A Knowledge Graph for Automated Construction Workers' Safety Violation Identification

Y. Zhu[a,b] and X. Luo[a,b]

[a]Department of Architecture and Civil Engineering, University of Hong Kong, Hong Kong, China
b Research Center of Architecture and Civil Engineering, Shenzhen Research Institute, City University of Hong Kong, Shenzhen, China
E-mail: yifanzhu4-c@my.cityu.edu.hk, xiaowluo@cityu.edu.hk

Abstract

Identifying workers' safety violations on construction job sites is critical for improving construction safety performance. The advancement of sensing technologies makes automatic safety violation detection possible by encoding the safety knowledge into computer programs. However, it requires intensive human efforts in turning safety knowledge into computer rules, and the hard-coded rules limit the expandability of the developed applications. This study proposes a condition-based knowledge graph for the safety knowledge representation to support the reasoning on safety violations. The improved knowledge graph's structure solves the limitation by presenting the public knowledge and safety rules for condition structure, respectively. A natural language processing supported automatic knowledge graph development approach is developed in this paper to extract the safety knowledge from safety knowledge texts automatically and to construct the knowledge graph. To validate this construction framework, an initial knowledge graph containing 1,200 rules is developed based on construction safety regulations. The proposed automatic safety knowledge extraction model achieves an F1 value of 67%.

Keywords

Knowledge Graph; Natural Language Processing; Construction Safety; Workers' Safety Violation

1 Introduction

Timely identifying workers' safety violations onsite is essential to construction safety, and safety knowledge provides guidance to such identification tasks. Recently, research efforts have been made to automatically identify the workers' onsite safety violations using sensing technologies and artificial intelligence. Each of those research focuses on specific safety rules and hard-coded them into the computer applications, limiting the expandability of those developed applications. A generic safety representation that the computer could understand is needed to support the development of automatic worker safety violation identifications.

This research proposes a condition-based knowledge graph to store the safety knowledge, which the computer could understand to support automatic safety violation identification. The proposed knowledge graph consists of two main components: a Rule Knowledge Graph (RKG) used to store the safety rules and an Association Knowledge Graph (AKG) to save corresponding safety knowledge. Furthermore, a knowledge graph development framework based on Natural Language Processing (NLP) is proposed to support the automatic knowledge graph development using safety regulations, reducing the labor cost, and improving the efficiency in constructing such knowledge graph. In this case study, an initial knowledge graph will be established to demonstrate the knowledge graph development and validate the effectiveness of the development framework.

2 Literature Review

2.1 The Application of Artificial Intelligence in Construction Safety

In the construction industry, automatic safety inspection has been applied in various fields related to worker safety, with the most prevalent domain divided into three aspects: Personal Protective Equipment (PPE) detection, exposure to hazardous areas, and unsafe behavior [5, 6]. In the area of PPE detection, Zhang et al. proposed an improved BiFPN-based deep learning method to detect workers and their hardhats [3]. Mneymneh et al. provided an intelligent monitoring framework for hardhats detection by applying motion detection algorithms and object detection tools to capture the required data [4]. In the aspect of identifying worker's exposure to hazardous areas, Fang et al. have tried to detect whether workers are on or across structural support using a convolutional neural network [5]. Konstantinou et al. designed a vision-based approach to track workers with similar appearances and abrupt changes in a complex environment due to congestion, background clutter, and occlusions [6]. In the field of unsafe behavior, Yan et al. established an ergonomic posture recognition technique to capture injury-prone postures so as to prevent accidents and injuries [7]. Distinguishing workers' dangerous and fatigable postures, Seo and Lee tried to prevent work-related musculoskeletal disorders using an ergonomic assessment system in a computer vision-based assessment method [8].

Despite various studies in automatically detecting workers' unsafe behavior, only specific safety rules are used in those researches, without much consideration given to exhaustive references. The knowledge application lacks versatility since the selection of rules is targeted at fitting only a single construction activity rather than a broader range of areas.

