School of Design and the Built Environment

A Semantic Information Management Approach for Improving Bridge Maintenance based on Advanced Constraint Management

Chengke Wu

This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University

July 2021

Declaration

To the best of my knowledge and belief, this thesis contains no material previously published by any other person except where due acknowledgment has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Signature:

Date:

(left blank)

Acknowledgments

I would like to express my gratitude to all those who have helped me complete the thesis. First, I want to thank my main supervisor, Professor Xiangyu Wang, who gave me wise advice at every step in my research, and encouraged me all along the way whenever I felt frustrated.

I am also extremely fortunate to have Professor Peng Wu as my co-supervisor. He gave me lots of useful suggestions, helping me gain a deeper understanding of how to do research and how to be a good researcher. He spent much time and effort reviewing and revising every academic paper I drafted. It is impossible to get these papers published without his support.

I am very grateful as well to have the opportunity to work with the research colleagues in the BIM Centre: Mrs Rui Jiang, Mr Xiang Lei, Mr Weixiang Shi, Mr Wenzheng Ying, Mr Kai Luo, Mrs Shuyuan Xu, Dr Yongze Song, Dr Jian Chai, Dr Hung-Lin Chi, and Dr Heap-Yih Chong. Special thanks to Dr Junxiang Zhu, Dr Jun Wang, Professor Chao Mao, and Professor Chimay Anumba, who offered technical advice for the hybrid deep learning model training and ontology development.

Most data used in this research were collected from a public online database. However, it is still important to collaborate with the industry to conduct the focus group. Thus, I must thank all participants who provided their valuable knowledge, despite that I cannot mention their identities due to the ethic issue.

This study would not be possible without the financial support from the Australian Research Council discovery project (grant number DP170104613).

I am also indebted to the colleagues in the School of Design and Built Environment, such as Christine Finally, thanks for assisting me with my study and my life at Curtin.

Finally, never to be forgotten are my warmest family and the dearest Mrs Rui Jiang. I fall short of words to properly express my gratitude for their love and support. I wish to proudly dedicate this piece of work to them all.

(left blank)

Abstract

Bridges are critical infrastructures, and effective maintenance is critical to keep them in a good condition, which includes inspection, condition evaluation, decision-making, and rehabilitation. Bridge rehabilitation, where most resources are spent, to-a-large-extent determines the success of the bridge maintenance. However, it is challenging to manage bridge rehabilitation projects due to the complex constraints involved (e.g., various materials, equipment, and drawings). Constraint management approaches, e.g., Advanced Working Packaging (AWP), are good at managing complex constraints by defining and managing work packages with different granularities. Work packages group constraint entities affecting one another (e.g., tasks and resources) and minimise work interruption and delay by ensuring all constraints are removed before work. In bridge rehabilitation projects, a work package for replacing deck pavement can include materials (e.g., asphalt), labour, paving machines, and drawings (e.g., the original and new pavement designs).

AWP includes constraint modelling, monitoring and analysis, and removal. Constraint modelling is a prerequisite of AWP, which creates work packages, identifies constraint entities, establishes relations among entities and packages, and form knowledge bases (KBs). Constraint modelling requires timely extracting constraint entities and relations from texts. However, this is manually performed in practice, which cannot meet the demands of AWP. Some studies to-some-extent automate the process. However, they focus on entity extraction and largely ignore relation extraction. State-of-the-art deep learning (DL) models (e.g., Knowledge Representation Learning (KRL)) models can extract relations. Such models are designed for general knowledge (e.g., Jobs-founder-Apple). They do not use domain knowledge (e.g., domain classes and relations among the classes) that can restrict data semantics (e.g., by disambiguation) and reduce data demands for training, making them impractical to be applied in bridge projects.

Efficient information integration is another prerequisite of AWP. Current approaches rely on relational databases which cannot effectively model interconnections among constraint entities (i.e., entity-relation-entity triples). Emerging graph databases (e.g., ontologies) are good at handling unstructured triples. However, ontologies in the area focus on static information (e.g., geometries) but cannot handle dynamic constraint information (e.g., tasks/procedures, constraints, their attributes), as their underlying syntax does not support required computation, e.g., traversing, iteration, and temporal

computation. Besides, most KBs are incomplete, which can be measured by the existence and correctness of searched information. This can damage AWP functions while the reasons include 1) it is very difficult to extract all information from texts, either manually or automatically, and 2) project documents may not cover all needed information themselves.

AWP and KB-based management have been applied in buildings and energy projects. However, bridge rehabilitation has unique challenges. First, there are different types of constraints, including underwater task devices, special materials (e.g., cables), and social constraints (e.g., traffic). Second, many participants are scattered in the design, construction, maintenance, and rehabilitation stage. Thus, additional efforts must be made to extract and integrate information, e.g., defining different domain classes and reasoning rules. Third, applications of information techniques do not cover managing bridge rehabilitation projects which still rely on manual management methods. Thus, similar to the bridge information modelling (BrIM) is a specialisation of building information modelling (BIM), novel methods are needed to modify and apply AWP and information management approaches in bridge rehabilitation projects, which can bring the most significant benefits and improvements.

This research develops a novel information extraction and integration approach for implementing AWP in bridge rehabilitation projects. It includes three components: 1) A hybrid DL model to extract constraint information from documents, where a bi-directional long-short-term memory and conditional random field (Bi-LSTM-CRF) model extracts constraint entities and a KRL model identifies relations (i.e., triples). Domain classes of entities are identified, while their representing vectors are stacked in the model to increase performance. The hybrid information extraction (IE) model reaches 0.936 and 0.891 F1 scores when extracting entities and relations respectively. Considering domain classes can increase relation extraction F1 score by 6.63%. The time for AWP constraint modelling is reduced to 1/29 of manual modelling. 2) Bridge rehabilitation management ontologies (BRMO), i.e., project KBs, which is built by comprehensively collecting domain knowledge. Its novelty lies in the combination of logic rules and an application programming interface to address syntax limitations in ontologies. The KBs can integrate static and dynamic constraint information for AWP management functions, e.g., evaluating task progress, constraint statuses, and project participants' performance. The BRMO reduces information searching time to 1/50 of

manual searching. 3) A knowledge base completion (KBC) model to predict missing information in KBs, which includes a data enriching module, a graph neural network (GNN) encoder, and a convolutional neural network (CNN) decoder. The enriching module adopts logic reasoning to increase data semantics. The encoder learns vectors of entities and relations using enriched data, and the decoder predicts missing triples. Domain-specific information (i.e., classes and working contexts of entities) are used in the encoder and decoder, respectively. The model reaches 0.844 hit@1. Enriching data and adding domain classes and working contexts gain 0.112, 0.277, and 0.129 additional hit@1, respectively. Besides, the model reduces the time for checking and completing KBs to 1/6-1/40 of manual methods while gaining higher accuracies.

