Journal of Computational Design and Engineering, 2020, 7(5), 563–576

doi: 10.1093/jcde/qwaa046
Journal homepage: www.jcde.org
Advance Access Publication Date: 14 May 2020

RESEARCH ARTICLE

Deep learning-based extraction of predicate-argument structure (PAS) in building design rule sentences☆

Jaeyeol Song[1], Jin-Kook Lee[1,*], Jungsik Choi[2] and Inhan Kim[3]

[1]Department of Interior Architecture and Built Environment, Yonsei University, Seoul, 03722, Republic of Korea; [2] Department of IT Convergence Engineering, Hanyang University ERICA, Gyeonggi, 15588, Republic of Korea and [3]Department of Architecture, Kyung Hee University, Gyeonggi, 17104, Republic of Korea

☆Selected paper from the International Congress and Conferences on Computational Design and Engineering 2019, 7-10 July 2019, Penang, Malaysia
*Corresponding author: E-mail: leejinkook@yonsei.ac.kr

Abstract

This paper describes an approach to extracting a predicate-argument structure (PAS) in building design rule sentences using natural language processing (NLP) and deep learning models. For the computer to reason about the compliance of building design, design rules represented by natural language must be converted into a computer-readable format. The rule interpretation and translation processes are challenging tasks because of the vagueness and ambiguity of natural language. Many studies have proposed approaches to address this problem, but most of these are dependent on manual tasks, which is the bottleneck to expanding the scope of design rule checking to design requirements from various documents. In this paper, we apply deep learning-based NLP techniques for translating design rule sentences into a computer-readable data structure. To apply deep learning-based NLP techniques to the rule interpretation process, we identified the semantic role elements of building design requirements and defined a PAS for design rule checking. Using a bidirectional long short-term memory model with a conditional random field layer, the computer can intelligently analyze constituents of building design rule sentences and automatically extract the logical elements. The proposed approach contributes to broadening the scope of building information modeling-enabled rule checking to any natural language-based design requirements.

Keywords: automated rule checking; building information modeling (BIM); natural language processing (NLP); predicate argument structure

1. Introduction

Rule checking is conducted to assess the quality of building design generated during the design process. The building design must comply with regulatory requirements that have a binding force to obtain permission during the administration process. In addition to the regulations, following the owner's specifications or request for proposal (RFP) are also important for developing the building design and proceeding into subsequent stages. A failure to assess the building design accurately leads to a delay in the construction process and wasted budgets (Macit İlal & Günaydın, 2017). Despite its importance, the conventional rule-checking process was time consuming and error prone because of the dependence on 2D drawing plans and manual examination. Automated rule checking has been researched to address these problems and building information modeling (BIM) enables quantitative and precise compliance checking using computable information of building objects (Sacks, Eastman, Lee, & Teicholz, 2018).

With interest in automated rule checking increasing, several problems have emerged for the real-world implementation of rule-checking systems. In a review of diverse automated rule-checking systems, Eastman et al. identified that the BIM-based rule-checking process is widely implemented with four steps: (i) rule interpretation and logical structuring, (ii) building model preparation, (iii) rule execution, and (iv) reporting of the

Received: 5 January 2020; **Revised:** 26 March 2020; **Accepted:** 7 April 2020

checking results (Eastman, Lee, Jeong, & Lee, 2009). Encoding the human-readable building design rule sentences into machine-readable rules is one of the challenges in implementing a rule-checking system. The accuracy of rule interpretation and logical structuring (the rule-making process) is crucial to precise rule checking. However, the vagueness and complexity of natural language sentences are bottlenecks for translating natural language sentences into structured data (Nawari, 2012).

Since the early period of rule-checking research in the 1960s, various approaches to encoding natural language have been developed. However, rule-making approaches were still based on manual interpretation and translation. Manual interpretation can ensure the accuracy of the rule-making process, but complicates generating or modifying computer-readable rules. Numerous requirements for building design have recently emerged as complexity and building size increases. The application of an automated rule-checking system for each building design project requires the translating of diverse requirements for each project. Computer-readable rules for these regulations must also be updated based on the revised building codes and by-laws. Tracking the requirement changes and updating the corresponding computer-readable rules reduce the efficiency of the rule-making process. To address these problems, a more intelligent approach to the rule-making process must be developed.

Accordingly, this paper proposes an intelligent rule-making process that uses deep learning-based natural language processing (NLP) techniques. Application of artificial intelligence (AI) technology in architecture, engineering, and construction industry has been carried out since the computer-aided design was proposed (Mitchell, 1975). As AI technology has been developed, the adoption of machine learning methods is increasing in architecture, engineering, and construction domain. Research for developing rule-checking system also has been attempted to adopt the machine learning methods. The representative example is a research area called "semantic enrichment" that aims to check the semantic integrity of BIM objects or infer the semantic information (Rafael et al., 2017; Bloch & Sacks, 2018; Koo & Shin, 2018; Koo, La, Cho, & Yu, 2019). Several studies have employed NLP techniques for information extraction in the rule-checking process. J. Zhang and El-Gohary proposed an automated code compliance system and automated rule translation with NLP techniques (J. Zhang & El-Gohary, 2017). However, the previous research used rule-based information extraction, which is only applicable to limited scope of design rule sentences.

This paper proposes a deep learning-based information extraction model that can process building design rule sentences from more diverse sources. From the previous research, we found that machine learning-based NLP techniques could help the computer learn the relevancy of semantic meaning with neural network-based word embedding, and the word-embedding results could be used in the rule-making process (Song, Kim, & Lee, 2018). The relevancy training results are used to train the deep learning-based information extraction model for an automated design rule-checking system. As an early phase of the research, this paper focused on extracting the semantic role of each component in a given sentence based on the predicate-argument structure (PAS). The scope of this paper is to propose a deep learning-based PAS extraction process and validate the proposed process by implementing the process.

This paper is organized into six sections, with the remainder of this paper as follows. Section 2 reviews the rule-making process and information extraction techniques in general domains. Section 3 describes the approach to applying information extraction to automated design checking, defines the semantic role in regulatory sentences, and classifies the PAS types. Section 4 proposes the deep learning-based PAS extraction process for Korean building design sentences. Section 5 validates the performance of the proposed PAS extraction model with a demonstration. Section 6 concludes the paper with contributions and discussion of the application of intelligent techniques.

2. Background

2.1. Previous studies on rule-making approaches

BIM can facilitate the quantitative evaluation for building design based on the computational data of building objects and their associated properties. The computational information of building elements helps to conduct not only the visual inspection of 3D model but also the variable assessment in design phase. Simply deriving or calculating the data of building elements, architects, or other stakeholders can assess the construction safety (Zhang, Teizer, Lee, Eastman, & Venugopal, 2013), building circulations for evacuation, and walkability of given building design (Choi, Choi, & Kim, 2014; Shin & Lee, 2019). As the adoption of rule checking has been expanded to various domains, the range of design rule also expanded (Solihin & Eastman, 2015).

Interpretation and translation of natural language-based regulations for automated rule checking have been researched since 1966 when Fenv investigated decision tables to represent structured regulatory codes (Preidel & Borrmann, 2016; Ismail, Ali, & Iahad, 2017). In early development projects, most of the rules were hard-coded into the rule-checking software. Solibri Model Checker, which is one of the most well-known software program for design assessment, was also implemented with hard-coded rule sets. In Solibri Model Checker, end-users can adjust certain parameters with given rule templates using the Solibri Ruleset Manager, but it is difficult to generate or modify rule sets because the software was implemented with predefined checking functions and a native data format (J. Zhang & El-Gohary, 2017). To overcome the limitation of the "black box" implementation, several approaches to enhancing the transparency and functionality of computerized building codes and regulations have been proposed.

Most of the rule interpretation approaches are based on logic rules defined by domain experts. To eliminate vagueness and clarify the semantics of natural languages, logical rules were defined with domain knowledge about code checking and general linguistics. The first-order logic, conceptual graph, and deontic logic were used for interpretation (Ismail et al., 2017) and the interpreted information was represented through domain-specific languages or open-standard data schemas. A recent review and analysis of language-driven rule-checking systems (Solihin, Dimyadi, & Lee, 2019) reported an assessment of language-based methods. The use of open-standard data schemas with RuleML or LegalRuleML has also been increasing (Ghannad, Lee, Dimyadi, & Solihin, 2019). Recently, Nora El-Gohary suggested using NLP and machine learning techniques based on the logic-based rule interpretation approach (Pauwels & Zhang, 2015; Ruichuan & El-Gohary, 2019).