2.2 Knowledge Graph

As a representation technique derived from the semantic web, a knowledge graph is widely recognized as one of the essential technologies for knowledge storage and management. Compared with traditional knowledge structures, a knowledge graph allows complex queries across multiple data sources so as to manage construction information on site [9], thereby realizing its applications in complicated and dynamic construction environments. Besides, a knowledge graph can save more time and labor costs by providing in-depth knowledge management methods [10]. In rule-based methods, constructors set up isolated rules, which do not have connections with each other. For the knowledge graph-based methodology, users can flexibly set their mechanisms to find out the results required. In addition, the graph-based knowledge representation can facilitate the discovery of new knowledge hidden behind. Therefore, the knowledge graph representation is more flexible and economical in management while enabling quick reflection of answers through a comprehensive retrieval.

Some researchers have designed several structures for knowledge graphs. For example, Ding et al. proposed an event logic graph, with its nodes being the event, and the edges representing the sequential, casual, is-a relations [11]. Li et al. suggested an AND/OR graph-based knowledge point organization model to represent the selective knowledge that is hard to describe previously [12]. Yu et al. proposed a tax graph to express the calculation logic about specific tax topics; it includes thousands of interconnected calculation models to indicate the calculation statement and contains calculation function nodes and input/output data nodes [12]. Such structures of knowledge graphs above generally have a specific application domain and cannot be used in construction safety fields. Hence, a generalized knowledge graph is essential for furnishing a dependable reference in the construction safety domain.

3 Methodology

To apply numerous construction safety-related knowledge to aid artificial intelligence in spotting safety vilations in an automatic, swift and correct manner, this study refines new knowledge graph representation and suggests a relevant automatic construction model. The results of the study could be used to create a knowledge graph that can be used for the automatic identification of safety violations on construction sites. The framework consists of two major components: a knowledge graph storing the knowledge and an automatic construction framework used to generate the knowledge graph. Figure 1 depicts the structure of the construction framework. The input is a series of one-sentence rules and public knowledge, and the output is the constructed knowledge graph.

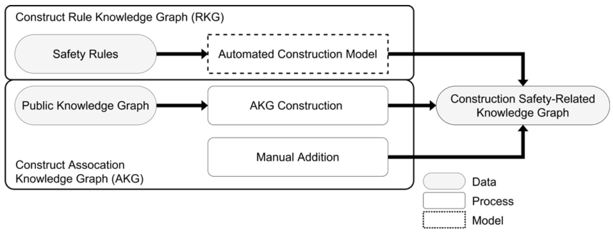

Figure 1. The construction framework structure

3.1 The Structure of the Construction Safety-Related Knowledge Graph

After training the artificial intelligence to automatically identify the safety violation, the artificial intelligence should understand which safety rules the workers should follow. In this process, it is necessary to construct a knowledge graph that can represent rules. Sensing technology, on the other hand, has to learn the public knowledge connected with these rules, meaning that a knowledge graph linked with these rules shall be built up. Figure 2 shows the overall structure of the knowledge graph conceived in this study. It comprises two main knowledge graphs: RKG expresses the construction-related rule knowledge graph, and AKG indicates the public knowledge graph associated with the knowledge on construction safety. The two main knowledge graphs enable a more flexible representation of safety rules and public knowledge and promote knowledge expandability.