The theoretical contribution is twofold. First, the BRMO overcomes syntax limitations in ontologies, enabling integration, updating, and searching of dynamic information in AWP KBs. Second, the research improves IE and KBC models by proposing ways to utilise domain information in DL models, which increases model performance. This research also has practical implications. First, the hybrid IE model partially automates constraint modelling, releasing engineers from intensive work. Second, the research expands the coverage of ontologies in the industry to bridge rehabilitation. Thus, the ontological KBs become a practical platform for different participants, which handle constraint information of both building construction and bridge rehabilitation while supporting management functions. Third, the KBC model can enrich KBs and further facilitate information searching and management. The approach improves practicality and usefulness of AWP with improved information extraction and integration. Much time for modelling and information searching can be saved, and more attention can be paid to constraint monitoring and removal thus contributing to project success.

Table of contents

List of figures

List of tables

List of abbreviations

AEC	Architectural, Engineering, and Construction
AI	Artificial Intelligence
API	Application Program Interface
AT	Attribute Entities
AWP	Advanced Work Packaging
BIM	Building Information Modelling

Bi-LSTM-CRF	Bi-directional LSTM-CRF
BMS	Bridge Management System
BrIM	Bridge Information Modelling
BRMO	Bridge Rehabilitation Management Ontology
CNN	Convolutional Neural Network
CONS	Constraint Entities
CRF	Conditional Random Field
CWP	Construction Work Package
DDBM	Data Driven Bridge Maintenance
DL	Description Logic
DNA	Dynamic Network Analysis
DOF	Degree of Overfitting
ERP	Enterprise Resource Planning
EWP	Engineering Work Package
F1	F1 score
FOL	First Order Logic
GA	Genetic algorithm
GIS	Geographic Information System
GNN	Graph-based Neural Network
GPS	Global position system
GRU	Gated Recurrent Network
ICTs	Information Communication Technologies
IE	Information Extraction
IFC	Industrial Foundation Class
IoT	Internet of Things
IWP	Installation Work Package
KB	Knowledge Base
KBC	Knowledge Base Completion
KRL	Knowledge Representation Learning
LiDAR	Light Detection and Ranging
LNG	Liquid Natural Gas
LPG	Labelled Property Graph
LPS	Last Planner System
LSTM	Long-short-term-memory
MR	Mean Rank
NER	Named Entity Recognition
NLP	Natural Language Processing
NN	Neural Network
NoSQL	Not Only Structured Query Language
O	Other Entities
PAC	Probably Approximately Correct
Pr	Precision
RDF	Resource Description Framework
RFID	Radio Frequency Identification
Re	Recall
SCM	Supply Chain Management
SN	Sensor Network
SNA	Social Network Analysis
SPARQL	SPARQL Protocol and RDF Query Language
SQL	Structured Query Language

SQWRL	Semantic Query-enhanced Rule Language
SWRL	Semantic Web Rule Language
SWT	Semantic Web Technologies
TP	Task/Procedure Entities
TransE	Translation Embedding Model
TransH	Translating Embeddings in Hyperplanes
TransR	Translating Embeddings in Relation Space
UAV(s)	Unmanned Aerial Vehicle(s)
VC	Vapnik-Chervonenkis
WFP	Work-face Planning
XML	eXtensible mark-up language

List of publications

A. Journal paper (* corresponding author)

1. **Wu, C.,** Wu, P., Wang, J., Jiang, R., Chen, M., & Wang, X*. (2021). Developing a hybrid approach to extract constraints related information for constraint management. Automation in Construction, 124, 103563.
2. **Wu, C.,** Wu, P., Wang, J., Jiang, R., Chen, M., & Wang, X*. (2020). Critical review of data-driven decision-making in bridge operation and maintenance. Structure and Infrastructure Engineering, 12, 1-24.
3. **Wu, C.,** Wu, P., Wang, J., Jiang, R., Chen, M., & Wang, X*. (2020). Ontological knowledge base for concrete bridge rehabilitation project management. Automation in Construction, 121, 103428.
4. **Wu, C.,** Wu, P., Jiang, R., Wang, J., & Wan, M*. (2020). Evaluating the economic and social benefits of multiutility tunnels with an agent-based simulation approach. Engineering Construction & Architectural Management, (ahead-of-print).
5. **Wu, C.*,** Chen, C., Jiang, R., Wu, P., Xu, B., & Wang, J. (2019). Understanding laborers' behavioural diversities in multinational construction projects using integrated simulation approach. Engineering construction & architectural management, 26(9), 2120-2146.
6. Li, X., **Wu, C.*,** Wu, P., Xiang, L. Q., Shen, G. Q., Vick, S., & Li, C. Z. (2019). SWP-enabled constraints modelling for on-site assembly process of prefabrication housing production. Journal of Cleaner Production, 239, 117991.
7. **Wu, C.*,** Xu, B., Mao, C., & Li, X. Overview of BIM maturity measurement tools. Journal of Information Technology in Construction (ITcon), 27, 34-62.

B. Conference paper

1. **Wu, C.,** Li, X., Li, S., & Xu, B. (2017). An Evaluation of Project Management System of Public Construction Sector in Shenzhen, China. Proceedings of the 20th International Symposium on Advancement of Construction Management and Real Estate. Hangzhou, China, 563-573.
2. **Wu, C.,** Jiang, R., & Li, X. (2016). Integration of BIM and computer simulations in modular construction, a case study. Modular and Offsite Construction (MOC) Summit Proceedings. Edmonton, Canada, 59-66.

Chapter 1: Introduction

1.1 Background

Bridges are critical infrastructures because they are links in transportation networks. Bridges are expected to serve about 50-100 years. However, the speed of bridge aging is increasing due to various reasons, e.g., the growing traffic volumes, high vehicle loading, and harsh environments (Lounis, 2007). Aging can make bridge structures deficient and unsafe, which has become a challenge in many countries. For instance, the average age of a bridge in the U.S. is 43 years, and 13% of them are structurally deficient (ASCE, 2017). In the UK, more than 2,000 bridges are not suitable to carry the heaviest vehicles and thus require major rehabilitation (WTW, 2018). Accordingly, governments around the world must make huge investments in bridge maintenance. For example, the Australian government has invested 480 million dollars for bridge renewal, with an ongoing commitment of $60 million/year (Infrastructure, 2018), and the U.S. government has raised 20.5 billion to repair highway bridges (OCIA, 2015).

Bridge maintenance includes four stages: inspection, condition evaluation, decision-making, rehabilitation (Wu et al., 2020a). In the digital era, the value of information is recognised, and it is found that 75% of the participants involved in bridge projects believe that information is the key to successful bridge maintenance (Woldesenbet, 2014). Many studies employ information and communication technologies (ICTs) to assist bridge maintenance, e.g., sensor-based real-time structure health monitoring as well as accurate structure condition evaluation and optimal maintenance decision-making based on advanced algorithms (Woldesenbet, 2014). In addition, bridge maintenance can involve many stakeholders who use isolated databases, which can result in the ‘data island’ problem. To address the problem, existing efforts are expanding and refining data schemas (e.g., mark-up languages and the industrial

foundation class (IFC) schema) to better integrate bridge data. Bridge Management System (BMS) and Bridge Information Modelling (BrIM), an extension of Building Information Modelling (BIM), have also been adopted to facilitate data storage and sharing. Those studies are referred to as data-driven bridge maintenance (DDBM) studies In this research (Sabatino et al., 2016).