KBimLogic is a logic rule-based mechanism that was developed to translate Korean Building Act sentences into a computer-readable script language (Lee, Lee, Park, & Kim, 2016; Kim, Lee, Shin, & Choi, 2019). KBimLogic is composed of three logical elements: (i) building objects and properties (noun phrases), (ii) methods for checking (predicates), and (iii) logical relationships between sentences. KBimLogic enables architects and rule-checking experts who are not familiar with

programming to translate natural language-based legislations into a computer-readable format, named KBimCode. Furthermore, KBimCode is managed with a meta-database that accumulates the translated script code data for each logical element. However, KBimCode database is limited in the specific corpus derived from the target sentences. The corpus data should be expanded manually like other logic rule-based approaches. Machine learning techniques can contribute to alleviating the time and cost for expanding the scope of KBimLogic.

2.2. General information extraction process using the NLP technique

Information extraction is a research topic in the NLP discipline and aims to transform unstructured data into structured information (Pollock, Waller, & Politt, 2010). Information extraction is focused on identifying the instance of a class of events or the relationship between entities (Cowie & Wilks, 1996). To achieve this goal, information extraction has several sub-tasks, such as named entity recognition (NER), semantic role labeling (SRL), and relations extraction. These techniques were adopted for understanding news and messages about general events or terrorist events. Transforming the information into a structured format, information extraction techniques could be used to construct and manage the information within the database. The output of information extraction can be used for other NLP tasks such as machine translation, question understanding, and answering with a computer-readable data format.

Extracting structured information from a natural language sentence comprises several sub-tasks. NER is regarded as the first step in information extraction. A named entity refers to a specific noun word that can be classified into specific categories, such as a person's name, a country, an organization, or a numeric expression. Extracting and classifying entities from sentences can help to identify the exact information associated with a specific entity mentioned in a sentence. SRL is more focused on the semantic relationship between entities and is regarded as a shallow semantic processing task. SRL extracts the semantic role of each entity based on the meaning of the predicates. The output of the SRL is a PAS that represents the relationship and semantic role of each argument.

In the early period of machine learning-based approaches, the input features of languages were manually constructed with language-specific knowledge. Although the hand-crafted features could automate NER tasks, the development of new resources and features for other languages and new domains remained challenging (Lample, Ballesteros, Subramanian, Kawakami, & Dyer, 2016). To address the problem with extracting features, Collobert et al. (2011) proposed a neural network model using unsupervised features instead of hand-crafted features. Advancements in unsupervised learning of word embedding (Mikolov, Sutskever, Chen, Corrado & Dean, 2013) enabled the development of neural network-based architectures. However, this model uses a simple feed-forward neural network and depends solely on word embedding, which is unable to process various sentence lengths and exploit explicit features (Chiu & Nichols, 2016). The recurrent neural network (RNN) and the long short-term memory (LSTM) model can process variable lengths of sentences and use long-term memory. Leveraging these features, RNN, LSTM, and bidirectional LSTM (Bi-LSTM) have recently been used for sequence labeling tasks (including NER tasks) and have demonstrated outstanding performance (Huang, Xu, & Yu, 2015; Gridach, 2017). Machine learning-based NER techniques also have been applied to specific domains, especially for biomedical research (Gridach, 2017). In this study, we propose using a Bi-LSTM model to extract required regulatory information from Korean Building Act sentences.

2.3. NLP-based information extraction for design requirement analysis

To develop a more intelligent rule-making process, this paper proposes an NLP-based information extraction technique for rule interpretation and translation. The purpose of rule interpretation and translation in a BIM-enabled rule-checking system is to precisely generate computer-readable rules based on given design requirements, which are based on the explicit and structured data of building objects and their properties. Accordingly, the information expressed in the given sentence should be extracted seamlessly and represented in a structured format. The words representing the building objects and their properties can be captured by a lexical analysis comparing the input words and the defined words in a database. However, structuring the relationship between the words requires a semantic understanding of languages.

Information extraction in the NLP discipline can be used as an intermediate process to translate natural language-based design requirement sentences into computer-executable code. The goal of information extraction is to capture information with pre-defined templates composed of frame-like structures representing specific events such as actors, times, and locations (Surdeanu, Harabagiu, Williams, & Aarseth, 2003). When applying information extraction to a specific domain, appropriate templates must be defined because the classification of data and the relationship of entities imply domain knowledge. The biomedical discipline is one of the most active areas for employing information extraction techniques (Gaizauskas, Demetriou, Artymiuk, & Willett, 2003) to extract domain-specific information such as the relationships of protein and disease, where the templates for expressing the relationship are defined. In the same context, we need to identify the required information and more specific classifications for adopting information extraction for automated design rule checking. This paper applies the predicate-argument extraction from the sub-task of information extraction to extract the logic rule components in sentences as shown in Fig. 1.

3. PAS Extraction in Building Design Rule Sentences for Design Rule Checking

3.1. Classification of regulatory information in building design rule

When generating computer-readable codes from building design rule sentences, the relationship between the target object and their properties must be clarified based on the checking methods. Checking methods can be derived from the semantic meaning of predicate parts in sentences and require mandatory arguments based on the semantic meaning. From the given building design rule sentences, specific constituents are translated into input arguments of checking methods reflecting their semantic roles. To enable the computer to understand the semantic role of each constituent and their relationships, the classification of semantic roles must be defined. We adopt the concept of PAS from the linguistic field to represent the semantic relationship between the objects, properties, and other parameters required to compose computer-readable rules.

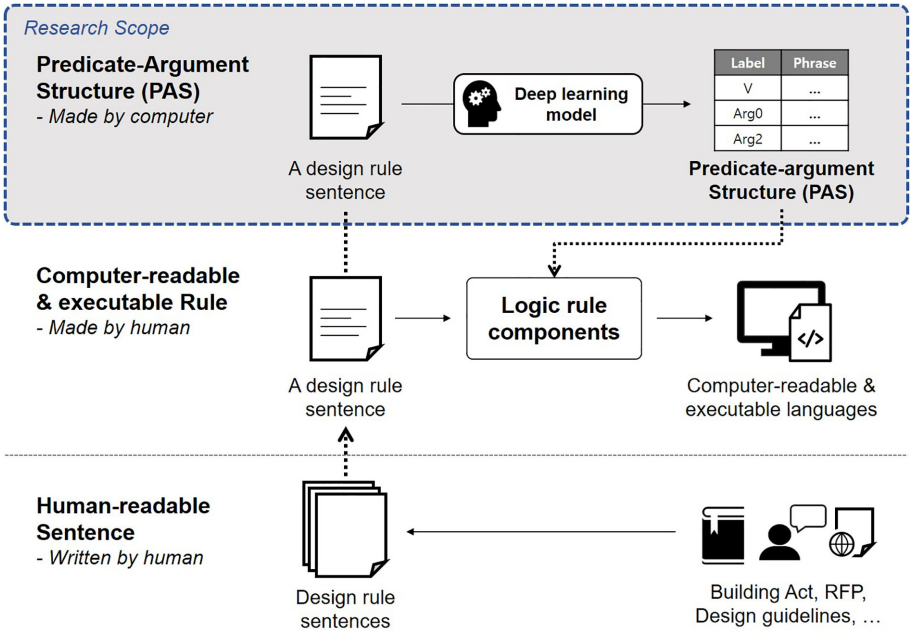


Figure 1: Overview of NLP-based regulatory information extraction for a building design rule-checking system.

Table 1: Semantic role classification for design rule checking.

Type	Semantic role	Definition
Core arguments	Object	Target object of checking rules
	Checking properties	Specific properties of building objects
	Required value	A specific value prescribed in a building design rule
	Relational object	Additional objects that have relationships with target objects
Modifiers	Secondary predication	Additional verb or adjective used to modify the objects
	Reference	Referred rules (acts, guidelines) in a building design rule
	Transition	Objects placed between the subject and relational objects
	Negation	A phrase for negating the predicate
	Condition	Logical condition for checking the design rules
	Methods	Additional parameters for counts or calculations

The classification of the semantic roles in this paper was derived from the general PASs of natural language sentences. We required domain-specific information extraction suitable for the rule-making process; accordingly, the classification was modified reflecting the context of generating computer-readable design rules. PropBank, FrameNet, and other similar corpora for general SRL have annotations for semantic roles of each argument used for analyzing the semantic structure of sentences (Kingsbury & Palmer, 2003). The arguments are classified into core arguments (numbered arguments) and modifiers to construct the semantic structure. PropBank has five numbered arguments (agent, patient, instrument, starting point, and ending point) and several other modifiers. Based on the concept of PropBank annotations, we derived the semantic role classification for design rule checking as presented in Table 1.