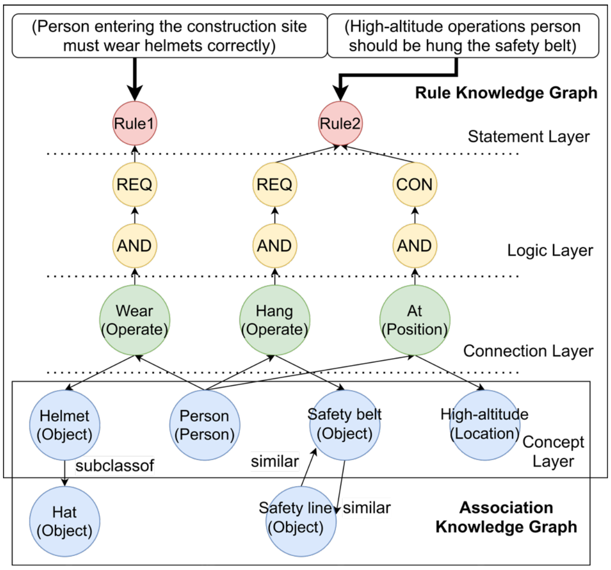

Figure 2. The overall structure of the knowledge graph

3.1.1 The Structure of Rule Knowledge Graph

Because of the dynamic and complex nature of construction environments, the safety criteria and associated circumstances for various types of job contents are diverse, with examples including to use supplementary PPE while conducting different tasks. The following issues are fundamental for safe operation by workers in the context of construction. Which kinds of PPE should be worn by workers? Which workplaces and conditions are permitted for workers to stay? Which process or activity is considered to be safe for workers? [6, 20]. Thus, the primary safety aspects in aiding artificial intelligence to perceive safety events that demand attention are related above. The above-mentioned elements serve as the basis for the RKG's construction. In addition, construction regulations and manuals have codified the safety elements that should be observed into the rules that must be followed. Therefore, artificial intelligence should understand the rules by using a knowledge graph to automatically detect workers' risky activities.

The RKG is made based on the condition-based knowledge graph offered by Jiang et al. [14]. It is composed of a sequence of rules retrieved from safety regulations, with each rule defined by a collection of triplets, shown in equations (1) and (2):

$$t_{u_1} = (\{n_1: a_1\}, n_2, \{n_3: a_3\}) \quad (1)$$

$$t_{u_2} = (\{n_1: a_1\}, r, \{n_3: a_3\}) \quad (2)$$

where $n_1, n_3 \in C$, and C is a set of concept nodes in the triplet; $n_2 \in O$, and O is a set of connection nodes in the triplet; $a_1, a_3 \in A$, and A is a set of attributes. '1' is for the subject, and '3' is for the object; r is the relation between n_1 and n_3. Triplet t_{u_1} uses the connection node as a connection entity to represent the relation, and t_{u_2} applies the relations 'Con_Belongto' to show the connection. Furthermore, triplet t_{u_1} represents an event such as a worker standing at a height, a worker moving rebar, or a worker wearing the hardhat. Triplet t_{u_2} denotes the affiliation between entities in a specific rule. Furthermore, in an RKG, the attribute and affiliation of an entity are valid only in the rule it is in.

Jiang et al. have introduced a condition-based knowledge graph [14], but it has no thorough description of the logic execution and sequence of triplets in the requirement and condition components. Furthermore, there are no detailed type definitions for the triplets' subjects, objects, and relation parts in triplets, leading to ambiguous expressions, i.e., some entities owning several meanings. Thus, this study adds a logic layer to the graph structure to showcase the details of the rules' logical execution and the relationship between the triplets in the requirement and condition part. In addition, this study adds the entity type to the subjects, objects, and relation in triplets, which will play an assistance role in the follow-up querying procedure. The proposed structure of the knowledge graph after the modification is shown in Figure 2, which is categorized into four layers: concept layer, connection layer, logic layer, and statement layer.

The concept layer contains concept nodes, which are the subject and object entities in triplets. The concept nodes have five types to display five categories of entities: ***Person***, ***Work***, ***Object***, ***Location***, and ***Environment***. ***Person*** indicates the roles and professions on the construction sites, such as the Person concept node in Figure 2; ***Work*** represents the behavior and action of the people and machinery like Climb; ***Object*** indicates the objects usually appear on the sites, and this type of entity has its own attributes to additionally define its requirements or circumstances within the rules they are in like Helmet concept node in Figure 2; ***Location*** indicates a range belonging to the machinery, object, people or region like High-altitude concept node in Figure 2; ***Environment*** indicates the weather or times on construction sites such as the wind, and it uses three predefined attributes to show the degrees and levels: ***Unit***, ***Value***, and ***Property***. These three attributes are illustrated in the selection range of weather conditions in the current rule. Particularly, ***Unit*** presents the weather's unit including level, m/s; ***Value*** determines the value of the weather's level, which is composed of digital form like 6 or six; and ***Property*** exhibits the weather's value range, such as larger than, not less than. Furthermore, the Object- and Location-typed nodes have 'Con_Belongto' links to describe the attribution relationships.