Most maintenance resources are spent in the rehabilitation stage. Thus, the success of bridge rehabilitation projects largely determines the success of bridge maintenance programs (Wu et al., 2020a). However, it is difficult to manage bridge rehabilitation projects. First, bridge rehabilitation often faces complex constraints. Constraints are things that can prevent work from being smoothly executed, e.g., labour, materials, equipment, and permits. Work can be delayed or rework can happen if constraints are not timely removed (Şimşit et al., 2014). Bridge rehabilitation can encounter different and usually more constraints than conventional vertical building projects (especially when rehabilitation is performed on large river-crosses). Bridge rehabilitation needs more special resources (e.g., cables and equipment for underwater tasks) in addition to common resources (e.g., concrete, steel, and mechanical, electrical, and plumbing systems) in building projects. Some resources may need to be procured remotely from other counties where delays are more likely to happen (Wang, 2018). Second, bridge rehabilitation has more participants from different backgrounds, e.g., specialised suppliers, inspection teams, original and maintenance design teams, and external authorities (e.g., transportation department (DoTs) and municipal bureaus) which grant permits (e.g., the traffic control and water protection permits). The complex network of participants requires more efficient information integration and exchange for management (Woldesenbet, 2014). Thus, the approaches to extract, integrate, and analyse information must be modified to accommodate the differences, e.g., defining different domain classes and reasoning rules according to specific relations among constraints and participants. Third, despite the increasing number of DDBM studies, one gap is that most DDBM studies are restricted to pre-rehabilitation stages (i.e., inspection, condition evaluation, decision-making) of bridge maintenance. This can cause poor information management in that stage and affect the effectiveness of management methods. Finally, to ensure smooth traffic, bridge rehabilitation projects usually have tight schedules, making constraint management challenging, since any delay of constraint removal can cause project delay and more congestion. As such,

improving the management in bridge rehabilitation projects with ICTs can not only increase project profits by avoiding work interruptions but also bring social benefits, e.g., minimizing congestion.

Several modern constraint management approaches, e.g., Last Planner System (LPS), Workface Planning (WFP), and Advanced Working Packaging (AWP) can be applied to better manage constraints in bridge rehabilitation projects. The principle of these approaches is to ensure all constraints are removed before starting work. AWP gains increasing popularity recently, as it covers constraints in both the construction stage and early project stages. It has been recognised that AWP can significantly improve project quality, productivity, and predictability (Halala & Fayek, 2019). AWP has three steps: constraint modelling, constraint monitoring and analysis, and constraint removal. During constraint modelling, AWP breaks down the construction work into small manageable packages, identifies constraint entities (i.e., constraints, constraints' attributes, and tasks/procedures), and models relations among the entities (CII, 2013a, 2013b, 2020). For instance, 'water-reducing agent' and 'concrete' are two entities, and there is likely a constraining relation between them, i.e., 'water reducing agent constrains concrete' if the two entities appear in consecutive texts in one document (e.g., a working plan or technical specification), because water reducing agent can affect the performance of concrete. A work package cannot be released until all constraints linked to it are removed. These entities and relations form numerous entity-relation-entity triples, and many triples form a large graph which is a project knowledge base (KB). The KB enables various management functions in AWP, e.g., information searching and decision-making using graph analysis (e.g., finding critical constraints and tasks) (Fayek & Peng, 2013). In other words, these triples describe complex interconnections among constraint entities and are bottom-level components for implementing AWP.

Nevertheless, to realise all three steps (constraint modelling, monitoring/analysis, and removal), AWP should satisfy two prerequisites: efficient constraint modelling and information integration. AWP modelling can be required weekly, whereas constraint information often varies when a project proceeds. Thus, constraint information must be quickly extracted, modelled, and updated. Unfortunately, current AWP modelling is manually performed, where all entities and triples are extracted by humans. Besides, different stakeholders are supposed to remove different constraints by collaboration

and communication in bridge rehabilitation projects. Efficient information integration should allow them to access and exchange information in AWP graphs (i.e., KBs). Hence, advanced information collection and integration methods should be proposed to support the functions, which belong to the field of information management.

Information management in bridge maintenance projects has four critical steps: data collection and conversion, information integration and sharing, information analysis (to obtain certain assessing criteria), and decision-making (Venkatraman, 1997). Figure 1-1 shows the steps and their relationships (steps that this research focuses on are highlighted using bold borders). These steps have been significantly improved by existing DDBM studies.

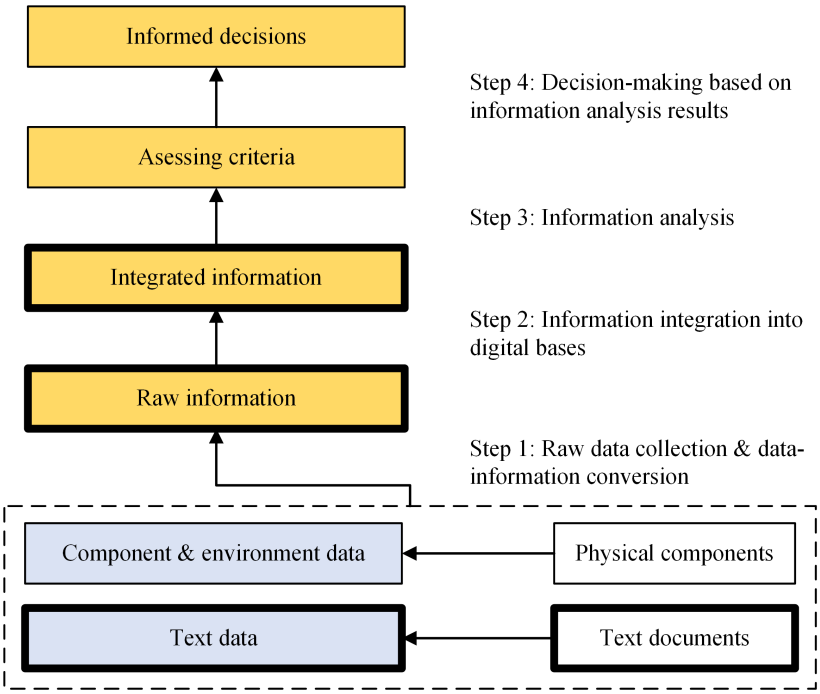


Figure 1-1 Information management steps

The first step gathers raw data and converts data into information that is useful for AWP (Wu et al., 2020a). There are two data types in the architecture, engineering, and construction (AEC) industry, i.e., structured data of components (e.g., geometries of a bridge pier) and environment (e.g., temperature and humidity) and unstructured data (e.g., texts in working plans and standards). Structure data can be conveniently stored in tables (e.g., rows and columns), which are collected through manual inspection and monitoring devices (e.g., non-destruction techniques (NDTs) and sensors). On the other hand, unstructured text data can be extracted by information extraction (IE) approaches that identify valuable information from free-written texts by extracting

entities and setting-up relations among entities (i.e., relation extraction). Although the term 'relation extraction' is widely used in studies, it is entity-relation-entity triples that are extracted. As such, the terms 'relation extraction' and 'triple extraction' are used interchangeably and mean the same concept in this thesis.