This research is focused on BIM-enabled design rule checking; thus, the primary target is building objects and their properties. Building design rule sentences, especially in Korean Building Act sentences, are typically written as imperative sentences with no subjects for actions. Furthermore, even if the subjects are mentioned in a sentence, the information is unnecessary for checking the design of building objects. Therefore, the subject (agent) is excluded from numbered arguments in the proposed classification.

3.2. PAS for building design rule sentences

A PAS is a representation of the semantic relationship between constituents of a sentence. A PAS can express semantic relationships expressed in various forms of sentences in a unified form. Predicates require syntactic or semantic arguments to construct a complete natural language sentence. To generate computer-readable codes, the meanings of predicates are translated into checking functions that require a set of arguments for execution. We identified the six PAS types by analyzing the syntactic constituent of building design rule sentences and their semantic meaning (Fig. 2; Table 2):

(i) PAS Type 1 is a structure that only requires the target objects (Argument 0). Building design rule sentences that exclusively verify the existence of objects are classified in this type. Using natural language, these types of sentences are expressed with an S + V + O structure or as imperative sentences, such as “architects have to install some-

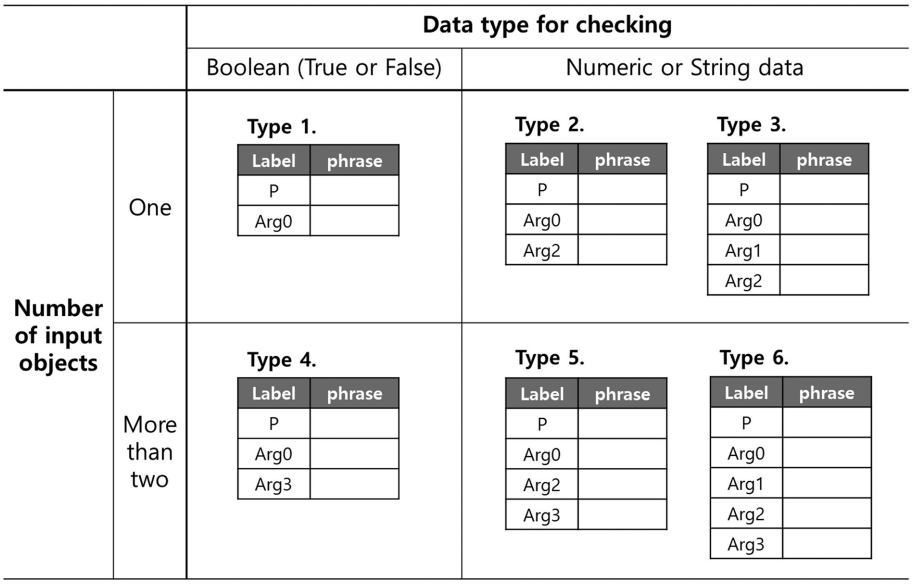


Figure 2: Conceptual graphical representation and classification of PAS.

Table 2: PAS-type classification and their properties for building design rule checking.

PAS type	Argument	Checking property type	Return data type	Required value
1	Arg0 (Target objects)	Object instance	Boolean	X
2	Arg0 (Target objects) Arg2 (Required values)	Object instance (count), General property	String or numeric	O
3	Arg0 (Target objects) Arg1 (Properties) Arg2 (Required values)	General property, Geometry property	String or numeric	O
4	Arg0 (Target objects) Arg3 (Relational objects)	Relational property	Boolean	X
5	Arg0 (Target objects) Arg2 (Required values) Arg3 (Relational objects)	Relational property (distance)	Numeric	O
6	Arg0 (Target objects) Arg1 (Properties) Arg2 (Required values) Arg3 (Relational objects)	Relational property (distance)	Numeric	O

thing" or "install walls." This structure is a translated structure for checking the installation or existence of building objects.

(ii) PAS Type 2 is a structure composed of target objects (Argument 0) and the required values (Argument 2). In this case, a sentence has a syntactic structure composed of a complement and a subject. The building design rule sentences that regulate the number of objects or specify the general properties such as material and functional usage are translated into Type 2. The specific checking property is not mentioned in this type of sentence, but can be inferred by the semantic meaning of the required values.

(iii) PAS Type 3 has specific checking properties (Argument 1) with a target object (Argument 0) and its required value (Argument 2). Similar to Type 2, Type 3 is also translated from the sentence that regulates the specific value of the properties. Some required values for general properties imply which properties are related to the values. However, geometric properties must be specified in sentences because quantitative values for geometric properties can be used without any distinction. The geometric property is used to determine how to implement the low-level algorithm that calculates the required properties. Target objects and their properties are usually expressed in the possessive form in Korean sentences, such as "interior finishing of wall" and "width of door." These phrases must be separated to objects and their properties to clarify the purpose of sentence.

(iv) PAS Type 4 is a structure for checking the relationship between different objects, which requires a pair of building objects to process the checking (Argument 0 and Argument 3). This paper focused on the physical relationships such as inclusion, connection, and adjacency. The results of checking relationships are Boolean (true or false) values, thus not needing the required value for checking.

(v) Both Type 5 and Type 6 PASs appear in sentences that review the distance between objects. The difference between Type 5 and Type 6 is the different grammatical shapes of sentences. Sentences translated into Type 5 do not clarify

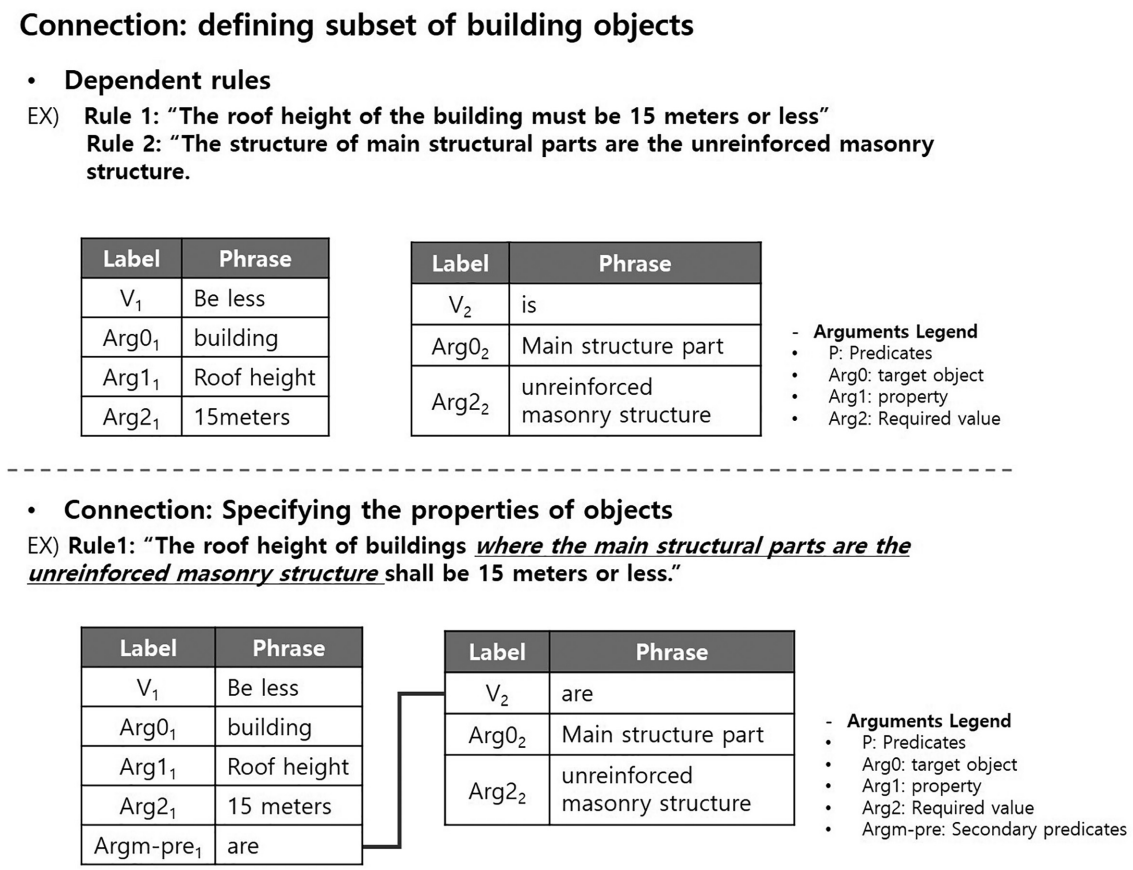


Figure 3: The concept of PAS connection for object modifications.

the measurements between two objects, such as "A and B should be more than 3 meters apart." In contrast, Type 6 specifies how to measure the distance between objects with a noun phrase, such as "The vertical distance from A to B shall not be less than 1 meter." In a former example "vertical distance" is a specific property (Argument 1) for checking the distance between the target object (Argument 0) and relational object (Argument 3).

