Deep learning-based relation extraction and knowledge graph-based representation of construction safety requirements

Xiyu Wang[1] and Nora El-Gohary[2]

[1] Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, 205 N. Mathews Ave., Urbana, IL 61801, United States. Email: xiyuw2@illinois.edu
[2] Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, 205 N. Mathews Ave., Urbana, IL 61801, United States. Email: gohary@illinois.edu

Abstract

Field compliance checking aims to check the compliance of site operations with applicable construction safety regulations for detecting violations. Relation extraction provides an automated solution to extract relations that describe construction safety requirements from unstructured text. However, previous relation extraction efforts are limited in their extraction capabilities, representation, and automation. To address this gap, this paper proposes a deep learning-based method to automatically extract and represent relations that describe fall protection requirements. The proposed method: (1) uses a CNN-based model, with pre-trained word and position embeddings, to automatically extract domain-specific relations, and (2) represents the extracted requirements in the form of knowledge graph-based queries, which helps decompose complex requirements into manageable units while keeping these units connected in a scalable graph structure. The proposed method was tested on 20 OSHA sections, and has achieved 87.5% precision, 83.4% recall, and 85.4% F-1 measure, which indicates good relation extraction performance.

Keywords: Relation extraction; Construction safety; Fall protection; Field compliance checking; Deep learning; Knowledge graphs; Word embeddings.

1 Introduction

Field compliance checking aims to detect violations to construction safety regulations to protect workers from potential safety incidents. This is because a large portion of construction site accidents occur as a result of field noncompliances, which usually include damaged or no personal protective equipment (PPE); inoperative or inappropriate equipment; and wrong poses, operations, or work sequences (Chi and Lin 2018). For fall fatalities in particular, according to an analysis of the Construction FACE Database (CFD), noncompliance of personal fall arrest systems has caused more than 77% of all deaths (Dong et al. 2017). To identify such field noncompliances, different solutions have been proposed for mitigating the risks and consequences of potential safety incidents, among which utilizing computer vision techniques for safety checking has attracted an increasing amount of research attention. These research efforts include using computer vision techniques to detect personal protective equipment (PPE) (Nath et al. 2020; Fang et al. 2020a), recognize workers' operations (Roberts et al. 2020; Tang et al. 2019), and track the trajectory of labor and equipment onsite (Tang et al. 2020).

However, compared with the rich body of literature in applying computer vision techniques, limited studies have explored the use of natural language processing (NLP) techniques to analyze construction safety documents. For example, Feng and Chen (2021) proposed a deep learning-based framework to extract extract event-related information (e.g., date, location, and accident type) from accident news reports for construction safety management. Rupasinghe and Panuwatwanich (2021) proposed a rule-based method to mine hazard information from accident reports. Baker et al. (2020) proposed approaches to extract injury precursors using NLP techniques (a set of text patterns). Zhong et al. (2020c) presented a deep learning-based framework to mine hazard knowledge from hazard records. Chi et al. (2017) proposed a semi-automated approach to develop a gazetteer that can eventually support information extraction from construction safety regulations. Collectively, these efforts either focused on the analysis of injury and accident reports or focused on the extraction of hazard factors. There is, thus, a lack of research efforts that focused on automatically extracting requirements from construction safety regulations for supporting field compliance.

Information extraction offers an opportunity to automatically extract safety requirements. In recent years, there is a growing body of literature in the construction domain that aims to propose different methods for extracting information from various construction regulatory documents, including but not limited to energy conservation codes, quality standards, and general building codes (Zhong et al. 2022; Moon et al. 2022; Zhong et al. 2020d; Schönfelder and König 2021; Zhang and El-Gohary 2021a; Ren and Zhang 2021; Moon et al. 2021; Guo et al. 2021; Zhong et al. 2020a; Song et al. 2018; Zhou and El-Gohary 2017; Zhang and El-Gohary 2013). These methods mainly vary in two aspects: (1) the approach for extraction, e.g., rule-based (Zhang and El-Gohary 2013; Lee at al. 2019b; Ren and Zhang 2021), machine learning-based (ul Hassan et al. 2020; Kim and Chi 2019), or deep learning-based (Zhang and El-Gohary 2021b; Schönfelder and König 2021; Zhong et al. 2020d). These efforts have laid a solid foundation for extracting information from construction safety regulations, with deep learning approaches offering the highest potential for limiting human involvement in developing the information extraction methods (e.g., eliminating the need for handcrafted rules); and (2) the targeted semantic information elements to be extracted, e.g., building-code requirements (Zhang and El-Gohary 2021a), utility spatial configurations from utility accommodation policies (Xu and Cai 2019), or construction procedural constraints and attributes for quality compliance (Zhong et al. 2022).

However, most of the aforementioned efforts mainly either focused on extracting named entities or considered relations as a type of information element (Wu et al. 2022; Zhong et al. 2020d; Ren and Zhang 2021; Schönfelder and König 2021), which makes them limited in capturing the different types of relation classes and in expressing the rich semantics in the original documents. More research studies are therefore needed to bridge five main knowledge gaps in relation extraction for construction domain applications. First, existing relation classes in other relation extraction efforts (e.g., "Place_Of_Birth" in the linguistics domain) cannot be directly transferred, because they are not suitable to describe situations in the construction safety domain. Thus, relations that describe construction safety requirements need to be identified and extracted. Second, existing efforts (e.g., Wu et al. 2022) in the construction domain are limited in considering non-verbal-predicate relations, redundant relations, and relation directions. Third, most of the existing efforts in the construction domain fall short in their scalability and generalizability, because they require a heavy amount of human assistance. Fourth, limited efforts in the construction domain have explored generating query graphs for knowledge graph-based reasoning directly from text, which would help identify new or missing information not explicitly expressed in the original text for subsequent analysis. Fifth, limited attention has been paid to developing queries that can support deep learning-based automated reasoning, especially hyper-relational queries for representing nested relations.

To address these gaps, this paper proposes a new method to automatically extract and represent safety requirements from construction safety regulations. The proposed method: (1) uses deep learning to automatically extract domain-specific relations (e.g., Located_At and Engage_In) about fall protection requirements from the regulations to add interlinks to the isolated named entities extracted in Wang and El-Gohary (2022); (2) uses word- and position-embedding features to improve the relation extraction performance; and (3) represents the extracted safety requirements (both relations and named entities) in the form of query graphs to facilitate future discovery of implicit or missing information, as well as knowledge graph-based compliance reasoning. The proposed method was tested using fall-related sections from Occupational Safety and Health Administration (OSHA) 29 CFR 1926 (OSHA 2020).

2 Background

2.1 Current Practices for Automated Field Compliance Checking

Automated field compliance checking aims to automatically check whether workers' behaviors and their surrounding environment are adhering to applicable safety regulations, norms, procedures, and guidelines (Khalid et al. 2021). In addition to the research efforts that utilize computer vision techniques (described in Section 1), existing research efforts and construction site practices have explored using various emerging technologies, such as BIM, wearable devices, and smart systems (e.g., Awolusi et al. 2018; Jebelli et al. 2018; Cheung et al. 2018; Zhang et al. 2017; Zou et al. 2017; Park et al. 2013). However, the majority of these systems or applications are limited in supporting automated field compliance checking because they (1) mainly focused on checking violations in the design to identify potential hazards or risks (e.g., Kincelova 2020); (2) developed ontologies to represent a set of safety checking rules, but often lacked in capturing information about real-time site operations to detect field noncompliance (e.g., Li et al. 2022); and/or (3) were designed as management tools [e.g., HCSS Safety field app (HCSS 2022)] for organizing incident reports, documenting observations from coworkers, and/or collecting workers' physical states, rather than capturing and comparing site information with safety requirements to detect noncompliance.

On the other hand, the majority of BIM software systems available on the market that aim to conduct automated compliance checking are limited in supporting construction field safety checking scenarios. For example, safety checking using Solibri (Solibri 2021) relies heavily on (1) BIM models that are typically covering design information and are lacking real-time field information on construction operations (equipment, labor, etc.); and (2) hard-coded safety requirements embedded in the software, which require manual effort to read the safety regulations and encode the requirements in computable rule formats. Other commercial software systems that can be used in the construction phase are limited in compliance checking scope and generalizability to address various scenarios. For example, smartvid.io (ECT Team 2021) mainly considers checking the existence of PPE such as gloves, reflective vests, and footwear (Nath et al. 2020), but it cannot check compliance with applicable safety requirements, which express different conditions and exceptions for various fall-related scenarios. Therefore, more research efforts are needed to develop methods for identifying noncompliances automatically for diverse accident scenarios in the field. Extracting requirements from construction safety regulations and representing them in a structured and computer-processable format is the first step towards such an automated field checking process.

2.2 Relation Extraction

Relation extraction is the task of recognizing and classifying semantic relations from unstructured text into several predefined classes (Nguyen and Grishman 2015). For example, in the sentence “Defective safety net components shall be removed from service”, relation extraction would recognize and classify the relation between “safety_net_component” and “defective” as “Is”, and the relation between “safety_net_component” and “service” as “Keep_From”. Early relation extraction efforts have proposed various rule-based and traditional machine learning-based (as opposed to deep learning-based) methods (Wang et al. 2012; Zhang et al. 2009; Culotta and Sorensen 2004), which have achieved good performance but have typically required much human effort to develop the extraction rules or conduct feature engineering.

In recent years, outside the construction domain, deep learning-based methods have been used for relation extraction and many novel neural network models have been proposed (Jiang et al. 2020). Among them, convolutional neural networks (CNN)-based models and recurrent neural networks (RNN)-based models have received high popularity and reached good and comparable performance levels (Miwa and Bansal 2016). For example, Hendrickx et al. (2019) achieved an F-1 measure of 84.1% in relation extraction from the SemEval-2010 dataset using a CNN-based model and 84.0% using an RNN-based model. Similar performance was shown for variants of the two model types as well. For example, Shen and Huang (2016) achieved an F-1 measure of 85.9% on the same dataset using an Attention-based CNN model. The entity-aware Attention bidirectional long short term memory (BiLSTM) (Lee et al. 2019a), an RNN-based model, achieved an F-1 measure of 85.2% using the same dataset. Another branch of research efforts has also attempted to improve performance by adding sentence hierarchies such as dependency paths as additional features (Yu et al. 2020; Cai et al. 2016).

Depending on the types of supervision received, those deep learning-based methods can be further divided into two categories: distant and fully supervised methods. Distant supervised methods learn from unlabeled data with the help of some external knowledge bases. For example, Mintz et al. (2009) used the Freebase (Bollacker et al. 2008), a semantic knowledge base, for distant supervised learning. In general, research on distant supervised methods attempts to experiment with different deep learning architectures or different knowledge bases for performance improvement. Example efforts that adopt either CNN-based models or RNN-based models but receive different levels of supervision include heterogeneous representations for neural relation extraction (HRERE) (Xu and Barbosa 2019), language understanding with knowledge-based embeddings (LUKE) (Yamada et al. 2020), and advanced prototypical networks

(Proto-ADV) (Gao et al. 2019). Fully supervised relation extraction methods, on the other hand, are more suitable for construction applications, because (1) they do not require external knowledge bases, which are currently unavailable in the construction domain; and (2) customized relation classes can be easily incorporated through additional classes/labels.

2.3 Knowledge Graphs

A knowledge graph is a multi-relational graphical network that uses different relations as directed edges to connect concepts or entities for representing information (e.g., information extracted from text or databases) in a semantically rich and structured way (Bellomarini et al. 2019). Such graphical network structure not only helps express relational connectivity in an intuitive way, but also helps discover implicit, missing, or new information through edge traversal, because some relations may not be explicitly expressed or some entities may be omitted in the original data (e.g., natural language sentences) (Chen at al. 2020c; Ji et al. 2021). Knowledge graphs thus show three significant advantages, which are especially beneficial for field compliance checking applications. First, knowledge graphs can store relations between entities explicitly due to its graph-like structure, unlike other representation approaches such as traditional relational databases in which entities are stored in the form of tables and are linked by separate linking tables that cannot represent the exact semantic relations. Second, knowledge graphs typically allow more flexibility to easily add or remove classes and relations from the knowledge-graph schema. Since construction safety regulations are becoming more stringent as safety knowledge improves (Fang et al. 2020a), using knowledge graphs requires less manual work to keep up with updates. Third, knowledge graphs allow for faster information retrieval due to their structure, compared with other representation approaches (Holzschuher and Peinl 2013). For example, Chen et al. (2020a) showed that knowledge graphs have outperformed traditional relational databases in querying and retrieving transportation data (Chen et al. 2020a). This is because information retrieval using knowledge graphs usually starts from the related named entities, and only scans relations in their neighborhood for desired information.