The connection layer includes connection nodes and links connected with the concept nodes, and the link direction shows that a concept node is a subject or an object in triplets. Connection nodes have three types applied in three distinct domains: ***Operate***, ***Position***, and ***Predicate***. Generally, ***Operate*** is applied in a worker's PPE domain, demonstrating the relationship between worker and PPE, such as wear, hang. ***Position*** regulates the place and weather relation like up, down, or around. ***Predicate*** represents the working condition of workers and things and how they operate other things including use and operate.

The logic layer denotes the logic execution in the requirement and condition part as well as the logic order between triplets. The logic layer is like an event tree, in which the nodes are similar to tree branches. A larger branch can continue to be extended to smaller branches, with most terminals being the connection nodes. The nodes in the logic layer have two categories: the logic node and the part node. The former one plays a crucial role in the logic execution representation; in safety rules, it is an objective existence, meaning that some requirements and conditions have distinct constraints. The logic node is indispensable to describe this circumstance. The logic node is divided into three categories: AND node, OR node, and NOT node. The AND node means that all child nodes should be followed. The OR node means that at least one of the child nodes should be obeyed. The NOT node means that all children nodes should not be followed. The result of logic nodes will be concluded to part nodes, i.e., the Req and Con nodes. The part node denotes which triplets belong to the requirement and condition part through connection to the Req and Con nodes. For example, in Figure 1, the triplet [Person, Hang, Safety Belt] is the requirement part due to the connection to the Req node, and this triplet should be followed as it is connected with the AND node.

The statement layer uses statement nodes to define each rule, and is the root node in the knowledge graph to represent rules. The statement node indicates which requirement and condition part belongs to a certain rule by connecting the requirement and/or condition-typed nodes.

3.1.2 The Structure of Associated Knowledge Graph

Except for rules, artificial intelligence is also required to learn the knowledge related to construction safety so as to facilitate the safety violation identification ability. For instance, the relationship between the subclasses of hats or objects has a similar meaning. Therefore, this study constructs the knowledge graph associated with the construction safety-related rules (AKG). The entities come from the concept layer of RKG and artificial additions. Triplets in AKG work in any rule. In contrast, some triplets in RKG only work in the rules they belong to. Compared with RKG, the relation in triplets of AKG exists in the form of edges, not nodes. Formula (2) denotes triplets in AKG.

AKG implements two links: 'Similar' and 'subclassof'. The former one indicates the entities that have a similar meaning; the latter one means that a subject entity is the subclass concept to the object entity. The relation 'subclass' is an edge in AKG to connect the subject and object. Likewise, the relation 'similar' is an edge in a triplet, e.g., [Safety line, similar, Safety belt]

3.2 The Construction Framework of the Knowledge Graph

In the construction framework, in addition to implementing the automatic extraction model as proposed by Wei [15], this study designs a knowledge graph construction procedure. The construction procedure is responsible for transforming the extracted file into a knowledge graph based on the predefined rules and methods.

In practice, some texts describing workers' normative requirements lack the subject, while supplementing the subject words to each text is a time-consuming task. Additionally, in RKG, triplets are composed of three parts: subject entity, connection entity, and object entity. If a connection entity is fixed as several relations to link the subject and object entities, it will not represent more elements flexibly, and the complexity will increase dramatically. Meanwhile, the representation of belonging parts and types for triplets in a tuple label is indispensable. Therefore, the extraction model needs the specific relation label representation to suit the RKG with multiple layer structures. To solve this issue, this study divides the relation label into five types: the relation label for labelling subject and connection entities; the connection entity for object label; the entity for attributes; the subject for objects; and the connection entity for the connection entity.

The information of relation labels includes the triplet type, a requirement or a condition part, and two entities' types (the subject and object). For example, the tuple label 'Operate-Req_AND-Object' means that the subject type is Operate, the object type is Object, and the relation between the two entities is Req_AND, where 'Req' indicates this tuple belongs to the requirement part, and 'AND' indicates the tuple's execution logic. Thus, the relation label can show the variety of data in the relation label.