In the industry, the current focus is on collecting structured data. Many studies are conducted to develop advanced sensors, sensor networks (SN), and NDTs. Given the development of Natural Language Processing (NLP) methods, an increasing number of studies begin to develop text IE approaches for extracting entities and relations among the entities. However, these approaches are inadequate and do not satisfy the demands of AWP, as they cannot extract both entities and relations. Instead, current studies focus on entity extraction and ignore relation extraction (especially semantic-rich ones). Semantic-rich relations indicate that a relation between entities contains specific meanings in the AWP domain, e.g., 'asphalt-has-attribute-500m2'. This is opposed to simple relations (e.g., 'related-to' or 'similar-to'). Extracting semantic-rich relations is challenging, as 1) it requires accurate entity extraction as a pre-requisite, where relation extraction relies on recognizing valid triples from candidate triples formed by extracted entities; 2) it should handle ambiguity and noise of entities, where entities with different names imply the same relation, e.g., the entities 'asphalt' and 'paving material' are the same constraint and they both have a 'has-attribute' relation with the attribute '500m2'; 3) it is difficult to collect enough training data if ML/DL models are used for the task (Jiang et al., 2020).

In addition, current IE in the industry depends on complex rules which extract information when the rules are matched in texts. Developing rules manually is very time-consuming. More importantly, rules are only applicable to certain data, once the data is changed, the performance can drop drastically. In recent years, a few machine learning (ML) models, e.g., the Hidden Markov Model (HMM) and Conditional Random Field (CRF) model, and deep learning (DL) models, e.g., the bi-direction long-short term memory and conditional random field (Bi-LSTM-CRF) model, are proposed for entity extraction and are applied in several scenarios, e.g., extracting structure conditions in bridge inspection reports (Liu & El-Gohary, 2017a) and task dependencies in quality codes (Zhong et al., 2020b). DL models are found to be more accurate and practical, as 1) they can capture richer semantic features hidden in texts than ML models (e.g., CRF), thus gaining better extraction performance, and 2) they

do not require manually designed features but can extract features automatically, largely reducing the training time and difficulty (Murphy, 2012). As for relation extraction, most studies in the industry use rule matching. However, such methods either can only extract relations with simple semantics or heavily rely on handcrafted rules, making them subjective and impractical (Chi et al., 2019; Le & David, 2017; Wu et al., 2020b). On the contrary, state-of-the-art knowledge representation learning (KRL) models use DL structures (e.g., the convolutional neural network (CNN) and LSTM structure) which can capture triple features and extract semantic-rich relations. However, they are designed for general world knowledge (e.g., 'Tolkien-occupation-writer'), which require millions of entities and billions of triples for model training (Zhang et al., 2018b). They do not cover enough training data for AWP modelling in bridge rehabilitation, while not utilising domain knowledge in the construction sector to restrict data semantics and reduce training data demand. In addition, the industry lacks suitable ways to incorporate domain knowledge in DL models. Therefore, the models cannot reach high performance if being directly used for AWP modelling.

As for information integration and storage, most databases in the sector are relational databases that employ data tables and take table columns as keys. To store data of different types, many tables must be developed and linked using keys. Relational databases are very good at integrating structured data. However, entities and triples extracted from texts are unstructured data. Although there is a word 'relational' in the name, relational databases are not good at storing interconnections among entities (Medhi & Baruah, 2017). The word 'relational' refers to relating columns in a table, not relating data in different tables. The relationships among columns exist to support database operations, which is different from relations among entities (e.g., constraint entities). Hence, text data are often stored in .txt or .csv files in practice, making their retrieval and analysis difficult (Wu et al., 2020b).

The emerging technology, graph databases, can effectively handle data that involve many mutual relationships. Graph databases can be flexibly updated while knowledge facts (e.g., triples) are retrieved in real-time using specialised queries (Vukotic et al., 2014). Therefore, graph databases can implement AWP KBs and address the data integration issue. Typical graph databases include ontologies based on the resource description framework (RDF) and labelled property graph (LPG) databases (e.g., Neo4j). Ontologies can model triples and support semantic reasoning, which are used

to manage various construction information, e.g., geometries (Niknam & Karshenas, 2017), structure conditions (Ren et al., 2019), and structure defects (Park et al., 2013). Ontologies need more computer memory while do not support complex operations, such as iteration, enumeration, and temporal computation. For instance, identifying the constraint with the most severe removal delay requires traversing all constraints, comparing their removal progress, and returning the maximum value. Furthermore, constraint information can regularly change, which requires the above process to be repeatedly performed to update the ontologies. The computation and updating can be easily realised through object-oriented programming languages (e.g., Java), which however is very difficult (if not impossible) to be realised in ontologies due to the limitations of underlying syntax (i.e., Ontology Web Language (OWL)). In contrast, LPG databases are lighter thus can manage big data, however, the reasoning capacity is weak (Gong et al., 2018). Reasoning is important for management (e.g., finding potentially delayed work). Thus, ontologies are the better tool to integrate constraint information, but the gap of managing dynamic information needs to be bridged.

Another challenge for using graph-based KBs is that most KBs are incomplete. The main reason is that a project contains numerous constraint entities, and it is extremely difficult to extract all information (entities and relations) using existing IE methods. For one thing, manual extraction is inefficient and error-prone when human engineers lose focus. For another, although there are automated IE approaches (such as the one proposed in Section 3.4), they still make mistakes, e.g., missing triples or extracting irrelevant triples (Dettmers et al., 2017). Manually checking and completing KBs are impractical, and the industry lacks a computationally efficient method for knowledge base completion (KBC) and updating. However, it is difficult to realise automated KBC methods, as they (often adopting DL models) must effectively capture not only information in separated nodes/edges but also features and patterns of linkage and paths among all nodes and edges in KBs (Ji et al., 2020).

The last two steps of information management are information analysis and decision-making. Information analysis applies quantitative and qualitative methods to analyse information and obtain assessing criteria for decision-making. For instance, engineers employ mechanical models to compute or predict bridge structure condition ratings based on damage information (e.g., crack area and length); and project managers identify critical tasks and constraints by investigating the interconnections in AWP

graphs. Decision-making makes final judgments based on information analysis results (e.g., the assessing criteria). For instance, a buffer can be assigned to tasks whose constraints are not timely removed, while more attention can be paid to the delayed constraints to closely monitor their removal.

1.2 Problem statement

Given the growing number of constraints and complexity of bridge rehabilitation, the importance of constraint management should be recognised. Constraint management approaches, e.g., AWP, have been successfully adopted in different complex projects, e.g., maintenance and construction of natural and liquid natural gas (LNG) plants and modular buildings (Li et al., 2019; Wang et al., 2016). Case studies of such projects have shown that AWP is very effective, especially for projects with tight schedules and multiple participants (CII, 2020). Hence, AWP can contribute to the success of bridge rehabilitation projects.