Due to their aforementioned characteristics, and hence the promising performance in various applications such as knowledge retrieval, question-answering, knowledge recommendation, and knowledge visualization, knowledge graphs have been successfully deployed by many leading companies to organize their business data such as Google, Amazon, eBay, IBM, and LinkedIn (Chen at al. 2020c; Chen et al. 2020b). Multiple open knowledge graphs were published as well, such as DBpedia, YAGO, and Google's Knowledge Graph (Ji et al. 2021). Recently, knowledge graphs have been applied in more research fields. For example, some research efforts have attempted to develop knowledge graphs to model hazardous chemical knowledge for risk management (Zheng et al. 2021). In the construction domain, a few efforts focused on developing knowledge graphs for a number of applications. For example, Chen and Luo (2019) constructed an ontology-based knowledge graph using noun phrases extracted from different abstracts in the construction literature for bibliometric analysis. Fang et al. (2020a) developed a small-scale knowledge graph for modeling detected site objects with spatial relations. Jiang et al. (2021) constructed a small-scale knowledge graph for representing the connections among different construction safety standards (e.g., "Specification of Inspection of Construction Hoist Equipment" "instance_of" "Machinery Management").

2.4 Query Representation for Knowledge Graph-Based Reasoning

Reasoning over knowledge graphs aims to infer new information or identify the target information from large amounts of available facts represented in a knowledge graph (Chen et al. 2020b). Traditional reasoning methods depend heavily on external databases and query languages, which can be time-consuming and subject to the quality and coverage of existing knowledge graphs. On the contrary, neural reasoners are faster and can better adapt to the incompleteness in existing knowledge graphs, which makes them potentially efficient for field compliance checking. A research area that has recently attracted research attention is query representation for neural reasoning (Alivanistos et al. 2021; Yu and Yang 2021), especially to represent arbitrary logic operators such as conjunction ($\wedge$), disjunction ($\vee$), and negation ($\neg$) together with other triples in complex query graphs. This is particularly important for safety compliance reasoning, because construction safety requirements cover different situations and are typically represented using multiple logic operators. For example, the query graph for the sentence "the attachment point of the body harness shall be located in the center of the wearer's back near shoulder level, or above the wearer's head" would contain two locations for the attachment point connected by a disjunction operator.

There exist two ways to represent query graphs that involve logic operators: (1) as directed acyclic graphs with symbolic logic operators. Knowledge graph-based reasoning methods then seek to retrieve subgraphs that match with the query graphs. However, subgraph matching is relatively sensitive to data quality, producing correct answers largely when facts in the knowledge graph are complete and accurate, which is not the case in real-world knowledge graphs (Ren et al. 2020; Chen et al. 2020b; Zhu et al. 2022); and (2) as dependency or computation graphs to be mapped to an embedding space together with facts from the knowledge graph (Ren et al. 2020; Ren and Leskovec 2020). Knowledge graph-based reasoning methods then seek to identify entities or relations which are nearest to the queries in the embedding space as answers to be returned. Such embedding methods can robustly handle missing relations and have achieved good performance in various reasoning tasks such as knowledge graph completion (Zhu et al. 2021).

3 State of the Art and Knowledge Gaps in Relation Extraction

In the area of relation extraction, in addition to the efforts outside the construction domain as discussed in Section 2, there is a growing number of research efforts undertaken to extract relations from construction documents. For example, Zhang and El-Gohary (2013) proposed a semantic rule-based natural language processing approach to automatically extract requirements, including quantitative relations and comparative relations, from building codes. Ren and Zhang (2021) proposed a semantic rule-based method with a set of natural language processing techniques to extract successive and parallel relations from construction procedural documents. Liu and El-Gohary (2021) proposed a semantic neural network ensemble–based dependency parsing method to automatically extract dependency relations between bridge-related entities. Zhong et al. (2020d) proposed a deep learning-based method to classify relations between entities about construction procedural requirements into seven predefined categories. Despite the contributions of these efforts, five gaps of knowledge exist.

First, existing relation classes in other relation extraction efforts are not sufficient/suitable to describe the complex situations in the construction safety domain, and thus cannot be directly applied. For example, only nine relation classes such as “Cause-Effect” and “Part-Whole” have been considered in the SemEval-2010 dataset, six relation classes such as “Agent-Artifact” and “Organization-Affiliation” have been considered in the ACE05 dataset (Walker et al. 2006), and 24 relation classes such as “Contain” and “Place_Of_Birth” have been considered in the New York Times dataset (Riedel et al. 2010). Also, as can be inferred from these examples, which are all from relation extraction efforts outside of the construction domain, many of the existing relation classes in these efforts can be either too general or irrelevant to describe construction safety requirements. Similarly, existing relation classes from other construction subdomains cannot be directly applied to construction safety applications either. This is because relations in domain-specific applications tend to be specific and can vary in semantics and detail from one subdomain/application to another. For example, construction safety regulations can include interactions between the workers and their environment. Thus, for instance, relations describing construction procedural constraints [e.g., the seven relations such as “Before”, “Start”, and “During” in (Zhong et al. 2020d)] are not sufficient to describe those interactions. In addition, most of the existing research efforts in the construction domain paid limited attention to the directions of relations. However, in natural language sentences, each relation often has two associated directions (e.g., "support" and "supported by" can both indicating a relation of "Support"), which need to be inferred from the context for accurate representation of the requirement semantics.

Second, most of the existing relation extraction methods, especially the research efforts in the construction domain, are limited in considering non-verbal-predicate or redundant relations. On one hand, existing efforts mainly focus on extracting simple predicates (i.e., verbs) as relations (e.g., Wu et al. 2022). However, relations sometimes exist not only in the form of predicates, in which case extracting merely predicates can lead to missing information. For example, "Same_As" is an important type of relation that describes the comparison between entities, but it cannot be extracted using methods that only consider predicates as relations. Missing such information in the extracted safety requirements could lead to missing the detection of noncompliance instances and eventually serious accidents onsite. On the other, current research efforts extract relations without considering that different expressions can be used to refer to the same relation (e.g., Wu et al. 2022). For example, relations such as "Conform_To" and "Meet" can both express that the compliance checking subject should be compliant to a specific requirement; however, they are considered as different relations in the extracted requirements using the existing extraction methods.

Third, most of the existing relation extraction methods in the construction domain still rely heavily on human assistance, thus can fall short in their scalability and generalization. On one hand, rule-based information/relation extraction methods (e.g., Zhang and El-Gohary 2015; Xu and Cai 2019; Ren and Zhang 2021; Wu et al. 2022) require hand-crafted rules. On the other hand, traditional machine learning-based (as opposed to deep learning-based) extraction methods (e.g., Liu and El-Gohary 2017; ul Hassan et al. 2020) require highly engineered features that are obtained through trial and error. Only a limited number of efforts have explored deep learning approaches (e.g., Zhong et al. 2020d; Zhang and El-Gohary 2021b; Schönfelder and König 2021). More research efforts are needed to further explore the use of deep learning (especially fully supervised deep learning, as described in Section 2.2) in relation extraction – for example to explore the use of feature embeddings to enhance the extraction capabilities and performance.

Fourth, there is a lack of information extraction efforts that allow a direct pipeline to generate queries for discovery of new information and improved analytics using knowledge graphs, which is especially needed for regulation analytics. Knowledge graphs use a graphical network to represent relations as interlinks to connect concepts or entities for maintaining the rich semantics in the original data (Bellomarini et al. 2019). Due to the ability to traverse through edges, reasoning over such graphical structure can help discover new relations or entities that are not explicitly expressed (e.g., spatial relations that were not identified from site images, or omissions of named entities in the text) (Chen at al. 2020c; Ji et al. 2021), which are important in identifying noncompliance instances. Thus, knowledge graphs and queries have been used in compliance-related applications outside of the construction domain. For example, Kaltenboeck (2022) developed queries based on different European laws for international-business applications. In the construction domain, Fang et al. (2020b) converted a checklist of unsafe behavior rules into Cypher queries to identify hazards in a knowledge graph, which stores detected site information, for improved hazard identification. However, most queries from previous knowledge graph-based reasoning efforts were developed manually, thus are small in scale [e.g., six rules in Fang et al. (2020b)]. There is a lack of efforts that use NLP techniques to directly create queries from construction-domain text for supporting knowledge graph-based reasoning.

Fifth, studies in query representations are needed for developing query graphs that can support deep learning-based field compliance reasoning. Queries in most of the previous efforts were developed in a traditional way using some particular query languages such as SPARQL and GQL. Reasoning using such queries relies heavily on external databases and query engines, requires longer processing time, and can be dependent on the quality of knowledge graphs, which can eventually impede the effectiveness of field compliance checking. There are a few research efforts on developing query graphs to perform deep learning-based reasoning directly in the embedding space. However, despite considering the conjunction and disjunction logic operators, these efforts paid less attention to hyper-relational queries or nested relations (Alivanistos et al. 2021; Yu and Yang 2021), which are important for accurately representing the extracted safety requirements.

4 Proposed Method for Relation Extraction

The proposed information extraction and information modeling method uses deep learning models to automatically extract domain-specific relations and represent the extracted safety requirements in a semantically rich and structured way. The proposed relation extraction method seeks to automatically identify semantic relations such as “Break” and “Tip_Over” from unstructured text, and classify them into several predefined relation classes such as “Fail” for normalizing different expressions that refer to the same relation, as illustrated in Fig. 1. A total of 56 relation classes were first identified based on a thorough review of relevant documents and research efforts. After predefining the relation classes and preprocessing the raw text, a deep learning-based model was developed to automatically recognize and classify relations based on their syntactic and semantic features. In developing the relation extraction model, two alternative deep learning models were tested: CNN-based (Attention-based CNN) and RNN-based model (Entity-aware Attention BiLSTM). These two types of models were selected for testing because they are two mature types of deep learning models that have achieved comparable performance in the computational linguistics domain (see Section 2.2) but that also have different focuses and merits (see Section 2.2), thus a comparison of the two can help provide insights in terms of which structures, layers, or techniques are more effective in addressing complex domain-specific text, which can ultimately lead to optimized model structures specifically for tasks in the construction safety domain. Additional deep learning models, such as transformer-based models, can also be tested in future work, as discussed in Section 6. Pre-trained features were used to leverage the rich semantics of these features, which were obtained using a large amount of annotated data from the computational linguistics domain. Two state-of-the-art static word embeddings were selected and incorporated into the proposed method for comparative evaluation: the continuous bag-of-words (CBOW) embedding (Mikolov et al. 2013) and the global vector (GloVe) embedding (Pennington et al. 2014). After relation extraction, all the extracted requirements [including relations extracted in this study and named entities extracted in (Wang and El-Gohary 2022)] were represented in the form of knowledge graph-based queries. Fig. 2 summarizes the research methodology, which includes six primary tasks: relation identification, text preprocessing, feature preparation, relation extraction, knowledge graph-based relation representation, and evaluation. An example to further illustrate the application of the proposed relation extraction method is shown in Fig. 3.

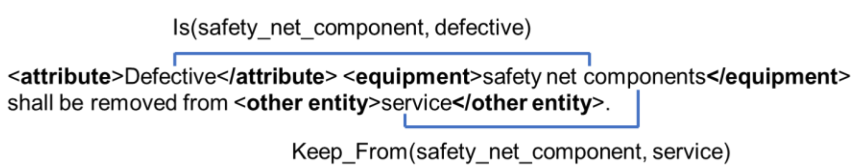


Fig. 1. Example relations extracted from an OSHA clause.

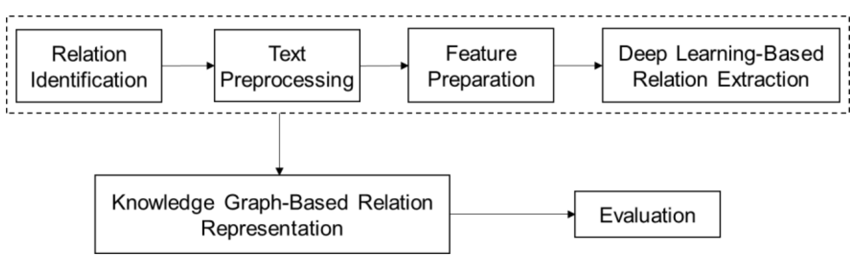


Fig. 2. Research methodology.