In addition, some relation labels are disparate from the labels mentioned above. For instance, 'Object-Con_Belongto-Object' means that the object and subject entities' type is Object, and the subject entity belongs to the object. Besides, some triplets do not need a connection entity, and thus their relation label will be like 'Object-Con_AND-Work' to describe this circumstance, while a connection entity will be added in the triplet in the construction of the knowledge graph. Likewise, the relation type also has the attribute representation like 'Object-Con_AND-Attribute', and the connection entity's sequence representation of 'Predicate-Connect_Dis-Predicate' indicates that the triplet belonging to each connection entity should be judged separately.

3.2.1 The Automated Construction Procedure of Rule Knowledge Graph

After finishing the part of triplet extraction, the next step is to construct the knowledge graph based on the extracted partial triplet. Firstly, for the extraction content of the automated extraction model this study calls tuples, the tuple sets will be inputted for expansion. For instance, the type 'Object-Con_AND-Work' means the condition that the object is doing some work. Nevertheless, this form is not appropriate for the RKG, so a division is necessary. This tuple will be divided into 'Object-Con_AND-Predicate' and 'Predicate-Con_AND-Work'; the Predicate-typed entities in two new tuples through the predefined procedure are the same, both the entity 'conduct'. Secondly, some triples represent the entity's attributes, such as the 'Environment-Con_OR-Property' and 'Object-Con_AND-Attribute'; they will not be constructed as the triples in the knowledge graph. In contrast, the attribute entity will show the attributes to the subject in the corresponding rules. Thirdly, the two tuples with the same connection entities and same relations will be combined with the new triplet in the knowledge graph. In addition, some rules do not own the subject, so the labeled triples in these rules will all be the latter part of the triple in the RKG, and the new first half of the corresponding triple will be added. Fourthly, the corresponding logic layer will be generated, followed by the relation in the triple, and the triple containing two connection entities will also participate in the logic layer's generation. Finally, after finishing the four steps above, each text will create a statement node to associate with the logic node and indicate the belonging of the requirement and condition parts.

3.2.2 The Construction of Association Knowledge Graph

For the generation of AKG, the data usually come from two sources: the public knowledge graph and the knowledge added manually. For the public knowledge graph, this research will search each concept node in the public knowledge graph for similar entities and the sub-class or upper-class entities. Furthermore, this study will observe the relations between the same-type entities in the concept layer and add the SubClassof, with similar relations in these entities.

4 Case Study

To demonstrate the construction process of the knowledge graph, a case study is carried out mainly in two steps to verify the feasibility. First, the case study will manually establish the construction safety-related knowledge graph by relying on safety documents and the public knowledge graph. Second, this study will develop an automated extraction model to improve the automatic construction model of the knowledge graph. Figure 3 shows the overall process, with these steps explained in detail below.

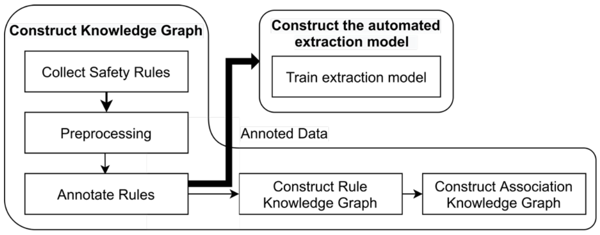

Figure 3. The process of case study

4.1 Construct the Knowledge Graph

4.1.1 Collect Safety Rules

Nearly 97 specifications have been collected, including Chinese National Standards (GB), construction industry standards (JGJ), and safety manuals. These documents were chosen because they all involve construction safety and workers' behavior safety, while showcasing a certain universality.

4.1.2 Preprocessing

This study starts with data preprocessing to handle the construction codes collected. First, this study determines the extraction range of documents, eventually picking out provisions in the three areas mentioned in Section 3.1, and excluding those requirements related to other construction safety conditions. To represent it clearly, this study divides the relatively complex rules into simple contents, and rules will be replenished with the subject and object, if missing, based on the chapter titles. Finally, 1,236 rules are collected as the corpus used for the next step.