However, in practice, there are two challenges for implementing AWP while fully reaping its benefits. First, efficient constraint modelling is a prerequisite of AWP, which requires timely extracting constraint entities and relations from documents. Unfortunately, due to the lack of IE methods that can extract both constraint entities and semantic-rich relations, information extraction and modelling are still manually performed and cannot meet the demands of AWP. Second, due to the poor ability to handle unstructured and dynamic project information using mainstream relational databases, constraint information cannot be effectively integrated and reused, which largely damages AWP management functions. Detailed problems of current AWP and IE in construction projects are summarised below.

1.2.1 Inadequate research attention for managing bridge rehabilitation projects

Bridge rehabilitation consumes most maintenance funding and resources, which is complex owing to a large number of constraints, multiple participants, and tight schedules. Implementing modern constraint management methods such as AWP can contribute to the success of such projects. However, studies of bridge rehabilitation concentrate on engineering techniques and approaches, while few efforts are made to improve the management aspect in such projects. On the other hand, implementing AWP must handle intensive information exchange among multiple project-level and

external participants (e.g., DoTs), which makes effective information management necessary. However, DDBM studies are restricted to the pre-rehabilitation stages of bridge maintenance, namely, collecting structured data, analysing information while computing assessing criteria, and optimising decision-making. This can be proved by the distribution of DDBM studies in Figure 1-2, which covers 485 peer-reviewed articles (Wu et al., 2020a). Poor information management can hinder the effectiveness of AWP and hinder project success.

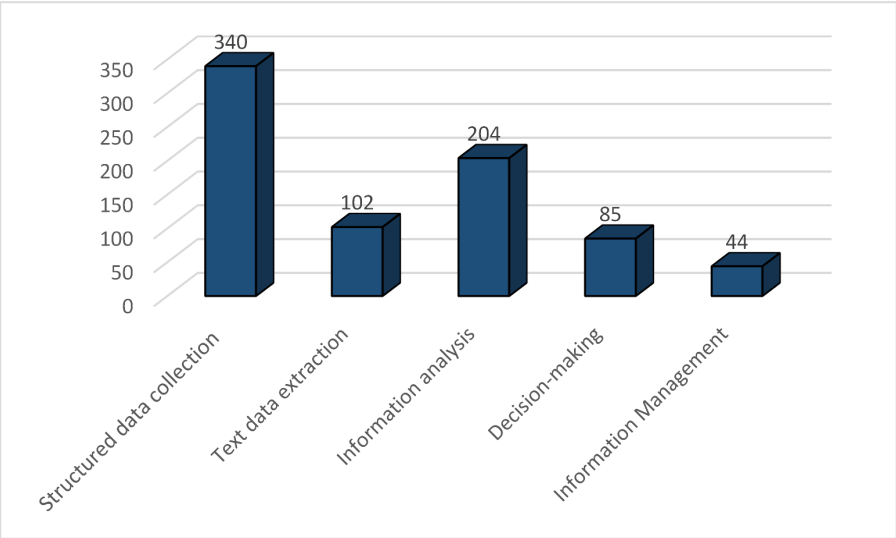


Figure 1-2 Distribution of existing DDBM studies (please note one study can cover one or more topics)

1.2.2 Inefficient AWP modelling

Developing AWP graphs through constraint modelling is the prerequisite of AWP. However, the current AWP relies on manually extracting constraints and establishing relations by reviewing documents and consulting project teams. Although engineers can perform constraint modelling based on their experience, it is very time-consuming and cannot meet the demands of AWP. The AWP in practical projects is iterative and repetitive, where constraint modelling can be required weekly (Li et al., 2019). Thus, manual modelling can result in very short windows for constraint monitoring and removal (Fayek & Peng, 2013). In construction projects, constraint information is buried in different types of documents (e.g., bill of quantities, working plans, and meeting records), which can worsen the situation (Wang et al., 2016). In addition, constraints commonly come from many backgrounds (e.g., mechanical, chemical, and engineering). As such, it is difficult for engineers (especially inexperienced ones) to identify all constraints and relations from the large volume of text data. Experienced

engineers can also make unpredictable mistakes when setting up complex relations, as a human easily loses concentration during the modelling task (Stallkamp et al., 2012).

1.2.3 Inefficient text information extraction

As mentioned, compared to extracting structured data, there are fewer efforts for extracting text data with NLP. This can make AWP modelling difficult, as constraint information (mainly constraint entities and semantic-rich relations among the entities) is usually buried in texts, e.g., working plans, meeting records, manuals, standards, and specifications (Hamdi, 2013; Wu et al., 2020b). Nevertheless, text data extraction is different from collecting component and environment data, and Table 2-1 lists the challenges, existing methods, and open issues.

Table 1-1 A summary of text data extraction in the construction industry

Text IE challenges	Existing solutions	Issues remaining open
1) Useful entities are hidden in texts. For instance, in a working plan, constraint entities are buried in irrelevant contents, such as the local policies and team organisations. 2) Entities can be ambiguous, as text documents are often written freely, and different expressions can be used for the same entity.	• Rule-based matching • ML models (e.g., CRF and HMM)	• Rules are often inflexible and difficult to be generalised to different projects • ML models require manually designed features, which is inefficient and impractical for AWP in real projects • ML models cannot gain high entity extraction performance due to their inability to handle noisy data
3) Relation extraction is difficult, as it requires extracting entities while interpreting dependencies among as well as specific and ambiguous meanings of entities.		• Rule-based relation extraction is inefficient and subjective, which is restricted to simple-semantic relations, e.g., the existence of relations and synonyms
4) Relations can be implicit. For example, in the sentence 'the supervisor checks safety belts of workers', there is an implicit 'constrains' relation between 'safe belts' and 'workers' as workers cannot start work until safety belts are provided. Current studies rely on hand-crafted rules,	• Rule-based matching • KRL models	• Existing KRL models cannot gain high performance if being directly used in AWP. They are designed for general KBs, which require enormous training data. They do not consider domain knowledge (e.g., entity classes) which can restrict data semantics so that the model is unlikely to be distracted by ambiguous entity names, thus reducing data demand. The reason

Text IE challenges	Existing solutions	Issues remaining open
		is that no effective methods exist that can integrate such domain knowledge into KRL for AWP. • There are inadequate triples about AWP for training KRL models.

1.2.4 Inefficient unstructured information integration

Information in bridge rehabilitation projects comes from multiple sources (e.g., file systems) which can be isolated. Hence, after being extracted, there should be suitable digital bases to integrate the information. AWP requires constraint information that takes the form of entity-relation-entity triples. Thus, the databases for AWP should serve as project KBs that can integrate, search for, and exchange such triple data. However, information integration approaches in the industry are again inadequate in the following aspects.