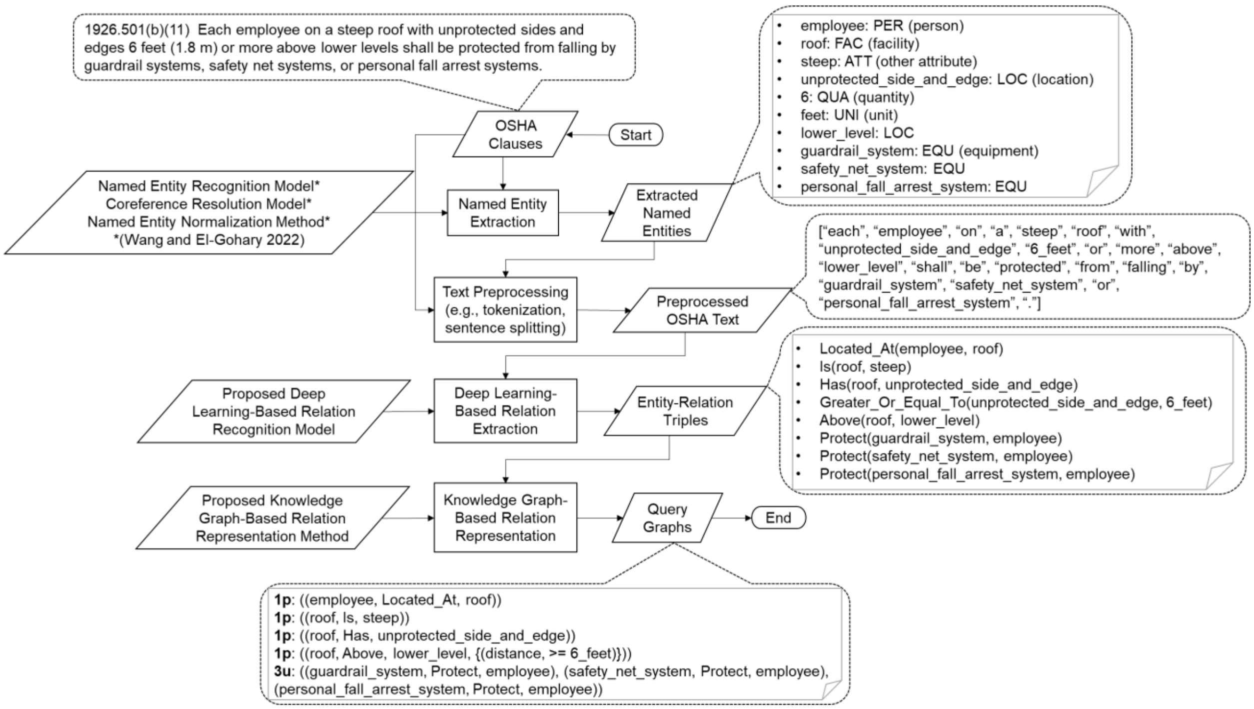


Fig. 3. Application of proposed relation extraction method, with example.

4.1 Relation Identification

A review of 20 OSHA sections related to fall protection and of previous efforts on ontology-based modeling of construction safety knowledge was conducted to identify the main semantic relations that are needed to represent fall protection requirements (Zhong et al. 2020b; Xing et al. 2019; Lu et al. 2015; Zhang et al. 2015). A total of 56 relation classes were identified, which aim to cover the main relations without redundant expressions. They were further grouped into five main types: (1) comparative or spatial relations, which describe comparisons or spatial locations, such as "Above" and "Below"; (2) interaction relations, which describe the interactions of the workers with their environment, such as "Face", "Access", and "Change"; (3) constraint relations, which describe conditions or situations, such as "Except" and "Conditioned_On"; (4) descriptive relations, which describe the the properties, characteristics, or components of the entities, such as "Is" (e.g., is steep) and "Has" (e.g., has unprotected side and edge); and (5) logic relations such as "And" and "Or". Most of the negated relations (e.g., "does not create a hazard") were treated as separate relations (e.g., "Not_Cause") to minimize the number of negation operations for enhancing the efficiency of the knowledge graph-based reasoning. Table 1 lists all the identified relation classes with examples and their corresponding relation types. Most of these relations are bidirectional (except relations such as "And", "Or", "Same_As", "Whichever_Greater", and "Whichever_Less"): either direction from head entity to tail entity or direction from tail entity to head entity. For example, in the sentence "*anticipated loads*(1) caused by *ice buildup*(2) …", the relation is that tail entity (2) causes the head entity (1). In the sentence "*ladder deflection*(1) cause the *ladder*(2) to …", the relation is that head entity (1) causes the tail entity (2).

Table 1. Identified Relation Classes with Examples

Relation	Example(s)	Relation type
Located_At	in, on the face of, at the edge of	Comparative or spatial relation
Part_Of	steps of portable ladders, wells for fixed ladders	
Less_Or_Equal_To	nor beyond, less than, nor more than, not to exceed	
Greater_Or_Equal_To	more than, at least, not less than, exceed	
Same_As	same as, i.e.	
Whichever_Greater	whichever is greater, whichever is later	
Whichever_Less	whichever is less	
Related_To	related, about	
Close_To	is closer to, near	
Above	above	
Below	below	
Over	over the edge of …	
Not_Over	would not go over, not overhang	
Into	into or through, falling through	
Behind	behind	
Between	within, at intervals	
From	from, between ... and ..., start at	
To	to, between ... and ..., to which	
After	after	
Before	until, before, prior to	
Cause	cause, so that, such that, because of	Interaction relation
Not_Cause	in no case … be such that, it will not create a greater hazard to …	
Conform_To	shall conform to, in conformance with, meet	
Provide	provide, to provide, shall be provided	
Support	supported, to support, shall be capable of supporting	
Not_Support	shall not be used to support, without supporting	
Decide	decide, shall determine	
Protect	shall be protected by, protect	
Keep_From	keep from, prohibit from, be removed from, be withdrawn from	
Allow	shall permit, to allow	
Use	by the use of, by, through	
Use_For	for, apply to, are used for, are designed for	
Not_Use_For	not apply to, used … not for	
Use_As	as, is used as	
Not_Use_As	shall not be used as	
Engage_In	engaged in, performing	
Not_Engage_In	who is not engaged in, not in	
Change	change, affect	
Match	be compatible with, match	
Fail	break, tip over, fail, fall	
Not_Reduce	shall not reduce	
Access	to reach, access	
Parallel	that parallels, shall be parallel, along	
Surround	around, encircle	
Face	shall face, face	
Conditioned_On	only when, if, provided that	Constraint relation
Except	unless, except, excluding	
Because	because, because of, for, as	
Otherwise	otherwise, or	
Has	shall have, to have, have, contain	Descriptive relation
Not_Has	without, shall not have	
Is	shall be, were, are	
Is_Not	shall not be	

But	but, however	Logic relation
Or	or	
And	and, in addition to, besides	

4.2 Text Preprocessing

Text preprocessing aims to prepare the raw text in a format that would be ready for subsequent analysis. Preprocessing consists of correcting misspelling, removing redundant punctuation, tokenization, and sentence splitting. Correcting misspelling and removing redundant punctuation aims to reduce the noise in the text. Tokenization aims to divide each sequence from the text into units of words. Sentence splitting aims to recognize the boundaries of sentences and divides them into chunks.

4.3 Feature Preparation

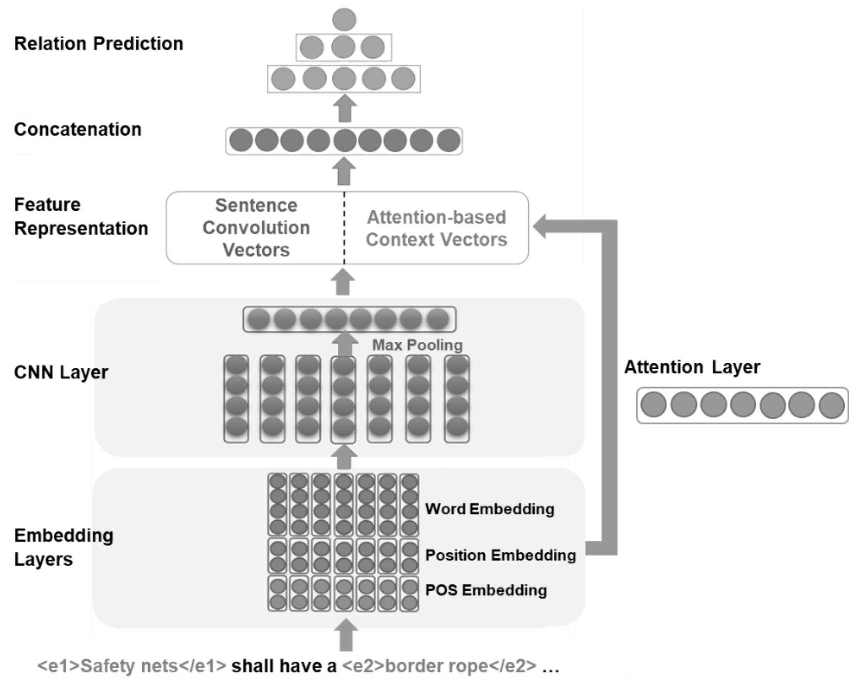
Two types of features were used for relation extraction: word embedding and position embedding. For word embedding, two state-of-the-art static word embeddings were selected for comparison: CBOW and GloVe embeddings. The CBOW embedding is pre-trained on 100 billion words from Google News. However, it does not encode explicit global information. The GloVe embedding, on the other hand, develops a global co-occurrence matrix to represent probabilities that a given word will co-occur with others. It is pre-trained on Wikipedia and Web text of 6 billion words. Both word embeddings can capture the semantics of each word, with its context, and represent them in the form of continuous and dense feature vectors, so that words similar in meaning are closer to each other in their embedding space. Compared to other static word embeddings (e.g., Skip Gram), CBOW and GloVe were selected for testing because they typically show better performance in relation extraction (İrsoy et al. 2020; Lai et al. 2018). Position embedding is used to differentiate the importance of each word due to its location in the sentence. This is because usually words closer to the given entities are more informative. Position information is thus calculated with reference to the head entity. For example, in the sentence "All <e1>fall protection</e1> required by <e2>1926.501</e2> shall …", the relative distance from the word "required" to the head entity is 1, and the relative distance from the tail entity "1926.501" to the head entity is 3, which are encoded in the position embedding.

4.4 Deep Learning-Based Relation Extraction

Two deep learning-based relation extraction models, a CNN-based model (Attention-based CNN) and an RNN-based model (Entity-aware Attention BiLSTM), were developed and tested for comparative evaluation. CNN and RNN were selected for the reasons outlined at the beginning of Section 4. A fully supervised learning approach was adopted for the reasons outlined in Section 2.2.

4.4.1 Proposed Attention-Based CNN Model

The proposed Attention-based CNN model contains four main types of layers: embedding layers, convolution layer, attention layer, and multi-layer perceptron layers. The embedding layers consist of three components: word embedding, position embedding, and part-of-speech (POS) embedding. The word embedding layer starts from the pre-trained embedding (CBOW or GloVe, as per Section 4.3), then adjusts itself to the semantics in the construction safety domain during training. The position embedding layer provides the relative location information of each word, as described previously. The POS embedding layer aims to encode the POS tag of each word, which indicates the lexical category of that word, such as noun, verb, and adjective. A total of 15 POS categories were considered and obtained using the Stanford CoreNLP Toolkit (Manning et al. 2014). With the lexical category of each word encoded, the model can capture more relation classes than predicates. The outputs for each word from these three embeddings are then concatenated before being fed into the CNN layer and the attention layer. The CNN layer, consisting of a convolution layer and a max-pooling layer, is used to extract local character-level features. The convolution process in the convolution layer aims to extract features by applying different filters. The max-pooling layer aims to keep the most important features for sentences with variable lengths. The outputs from the CNN layer are represented as sentence convolution vectors. For the attention layer, attention weights are calculated to quantitatively model the contextual relevance of the words. Then attention-based context vectors are calculated as a weighted sum of the words based on their attention weights. The outputs from both the CNN layer and the attention layer, namely sentence convolution vectors and attention-based context vectors, are concatenated together for a full representation of an input sentence. The multi-layer perceptron layers take in all the concatenated vectors and transform them into probabilities. Relation class tags with the highest probabilities are then selected as predictions. The Attention-based CNN architecture is illustrated in Fig. 4.

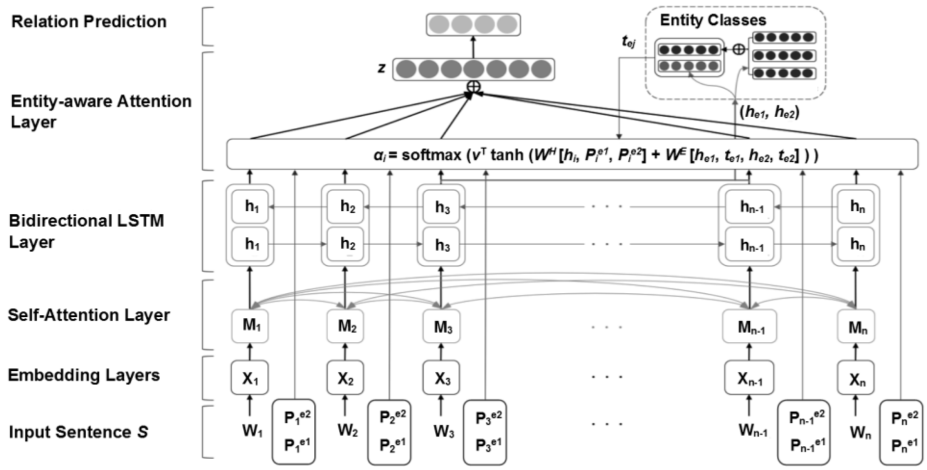


Note: CNN=convolutional neural network; POS=part-of-speech.

Fig. 4. Architecture of proposed Attention-based CNN model.