Figure 3 illustrates the concept of connections between PASs. A PAS is generated based on the predicate, and if a given sentence has multiple predicates, they are extracted as a dependent PAS. In the connections between PASs, one PAS modifies the numbered arguments in the other PAS. In some building design rule sentences, certain modifying phrases declare the specific constraints of objects. General-purpose PAS extraction models deal with the modifying phrases as independent labeling results and are not concerned with the relationship between them. However, the modifying information is critical for the design rule-checking process because the target objects are different whether or not they have a specific condition. The objects without any modifying phrases are translated into an entire set of object instances in a given BIM model, while the objects with one or more modifiers are translated into a subset of the object class. Consequently, the proposed PAS expresses the modification of relationships with the connection. Only the numbered arguments representing the objects, Argument 0 and Argument 3, can have the connection. Other modifying phrases such as adverbs are treated as modifying arguments, as presented in Table 1.

3.3. Deep learning-based PAS extraction

We apply a deep learning model that makes advances in sequence labeling tasks. The proposed PAS extraction process aims to broaden the range of rule-making from the limited building code sentences to various sources such as design guidelines, RFPs, and even web documents. Our primary goal is to extract the design rule checking-related information from a variety of sentences. The proposed method was developed with a assumption that input sentences are entered by human users and it does not guarantee that the given sentence is related to building design. Accordingly, the proposed process must be able to classify whether the given sentence is architecture related first and recognize the semantic role of each word. The PAS extraction and sentence classification are required semantic information of given sentences. There are several NLP techniques to enable computer to process and understand the semantic meaning of languages. Recently, neural network and deep learning are used to train computer to learn the semantics of languages.

As illustrated in Fig. 4, the quantity of data and methods to implement training models differentiate classical NLP models from deep learning-based NLP models. As the performance of hardware and data collection techniques such as web crawling have improved, machine learning studies have focused on deep learning methods. Machine learning-based NLP approaches enable computers to learn how to solve given NLP problems, which can address the limitation of the rule-based methods. One of the bottlenecks in machine learning-based methods is how to represent input text data. Machine learning models consume numeric vector data for training; therefore, natural language data

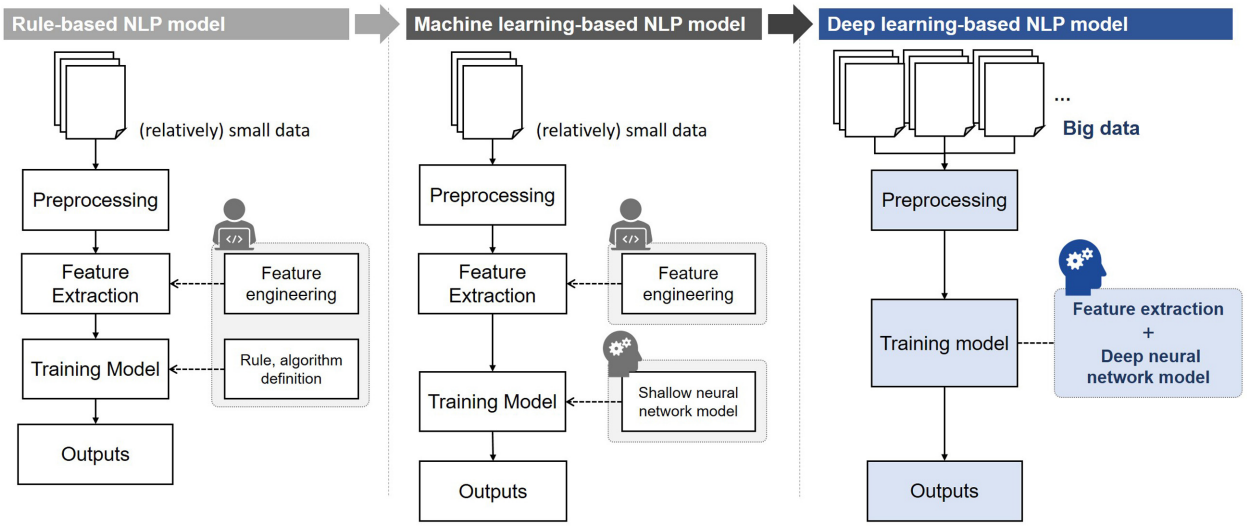


Figure 4: Comparison between classical NLP model and deep learning-based NLP model.

cannot be used directly for machine learning model. The feature need to be encoded into numeric vectors and it was based on hand-crafted features by domain experts. Using a count-based representation or grammatical features, training data were encoded with very high-dimensional and sparsely featured vectors. This type of feature representation requires significant time and computing resources to process the model. Furthermore, the context and semantic features of natural language are difficult to encode with defined rules; therefore, the training results also experienced difficulty reflecting the semantic elements. Deep learning-based NLP models marked a breakthrough with neural network-based vector representation (word-embedding techniques) and RNN models (Young, Hazarika, Poria, & Cambria, 2018).

Neural network-based vector representation enables computer to learn the semantic meaning of each word by analyzing the concurrent word's information. It also enables to infer how to translate the compound words or words from out of corpus. Through these features, neural network-based representation algorithms automate a feature extraction and reduce the dimension of vector representation. This eliminates the need for manual feature engineering and enhances the computation efficiency.

The proposed PAS extraction needs semantic analysis of each word and it has to cover the unused words in the Building Act sentences. By using the large dataset of architectural documents, we can also train the neural network model to encode words for architecture objects and their associated properties into vector format. The translated vector representation can be used as a input of deep learning model that classifies the semantic role of input words. It helps to extend the scope of logic rule for translating building requirements into computer-readable format with minimal human intervention.

4. Development of Deep Learning and NLP-Based PAS Extraction Process

4.1. Overall process

The proposed PAS extraction model has three steps: (i) pre-processing, (ii) extracting the PAS, and (iii) printing the extraction results. Pre-processing includes syntax analysis and sentence classification before PAS extraction. The goal of the proposed model is to extract the PAS for generating computer-readable design rules from various natural language sentences. Pre-processing is used to filter the non-target sentences and transform the input sentences into the proper form. After pre-processing, the deep learning-based PAS extraction model consumes the input data and prints the extraction result. The detailed process is illustrated in Fig. 5 and the details are described in the following sections.

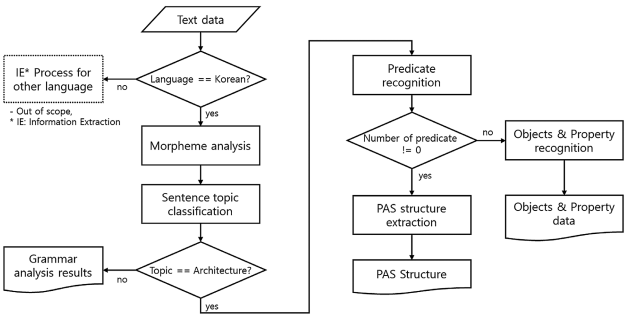


Figure 5: Deep learning and NLP-based building design PAS extraction process.

4.2. Pre-processing

Pre-processing involves basic grammatical analysis for input sentences and topic classification modules. The pre-processing step aims to enhance the accuracy of subsequent procedures by decomposing phrases into atomic units and eliminating unnecessary words. Morpheme analysis and part-of-speech (POS) tagging enable recognition of the grammatical use of each word. In the Korean language, there are post-position particles that are attached behind other words and represent their semantic roles. By separating post-position particles with morpheme analysis, exact nouns or verb words are extracted and used for training. POS tagging identifies the grammatical role of each word (e.g. noun, verb, and punctuation). After POS tagging, we excluded stop words such as punctuations and Chinese characters, which are not used for the subsequent step. For the simplicity of input sentences, we also excluded the phrases that are described in

parentheses. In Korean Building Act sentences, some additional explanation about specific words or conditions is described in parentheses. These phrases help to understand more detailed contents of design requirements. However, parentheses phrases were inserted inside the words or phrases, for example, "A. Stair room is partitioned from other parts of the building except for windows, entrance and other openings (hereafter referred to as "window etc.") with a wall of fire-proof structure." These inserted phrases make it hard to figure out the context of sequence of sentence. The pre-processing module is implemented with a Korean morpheme analysis library called KoNLPy.