1.2.4.1 Lack of suitable project knowledge bases

Constraint triples are critical for AWP modelling but cannot be efficiently integrated into conventional relational databases for structured data. When unstructured triples are stored in a relational database, much more tables should be developed compared to that of storing structured data. Specifically, a triple can need two data tables and one key, while a data table only contains one data entry, which is very inefficient and requires much more computer memory. The drawbacks of relational databases are more evident during updating. For instance, when one wants to add a node (i.e., a constraint entity), multiple relations must be established between the added node and existing ones. As such, he/she must search for multiple tables to add keys. Moreover, data tables cannot intuitively represent links among entities, which makes information searching inefficient (Vukotic et al., 2014). On the other hand, graph databases are the more suitable tool for integrating constraint information. A graph database can be regarded as a project KB, where a relation between two entities is stored using only one triple, and data updating can be realised by simply adding, deleting, and modifying triples. Besides, graph KBs are highly intuitive information management tools, as information can be conveniently searched by navigating among nodes and edges.

AWP graphs have many triples which can change when the project proceeds. Thus, graph KBs (e.g., ontologies used in this research) are the better option to integrate constraint information. Unfortunately, ontologies in the AEC industry do not model

knowledge for AWP and bridge rehabilitation due to the lack of DDBM studies in the stage. Besides, most of them can only handle information of static objects and facts (Niknam & Karshenas, 2017; Zhang et al., 2015). In real-world projects, the change of one piece of information (e.g., the delay of removing one constraint) can cause the change of other information (e.g., task progress). Failing to capture such change can cause missing or erroneous information when information searching and affect AWP effectiveness. However, due to syntax limitations, current ontologies do not support important functions for handling such dynamic project information, e.g., traversing, iteration, and temporal computation.

1.2.4.2 Incomplete knowledge bases

Even information can be extracted and integrated into ontological KBs, the KBs are often incomplete, i.e., some constraint triples are missing. Incomplete KBs can hinder downstream management functions such as information searching and graph-based decision-making, e.g., identifying key constraints or constraints that can be delayed by analysing the topology (i.e., relations among and neighbourhoods of entities) of KBs.

Given the large and complex AWP graphs in real-world projects, manually checking and completing KBs is impractical, and the industry lacks a computationally efficient method for automated KBC. In the field of computer science, some KBC models can predict missing triples based on features of KBs' topology (Velickovic et al., 2017). However, similar to KRL models, KBC models cannot be directly used because 1) they are designed for completing general KBs (e.g., Wikipedia), and model training requires enormous data, making it impractical for practical project management, 2) they ignore the value of domain knowledge which can improve model performance and reduce training data demand by restricting data semantics, and 3) the industry lacks an effective mechanism to incorporate domain knowledge to improve general KBC models.

1.3 Scope and aim/objectives

To address the above issues, this research aims to develop an information extraction and integration approach which can develop complete KBs for bridge rehabilitation projects based on automatically extracted constraint entities and relations using novel ontologies and DL models. The approach can improve the implementation of AWP in bridge rehabilitation projects (especially concrete-reinforced bridges). The research

focuses on concrete-reinforced bridges because such bridges (e.g., cable-stayed and suspension bridges) are the most common bridge type (Wu et al., 2020a). To achieve the aim, four objectives are established.

Objective 1: To investigate topics, trends, and limitations of information management in bridge maintenance projects, implementation of AWP in the AEC industry, and information extraction and integration approaches.

The number of studies of bridge maintenance and information extraction/integration is large. Thus, a critical review will be conducted in the two areas. The articles will be collected from the Web of Science database. In contrast, package-based constraint management (i.e., WFP and AWP) is not well-studied currently. First, articles will be collected by searching the Web of Science database. Then, standards and case reports will be collected from online databases of the Construction Industry Institute (CII) and Constructions Owners Association of Alberta (COAA), the initiators and main implementors of AWP.

Objective 2: To develop a novel deep-learning-based information extraction model to automate AWP constraint modelling by extracting constraint entities and relations from text documents.

This thesis will propose a hybrid DL model, which applies a bi-directional long-short term memory and conditional random field (Bi-LSTM-CRF) model to extract entities and a knowledge representation learning (KRL) model to extract relations among the entities. As such, the hybrid model can simultaneously extract entities and semantic-rich relations, a significant IE challenge in the industry. Given bridge rehabilitation projects have specific constraints, the thesis will review manuals, standards, and case reports of both conventional building construction and concrete-reinforced bridge rehabilitation projects and then identify typical constraint types (i.e., domain classes) of bridge rehabilitation. A focus group will be organised to refine the initial findings. Constraint types provide important domain knowledge, based on which the capacity and performance of the KRL model can be improved.

Objective 3: To develop ontological knowledge bases to integrate the constraint information in bridge rehabilitation projects.

The thesis will develop bridge rehabilitation management ontologies (called BRMO) to integrate constraint information (i.e., information extracted by the hybrid model

proposed in Objective 2). The ontological knowledge bases (KBs) will be developed based on standard guidelines and domain knowledge collected in Objective 2. In addition, reasoning rules and a specialised Application Programming Interface (API) will be combined to overcome syntax limitations in conventional ontologies so that the ontological KBs can manage dynamic constraint information in ongoing projects.

Objective 4: To develop a novel knowledge base completion (KBC) model to automatically identity missing triples in AWP KBs.

To address the incompleteness problem in AWP graphs (i.e., the ontological KBs created in Objective 3), this thesis will develop a novel KBC model which consists of a data enriching module and an encoder-decoder structure. The relations for AWP modelling can have different levels of detail, which form hierarchies. To gain high relation extraction performance, the KRL model developed in Objective 3 can only extract top-level relations with abstract semantics (e.g., 'person constrains task_1'). However, more detailed relation types can express rich semantics and are important for training the KBC model, as KBC models predict missing triples by interpreting semantic information expressed in entities and relations. For instance, the 'constrains' relation in above example can be divided into 'works-in' and 'supervises' according to domain classes of the entities. As such, the data enriching module uses reasoning rules to convert simple triples to semantic-rich ones, e.g., 'crew_1 works-in task_1'. This helps the model better distinguish data and improves performance. In KBs for AWP, a node is a project entity that has a neighbourhood consisting of several nodes linked to the central one. All nodes and relations are represented by numerical vectors (called embeddings). The encoder of the KBC model computes new embeddings for each node by interpreting and integrating semantic information of all neighbour nodes and relations with the graph-based neural network (GNN). Then, the decoder predicts missing triples as follows: 1) takes new embeddings as inputs, 2) for each node in the KBs, identifies nodes to which it does not have relations, 3) traverses those nodes and pre-defined relations, forming candidate triples, 4) compute a validation score for each triple using a CNN structure similar to the KRL model, 5) establishes validate triples in the KBs. The KBC model is improved by adding domain information. Specifically, domain classes and working contexts of constraint entities are identified and utilised in the encoder and decoder, respectively (see Section 3.6 for more details).

1.4 Significance

It is difficult to manage concrete-reinforced bridge rehabilitation projects, thus, the projects are suitable for applying AWP and information management approaches. Unfortunately, information extraction and integration methods in the AEC industry cannot support effective constraint modelling and information integration, which can hinder AWP functions and project success. This research addresses these issues by developing an information management approach, where DL models are applied to automatically extract entities and relations, ontological KBs are created to integrate extracted information, and a KBC model is proposed to identify missing information and enrich the KBs. Accordingly, there are three main contributions.