4.4.2 Proposed Entity-Aware Attention BiLSTM Model

The proposed Entity-aware Attention BiLSTM model contains five main types of layers: embedding layers, self-attention layer, bidirectional LSTM layer, entity-aware attention layer, and the multi-layer perceptron layers. The embedding layers consist of two components that correspond to the aforementioned two features. The outputs from the two embedding layers are concatenated before being fed into the self-attention layer, which is implemented using the multi-head attention formulation. The self-attention layer is used to capture the distinctive information in a sentence by measuring the correlation between words. Then the outputs from the self-attention layer are fed into the bidirectional LSTM layer for computing the feature values and capturing the context information of each word. The entity-aware attention layer is used afterwards to calculate the attention weights by considering three factors: (1) the semantic and syntactic features of each given entity pair, (2) the relative positions of the surrounding words to the target entity pair, and (3) the entity classes of the target entity pair. The multi-layer perceptron layers then transform the outputs from the entity-aware attention layer into relation class predictions, in the same way as the proposed Attention-based CNN model. To prevent overfitting, L2 (squared) regularization was added to the model. The Entity-aware Attention BiLSTM architecture is shown in Fig. 5.



Note: LSTM= long short term memory.

Fig. 5. Architecture of proposed Entity-aware Attention BiLSTM model.

4.5 Knowledge Graph-Based Relation Representation

4.5.1 Query Structure Development

The extracted safety requirements were represented as query graphs, using computation graphs, for supporting subsequent field compliance reasoning using knowledge graphs, where nodes correspond to named entities and edges correspond to relations, as illustrated in Fig. 6(a). Extracted entity-relation triples (i.e., output from relation extraction) were treated as atomic components of query graphs, which were connected using a set of logic operators. Conjunction and disjunction operators are handled using additional blank nodes and auxiliary edges, as illustrated in Fig. 6(b) and Fig. 6(c). There are three types of edges: (1) projection, which uses the semantic relations predefined in Section 4.1 (except logic relations) to connect the nodes; (2) union, which indicates a disjunction operation at the additional blank node it points to; and (3) intersection, which indicates a conjunction operation at the additional blank node it points to. For example, in the sentence "employee … shall be protected by guardrail systems or personal fall arrest systems", two nodes, "guardrail_system" and "personal_fall_arrest_system", were first projected to their corresponding blank nodes using the "Protect" relation, then connected to a single blank node (where the disjunction will be executed) using union edges. Similarly, in the sentence "Articulating boom platforms …shall have both upper and lower controls", the "articulating_boom_platform" node is connected to a blank node for representing the conjunction operation, which is further connected to two blank nodes using intersection edges and then further connected to the two nodes "upper_control" and "lower_control". Qualifiers were then added, to convert the triple-based queries into hyper-relational queries, which can provide further fine-grained constraints for reasoning (Alivanistos et al. 2021; Yu and Yang 2021). This is especially necessary in representing construction safety requirements because relations are sometimes nested. For example, in the phrase "employee on a walking/working surface 6 feet or more above a lower level", three triples were extracted using relation extraction methods that assume flat relations: "Located_At(employee, walking_working_surface)", "Above(walking_working_surface, lower_level)", and "Greater_Or_Equal_To(walking_working_surface, 6_feet)". However, the relation "Greater_Or_Equal_To" is in fact constraining the relation "Above", with a "distance" attribute and some particular value. Therefore, a qualifier of distance was added to the relation of "Above", with a qualifier value of ">=6 feet", as follows: "$\text{Above}_{\{(\text{distance: >= 6 feet})\}}$(walking_working_surface, lower_level)".

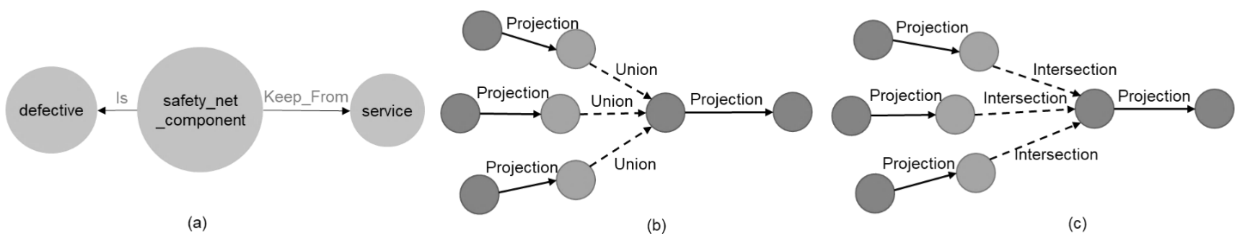


Fig. 6. Example of query structure development: (a) Components of query graphs; (b) Query graph with disjunction operation; and (c) Query graph with conjunction operation.

4.5.2 Query Graph Coding

Query graph coding aims to represent query graphs with the structure described previously into a computer-processable format. The atomic components of the query graphs are in the form of (*h, r, t, qp*), where *h* means head entity, *r* means relation, *t* means tail entity, and the optional $qp = \{(qa_1, qe_1), \ldots\}$ means the set of qualifiers, with $\{qa_1, qa_2, \ldots\}$ as qualifier attributes and $\{qe_1, qe_2, \ldots\}$ as qualifier values. Query graph coding includes four main steps. First, the entity-relation triples obtained from the relation extraction were converted to the correct form, i.e., directions corrected to be from head entity to tail entity, and relation corrected to be in the middle. For example, the triple from the sentence "*anticipated loads*(1) caused by *ice buildup*(2) …" was converted to (ice_buildup, Cause, anticipated_ice), with head entity and tail entity switched. Second, entities and relations in the query graph were assigned with an index, such that the developed query graphs can be more simplified and less repetitive for subsequent compliance reasoning. For example, the triple "(employee, Above, dangerous_equipment)" was mapped to "(11, 11, 65)". Third, qualifiers were added to the main triples for providing additional constraints. Especially, triples indicating comparisons or spatial relations with values were checked for their validity as qualifiers. For example, as per Fig. 7(a), "Above(walking_working_surface, lower_level)" and "Greater_Or_Equal_To(walking_working_surface, 6_feet)", which were extracted from the sentence "each employee on a walking/working surface 6 feet (1.8 m) or more above a lower level", are indicating a spatial relation with values and were thus merged into one triple with a qualifier "(walking_working_surface, Above, lower_level,{(distance, >= 6_feet)})" for a more accurate representation. Fourth, brackets were added to connect the atomic components in each clause, which include logic operators such as "And" and "Or", with a proper name to indicate the query types in terms of number of nodes and operators. For example, "employee … shall be protected from falling by the use of guardrail systems, safety net systems, or personal fall arrest systems", a name of "3u" was used to describe the query of "((guardrail_system, Protect, employee), (safety_net_system, Protect, employee), (personal_fall_arrest _system, Protect, employee))", which involves a disjunction operation among the three types of protection systems that are combined using brackets. Query graphs for each clause were coded using Python 3 and were stored in separate files.

This query-graph representation helps decompose complex requirements into manageable units, while keeping these units connected in a robust and scalable graph structure for supporting subsequent field compliance checking. The graph structure can also help identify missing information in the original regulations, due to occasional omissions in the natural language sentences. For example, for the sentence "employee on a walking/working surface with an unprotected side or edge which is 6 feet (1.8 m) or more above a lower level" [as in Fig. 7(a)], there exists a triple "Above(walking_working_surface, lower_level)" with a certain value. It can be inferred that since the employee is located on the "walking_working_surface", they can be above the "lower_level" as well. Thus, an edge from "employee" to "lower_level" can be identified and added through traversing the links, such that the represented safety requirements can be more complete and accurate in describing interconnections among the entities.

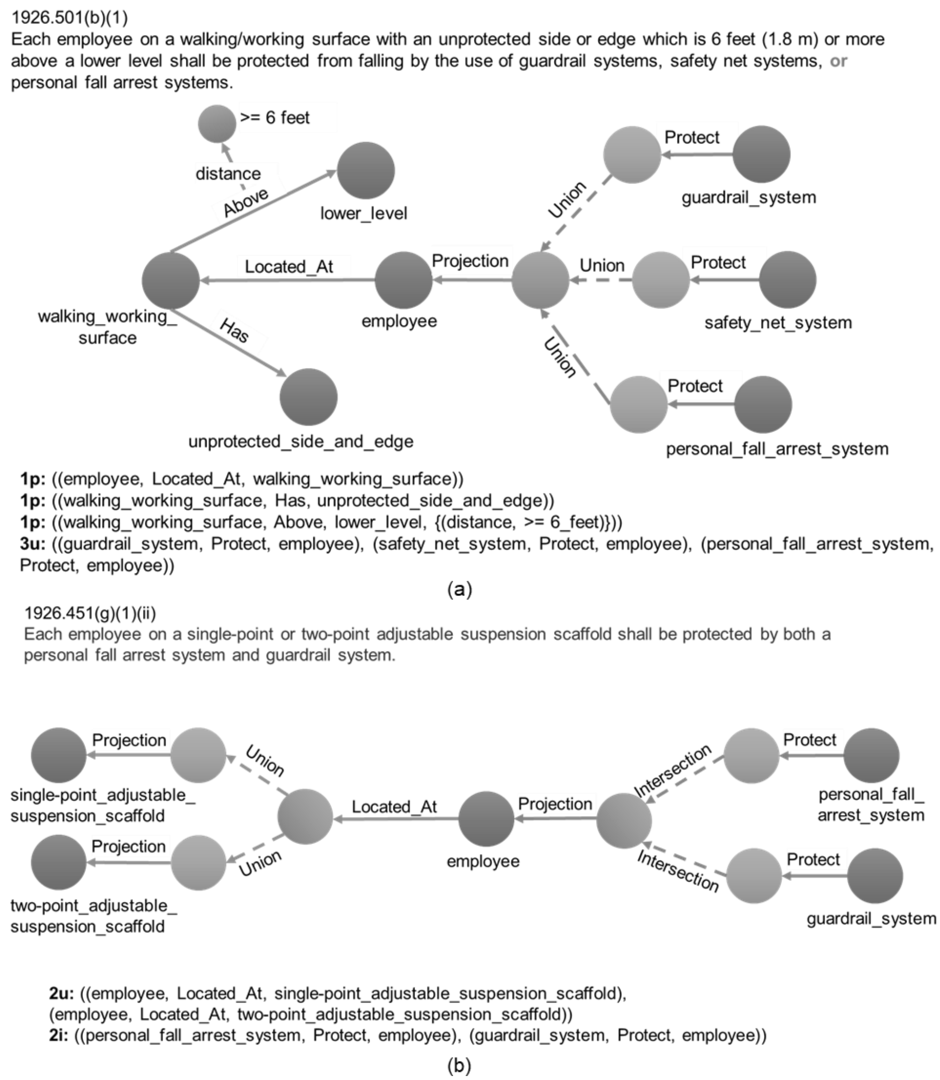


Fig. 7. Query graphs for representing safety requirements: (a) 1926.501(b)(1); and (b)1926.451(g)(1)(ii).

4.6 Evaluation Metrics

Precision (*P*), recall (*R*), and F-1 measure were used to evaluate the relation extraction performance. The three metrics were calculated by comparing the recognized relations with the annotated gold standard, as shown in Eqs. (1)-(3). Precision is defined as the number of correctly recognized relations divided by the total number of all recognized relations. Recall is defined as the number of correctly recognized relations divided by the total number of all relations that should be recognized. F-1 measure is the weighted harmonic mean of precision and recall. A precision-recall curve was also plotted to illustrate the tradeoff between precision and recall across different probabilities, and the area under the curve (AUC) was calculated. A higher AUC indicates that misclassification is less likely to happen. The margins of error at 95% confidence level were also calculated for the precision, recall, and F-1 measure to evaluate the sensitivity of the performance results.

$$P = \frac{\text{number of correctly recognized relations}}{\text{total number of all recognized relations}} \quad (1)$$

$$R = \frac{\text{number of correctly recognized relations}}{\text{total number of all relations that should be recognized}} \quad (2)$$

$$F\text{-}1 = \frac{2 \times P \times R}{P+R} \quad (3)$$

5 Experimental Results and Discussion

The proposed relation extraction method was tested using OSHA sections related to fall protection. A set of experiments were conducted to evaluate the proposed method, including: (1) comparing the performance of the two deep learning models in relation extraction (see Section 4.4), and (2) evaluating the impact of different word embeddings (see Section 4.3). The experiments were implemented using tensorflow and PyTorch on NVIDIA GeForce RTX 2070 SUPER.