The features used for extracting KBimCode elements and relationships must have numerical forms because the input of the training model is based on the calculation of numeric tensors. Word embedding becomes a common method for representing text data in numeric form. Although many methods can represent text in a numeric format, most are limited to representing the semantics of each word. Distributed representation is a breakthrough because it groups similar words and encodes the features into vector format (Mikolov et al., 2013). The concept of distributed representation has been recently implemented with variable neural network-based models, which enable the analysis of large amounts of text data. In this research, we obtained word-embedding vectors with a word-embedding model named fastText that was created by Facebook's AI Research lab (Bojanowski, Grave, Joulin, & Mikolov, 2017). The fastText model uses sub-word information to obtain word vectors, which enables a model to represent the words that do not appear in the training dataset.

We trained the word-embedding model with 24 313 sentences from building codes and RFPs. Building code sentences were collected from the KBimLogic database, which is based on data from the National Law Information Center (law.go.kr). Furthermore, 20 RFPs were collected from Nara-jang-teo, the South Korean online e-procurement system, for building projects from 2016 to 2018. The accuracy value of POS tagging is measured at 92.3% and the accuracy of sentence classification is measured at 88.21% with collected architectural sentence corpus.

4.3. Extracting the PAS using Bi-LSTM CRF model

Extracting the PAS of a given sentence requires the analysis of the semantic role of each phrase in the sentence. SRL aims to recognize and classify the semantic role of constituents for given predicates. In the proposed process, sentences that satisfy the preceding conditions—the topic is related to architecture and has one or many predicates—are inputs of the SRL model; the outputs are PASs of the input sentences.

The PAS extraction model used a Bi-LSTM with a conditional random field (Bi-LSTM CRF) model to label the sequence data (Huang et al., 2015). Bi-LSTM is a variation of RNN, which uses the state of previous data while processing the sequential data. LSTM was proposed to address the vanishing gradient problem—the signals of the sequence state vanish as the input sentences grow (Hochreiter & Schmidhuber, 1997). Bi-LSTM uses a pair of LSTM models to process the sequence data in forward and backward directions; by using bidirectional context data, it has contributed to improving the performance of various NLP tasks. CRF is a probabilistic model to tag sequence data focusing on sentence-level information rather than individual constituents of the sequence (Lafferty, McCallum, & Pereira, 2001). CRF considers the conditional probability and predicts the entire sequence of tagging jointly. In a Bi-LSTM CRF model, CRF consumes the output of the Bi-LSTM layer and makes predictions of the sequence label.

The structure of implemented Bi-LSTM CRF model is illustrated in Fig. 6. The input of the model is a sequence of morphemes generated by the pre-processing module; several semantic roles or entities are expressed with a series of morphemes. To capture the series of words as a single entity, the computer must recognize which words are at the beginning of the entity and whether the given words are part of an entity. Inside–outside–beginning (IOB) labeling is designed to clarify the position of each word within the entities. We label the training data using the IOB method, classified into 13 labels that combine the IOB label and semantic role classification. The Bi-LSTM CRF model consumes feature vectors as inputs and predicts a label for each input word. The input feature vector is composed of four features: embedding vector of an input word, predicate of sentence, POS tagging, and relational position from predicate phrases. The dimension of each LSTM layer is set to 300, using 600 dimension for bidirectional LSTM. We stacked two Bi-LSTM layers to improve the performance of training model.

4.4. Recognizing object and property information from noun phrases

The input sentences that do not have rule checking-related predicates cannot generate a PAS even though the sentence topic is related to architecture. Therefore, sentences that do not have rule checking-related predicates are sent to the objects and properties recognition model. The objects and properties recognition model is borrowed from NER in general NLP tasks.

NER is a task to recognize and classify entities from noun words. Based on the scope of the NER task, objects and property-related elements represented in noun phrases must be classified by their semantic meanings. Table 3 presents the classification of elements related to objects and their properties, represented with noun words. In building code sentences, building objects and their associated properties are expressed with noun words or phrases. The required value for specific properties is also included in the classification to capture the noun words or numbers such as specific quantitative or categorical values. Comparison operators referring to "more or less than" are also represented in a single noun word in Korean. The titles of law names or article numbers are also the target of extraction although it is not a building object or its associated properties, which are fundamental components for formalizing regulatory information. Extracting objects and properties also proceeds with the same deep learning model used for extracting the PAS, but the input feature includes only a word-embedding vector, and the annotation is labeled with object and property data.

5. PAS Extraction Model Training Results and Validation

5.1. The performance of the PAS extraction model

The SRL training dataset was established with 350 Korean building regulation sentences. From various Korean building regulation sentences, the target sentences were manually selected to collect the appropriate sentences for training. The training was conducted after pre-processing and feature extraction. Of the 350 sentences, 35 (10%) were used to test data, and the remaining sentences were divided using an 80:20 ratio for training and validation. The test data were used to validate the performance of the trained model, while the validation data were used during the training iterations. The number of epochs for the training was set to 50 because the model exhibits overfitting problems

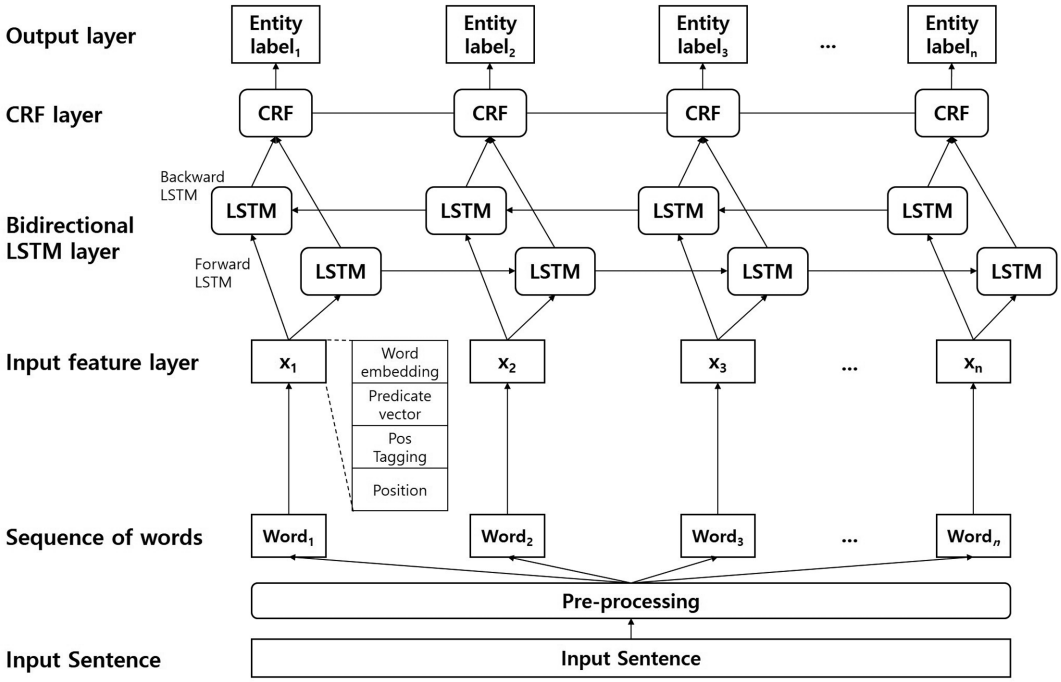


Figure 6: Deep learning-based semantic role labeling process.

Table 3: Classification of objects and property data represented with noun words.

Elements class	Definition	Example words
Object (OBJ)	A word that represents building objects	Wall, Stair, Door
Property (PRO)	A word that represents the name of the property	Height, Width, Material
Required value (RVA)	A specific required value of properties	10 meters, Non-combustible material
Relational operator (OPE)	A word that describes the quantitative relationship	More, Less, Within
Reference rule (REF)	Referred rules (acts and guidelines) in a requirement	Article 3. No 5,

beyond 50 epochs. The measure of validation accuracy with 50 epochs was 77.4%.

We also measured the F1 score to validate the performance of the proposed model. The F1 score is used to validate the performance of the machine learning model, especially for a model with an imbalanced dataset. Datasets used for SRL or NER models are naturally imbalanced because semantic roles and named entities are expressed with few words in a sentence, with all other words outside of the target labeled with "O." The accuracy of the training model can increase by predicting all words as "O," but this causes the training model to fail to extract the precise semantic role and named entity information. The F1 score is the harmonic average of precision and recall values, which are the ratios of true/positive responses to the total predicted positive observation (precision) or all observations in the class (recall). We validated the proposed model with 10-fold cross-validation. The summary of the results under 10-fold cross-validation is shown in Table 4. The F1 scores for each semantic role classes are shown in Table 5 with a test set #10.