4.1.3 Annotate Rules

The rule annotation extracts the imperative data from unstructured text and facilitates artificial intelligence-based safety inspection, and it further marks the entities and the relations between the entities in the rule. On one side, this study adopts the extracted data as the corpus to train the automated extraction model. On the other side, to ensure the accuracy and efficiency of the knowledge graph, this study uses the extracted data as the input to construct the RKG, without using the output of the automated extraction model. This study extracts information by applying brat rapid annotation tool (BRAT), which is a web-based tool for annotating by adding notes to the existing text documents [16]. Designed for systematic annotation, it has a defined structure that artificial intelligence can process and understand. BRAT involves two types of annotations: text span and relation annotation. The former one marks the entities and their types.

4.1.4 Construct the Rule Knowledge Graph

The annotated corpus will be fed into the automated construction procedure to construct the RKG. As one of the most popular tools for most knowledge graph storage, Neo4j is extensively used in many studies [10]. Therefore, this research uses Neo4j as the storage medium [17]. After annotation, this study outputs the annotated entities, including their name, type, and id number with interrelated relations; then, they are saved in the graph database management system Neo4j, as shown in Figure 4. The example rule shown in figure 4 has a set of triplets: [Person, at, high altitude] (Con_AND), [Person, Hang, safety belt] (Req_AND), indicating that when people are at a high altitude, they should wear the safety belt.

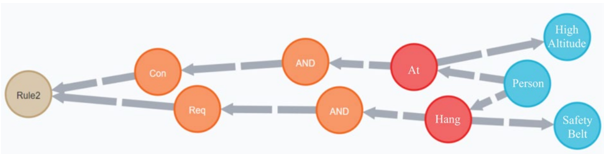

Figure 4. One rule in the Neo4j

4.1.5 Construct the Association Knowledge Graph

The corresponding AKG is constructed to be added to the relevant knowledge graph. OwnThink is a knowledge graph-based public knowledge graph, which incorporates twenty-five million entities with billions of entity-attribute relationships [18]. This study extracts similar word relations from the Ownthink knowledge graph. Searches for the Concept entity both in the knowledge graph and Ownthink find the similar entity through relations 'Also Known As (又名)' and 'Another Name (别名)'. Furthermore, this study manually identifies the 'subclassof' and 'similar' relations between person-type entities for better querying. Finally, The AKG is constructed in this step based on the public knowledge graph resources and manuals. The searched relations and entities are inputted into Neo4j manually.

4.2 Training and Performance of the Automated Extraction Model

The automated extraction model needs to be trained so as to make the model extract the data more precisely. After annotation of safety rules, the annotated files will be transformed into the json-type files and are regarded as the corpus for the extraction model. The BERT model in the extraction model adopts the Chinese_wwm_ext_L-12_H-768_A-12 pre-training model proposed by Cui et al. [19]. The corpus is divided into three components: train corpus, valid corpus, and test corpus, with a ratio of 7.5:1.5:1.5, respectively.

The results acquired after the training of the extraction model are shown in Table 1, illustrating the performance of the automated extraction model. The title 'Type' means the tuple's type; 'Precision' means the precision in each type; 'Recall' means the recall value in each type. The overall precision, recall, and F-1 value are 78.07%, 58.92%, and 67.15%, respectively. More specifically, this study divides the types into two parts: the first half of the triplet, and the second half of the triplet. The first half of the triplet is the subject, the connection, and the link between these two nodes, the second half is the connection, object, and the links between these two nodes. The performance of the first half is Precision (95.24%), Recall (60.60%), and F1-value (74.07%); the second half of the triplet is Precision (76.42%), Recall (59.54%), and F1-value (66.93%). Observation shows the performance of the first half is better than the second half. In addition, this study divides the tuple type into four parts based on the function: Operate, Position, Predicate, and Attribute. The performance of Operate is: Precision (79.66%), Recall (78.99%), and F1 (79.32%); the performance of Position is: Precision (85.31%), Recall (60.70%), and F1 (70.93%); the performance of Predicate is: Precision (71.11%), Recall (50.96%), and F1 (59.37%); the performance of Attribute is: Precision (86.05%), Recall (55.22%), and F1 (67.27%). These results indicate that the performance of Predicate and Attribute is the weakest, the performance of Position is strong, and the performance of Operate is the best.