5.1 Data Preparation and Gold Standard Development

Twenty (20) OSHA sections related to fall protection were used for developing the dataset. The selected sections cover a variety of fall-related topics such as general fall protection, fall protection systems, guardrail systems, and positioning device systems, as listed in Table 2. The dataset was annotated, following the tagging scheme of the SemEval-2010 dataset from the computational linguistics domain, to create the gold standard for training and testing the relation extraction model. During the annotation, special situations were considered for the "And" relation: (1) most requirements containing "And" were extracted as separate entity-relation triples according to algebraic properties. For example, in the sentence "Dee-rings and snaphooks shall have a minimum tensile strength of 5,000 pounds (22.2 kN)", triples of "Has(dee-ring, tensile_strength)" and "Has(snaphook, tensile_strength)" were annotated, instead of annotating a triple of "And(dee-ring, tensile_strength)" which does not accurately reflect the semantics in the original text. This can also help minimize the number of conjunction operations to simplify subsequent compliance reasoning; and (2) depending on the context in the sentence, the word "and" can sometimes indicate an "Or" relation, which was corrected during annotation. For example, in the sentence "Guardrail systems used on ramps and runways shall be erected along each unprotected side or edge", the requirement actually applies to guardrail systems at any of the two locations (i.e., ramps or runways), thus was corrected to "Or". The annotation process was conducted by three annotators who have background in both civil engineering and natural language processing. An inter-annotator agreement of 91.3% in F-1 measure was achieved, which indicates the reliability of the gold standard (Artstein 2017). Due to the complexity of OSHA clauses, one clause can contain multiple entity-relation triples. The resulting dataset, thus, included a total number of 7,927 entity-relation triples after the annotation (represented as 1,147 query graphs), which were split into training and testing datasets at a ratio of 85:15. The relation extraction performance was evaluated by comparing the extracted results with the developed gold standard, using the aforementioned evaluation metrics (Section 4.6). An example of the annotation is shown in Table 3, and the distribution of relation classes is illustrated in Fig. 8.

A normality test was then conducted to determine whether the dataset follows a normal distribution to further understand its characteristics. Two metrics for measuring the shape of the distribution were calculated for a statistical test: skewness and kurtosis (Jones 1969). Skewness is used to describe if the distribution is symmetrical. A symmetrical distribution will have a skewness of 0. A highly skewed distribution will have a skewness of less than 1 or greater than 1. The annotated dataset resulted in a skewness of 0.5107, which means that it is moderately skewed. Kurtosis is used to describe the height and sharpness of the central peak, compared to a standard bell curve. A normal distribution will have a kurtosis of 0 (Fisher 1992). The annotated dataset resulted in a kurtosis of -0.9994, which means that the distribution has thinner tails and fewer classes with extremely low frequency than a normal distribution. Therefore, the relation classes are not normally distributed. This matches with Zipf's law in the computational linguistics domain (Manning and Schutze 1999), which points out that the distribution of words is highly imbalanced, with some occurring very frequently and others occurring rarely.

Table 2. Selected OSHA Sections Related to Fall Protection

Topic	Section(s)
General requirements	1926.451, 1926.501, 1926.1051
Fall protection systems	1926.502, 1926.760, 1926.1423, 1926 Subpart R App G
Guardrail systems	1926 Subpart M App B
Personal fall arrest systems	1926 Subpart M App C
Positioning device systems	1926.104, 1926.105, 1926 Subpart M App D
Personal protective equipment	1926.95, 1926.96, 1926.100
Scaffolds	1926.452, 1926 Subpart L App A
Ladders	1926.1053
Aerial lifts	1926.453
Housekeeping	1926.25

Table 3. Examples of Annotated Entity-Relation Triples

Original sentence	Annotated sentence[1]	Relation class	Relation index[2]
Each employee on a walking/working surface shall be protected from objects falling through holes (including skylights) by covers.	Each <**e1**>employee</**e1**> on a <**e2**>walking working surface</**e2**> shall be protected from objects falling through holes (including skylights) by covers.	Located_At	1
No employee shall be allowed in an area where an employee is being protected by a safety monitoring system.	No employee shall be allowed in an area where an <**e1**>employee</**e1**> is being protected by a <**e2**>safety monitoring system</**e2**>.	Protect	48

[1] e1 = head tag; e2 = tail tag.

[2] odd number = relation direction is head to tail; even number = relation direction is tail to head.

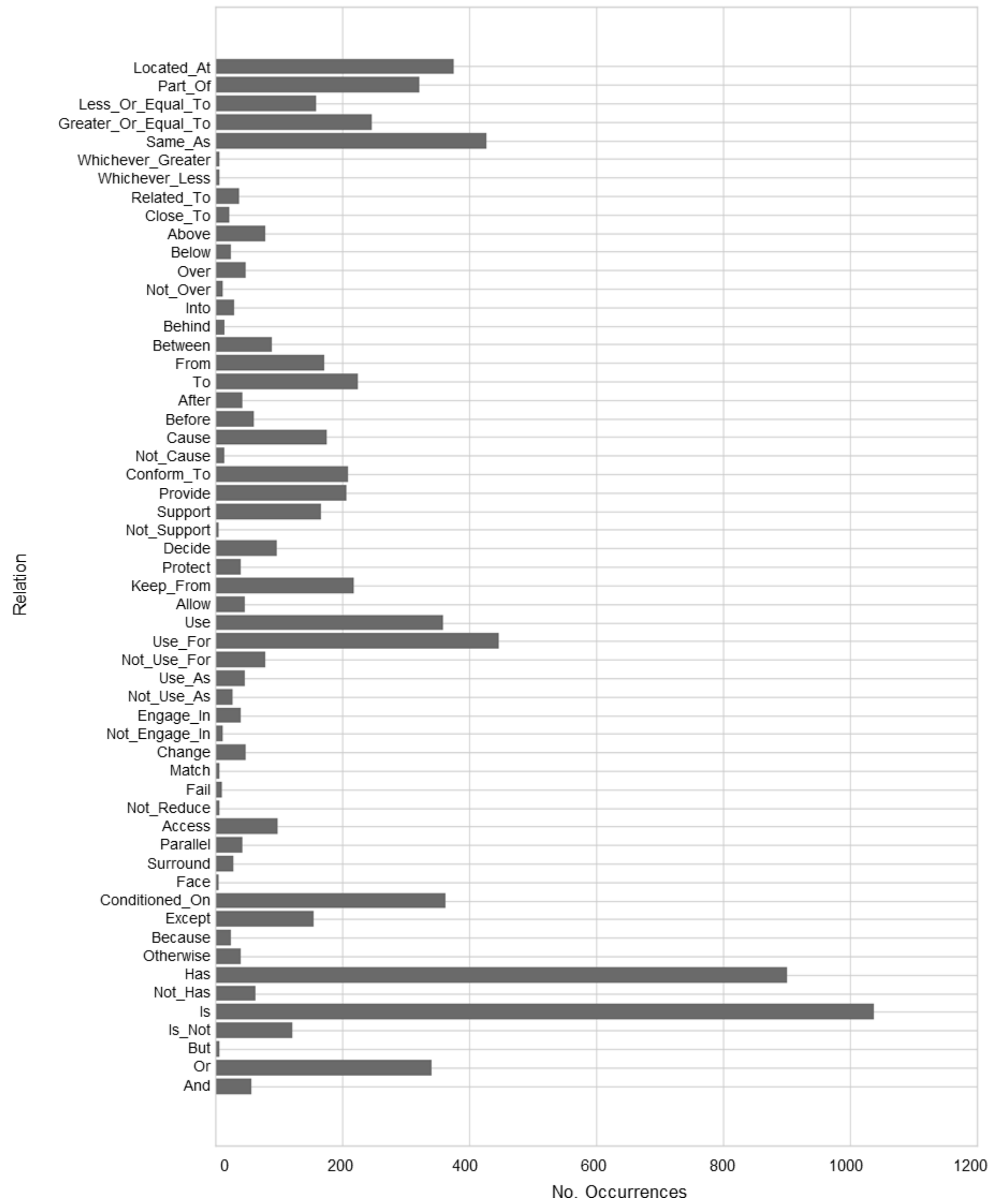


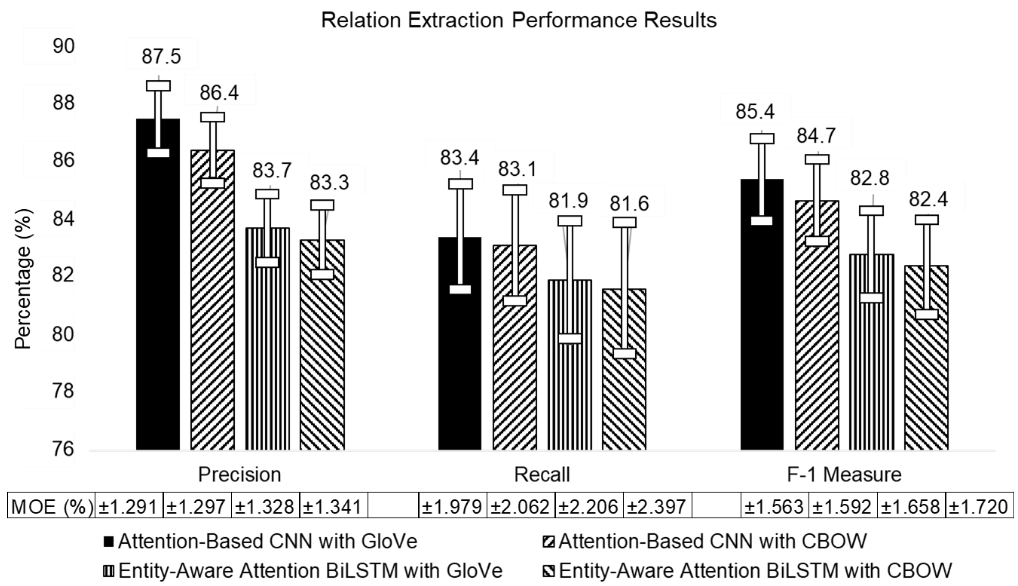
Fig. 8. Distribution of relation classes.

5.2 Relation Extraction Performance

A total number of 1,190 entity-relation triples were extracted from the testing dataset, resulting in 671 query graphs after query graph coding. Example computation graphs and coded queries for each query graph are illustrated in Fig. 7. The hyperparameters of the two models were finetuned for achieving optimized performance. The selected hyperparameters are listed in Table 4. The performance results and precision-recall curves for the two models are shown in Figs. 9 and 10, respectively, which show that both models achieved good relation extraction performance. The proposed Attention-based CNN model, with GloVe embedding, achieved the best results, 87.5% precision, 83.4% recall, and 85.4% F-1 measure (as per Fig. 9), and was hence selected. Comparatively, it also showed a slightly lower margin of error. In comparison, the proposed (RNN-based) Entity-aware Attention BiLSTM model achieved 83.7%, 81.9%, and 82.8%, respectively, as shown in Fig. 9. The superior performance of the proposed CNN-based model is likely because CNN better captures local features with small translations, while RNN better captures context with sequential features. In the construction safety regulations, informative words indicating relations are usually in the vicinity of the given entity pairs, which can be more useful than the dependency structure and sequential features captured by an RNN-based model.

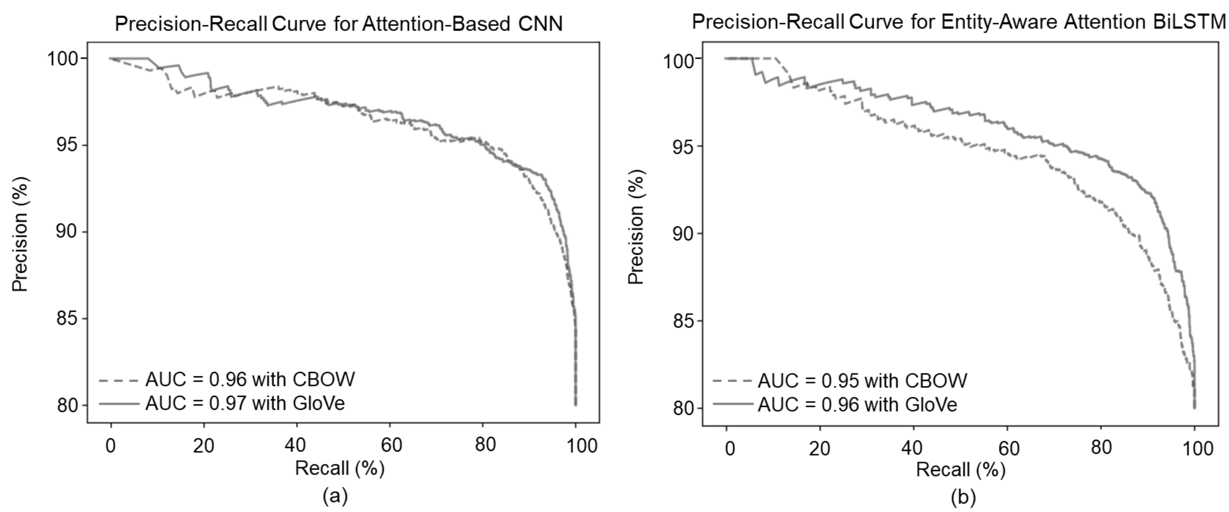
Despite the performance difference, both models were effective in capturing the distinctive semantics between or outside the given entity pairs using the attention mechanism. Example results for the attention mechanism are shown in Table 5. In each example sentence, words with the highest attention weight(s) from both models are marked in bold. For example, in the sentence "1926.501 sets forth requirements for *employers*(1) to **provide** *fall protection systems*(2)", the word "provide" between the given entity pair was assigned with the highest attention weight. In the sentence "Each employee who is constructing a *leading edge*(1) *6 feet*(2) (1.8 m) **or more**", the words "or more" outside the given entity pair were assigned with the highest attention weights.