The NER training was established using the same dataset as SRL training. The validation accuracy of the NER training model was 0.8336 and the average F1 score was 0.41 (Table 6). The precision value of the object class was much lower than the other classes, suggesting that the model predicts other classes to the object class. This can caused since words representing the building objects are used more frequently than other entities in building design rule sentences.

Table 4: 10-fold cross-validation results of SRL training model.

Test set	Precision	Recall	F1 score
1	0.42	0.59	0.44
2	0.46	0.55	0.43
3	0.49	0.56	0.45
4	0.39	0.46	0.36
5	0.35	0.38	0.29
6	0.38	0.49	0.38
7	0.38	0.52	0.37
8	0.35	0.49	0.37
9	0.36	0.43	0.36
10	0.49	0.65	0.49
Average	0.407	0.512	0.394

5.2. Discussion of training performance

The validation accuracy and F1 score values of the SRL task were lower than the general-purpose model because we conducted training with minimal training data. The existing training dataset for general SRL includes thousands of sentences in

Table 5: Precision, recall, and F1 score measures of each type of semantic role from test set #10.

Class[a]	Precision	Recall	F1 score	Count
Arg0	0.01	0.77	0.02	30
Arg1	0.82	0.78	0.80	18
Arg2	0.67	0.69	0.68	26
Arg3	0.50	0.60	0.55	5
REF	1.00	1.00	1.00	3
TRA	0.00	0.00	0.00	1
NEG	0.00	0.00	0.00	2
CON	0.00	0.00	0.00	10
MOD	0.33	0.50	0.40	2
V	0.81	0.87	0.84	30
Average	0.49	0.65	0.49	144

[a]*Class labels: Arg0 = subjects, Arg1 = properties, Arg2 = required values, Arg3 = relational objects, REF = reference rules, TRA = transition, ALT = alternatives, NEG = negation, CON = condition, MOD = method, V = verb.*

Table 6: Precision, recall, and F1 score measures of each type of entity.

Class[a]	Precision	Recall	F1 score	Count
OBJ	0.02	0.61	0.04	76
PRO	0.64	0.68	0.66	31
RVA	0.60	0.71	0.65	42
OPE	0.87	0.87	0.87	23
REF	0.33	0.50	0.40	2
Average/total	0.39	0.68	0.41	174

[a]*Class labels: OBJ = objects, PRO = properties, RVA = required values, OPE = operands, REF = reference rules.*

both English and Korean (Carreras & Márquez, 2005; Palmer, Ryu, Choi, Yoon, & Jeon, 2006). The performance of the deep learning model is affected by the quantity and quality of training data. While we developed the deep learning-based PAS extraction model, we also started to collect and process the training data because there is no existing natural language dataset for the architectural domain in Korea. Collecting the proper training data is essential, and could be the bottleneck, for establishing a domain-specific training model.

We measured the validation metrics while collecting the training dataset and verified the change in model performance based on the quantity of training data. As illustrated in the charts in Fig. 7, the performance of the model was enhanced with an increase in training sentences. When we used 50 sentences for training, the accuracy and F1 score were 0.6207 and 0.19, which increased to 0.8013 and 0.49 with 350 training sentences. The largest increase in learning performance was between datasets 1 and 2. There was no significant enhancement in performance beyond 100 sentences. The measured value was relatively lower than for general-purpose models, but we can expect the enhancement of training models by increasing the number of training datasets. Furthermore, the deep learning model and input feature used in this paper were implemented using a basic concept of deep learning model and word embeddings. Other more developed models use additional functions such as attention layers, highway connections in a Bi-LSTM model, or deep learning-based contextual word-embedding models such as BERT or ELMo for enhancing the performance of SRL tasks. The proposed model can be improved by properly adopting additional techniques in deep learning and NLP.

Furthermore, we compared the training results from variable parameter settings, as shown in Table 7. Dimension of word embedding was changed from 50 to 200, and we also compared the results of effects of predicate embedding features. The results show that performance with 100 and 200 embedding dimension is slightly enhanced than 50 dimension embedding vector. There was no significant difference in performance between 100 and 200 dimensions. Predicate embedding vector shows performance enhancement in all embedding 50,100, and 200.

5.3. Demonstration with GUI application prototype

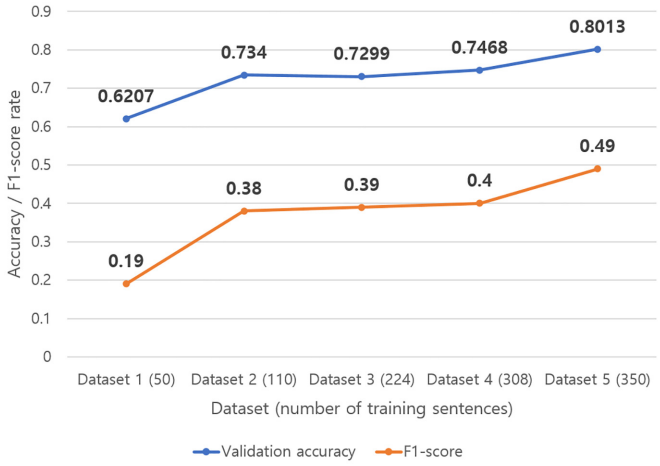
The detailed approach for the proposed method is illustrated in Fig. 8, based on the overview concept (Fig. 1). The information extraction model extracts the required information following the proposed process. The PAS proposed in this paper is part of information extraction, which focuses on word-level information and the semantic relationship between the predicate and arguments. Generating executable code in rule-checking software requires a higher level of semantic analysis in sentence units or even article units to clarify the logical relationship between each rule. The semantic information including the additional semantic information could be represented in an explicit data format such as XML. We can use the data format to generate the computer-executable code of rule-checking software. As a part of development of the entire process, PAS extraction model was implemented with a graphical user interface (GUI) application prototype as a sub-module for interpreting and translating building design rule sentences into computer-readable form.

As illustrated in Fig. 9, the user can input their own design rule sentences and extract the PAS information. A GUI interface provides the textual and graphical visualization of the PAS for intuitive understanding. The example in Fig. 9 depicts PAS used to describe the design requirements for effective width of stairs with a structure of PAS Type 3 in Table 2. Visualization with color labeling depicts the entire phrase of modified objects, and the graphical visualization in the upper panel depicts each PAS for modifying phrases. The interface for extracting object and property information is included as an additional tab, and the user can view the corresponding results.

Table 8 is one of the test results with GUI application. The input sentence in Table 8 is from Rule of Evacuation and Fireproof structure criteria article 9 (2) 2, which regulates the effective width of evacuation stairs installed at the outside of buildings. The intent of the regulation is to check the geometric shape of the target object (stair) to ensure the sentence is categorized as PAS Type 3. The target object, properties for checking, and required values are “stair,” “effective width,” and “0.9 meters,” respectively. This example is a general structure of building design rule sentences; the trained model demonstrates it can recognize the semantic role of each word correctly.

6. Conclusion

This paper proposed an application of deep learning and NLP-based information extraction techniques for translating building design rule sentences into a computer-executable format. With the deep learning-based method, the computer can learn how to interpret the semantic meaning of natural languages and translate the information into an explicit data format. Automated design rule interpretation can alleviate wasted labor and time for manual interpretation. We developed a PAS for



	Validation Accuracy	Precision (average)	Recall (average)	F1-Score (average)
Dataset 1 (50 sentences)	**0.6207**	0.21	0.24	**0.19**
Dataset 2 (110 sentences)	**0.7340**	0.51	0.45	**0.38**
Dataset 3 (224 sentences)	**0.7299**	0.40	0.45	**0.39**
Dataset 4 (308 sentences)	**0.7468**	0.42	0.51	**0.40**
Dataset 5 (350 sentences)	**0.8013**	0.49	0.65	**0.49**

Figure 7: Validation accuracy and F1 score measurements by dataset.

Table 7: The effect of parameter settings of word-embedding dimensions and additional parameter.