Table 1. The performance of the automated extraction model in part type

Type	Precision (%)	Recall (%)	F1 value (%)
Operate-Req_NOT-Object	72.73	66.67	69.57
Position-Con_AND-Object	75.00	40.00	52.17
Person-Req_NOT-Position	92.31	85.71	88.89
Person-Req_AND-Operate	100.00	77.78	87.50
Object-Con_Belongto-Object	50.00	10.00	16.67
Environment-	100.00	76.92	86.96

Con_OR-Property			
Environment-Con_OR-Unit	92.31	92.31	92.31
Environment-Con_OR-Value	92.31	92.31	92.31
Predicate-Req_NOT-Work	40.00	12.50	19.05
Position-Req_NOT-Location	81.82	47.37	60.00
Object-Con_AND-Attribute	50.00	18.18	26.67
Position-Con_AND-Location	87.50	60.87	71.79
Predicate-Con_AND-Work	63.64	36.84	46.67
Object-Con_AND-Work	78.26	64.29	70.59
Predicate-Con_OR-Object	65.52	65.52	65.52
Position-Con_OR-Environment	96.30	96.30	96.30
Position-Req_NOT-Object	76.32	55.77	64.44
Predicate-Con_AND-Object	71.74	52.38	60.55
Operate-Req_AND-Object	79.00	85.87	82.29
Predicate-Req_NOT-Object	72.48	54.48	62.20
Overall	78.07	58.92	67.15

5 Discussion

For the automated extraction model, this study excludes the tuple types with their number less than 30 to analyze the results since most of them have F1-values of zero. In the results, the tuple type with a relatively fixed structure will have a strong influence. The tuples like 'Position-Con_OR-Environment', 'Environment-Con_OR-Unit', 'Environment-Con_OR-Value' and 'Environment-Con_OR-Property' have a fixed text structure about 'when in some weather'. Thus, the performance of these tuples is better than other tuples. In addition, the tuple types 'Person-Req_AND-Operate' and 'Operate-Req_NOT-Object' have the fixed text structure 'should/shouldn't be equipped with something '. Nevertheless, the number of 'Operate-Req_NOT-Object' is smaller than that of other types, and it may impact the final performance. Furthermore, both types of 'Person-Req_NOT-Position' and 'Person-Req_AND-Operate' have the fixed structures, where 'Person-Req_NOT-Position' has the structure of 'Non-worker cannot go into it ', while 'Person-Req_AND-Operate' has the text structure of 'A certain types of worker needs to be equipped with it'. Therefore, the tuple types with fixed structures will deliver a good performance. The types of 'predicate-Req_NOT-Work', 'Object-Con_Belongto-Object' and 'Object-Con_AND-Attribute' give the worst performance, with variable text structures; and the extraction model faces some challenges in the annotation. Some potential solutions are proposed. On one side, more corpus could be furnished for training the model. On the other side, using the hidden connections between different concepts in post-processing can also enhance accuracy.

6 Conclusion

This research develops a unique knowledge graph and corresponding construction framework to assist artificial intelligence-based identification of workers' safety violations. The knowledge graph includes two parts: a Rule Knowledge Graph (RKG), including four layers (Statement layer, the Logic layer, the Connection layer, and the Concept layer), and an Association Knowledge Graph (AKG). The F1-value performance of the automated extraction model in the construction framework can reach 67%.

The proposed framework showcases the following advantages. First, it can assist artificial intelligence to automatically identify safety violations by searching for all rules related to the scene description. Second, users can flexibly adjust the identification range in constructing the knowledge graph. Third, the proposed construction framework is beneficial for the automatic construction of knowledge graphs, while reducing the labor cost incurred in such constructing. In practical applications, the acquired information from the variety of sensors will be transformed to a text-based description. The artificial intelligence identifies the worker's violation in description based on the constructed knowledge graph.

Despite the successful advantages above, several limitations with this study have to be overcome. First, the automated extraction model still needs to be enhanced for better performance. Second, the knowledge graph shall collect more related rules, whether on construction codes or even corporate regulations. Third, the proposed framework cannot convert construction-site videos into textual description, and it has not been verified in a practical application.