Fig. 9 illustrates the results of testing the two word embeddings, CBOW and GloVe. The results showed a small difference in the performance of the CBOW and GloVe embeddings. For example, an average precision, recall, and F-1 measure of 87.5%, 83.4%, and 85.4%, respectively, were obtained for the proposed Attention-based CNN model with GloVe embedding, compared to 86.4%, 83.1%, and 84.7% with CBOW embedding. Similarly, comparing the GloVe and CBOW embeddings, in Fig. 10(a) and (b), also shows that both achieved comparable performance (slightly better for the GloVe). These results may indicate that different types of static word embeddings might show similar performance levels for this domain-specific application, and hence it might be beneficial to further explore dynamic word embeddings such as ELMO (Embeddings from Language Models) and BERT (Bidirectional Encoder Representations from Transformers) embeddings in future work.



Note: MOE=margin of error at 95% confidence level.

Fig. 9. Relation extraction performance results using CNN-based and RNN-based models with different word embeddings.



Note: AUC=area under the curve; CNN=convolutional neural network; BiLSTM=bidirectional long short term memory.

Fig. 10. Precision-recall curve: (a) Curve for proposed Attention-based CNN model; and (b) Curve for proposed Entity-aware Attention BiLSTM model.

Table 4. Hyperparameters of the Models

Hyperparameter	**Attention-based CNN model**	**Entity-aware Attention BiLSTM model**
Dropout rate	0.5	0.7
Word-embedding dimension	300	300
Position-embedding dimension	50	50
POS-embedding dimension	15	N/A
Max-sentence length	150	150
L2 weight	N/A	0.00001
Epoch	40	40
Optimizer	Stochastic gradient descent	Adadelta

Table 5. Example Results for Attention Mechanism

Relation	Example results of attention weights
Provide	1926.501 sets forth requirements for <e1>employers</e1> to **provide** <e2>fall protection systems</e2> .
Conditioned_On	<e1>Employees</e1> shall be allowed to work on walking working surfaces **only when** <e2>walking working surfaces</e2> have the strength .
Greater_Or_Equal_To	Each employee who is constructing a <e1>leading edge</e1> <e2>6 feet</e2> (1.8 m) **or more** ,
Less_Or_Equal_To	<e1>length of climb</e1> is **less than** <e2>24 feet</e2>
Not_Has	The <e1>cantilevered portion</e1> of the platform is able to support employees **without** <e2>tipping</e2> .
Except	**Except when** <e1>portable ladders</e1> are used to gain access to fixed ladders, the <e2> portable ladders</e2> shall be offset with a platform .

5.3 Error Analysis

An error analysis was conducted to identify the sources of errors. Ambiguity is a major error source for both models, especially when the relations are indicated using prepositions only. For example, in the phrase "the *ability*(1) of a *ladder*(2) to sustain", the actual relation class is "Has", with a direction from (2) to (1), since (1) is one attribute (2) possesses. However, in the phrase "*steps*(1) of *portable ladders*(2)", the actual relation class is "Part_Of", with a direction from (1) to (2), since (1) is a component of (2). In both cases, there is only one preposition "of" that can provide relevant information for predictions, hence the difficulty to distinguish such cases. Similar situations can be found with other prepositions such as "for", "at", "in", "by", and "to". Therefore, extracting relations from the text with such ambiguities can be difficult.

Frequent omission is another source of error, in which case there is no sufficient information for the model to make the correct predictions. For example, in the phrase "leaving *both hands*(1) *free*(2)", the actual relation class is "Is", with a direction from (1) to (2), since (2) is an attribute of (1). However, there are no other words near the given entities supporting such prediction due to omission. Similarly, in the phrase "*one-eighth*(1) the *working length*(2)", words for indicating relations between the given entities are omitted, which makes it difficult to predict the correct relations.

A lack of domain knowledge can also lead to incorrect predictions. For example, in the sentence "When the *employee*(1) is ascending or descending a *ladder*(2)…", the actual relation is "Use", with a direction from (1) to (2), since both ascending and descending are the actions for (1) to use (2). Similarly, in the phrase "If the *slope*(1) is steeper than *one vertical in eight horizontal*(2)…", the actual relation class is "Greater_Or_Equal_To", with a direction from (1) to (2), since a steeper slope has a higher ratio. However, there is no sufficient context, background information, or term explanations for each OSHA clause. It is, therefore, difficult for the model to make the desired predictions.

Depending on the context, keywords that occur in certain relation classes sometimes do not indicate that relation, which makes it difficult for both models to produce correct predictions, even when the attention mechanism can effectively capture the most distinctive words. For example, the word "or" can indicate an "Otherwise" relation between "perpendicular" and "opposing_angle_tieback", rather than an "Or" relation in the sentence "tiebacks shall be installed perpendicular to …, or opposing angle tiebacks shall be installed", because it refers to the situation that the first condition is not met. The word "and" can indicate a "To" relation in the phrase "distance between the bottom horizontal band and the next higher band", because it indicates the end of that distance. Another example is related to word "but" in the sentence "safety nets shall be installed …, but in no case more than 30 feet below the walking/working surface", which does not indicate a "but" relation between "safety_net" and "30_feet". However, it is combined with its subsequent phrase of "in no case" to be a negation for phrase of "more than", which eventually indicates a "Less_Or_Equal_To" relation for describing the distance between the two levels.

There are two other sources of error for domain-specific relation extraction. First, there are significantly more relation classes in this study, with fewer training samples within each relation class. For example, the SemEval-2010 dataset using general-domain text considers nine relation classes, while in our application, a total of 56 relation classes were considered. Considering that our dataset size is smaller, it may not contain sufficient training samples for certain relation classes. Second, sentences in construction safety regulations are more complex, which makes relation extraction difficult. Such complexity includes longer sentences with a high density of information to be extracted, clauses with nested conditions to describe a particular scenario, and different text patterns across sections.

6 Limitations

Four main limitations of the work are acknowledged, which point to four directions of future work. First, the identified relations are not necessarily complete or exhaustive, especially if additional safety topics or contexts are considered. This is expected because relations in domain-specific applications can vary in semantics and detail from one subdomain/topic to another. Additional testing on different OSHA topics is needed to assess if the identified relations are sufficient, or if additional adaptation or extension effort is needed. Second, in developing the relation extraction model only two alternative deep learning models were tested, a CNN-based model (Attention-based CNN) and an RNN-based model (Entity-aware Attention BiLSTM). In future work, the authors plan to test additional types of deep learning-based models, especially transformer-based models, including different transformer variants and model architectures. Third, only two different static word embeddings were tested and compared in this study. Additional existing word embeddings could be tested in future work, including dynamic word embeddings (e.g., ELMO) or existing domain-specific word embeddings (e.g., Zhang and El-Gohary 2021a). In addition, in future work, the authors also plan to train a domain-specific word embedding using large quantities of construction regulatory documents or dictionaries from multiple construction subdomains, which can be more effective in further improving the relation extraction performance (and performance of other NLP applications in the construction domain). Fourth, the proposed query graph representation may be limited in representing cardinality (e.g., "both" is treated as an attribute not cardinality). The use of additional operators such as cardinality and aggregation can be considered and tested in future work. Fifth, the dataset size used in this study is limited. Given there are 56 relations classes in total, the developed dataset in this study may not contain sufficient training samples for certain relation classes. To further improve the relation extraction performance and generalizability, more text (including clauses from other sources of construction safety regulations) needs to be added to the current dataset.

7 Contributions to the Body of Knowledge

This research offers a new method for automatically extracting relations that describe fall protection requirements from construction safety regulations and representing the extracted information in the form of knowledge graph-based queries. From an intellectual perspective, the proposed method improves the information extraction methodology and application in the construction safety domain in four primary ways. First, it is the first effort to use a deep learning-based method with a combination of word and position embeddings to improve the domain-specific relation extraction performance. The proposed deep learning-based method can reduce the amount of human assistance required in the relation extraction process. The adopted two embeddings can bring rich semantics from the computational linguistics domain and distinguish informative words in a sentence for a deeper understanding of the text and better capturing of the domain-specific features. Second, this study considered non-verbal predicate relations, redundant relations, and the directions of the relations in the relation extraction, which helps accurately describe complex situations considered in the safety regulations without redundancy. The set of relation classes it identified was effective in describing fall-related requirements from OSHA. The relations could also be utilized – as is or with adaptation – for analyzing other construction safety documents such as the fall-related standards from the American National Standards Institute (ANSI), safety reports, etc. Third, the proposed method can directly generate a structured representation for the requirements extracted from construction-domain text. The query-graph representation helps decompose complex requirements into smaller manageable units that are connected in a robust and scalable graph structure. The graph structure can also facilitate the discovery of implicit information through edge traversal to allow for more complete and accurate representation of the safety requirements. Deep learning-based automated reasoning methods can also be developed based on such query graphs. Deep learning-based reasoning methods do not rely on external databases or query languages and can conduct reasoning in a dense and compact embedding space, which would allow for better reasoning performance, generalizability, flexibility, and speed than traditional query language-based compliance checking methods. The query graphs developed in this study can also be integrated with existing query graphs from other domains [e.g., the WD50K-Q (Alivanistos et al. 2021) and FB15k (Ren et al. 2020)] to support future potential efforts that may leverage out-of-domain large-scale graph structures with techniques like transfer learning for improved knowledge graph-based question answering and reasoning. Fourth, the proposed deep learning-based relation extraction method with the two types of features, as well as the method for developing query graphs, are adaptable to more safety topics and more accident types. Adapting and using the proposed method for multiple safety subdomains could help address different types, scenarios, and contexts of accidents – and possibly interdependencies and/or interactions among them – for improved field compliance checking.

From a practical perspective, this paper contributes to the practice of field compliance checking in three ways. First, the paper offers a relation extraction method to automatically extract safety requirements from construction safety regulations, which could be integrated into existing or future software applications for field compliance checking. The use of the proposed method could help eliminate (or reduce) the manual effort that would be needed to hardcode the extracted requirements into computable rules (which is the status-quo if using existing software). Second, the proposed method can extract and represent requirements that cover a variety of safety-related operation scenarios in the field, which could help in checking compliance for different situations onsite. The use of the proposed method could, thus, help improve the application, scope, and generalizability of existing field compliance checking systems and practices. Third, the resulting knowledge graph-represented safety requirements can be easily integrated with other data/information/knowledge or within other existing software systems. For example, safety checking software could easily add a module to represent the construction safety requirements in the form of knowledge queries, such that when real-time field information is collected, it could be checked for compliance with these requirements.

8 Conclusions and Future Work

This paper proposed a method to automatically extract domain-specific relations that describe fall protection requirements from construction safety regulations, as well as represent the extracted requirements as query graphs for supporting subsequent knowledge graph-based field compliance checking. The proposed relation extraction method uses a deep learning model to automatically identify relations from unstructured text and classify them into predefined relation classes. Two types of features were used to leverage the rich semantics from the computational linguistics domain and to help distinguish informative words in a sentence: pre-trained word embedding and position embedding. Two alternative deep learning models, a CNN-based model (Attention-based CNN) and an RNN-based model (Entity-aware Attention BiLSTM), were developed and comparatively evaluated. An attention mechanism was added to both models to better capture distinctive words. A query-graph representation was proposed to represent the extracted safety requirements with explicit semantics in a structured way that represents requirements in the form of smaller manageable units that are connected in a robust and scalable graph structure. The proposed method was tested on 20 OSHA sections related to fall protection.

The proposed Attention-based CNN model with GloVe embedding achieved an average precision, recall, and F-1 measure of 87.5%, 83.4%, and 85.4%, respectively, which showed higher performance than the Entity-aware Attention BiLSTM model with GloVe embedding (83.7%, 81.9%, and 82.8% in precision, recall, and F-1 measure, respectively). A small difference in the relation extraction performance was shown across the two word embeddings. For example, the Attention-based CNN model with CBOW embedding showed insignificantly lower results (86.4%, 83.1%, and 84.7%) than with GloVe embedding (87.5%, 83.4%, and 85.4%, respectively). Five conclusions can, thus, be drawn from the experimental results. First, the proposed relation extraction method was effective in automatically recognizing and classifying domain-specific relations from unstructured text with good performance and minimized human assistance. Second, the proposed CNN-based model (Attention-based CNN) showed better performance than the proposed RNN-based model (Entity-aware Attention BiLSTM), due to its ability to better capture local features with small translations. Third, the attention mechanism used in both models was able to capture the distinctive information located either between or outside the given entity pairs, which helps enhance the extraction performance. Fourth, the models using GloVe embedding achieved comparable performance in extracting the domain-specific relations compared with those using the CBOW embedding. Fifth, the developed query graphs were able to successfully represent the extracted safety requirements and logic operators.