Word embedding	Additional parameter	Total dimension	Precision	Recall	F1
50	None	50	21	29	21
	Predicate embedding	100	29	33	28
100	None	100	34	40	30
	Predicate embedding	150	32	43	33
200	None	200	27	44	29
	Predicate embedding	250	33	39	32

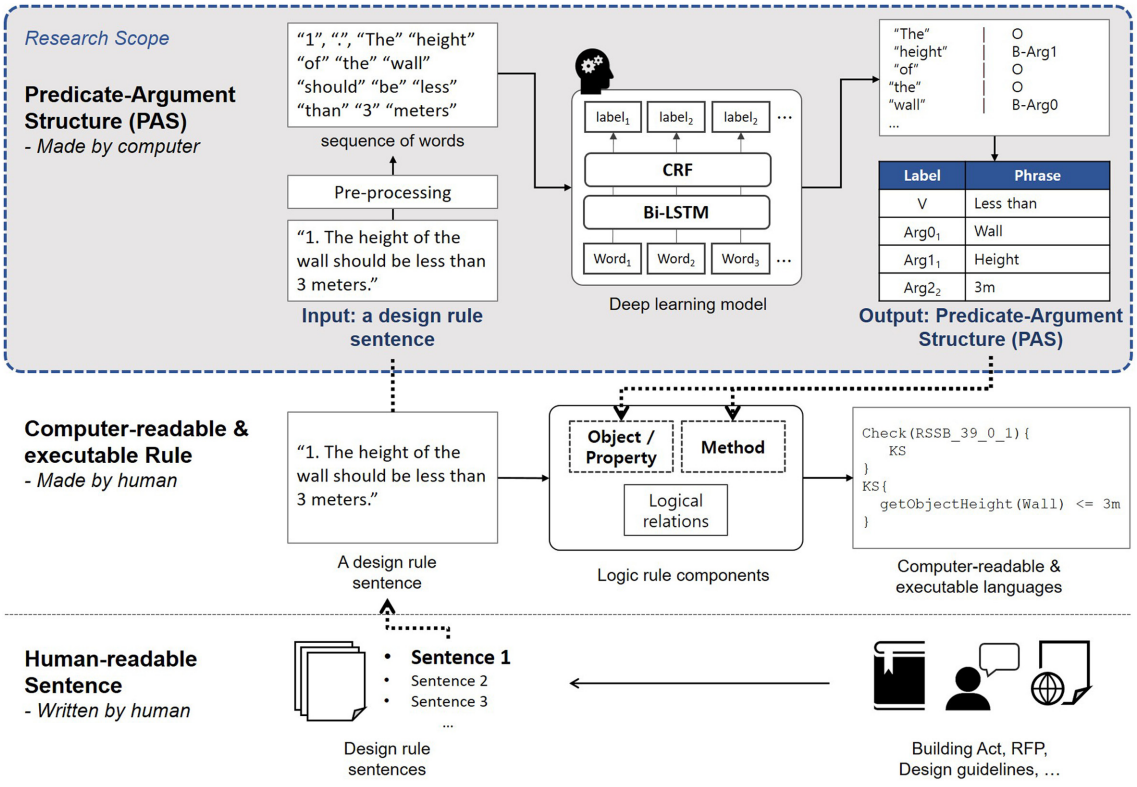


Figure 8: Deep learning and NLP-based PAS extraction in the rule-making process (extended from Fig. 1).

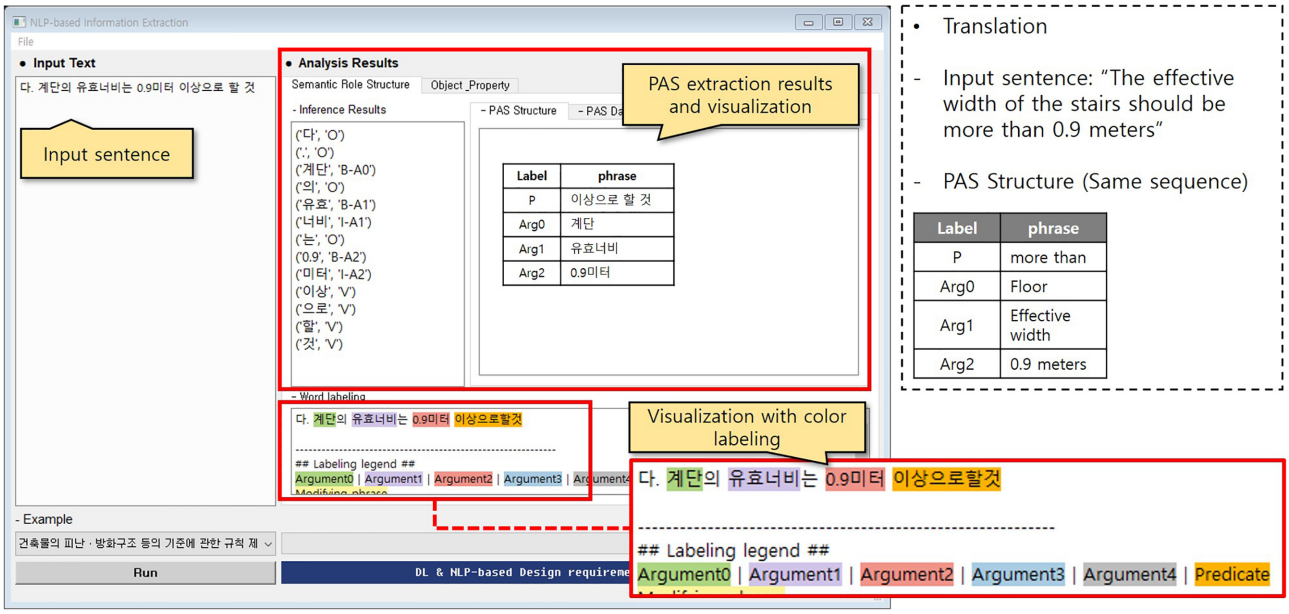


Figure 9: A snapshot of the GUI application prototype.

Table 8: Labeling results of the test sentence.

Input/output	Sentences	Translation[a]
Input sentence	"다. 계단의 유효너비는 0.9미터 이상으로 할 것"	"C. The effective width of stair should be more than 0.9 meter"
Labeling results ('Words', 'PAS label')	('다', 'O') ('.', 'O') ('계단', 'B-Arg0') ('의', 'O') ('유효', 'B-Arg1') ('너비', 'I-Arg1') ('는', 'O') ('0.9', 'B-Arg2') ('미터', 'I-Arg2') ('이상', 'V') ('으로', 'V') ('할', 'V') ('것', 'V')	('C', 'O') ('.', 'O') ('The', 'O') ('effective', 'B-Arg1') ('width', 'I-Arg1') ('of', 'O') ('stair', 'B-Arg0') ('should', 'O') ('be', 'O') ('more', V) ('than', 'V') ('0.9', 'B-Arg2') ('meter', 'I-Arg2)
Visualization with labeling	다. 계단의 유효너비는 0.9미터 이상으로할것 --- ## Labeling legend ## Argument0 | Argument1 | Argument2 | Argument3 | Predicate	

[a]*The translation sentence and labeling results for English sentences are manually written by the author based on the meaning of the original Korean sentence.*

building design rule sentences and classified the structure by the checking property type and required argument types. The deep learning model was trained to extract the PAS from the building design rule sentences, and the trained models can be used in the rule interpretation process. The extracted PAS was in the form of computer-readable data that implied shallow semantics of design rule sentences. The extracted data support to generate computer-executable design rules reflecting the meanings of building design rule sentences.

As an early phase of research, this paper focused on a method of capturing the semantic relationship between each word in a design rule sentence. The extracted PAS data can be translated for specific script languages with data parsing tools and mapping algorithms. To develop a fully automated rule-making process, there are challenges beyond PAS extraction and several limitations to be addressed in future research. The predicate and arguments in PAS are classified by their semantic roles but written using natural language words. Mapping these intermediate data to a BIM data format is another process that must be developed. Capturing the logical relationship between PAS units or sentences and structuring the logical order of checking are other remaining steps for translation.

The application of deep learning models can enhance the efficiency of labor-intensive work, but applying these methods requires consideration for real-world use. Deep learning models require a significant quantity of training data to guarantee the appropriate level of performance. Datasets for general AI models have been established for several years and used to develop the training model. However, datasets for domain-specific problems have not typically been established, which requires additional data collection and labeling to develop the training model. Domain adaptation techniques from the pre-trained model—such as semi-supervised learning and transfer learning—could be considered to develop the performance of the deep learning model.

Concerning design rule checking, especially for code compliance checking, the accuracy of rule translation is critical for the checking result to avoid errors in rule translation. A deep learning model, even for humans, cannot guarantee 100% accuracy of translation; consequently, an auxiliary process should be considered to integrate the rule-making process. To address these problems, user interfaces to review the automated information extraction and translation will be implemented as part of the rule-making interfaces.