(1) Improving current information extraction and knowledge base completion models

This research contributes to the knowledge body by proposing a novel hybrid DL model to extract unstructured constraint information and a KBC model to enrich project KBs. Previous AEC studies focus on entity extraction, whereas extracting semantic-rich relations is not well studied (Wu et al., 2021a; Wu et al., 2020a, 2021b). Besides, project KBs (i.e., AWP graphs) are often incomplete. The hybrid DL model can extract both entities and semantic-rich relations, and the proposed KBC model can automatically complete AWP KBs by identifying missing triples. These approaches are an early exploration in the industry. Furthermore, instead of simply using mature DL models, the research has computational novelty. It proposes ways to integrate domain knowledge into the structures of KRL and KBC models. Domain classes are stacked in the input end of the KRL model (more details are provided in Section 3.4), while both classes and working contexts of constraint entities are added in the neighbourhood of nodes to provide richer information for the GNN encoder (more details are provided in Section 3.6). The proposed ways are validated in experiments, and the results show that the performance of triple extraction (the KRL model) and missing triple prediction (the KBC model) is significantly increased when domain knowledge is added. This addresses the previously mentioned challenges of applying DL models designed for general knowledge. Thus, state-of-the-art KRL and KBC models can be enhanced by the research and become suitable for AWP.

(2) Improving information integration in bridge rehabilitation projects

The proposed Information integration approach has both theoretical and practical significance. Previous information integration efforts in the industry largely ignore unstructured information hidden in texts, which for AWP, refers to constraint triples. The lack of such integrated approaches can largely damage the value of IE tools and the usefulness of AWP, as extracted triples cannot be effectively integrated to build KBs and support management functions (e.g., information searching, reasoning, and decision-making). This research proposes ontological KBs (i.e., BRMO) to handle constraint triples for AWP. The BRMO is built based on a comprehensive collection of knowledge from relevant documents and experts in the bridge rehabilitation area while following standard procedures. Thus, it expands the scope of ontologies in the industry (focusing on buildings) to bridge rehabilitation. Unlike existing ontological KBs that are mainly applied to store static information (e.g., geometries and historical facts) (Niknam & Karshenas, 2017; Park et al., 2013; Ren et al., 2019). The novelty of BRMO lies in its ability to handle dynamic constraint information for AWP through combining reasoning rules and a specialised API to overcome syntax limitations in conventional ontologies. The BRMO can manage constraint information in ongoing projects and enables effective information searching, complex computation, dynamic updating, and semantic reasoning. Experiments and case scenarios have been used to prove the information capacity of BRMO (the details are provided in Section 5.4). With BRMO, project participants can effectively retrieve constraint information to perform essential management functions which can facilitate constraint monitoring and removal, i.e., the evaluation of project progress, constraint statuses, and project participants' performance (Wu et al., 2020b).

(3) Improving the management aspect of bridge rehabilitation projects and implementation of AWP by automatic constraint modelling

This research makes a practical contribution by providing an automatic constraint modelling tool for implementing AWP in bridge rehabilitation projects. AWP is an effective tool to manage bridge rehabilitation projects which usually have complex constraints. However, AWP is currently inefficient because its prerequisite, constraint modelling, still relies on manually extracting constraint information from texts. AWP modelling involves complex and semantic rich relations which cannot be extracted by current IE methods in the industry. This research proposes an information extraction and integration approach to largely automate constraint modelling, where a Bi-LSTM-

CRF model and a KRL model are combined to extract constraint information. All the information is integrated into ontological KBs, while the proposed KBC model can identify missing triples and enrich the KBs automatically and continuously. Given much constraint information of bridge rehabilitation is buried in text documents of different backgrounds, the proposed approach can significantly reduce the time for identifying and modelling constraints, checking and completing project KBs, and searching for relevant information. Thus, much time can be saved for downstream constraint monitoring, analysis, and removal, which makes AWP implementation in bridge rehabilitation projects more practical (details can be found in the experiments introduced in Section 4.5.3 and Section 6.3.3). The proposed approach also helps engineers (especially those lack experience) to understand interconnections among constraints and facilitate decision-making (details can be found in the experiments introduced in Sections 5.4.1-5.4.3).

1.5 Thesis structure

This thesis has seven chapters which are summarised below and in Figure 1-3.

Chapter 1 describes the background, research problems, aim and objectives of this thesis, as well as the thesis structure.

Chapter 2 summarises the literature on bridge maintenance, constraint management in the AEC industry, and information extraction and integration approaches (i.e., working mechanisms and applications of IE approaches, ontological KBs, and KBC models).

Chapter 3 introduces the research methodology. It outlines the research philosophy that underpins research methods first. Then, the chapter introduces the method for developing the hybrid IE model for entity and triple extraction, the method for constructing the ontological KBs, and the method for developing the KBC model.

Chapter 4 develops a hybrid IE model for constraint information extraction and automatic AWP constraint modelling. The model includes a Bi-LSTM-CRF model that extracts constraint entities (i.e., constraints, their attributes, and tasks) and a KRL model that identifies valid triples (i.e., relations) among entities. Domain classes are added to the model structure to improve performance. Detailed experiment results are summarised to compare the hybrid IE model with classical ML models in terms of

entity extraction while showing the effect of using domain information in the model structure for extracting semantic-rich relations.

Chapter 5 constructs ontological KBs to integrate extracted constraint entities and relations (i.e., entity-relation-entity triples). The development of the ontologies relies on a widely adopted guideline and comprehensive collection of domain knowledge. Meanwhile, semantic rules and a specialised API are applied to enable the ontologies to support constraint information searching, complex information computation and updating, as well as implicit information reasoning.

Chapter 6 develops a KBC model to identify missing triples in ontological KBs. The model consists of a data enriching module to improve data semantics, a GNN-based encoder to compute new embeddings for KB entities, and a CNN-based decoder to predict missing triples using the embeddings. Domain information (i.e., classes and working contexts) is utilised to improve model performance. Experiment results are investigated to demonstrate the model performance and effect of utilising domain information.

Chapter 7 concludes important findings in the thesis, highlights contributions and implications, discusses limitations in this research, and suggests future studies.