Future research directions include: First, to further boost the performance of the automated extraction model. Moreover, a more accurately semantic matching method relying on machine learning based on the current mechanism can be implemented in the querying model. Second, by collecting more relevant regulations to be stored in the knowledge graph, a more comprehensive range of construction safety domains will be suitable for this model. Lastly, the proposed framework can be integrated with some practical applications on construction sites, so as to verify its feasibility. For example, in order to improve detection efficiency, it can be combined with computer vision in camera monitoring.

Acknowledgment

This work was supported by the Shenzhen Science and Technology Innovation Committee Grant (PJ#JCYJ20180507181647320). The conclusions herein are those of the authors and do not necessarily reflect the views of the sponsoring agency.

References

[1] W. Fang, P. E. D. Love, H. Luo, and L. Ding, Computer vision for behaviour-based safety in construction: A review and future directions, *Advanced Engineering Informatics*, 43, August 2019, page 100980,2020

[2] W. Fang *et al.*, Computer vision applications in construction safety assurance, *Automation in Construction*, 110, February 2019, page 103013,2020

[3] C. Zhang, Z. Tian, J. Song, Y. Zheng, and B. Xu, Construction worker hardhat-wearing detection based on an improved BiFPN, page 8600–8607,2021

[4] B. E. Mneymneh, M. Abbas, and H. Khoury, Vision-Based Framework for Intelligent Monitoring of Hardhat Wearing on Construction Sites, *Journal of Computing in Civil Engineering*, 33, 2, page 04018066,2019

[5] W. Fang *et al.*, A deep learning-based approach for mitigating falls from height with computer vision: Convolutional neural network, *Advanced Engineering Informatics*, 39, January, page 170–177,2019

[6] E. Konstantinou, J. Lasenby, and I. Brilakis, Adaptive computer vision-based 2D tracking of workers in complex environments, *Automation in Construction*, 103, November 2018, page 168–184,2019

[7] X. Yan, H. Li, C. Wang, J. O. Seo, H. Zhang, and H. Wang, Development of ergonomic posture recognition technique based on 2D ordinary camera for construction hazard prevention through view-invariant features in 2D skeleton motion, *Advanced Engineering Informatics*, 34, June, page 152–163,2017

[8] J. Seo and S. Lee, Automated postural ergonomic risk assessment using vision-based posture classification, *Automation in Construction*, 128, March 2017, page 103725,2021

[9] P. Pauwels, S. Zhang, and Y. C. Lee, Semantic web technologies in AEC industry: A literature overview, *Automation in Construction*, 73, page 145–165,2017

[10] X. Li, M. Lyu, Z. Wang, C. H. Chen, and P. Zheng, Exploiting knowledge graphs in industrial products and services: A survey of key aspects, challenges, and future perspectives, *Computers in Industry*, 129, page 103449,2021

[11] X. Ding, Z. Li, T. Liu, and K. Liao, ELG: An Event Logic Graph, [Online]. Available: http://arxiv.org/abs/1907.080152019

[12] S. Li, X. Li, and L. Wang, Knowledge points organization model based on AND/OR graph in ICAI, *Proceedings - 2010 6th International Conference on Natural Computation, ICNC 2010*, 4, Icnc, page 2121–2124,2010

[13] W. Fang, P. E. D. Love, H. Luo, and L. Ding, Computer vision for behaviour-based safety in construction: A review and future directions, *Advanced Engineering Informatics*, 43, August 2019,2020

[14] T. Jiang *et al.*, Biomedical Knowledge Graphs Construction from Conditional Statements, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 18, 3, page 823–835,2021

[15] Z. Wei, J. Su, Y. Wang, Y. Tian, and Y. Chang, A Novel Cascade Binary Tagging Framework for Relational Triple Extraction, page 1476–1488,2020

[16] NLPlab. Online:https://brat.nlplab.org/introduction.html

[17] Neo4j. Online:https://neo4j.com/

[18] Ownthink. Online:https://www.ownthink.com/docs/kg/

[19] Y. Cui, W. Che, T. Liu, B. Qin, and Z. Yang, Pre-Training with Whole Word Masking for Chinese BERT, *IEEE/ACM Transactions on Audio Speech and Language Processing*, 29, November, page 3504–3514,2021