Acknowledgement

This research was supported by a grant (20AUDP-B127891-04) from the Architecture & Urban Development Research Program funded by the Ministry of Land, Infrastructure and Transport of the Korean government.

Conflict of interest statement

Declarations of interest: none.

References

Bloch, T., & Sacks, R. (2018). Comparing machine learning and rule-based inferencing for semantic enrichment of BIM models. *Automation in Construction*, *91*(July 2017), 256–272. doi:10.1016/j.autcon.2018.03.018.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5, 135–146. doi:1511.09249v1.

Carreras, X., & Márquez, L. (2005). Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *CoNLL 2005 – Proceedings of the Ninth Conference on Computational Natural Language Learning* (pp. 152–164), Ann Arbor, Michigan.

Chiu, J. P. C., & Nichols, E. (2016). Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4, 357–370. doi:10.1162/tacl_a_00104.

Choi, J., Choi, J., & Kim, I. (2014). Development of BIM-based evacuation regulation checking system for high-rise and complex buildings. *Automation in Construction*, *46*, 38–49. doi:10.1016/j.autcon.2013.12.005.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*, 2493–2537.

Cowie, J., & Wilks, Y. (1996). Information extraction. *Communications of the ACM*, *39*(1), 80–37.

Eastman, C., Lee, J. Min, Jeong, Y. Suk, & Lee, J. Kook. (2009). Automatic rule-based checking of building designs, *Automation in Construction*, *18*(8), 1011–1033. doi:10.1016/j.autcon.2009.07.002.

Gaizauskas, R., Demetriou, G., Artymiuk, P. J., & Willett, P. (2003). Protein structures and information extraction from biological texts: The PASTA system. *Bioinformatics*, *19*(1), 135–143. doi:10.1093/bioinformatics/19.1.135.

Ghannad, P., Lee, Y.-C., Dimyadi, J., & Solihin, W. (2019). Automated BIM data validation integrating open-standard schema with visual programming language. *Advanced Engineering Informatics*, *40*, 14–28. doi:10.1016/J.AEI.2019.01.006.

Gridach, M. (2017). Character-level neural network for biomedical named entity recognition. *Journal of Biomedical Informatics*, *70*, 85–91. doi:10.1016/j.jbi.2017.05.002.

Hochreiter, S., & Schmidhuber, J. (1997). *Long Short-Term Memory*. 9, 1735–1780.

Huang, Z., Xu, W., & Yu, K. (2015). *Bidirectional LSTM-CRF Models for Sequence Tagging*. http://arxiv.org/abs/1508.01991.

Ismail, A. S., Ali, K. N., & Iahad, N. A. (2017). A review on BIM-based automated code compliance checking system. *International Conference on Research and Innovation in Information Systems, ICRIIS* (pp. 1–6). doi:10.1109/ICRIIS.2017.8002486.

Kim, H., Lee, J.-K., Shin, J., & Choi, J. (2019). Visual language approach to representing KBimCode-based Korea building code sentences for automated rule checking. *Journal of Computational Design and Engineering*, *6*(2), 143–148. doi:10.1016/J.JCDE.2018.08.002.

Kingsbury, P., & Palmer, M. (2003). PropBank: The next level of TreeBank. *Proceedings of Treebanks and Lexical Theories*, Växjö, Sweden.

Koo, B., La, S., Cho, N.-W., & Yu, Y. (2019). Using support vector machines to classify building elements for checking the semantic integrity of building information models. *Automation in Construction*, *98*(June 2018), 183–194. doi:10.1016/j.autcon.2018.11.015.

Koo, B., & Shin, B. (2018). Applying novelty detection to identify model element to IFC class misclassifications on architectural and infrastructure building information models. *Journal of Computational Design and Engineering*, *5*(4), 391–400. doi:10.1016/J.JCDE.2018.03.002.

Lafferty, J., McCallum, A., & Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning* (pp. 282–289).

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural Architectures for Named Entity Recognition. In *Proceedings of the 2016 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 260–270). doi:10.18653/v1/N16-1030.

Lee, H., Lee, J.-K., Park, S., & Kim, I. (2016). Translating building legislation into a computer-executable format for evaluating building permit requirements. *Automation in Construction*, *71*, 49–61. doi:10.1016/j.autcon.2016.04.008.

Macit İlal, S., & Günaydın, H. M. (2017). Computer representation of building codes for automated compliance checking. *Automation in Construction*, *82*, 43–58. doi:10.1016/J.AUTCON.2017.06.018.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*.

Mitchell, W. J. (1975). Techniques of automated design in architecture: A survey and evaluation. *Computers & Urban Society*, *1*(1), 49–76. doi:10.1016/0305-7097(75)90005-8.

Nawari, N. (2012). The challenge of computerizing building codes in a BIM environment. *Journal of Computing in Civil Engineering*, 1, 285–292. doi:10.1061/9780784412343.0036.

Palmer, M., Ryu, S., Choi, J., Yoon, S., & Jeon, Y. (2006). *Korean Propbank*.

Pauwels, P., & Zhang, S. (2015). Semantic rule-checking for regulation compliance checking: An overview of strategies and approaches. *Proceedings of the 32nd CIB W78 Conference 2015, 27th–29th October 2015* (pp. 619–628), Eindhoven, The Netherlands. doi:10.1186/jbiol113.

Pollock, J., Waller, E., & Politt, R. (2010). Speech and language processing. In *Day-to-Day Dyslexia in the Classroom*. New York: Routledge. doi:10.4324/9780203461891_chapter_3.

Preidel, C., & Borrmann, A. (2016). Towards code compliance checking on the basis of a visual programming language. *Journal of Information Technology in Construction*, *21*, 402–421. https://doi.org/http://www.itcon.org/2016/25. ISSN 1874–4753.

Rafael, S., Ling, M., Raz, Y., Andre, B., Simon, D., & Uri, K. (2017). Semantic enrichment for building information modeling: Procedure for compiling inference rules and operators for complex geometry. *Journal of Computing in Civil Engineering*, *31*(6), 4017062. doi:10.1061/(ASCE)CP.1943-5487.0000705.

Ruichuan, Z., & El-Gohary, N. M. (2019). A machine learning approach for compliance checking specific semantic role labeling of building code sentences. In *Advances in informatics and computing in civil and construction engineering* (pp. 561–568). Berlin, Germany: Springer.

Sacks, R., Eastman, C., Lee, G., & Teicholz, P. (2018). *BIM handbook: A guide to building information modeling for owners, designers, engineers, contractors, and facility managers*, Hoboken, NJ, USA. John Wiley & Sons.

Shin, J., & Lee, J.-K. (2019). Indoor Walkability Index: BIM-enabled approach to quantifying building circulation. *Automation in Construction*, *106*, 102845. doi:10.1016/J.AUTCON.2019.102845.

Solihin, W., Dimyadi, J., & Lee, Y.-C. (2019). In search of open and practical language-driven BIM-based automated rule checking systems. In I., Mutis, & T. Hartmann (Eds.), *Advances in informatics and computing in civil and construction engineering* (pp. 577–584), Switzerland: Springer International Publishing.

Solihin, W., & Eastman, C. (2015). Classification of rules for automated BIM rule checking development. *Automation in Construction*, *53*, 69–82. doi:10.1016/j.autcon.2015.03.003.

Song, J., Kim, J., & Lee, J.-K. (2018). NLP and deep learning-based analysis of building regulations to support automated rule checking system. *Proceedings of the 35th ISARC* (pp. 586–592), Berlin, Germany. doi:10.22260/ISARC2018/0080.

Surdeanu, M., Harabagiu, S., Williams, J., & Aarseth, P. (2003). Using predicate-argument structures for information extraction. *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics* (pp. 8–15), Sapporo, Japan. doi:10.3115/1075096.1075098.

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent Trends in deep learning based natural language processing [review article]. *IEEE Computational Intelligence Magazine*, *13*(3), 55–75. doi:10.1109/MCI.2018.2840738.

Zhang, J., & El-Gohary, N. M. (2017). Integrating semantic NLP and logic reasoning into a unified system for fully-automated code checking. *Automation in Construction*, *73*, 45–57. doi:10.1016/j.autcon.2016.08.027.

Zhang, S., Teizer, J., Lee, J. K., Eastman, C. M., & Venugopal, M. (2013). Building information modeling (BIM) and safety: Automatic safety checking of construction models and schedules. *Automation in Construction*, *29*, 183–195. doi:10.1016/j.autcon.2012.05.006.