In their future work, the authors plan to explore four main directions. First, the authors will conduct additional research, implementation, and testing efforts to address the aforementioned limitations, as discussed in Section 6. Second, the authors will explore the integration of ontologies with knowledge graphs to enhance the representation and reasoning capabilities of the proposed query-graph representation. Integrating ontologies with knowledge graphs provides knowledge graphs with enhanced schema(s), enriched semantics, and improved reasoning capabilities. Third, the authors will also explore the use of the proposed graph-based representation to support further document and safety analytics. For example, we could leverage the graph analytics to uncover hidden links to discover the underlying reasons leading to noncompliances or common factors contributing to multiple violations. We will also focus on using the query graphs to discover missing requirements, which were hidden due to the implicitness in the natural language sentences, to improve the accuracy and completeness of the represented safety requirements and hence improved compliance assessment. Such analysis could also help bring new insights on how to further improve/refine the writing of the safety regulations to prevent/reduce requirements from being vulnerable to subjective (incorrect) interpretations that could compromise safety and lead to accidents. This is essential because current safety practices are in many cases noncompliant, because workers heavily depend on their own understanding/interpretation of the OSHA requirements and/or the direct guidance they receive from the safety manager – both which may not be fully compliant with OSHA. Fourth, beyond information extraction, the authors will devote their efforts to developing computer vision-based methods to detect site information from images or videos and using graph-based automated reasoning for supporting field compliance checking. Special attention will be paid to how to align these two sets of information properly and how to interpret compliance checking scenarios correctly. These factors are crucial in identifying noncompliances, sending feedback to relevant workers promptly, and improving overall field compliance. Our ultimate goal is to leverage deep learning as well as other artificial intelligence techniques, including natural language processing and computer vision, to automate the process of detecting violations to construction safety regulations promptly and consistently with minimized human assistance.

9 Acknowledgements

The authors would like to thank the National Science Foundation (NSF). This paper is based on work supported by NSF under Grant No. 1827733. 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.

10 References

Alivanistos, D., Berrendorf, M., Cochez, M., & Galkin, M. (2021). "Query Embedding on Hyper-relational Knowledge Graphs." *arXiv preprint arXiv*:2106.08166, https://doi.org/10.48550/arXiv.2106.08166.

Artstein, R. (2017). “Inter-annotator agreement.” *Handbook of linguistic annotation* (pp. 297-313). Springer, Dordrecht, https://doi.org/10.1007/978-94-024-0881-2_11.

Awolusi, I., Marks, E., & Hallowell, M. (2018). “Wearable technology for personalized construction safety monitoring and trending: Review of applicable devices.” *Automation in Construction*, 85, 96-106, https://doi.org/10.1016/j.autcon.2017.10.010.

Baker, H., Hallowell, M. R., & Tixier, A. J. P. (2020). “Automatically learning construction injury precursors from text.” *Automation in Construction*, 118, 103145, https://doi.org/10.1016/j.autcon.2020.103145.

Bellomarini, L., Fakhoury, D., Gottlob, G., & Sallinger, E. (2019). “Knowledge graphs and enterprise AI: the promise of an enabling technology.” *In 2019 IEEE 35th international conference on data engineering* (pp. 26-37). IEEE, https://doi.org/10.1109/ICDE.2019.00011.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008). “Freebase: a collaboratively created graph database for structuring human knowledge.” *In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data* (pp. 1247-1250), https://doi.org/10.1145/1376616.1376746.

Cai, R., Zhang, X., & Wang, H. (2016). “Bidirectional recurrent convolutional neural network for relation classification.” *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (pp. 756-765), https://doi.org/10.18653/v1/P16-1072.

Chen, H., & Luo, X. (2019). “An automatic literature knowledge graph and reasoning network modeling framework based on ontology and natural language processing.” *Advanced Engineering Informatics*, 42, 100959, https://doi.org/10.1016/j.aei.2019.100959.

Chen, J., Song, Q., Zhao, C., & Li, Z. (2020a). “Graph Database and Relational Database Performance Comparison on a Transportation Network.” *In International Conference on Advances in Computing and Data Sciences* (pp. 407-418). Springer, Singapore, https://doi.org/10.1007/978-981-15-6634-9_37.

Chen, X., Jia, S., & Xiang, Y. (2020b). “A review: Knowledge reasoning over knowledge graph.” *Expert Systems with Applications*, 141, 112948, https://doi.org/10.1016/j.eswa.2019.112948.

Chen, Z., Wang, Y., Zhao, B., Cheng, J., Zhao, X., & Duan, Z. (2020c). “Knowledge graph completion: A review.” *Ieee Access*, 8 (pp.192435-192456), https://doi.org/10.1109/ACCESS.2020.3030076.

Cheung, W. F., Lin, T. H., & Lin, Y. C. (2018). "A real-time construction safety monitoring system for hazardous gas integrating wireless sensor network and building information modeling technologies." Sensors, 18(2), 436, https://doi.org/10.3390/s18020436.

Chi, C. F., & Lin, S. Z. (2018). "Classification scheme and prevention measures for caught-in-between occupational fatalities." *Applied Ergonomics*, 68, (pp.338-348), https://doi.org/10.1016/j.apergo.2017.12.007.

Chi, N. W., Lin, K. Y., El-Gohary, N., & Hsieh, S. H. (2017). "Gazetteers for information extraction applications in construction safety management." *In Computing in Civil Engineering 2017* (pp. 401-408), https://doi.org/10.1061/9780784480847.050.

Corke, G. (2013). "Solibri model checker V8." AECMagazine, Building Information Modelling (BIM) for Architecture, Engineering and Construction. <http://aecmag. com/index. php.> (May 17, 2021)

Culotta, A., & Sorensen, J. (2004). "Dependency tree kernels for relation extraction." *In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics* (pp. 423-429), https://doi.org/10.3115/1218955.1219009.

Dong, X. S., Largay, J. A., Choi, S. D., Wang, X., Cain, C. T., & Romano, N. (2017). "Fatal falls and PFAS use in the construction industry: Findings from the NIOSH FACE reports." *Accident Analysis & Prevention*, *102* (pp. 136-143), https://doi.org/10.1016/j.aap.2017.02.028.

ECT Team. (2021). "SMARTVID. IO." https://docs.lib.purdue.edu/ectfs/229/. (May 15, 2021)

Fang, W., Ding, L., Love, P. E., Luo, H., Li, H., Pena-Mora, F., … & Zhou, C. (2020a). "Computer vision applications in construction safety assurance." *Automation in Construction*, 110, 103013, https://doi.org/10.1016/j.autcon.2019.103013.

Fang, W., Ma, L., Love, P. E., Luo, H., Ding, L., & Zhou, A. (2020b). "Knowledge graph for identifying hazards on construction sites: Integrating computer vision with ontology." *Automation in Construction*, *119*, 103310, https://doi.org/10.1016/j.autcon.2020.103310.

Feng, D., & Chen, H. (2021). "A small samples training framework for deep Learning-based automatic information extraction: Case study of construction accident news reports analysis." *Advanced Engineering Informatics*, 47, 101256, https://doi.org/10.1016/j.aei.2021.101256.

Fisher, R. A. (1992). "Statistical methods for research workers." In Breakthroughs in statistics (pp. 66-70). Springer, New York, NY, ISBN: 9351286584.

Gao, T., Han, X., Zhu, H., Liu, Z., Li, P., Sun, M., & Zhou, J. (2019). "FewRel 2.0: Towards more challenging few-shot relation classification." *arXiv preprint arXiv:1910.07124*, https://doi.org/10.48550/arXiv.1910.07124.

Guo, D., Onstein, E., & La Rosa, A. D. (2021). "A Semantic Approach for Automated Rule Compliance Checking in Construction Industry." *IEEE Access*, 9 (pp.129648-129660), http://doi.org/10.1109/ACCESS.2021.3108226.

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Séaghdha, D. O., Padó, S., ... & Szpakowicz, S. (2019). "Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals." *arXiv preprint arXiv:1911.10422*, https://doi.org/10.48550/arXiv.1911.10422.

HCSS. (2022). "Safety reporting". https://www.hcss.com/lp/safety-reporting/?utm_medium=cpc&utm_source=google&utm_campaign=2020_Search_NB_Safety&utm_term=construction%20safety%20reporting&utm_content=135128133489_624656767901_c&creative=624656767901&keyword=construction%20safety%20reporting&matchtype=p&network=g&device=c&product=safety&brand=nb&gclid=Cj0KCQjwnP-ZBhDiARIsAH3FSRfKsSb1o2naAsLv0MVe6wEYbeFWRj8tpPnA7a-Oy8fQoTKQl0FtpUYaAq0yEALw_wcB. (Oct 7, 2022)

Holzschuher, F., & Peinl, R. (2013). "Performance of graph query languages: comparison of cypher, gremlin and native access in neo4j." *In Proceedings of the Joint EDBT/ICDT 2013 Workshops* (pp. 195-204), https://doi.org/10.1145/2457317.2457351.

İrsoy, O., Benton, A., & Stratos, K. (2020). "Corrected CBOW Performs as well as Skip-gram." *arXiv preprint arXiv:2012.15332*, https://doi.org/10.48550/arXiv.2012.15332.

Jebelli, H., Khalili, M. M., Hwang, S., & Lee, S. (2018, January). "A supervised learning-based construction workers' stress recognition using a wearable electroencephalography (EEG) device." *In Construction Research Congress*, (Vol. 2018, pp. 43-53), https://doi.org/10.1061/9780784481288.005.

Jiang, H., Bao, Q., Cheng, Q., Yang, D., Wang, L., & Xiao, Y. (2020). "Complex Relation Extraction: Challenges and Opportunities." *arXiv preprint arXiv:2012.04821*, https://doi.org/10.48550/arXiv.2012.04821.

Jiang, Y., Gao, X., Su, W., & Li, J. (2021). "Systematic knowledge management of construction safety standards based on knowledge graphs: A case study in China." *International Journal of Environmental Research and Public Health*, 18(20), 10692, https://doi.org/10.3390/ijerph182010692.

Ji, S., Pan, S., Cambria, E., Marttinen, P., & Philip, S. Y. (2021). "A survey on knowledge graphs: Representation, acquisition, and applications." *IEEE Transactions on Neural Networks and Learning Systems*, 33(2) (pp. 494-514), https://doi.org/10.1109/TNNLS.2021.3070843.

Jones, T. A. (1969). "Skewness and kurtosis as criteria of normality in observed frequency distributions." *Journal of Sedimentary Research*, 39(4), 1622-1627, https://doi.org/10.1306/74D71EC9-2B21-11D7-8648000102C1865D.

Kaltenboeck, M., Boil, P., Verhoeven, P., Sageder, C., Montiel-Ponsoda, E., & Calleja-Ibáñez, P. (2022). "Using a Legal Knowledge Graph for Multilingual Compliance Services in Labor Law, Contract Management, and Geothermal Energy." *Technologies and Applications for Big Data Value* (pp. 253-271). Springer, Cham, https://doi.org/10.1007/978-3-030-78307-5_12.

Khalid, U., Sagoo, A., & Benachir, M. (2021). "Safety Management System (SMS) framework development–Mitigating the critical safety factors affecting Health and Safety performance in construction projects." *Safety Science*, 143, 105402, https://doi.org/10.1016/j.ssci.2021.105402.

Kim, T., & Chi, S. (2019). "Accident case retrieval and analyses: Using natural language processing in the construction industry." *Journal of Construction Engineering and Management*, 145(3), 04019004, https://doi.org/10.1061/(ASCE)CO.1943-7862.0001625.

Kincelova, K., Boton, C., Blanchet, P., & Dagenais, C. (2020). "Fire safety in tall timber building: A BIM-based automated code-checking approach." *Buildings*, 10(7), 121, https://doi.org/10.3390/buildings10070121.

Lai, S., Leung, K. S., & Leung, Y. (2018). "SUNNYNLP at SemEval-2018 Task 10: A support-vector-machine-based method for detecting semantic difference using taxonomy and word embedding features." *In Proceedings of the 12th International Workshop on Semantic Evaluation* (pp. 741-746), https://doi.org/10.1061/10.18653/v1/S18-1118.

Lee, J., Seo, S., & Choi, Y. S. (2019a). "Semantic relation classification via bidirectional lstm networks with entity-aware attention using latent entity typing." *Symmetry*, 11(6), 785, https://doi.org/10.3390/sym11060785.

Lee, J., Yi, J. S., & Son, J. (2019b). "Development of automatic-extraction model of poisonous clauses in international construction contracts using rule-based NLP." *Journal of Computing in Civil Engineering*, 33(3), 04019003, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000807.

Li, X., Yang, D., Yuan, J., Donkers, A., & Liu, X. (2022). “BIM-enabled semantic web for automated safety checks in subway construction.” Automation in Construction, 141, 104454, https://doi.org/10.1016/j.autcon.2022.104454.

Liu, K., & El-Gohary, N. (2017). “Similarity-based dependency parsing for extracting dependency relations from bridge inspection reports.” *In 2017 ASCE International Workshop on Computing in Civil Engineering* (pp.316-323), https://doi.org/10.1061/9780784480823.038.

Liu, K., & El-Gohary, N. (2021). “Semantic neural network ensemble for automated dependency relation extraction from bridge inspection reports.” *Journal of Computing in Civil Engineering*, 35(4), 04021007, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000961.

Lu, Y., Li, Q., Zhou, Z., & Deng, Y. (2015). “Ontology-based knowledge modeling for automated construction safety checking.” *Safety Science*, *79* (pp. 11-18), https://doi.org/10.1016/j.ssci.2015.05.008.

