-2105-12-486. ISSN 1471-2105.

[21] T.S. Li, B.M. Good, A.I. Su, Exposing ambiguities in a relation-extraction gold standard with crowdsourcing, Bio-Ontologies SIG 2016, ISMB, Bethesda, MY, 2015, pp. 1–4 http://arxiv.org/abs/1505.06256.

[22] Z. Li, K. Ramani, Ontology-based design information extraction and retrieval, J. Artif. Intell. Eng. Des. Anal. Manuf. 21 (2) (2007) 137–154, http://dx.doi.org/10.1017/S0890060407070199. ISSN 0890-0604.

[23] S. Malsane, J. Matthews, S. Lockley, P.E.D. Love, D. Greenwood, Development of an object model for automated compliance checking, Autom. Constr. 49 (Part A) (2015) 51–58, http://dx.doi.org/10.1016/j.autcon.2014.10.004. ISSN 0926-5805.

[24] C.D. Manning, H. Schutze, Foundations of Statistical Natural Language Processing, MIT Press, Cambridge, MA, 1999. ISBN 978-0262133609.

[25] J.P. Martins, A. Monteiro, LicA: a BIM based automated code-checking application for water distribution systems, Autom. Constr. 29 (2013) 12–23, http://dx.doi.org/10.1016/j.autcon.2012.08.008. ISSN 0926-5805.

[26] D. Maynard, W. Peters, Y. Li, Metrics for evaluation of ontology-based information extraction, Proc. WWW 2006 Workshop on Evaluation of Ontologies for the Web (EON), ACM, New York, NY, 2006, http://staffwww.dcs.shef.ac.uk/people/w.peters/eon.pdf (Nov. 14, 2016).

[27] J. Melzner, S. Zhang, J. Teizer, H.J. Bargstädt, A case study on automated safety compliance checking to assist fall protection design and planning in building information models, Constr. Manage. Econ. 31 (6) (2013) 661–674, http://dx.doi.org/10.1080/01446193.2013.780662. ISSN 0144-6193.

[28] M.F. Moens, Information Extraction: Algorithms and Prospects in a Retrieval Context, Springer, New York, Secaucus, NJ, 2006, http://dx.doi.org/10.1007/978-1-4020-4993-4. ISBN: 978-1-4020-4987-3.

[29] A. Moreno, D. Isern, A.C. López Fuentes, Ontology-based information extraction of regulatory networks from scientific articles with case studies for *Escherichia coli*, Expert Syst. Appl. 40 (8) (2013) 3266–3281, http://dx.doi.org/10.1016/j.eswa.2012.12.090. ISSN 0957-4174.

[30] N. Nawari, Automating codes conformance, J. Archit. Eng. (2012) 315–323, http://dx.doi.org/10.1061/(ASCE)AE.1943-5568.0000049. ISSN S1076-0431.

[31] J.P. Pestian, L. Deleger, G.K. Savova, J.W. Dexheimer, I. Solti, Natural language processing—the basics, Pediatric Biomedical Informatics: Computer Applications in Pediatric Research, Springer, Netherlands, Dordrecht, 2012, pp. 149–172, http://dx.doi.org/10.1007/978-94-007-5149-1_9. ISBN 978-94-007-5149-1.

[32] J. Piskorski, R. Yangarber, Information extraction: past, present and future, Multi-Source, Multilingual Information Extraction and Summarization, Springer, Berlin, 2013, pp. 23–49, http://dx.doi.org/10.1007/978-3-642-28569-1_2 ISBN 978-3-642-28568-4.

[33] J. Qi, R. Issa, J. Hinze, S. Olbina, Integration of safety in design through the use of building information modeling, Proc. 2011 Intl. Workshop on Computing in Civil Engineering 2011, ASCE, Reston, VA, 2011, pp. 698–705, http://dx.doi.org/10.1061/41182(416)86. ISBN 978-0-7844-1182-7.

[34] A. Robbins, L. Lamb, E. Hannah, Learning the Vi and Vim Editors, 7th Ed O'Reilly, Sebastopol, CA, 2008. ISBN 978-0-596-52983-3.

[35] D. Salama, N. El-Gohary, Automated compliance checking of construction operation plans using a deontology for the construction domain, J. Comput. Civ. Eng. (2013) 681–698, http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000298. ISSN 0887-3801.

[36] W. Solihin, C. Eastman, Classification of rules for automated BIM rule checking development, Autom. Constr. 53 (2015) 69–82, http://dx.doi.org/10.1016/j.autcon.2015.03.003. ISSN 0926-5805.

[37] W. Solihin, C. Eastman, A knowledge representation approach to capturing BIM based rule checking requirements using conceptual graph, Proc. 32nd CIB W78

Conf. 2015, Eindhoven Univ. of Tech, Eindhoven, Netherlands, 2015, pp. 686–695, http://itc.scix.net/data/works/att/w78-2015-paper-071.pdf (Nov. 14, 2016).
[38] E. Soysal, I. Cicekli, N. Baykal, Design and evaluation of an ontology based information extraction system for radiological reports, Comput. Biol. Med. 40 (11–12) (2010) 900–911, http://dx.doi.org/10.1016/j.compbiomed.2010.10.002. ISSN 0010-4825.
[39] X. Tan, A. Hammad, P. Fazio, Automated code compliance checking for building envelope design, J. Comput. Civ. Eng. 24 (2) (2010) 203–211, http://dx.doi.org/10.1061/(ASCE)0887-3801(2010)24:(203). ISBN 0887-3801.
[40] J. Tao, A.V. Deokar, O.F. El-Gayar, An ontology-based information extraction (OBIE) framework for analyzing initial public offering (IPO) prospectus, Proc. 2014 47th Hawaii Intl. Conf. on System Sciences, IEEE, Washington, DC, 2014, pp. 769–778, http://dx.doi.org/10.1109/HICSS.2014.103. ISSN 1530-1605.
[41] Y. Tateisi, Y. Shidahara, Y. Miyao, A. Aizawa, Annotation of computer science papers for semantic relation extraction, Proc. 9th Intl. Conf. on Language Resources and Evaluation (LREC'14), European Language Resources Association (ELRA), Paris, France, 2014, ISBN 978-2-9517408-8-4.
[42] D.C. Wimalasuriya, D.J. Dou, Ontology-based information extraction: an introduction and a survey of current approaches, J. Inf. Sci. 36 (3) (2010) 306–323, http://dx.doi.org/10.1177/0165551509360123. ISSN 0165-5515.
[43] J. Zhang, N. El-Gohary, Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking, J. Comput. Civ. Eng. (2013), http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000346, 04015014.
[44] J. Zhang, N. El-Gohary, Automated information transformation for automated regulatory compliance checking in construction, J. Comput. Civ. Eng. (2015), http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000427, B4015001.
[45] B.T. Zhong, L.Y. Ding, H.B. Luo, Y. Zhou, Y.Z. Hu, H.M. Hu, Ontology-based semantic modeling of regulation constraint for automated construction quality compliance checking, Autom. Constr. 28 (2012) 58–70, http://dx.doi.org/10.1016/j.autcon.2012.06.006. ISSN 0926-5805.
[46] P. Zhou, N. El-Gohary, Ontology-based multilabel text classification of construction regulatory documents, J. Comput. Civ. Eng. (2015), http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000530, 04015058.
[47] Z. Zhou, Y. Goh, L. Shen, Overview and analysis of ontology studies supporting development of the construction industry, J. Comput. Civ. Eng. (2016), http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000594, 4016026.