Manning, C., & Schutze, H. (1999). “Foundations of statistical natural language processing.” *MIT press*, ISBN 0262133601.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014). “The Stanford CoreNLP natural language processing toolkit.” *In Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics* (pp. 55-60), http://doi.org/10.3115/v1/P14-5010.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). “Efficient estimation of word representations in vector space.” *arXiv preprint arXiv:1301.3781*, https://doi.org/10.48550/arXiv.1301.3781.

Mintz, M., Bills, S., Snow, R., & Jurafsky, D. (2009). “Distant supervision for relation extraction without labeled data.” *In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP* (pp. 1003-1011), http://doi.org/10.3115/1690219.1690287.

Miwa, M., & Bansal, M. (2016). “End-to-end relation extraction using lstms on sequences and tree structures.” *arXiv preprint arXiv:1601.00770*, https://doi.org/10.48550/arXiv.1601.00770.

Montiel-Ponsoda, E., & Rodríguez-Doncel, V. (2018). “Lynx: Building the legal knowledge graph for smart compliance services in multilingual Europe.” *In Proceedings of 1st Workshop on Language Resources and Technologies for the Legal Knowledge Graph* (pp. 19-22), https://doi.org/10.3030/780602.

Moon, S., Lee, G., & Chi, S. (2022). “Automated system for construction specification review using natural language processing.” *Advanced Engineering Informatics*, 51, 101495, https://doi.org/10.1016/j.aei.2021.101495.

Moon, S., Lee, G., Chi, S., & Oh, H. (2021). “Automated construction specification review with named entity recognition using natural language processing.” *Journal of Construction Engineering and Management*, 147(1), 04020147, https://doi.org/10.1061/(ASCE)CO.1943-7862.0001953.

Nath, N. D., Behzadan, A. H., & Paal, S. G. (2020). “Deep learning for site safety: Real-time detection of personal protective equipment.” *Automation in Construction,* 112, 103085, https://doi.org/10.1016/j.autcon.2020.103085.

Nguyen, T. H., & Grishman, R. (2015). “Relation extraction: Perspective from convolutional neural networks.” *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing* (pp. 39-48), https://doi.org/10.3115/v1/W15-1506.

OSHA, U. (2020). “Construction Industry: OSHA Safety and Health Standards (29 CFR 1926/1910).” *US Department of Labor, Occupational Safety and Health Administration*, Washington, DC, https://www.osha.gov/laws-regs/regulations/standardnumber/1926. (June 20, 2022).

Park, C. S., & Kim, H. J. (2013). “A framework for construction safety management and visualization system.” *Automation in Construction*, 33, 95-103, https://doi.org/10.1016/j.autcon.2012.09.012.

Pennington, J., Socher, R., & Manning, C. (2014). “Glove: Global vectors for word representation.” *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (pp. 1532-1543), https://doi.org/10.3115/v1/D14-1162.

Ren, H., & Leskovec, J. (2020). “Beta embeddings for multi-hop logical reasoning in knowledge graphs.” *Neural Information Processing Systems*, 33 (pp. 19716-19726), https://doi.org/10.48550/arXiv.2010.11465.

Ren, H., Hu, W., & Leskovec, J. (2020). “Query2box: Reasoning over knowledge graphs in vector space using box embeddings.” *arXiv preprint arXiv*:2002.05969, https://doi.org/10.48550/arXiv.2002.05969.

Ren, R., & Zhang, J. (2021). “Semantic Rule-Based Construction Procedural Information Extraction to Guide Jobsite Sensing and Monitoring.” *Journal of Computing in Civil Engineering*, 35(6), 04021026, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000971.

Riedel, S., Yao, L., & McCallum, A. (2010). “Modeling relations and their mentions without labeled text.” *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp.148-163), https://doi.org/10.1007/978-3-642-15939-8_10.

Roberts, D., Torres Calderon, W., Tang, S., & Golparvar-Fard, M. (2020). “Vision-based construction worker activity analysis informed by body posture.” *Journal of Computing in Civil Engineering*, 34(4), 04020017, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000898.

Rupasinghe, N. K. A. H., & Panuwatwanich, K. (2021). “Understanding construction site safety hazards through open data: text mining approach.” *ASEAN Engineering Journal*, 11(4) (pp. 160-178), https://doi.org/10.11113/aej.v11.17871.

Shen, Y., & Huang, X. J. (2016). “Attention-based convolutional neural network for semantic relation extraction.” *In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics* (pp. 2526-2536), https://aclanthology.org/C16-1238 (June 20, 2022).

Schönfelder, P., & König, M. (2021). “Deep Learning-Based Entity Recognition in Construction Regulatory Documents.” *In Proceedings of the International Symposium on Automation and Robotics in Construction* (pp. 387-394). IAARC Publications, https://doi.org/10.22260/ISARC2021/0054.

Solibri. (2021). “Solibri Model Checker.” https://www.solibri.com/products/solibri-model-checker. (May 15, 2021)

Song, J., Kim, J., & Lee, J. K. (2018). “NLP and deep learning-based analysis of building regulations to support automated rule checking system.” *In Proceedings of the International Symposium on Automation and Robotics in Construction* (pp. 1-7). IAARC Publications, https://doi.org/10.22260/ISARC2018/0080.

Tang, S., Golparvar-Fard, M., Naphade, M., & Gopalakrishna, M. M. (2019). “Video-based activity forecasting for construction safety monitoring use cases.” *In ASCE International Conference on Computing in Civil Engineering 2019* (pp. 204-210), https://doi.org/10.1061/9780784482445.026.

Tang, S., Roberts, D., & Golparvar-Fard, M. (2020). “Human-object interaction recognition for automatic construction site safety inspection.” *Automation in Construction*, 120, 103356, https://doi.org/10.1016/j.autcon.2020.103356.

ul Hassan, F., Le, T., & Tran, D. H. (2020). “Multi-class categorization of design-build contract requirements using text mining and natural language processing techniques.” *In Construction Research Congress* (pp. 1266-1274), https://doi.org/10.1061/9780784482889.135.

Walker, C., Strassel, S., Medero, J., Maeda K. (2006). “ACE 2005 Multilingual Training Corpus.” *Linguistic Data Consortium*, https://doi.org/10.35111/mwxc-vh88.

Wang, C., Kalyanpur, A., Fan, J., Boguraev, B. K., & Gondek, D. C. (2012). “Relation extraction and scoring in DeepQA.” *IBM Journal of Research and Development*, 56(3.4), 9-1, http://doi.org/10.1147/JRD.2012.2187239.

Wang, X., and El-Gohary, N. (2022). “Deep learning-based named entity recognition and resolution of referential ambiguities for enhanced information extraction from construction safety regulations.” *Journal of Computing in Civil Engineering*, (Invited), accepted.

Wu, L. T., Lin, J. R., Leng, S., Li, J. L., & Hu, Z. Z. (2022). “Rule-based information extraction for mechanical-electrical-plumbing-specific semantic web.” *Automation in Construction*, 135, 104108, https://doi.org/10.1016/j.autcon.2021.104108.

Xing, X., Zhong, B., Luo, H., Li, H., & Wu, H. (2019). “Ontology for safety risk identification in metro construction.” *Computers in Industry*, *109* (pp. 14-30), https://doi.org/10.1016/j.compind.2019.04.001.

Xu, P., & Barbosa, D. (2019). “Connecting language and knowledge with heterogeneous representations for neural relation extraction.” *arXiv preprint arXiv:1903.10126*, https://doi.org/10.48550/arXiv.1903.10126.

Xu, X., & Cai, H. (2019). “Semantic frame-based information extraction from utility regulatory documents to support compliance checking.” *In Advances in Informatics and Computing in Civil and Construction Engineering* (pp. 223-230). Springer, Cham, https://doi.org/10.1007/978-3-030-00220-6_27.

Yamada, I., Asai, A., Shindo, H., Takeda, H., & Matsumoto, Y. (2020). “LUKE: deep contextualized entity representations with entity-aware self-attention.” *arXiv preprint arXiv:2010.01057*, https://doi.org/10.48550/arXiv.2010.01057.

Yu, B., Mengge, X., Zhang, Z., Liu, T., Yubin, W., & Wang, B. (2020). “Learning to Prune Dependency Trees with Rethinking for Neural Relation Extraction.” *In Proceedings of the 28th International Conference on Computational Linguistics* (pp. 3842-3852), https://doi.org/10.18653/v1/2020.coling-main.341.

Yu, D., & Yang, Y. (2021). “Improving hyper-relational knowledge graph completion.” *arXiv preprint arXiv*:2104.08167, https://doi.org/10.48550/arXiv.2104.08167.

Zhang, C., Zhang, X., Jiang, W., Shen, Q., & Zhang, S. (2009). “Rule-based extraction of spatial relations in natural language text.” *In 2009 International Conference on Computational Intelligence and Software Engineering* (pp. 1-4). *IEEE*, http://doi.org/10.1109/CISE.2009.5363900.

Zhang, J., & El-Gohary, N. M. (2013). “Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking.” *Journal of Computing in Civil Engineering*, 30(2), 04015014, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000346.

Zhang, J., & El-Gohary, N. M. (2015). "Automated information transformation for automated regulatory compliance checking in construction." *Journal of Computing in Civil Engineering*, 29(4), B4015001, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000427.

Zhang, R., and El-Gohary, N. (2021a). "A deep neural network-based method for deep information extraction using transfer learning strategies to support automated compliance checking." *Automation in Construction*, 132, 103834, https://doi.org/10.1016/j.autcon.2021.103834.

Zhang, R., and El-Gohary, N. (2021b). "Hierarchical representation and deep learning-based method for automatically transforming textual building codes into semantic computable requirements." *Journal of Computing in Civil Engineering*, 36(5), 04022022, https://doi.org/10.1061/(ASCE)CP.1943-5487.0001014.

Zhang, M., Cao, T., & Zhao, X. (2017). "Applying sensor-based technology to improve construction safety management." *Sensors*, 17(8), 1841, https://doi.org/10.3390/s17081841.

Zhang, S., Boukamp, F., & Teizer, J. (2015). "Ontology-based semantic modeling of construction safety knowledge: Towards automated safety planning for job hazard analysis (JHA)." *Automation in Construction*, 52 (pp. 29-41), https://doi.org/10.1016/j.autcon.2015.02.005.

Zheng, X., Wang, B., Zhao, Y., Mao, S., & Tang, Y. (2021). "A knowledge graph method for hazardous chemical management: Ontology design and entity identification." *Neurocomputing*, 430 (pp. 104-111), https://doi.org/10.1016/j.neucom.2020.10.095.

Zhong, B., He, W., Huang, Z., Love, P. E., Tang, J., & Luo, H. (2020a). "A building regulation question answering system: A deep learning methodology." *Advanced Engineering Informatics*, 46, 101195, https://doi.org/10.1016/j.aei.2020.101195.

Zhong, B., Li, H., Luo, H., Zhou, J., Fang, W., & Xing, X. (2020b). "Ontology-based semantic modeling of knowledge in construction: classification and identification of hazards implied in images." *Journal of Construction Engineering and Management*, 146(4), 04020013, https://doi.org/10.1061/(ASCE)CO.1943-7862.0001767.

Zhong, B., Pan, X., Love, P. E., Sun, J., & Tao, C. (2020c). "Hazard analysis: A deep learning and text mining framework for accident prevention." *Advanced Engineering Informatics*, 46, 101152, https://doi.org/10.1016/j.aei.2020.101152.

Zhong, B., Xing, X., Luo, H., Zhou, Q., Li, H., Rose, T., & Fang, W. (2020d). "Deep learning-based extraction of construction procedural constraints from construction regulations." *Advanced Engineering Informatics*, 43, 101003, https://doi.org/10.1016/j.aei.2019.101003.

Zhong, B., Wu, H., Xiang, R., & Guo, J. (2022). "Automatic Information Extraction from Construction Quality Inspection Regulations: A Knowledge Pattern–Based Ontological Method." *Journal of Construction Engineering and Management*, 148(3), 04021207, https://doi.org/10.1061/(ASCE)CO.1943-7862.0002240.

Zhou, P., & El-Gohary, N. (2017). "Ontology-based automated information extraction from building energy conservation codes." *Automation in Construction*, 74 (pp. 103-117), https://doi.org/10.1016/j.autcon.2016.09.004.

Zhu, Z., Zhang, Z., Xhonneux, L. P., & Tang, J. (2021). "Neural bellman-ford networks: A general graph neural network framework for link prediction." *Advances in Neural Information Processing Systems*, 34 (pp. 29476-29490), https://doi.org/10.48550/arXiv.2106.06935.

Zhu, Z., Galkin, M., Zhang, Z., & Tang, J. (2022). "Neural-Symbolic Models for Logical Queries on Knowledge Graphs." *arXiv preprint arXiv*:2205.10128, https://doi.org/10.48550/arXiv.2205.10128.

Zou, P. X., Lun, P., Cipolla, D., & Mohamed, S. (2017). "Cloud-based safety information and communication system in infrastructure construction." *Safety Science*, 98, 50-69, https://doi.org/10.1016/j.ssci.2017.05.006.