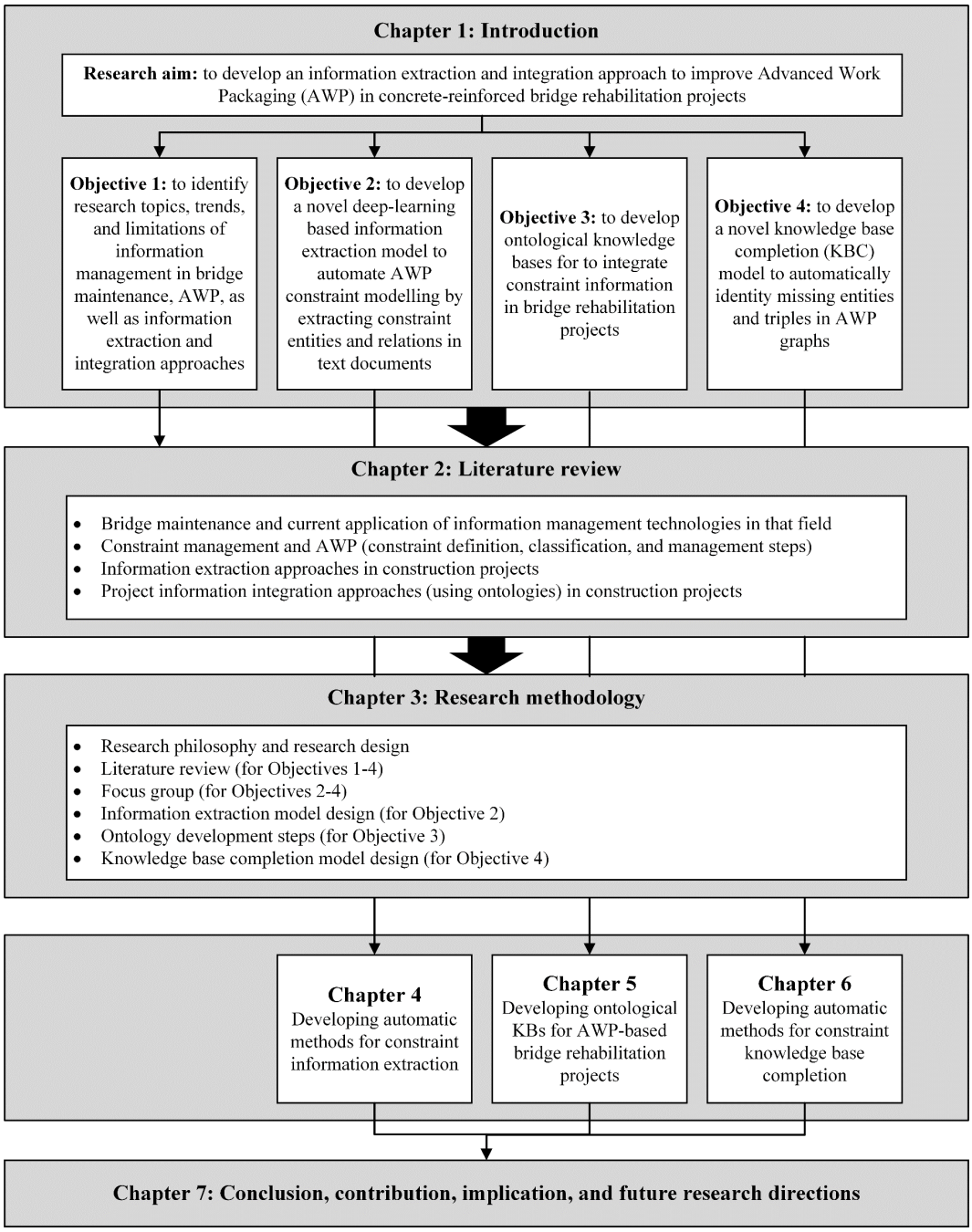


Figure 1-3 Thesis structure

Chapter 2: Literature review

2.1 Bridge maintenance

This section summarises the main stages of bridge maintenance, reviews efforts of applying information technologies for managing bridge maintenance projects (i.e., the DDBM studies), and explains the reason that bridge rehabilitation is selected as the implementation context of the proposed information management approach.

2.1.1 Bridge maintenance in different stages

2.1.1.1 Inspection

In this stage, data of bridge statuses are monitored and collected. Bridge inspection is an iterative process and can have different requirements subject to inspection details. Routine visual inspection should be performed periodically, while in-depth inspection and damage inspection are performed when suspicious damage is identified during a routine inspection or when damage happens, respectively (MLIT, 2015). For in-depth inspection, element-wise inspection is implemented by an increasing number of DoTs around the world, which will inspect each bridge element (e.g., a component and a group of components) (AASHTO, 2010). In addition, ad-hoc inspection is often performed after unusual events (e.g., earthquake or flood) (NCHRP, 2007).

2.1.1.2 Condition evaluation

Damage can happen suddenly, e.g., being damaged by a heavy vehicle, and gradually, e.g., being damaged by corrosion and repetitive loading. Damage can take various forms relying on materials meanwhile can happen on the surface of or inside bridge components. Common damage types include steel bar corrosion, cracks on concrete and steel, as well as concrete spalling, stain, honeycomb, and delamination (Gul et al., 2015; Turkan et al., 2018).

Condition evaluation estimates condition indexes of bridge components, considering different damage types and severity. A condition index can take many forms. The simplest index is a binary index, which indicates if damage happens or if the structure is out of service. The discrete rating (e.g., 0-9) is the most common index, where a lower number indicates more severe damage. Some bridge components (e.g., critical components) can need more sophisticated evaluation than discrete ratings. As such, more complex numerical indexes can be applied, such as 1) the failure probability which reflects the probability that gradual deterioration of a component exceeds its limit, 2) reliability which reflects the probability that a component does not fail before a specific time, 3) severability which reflects if the bridge can serve users normally, considering both structure safety and configurations (Thompson, 2012), and 4) durability which reflects the capability of a bridge component in terms of resisting deterioration (Anoop et al., 2012).

2.1.1.3 Decision-making

In this stage, maintenance decisions are made based on condition evaluation results, i.e., the condition indexes. The decision can include whether, when, and what tasks should be taken to restore a bridge, which can be divided into bridge-level and network-level decisions.

Bridge-level decisions determine bridge components and maintenance tasks that will be performed on the components. The aims of decision-making include 1) minimising life-cycle maintenance costs and delay of maintenance tasks before the failure of components, and 2) maximising structure conditions (Kabir et al., 2014). Decisions are made by assigning maintenance tasks to bridge components according to structure conditions and costs (including the costs for maintenance tasks and the social costs caused by structure failures, such as congestion and human life loss) (Sabatino et al., 2016). The importance of components can be further considered to ensure that critical components are treated first (Bolar et al., 2014). Network-level decision-making needs to prioritise multiple bridges in the transportation network and then schedule maintenance tasks for them. Goals for such decisions include minimising the overall costs and delay of maintaining all bridges and maximising the performance of the entire network, e.g., reducing total travelling time and distance as well as increasing network connectivity (Bocchini & Frangopol, 2013; Frangopol & Bocchini, 2012).

2.1.1.4 Rehabilitation

In this stage, restoration actions are performed following the decisions made. In this research, rehabilitation includes hazard treating, reinforcement, and replacement. Hazard treating fixes different types of damage. Reinforcement increases the load-carrying capacity of the bridge structure by adding materials or components, e.g., external prestressing. Replacement substitutes those severely damaged components. Rehabilitation tasks can be preventive or essential. Preventive rehabilitation tasks are scheduled regularly before structure failure. Preventive tasks can be proactive and reactive, which are performed before and after damage happens, respectively. On the other hand, essential rehabilitation tasks are carried out after structure failure, where the failed components are replaced or repaired completely (Okasha & Frangopol, 2010). Current research on bridge rehabilitation focuses on improving engineering approaches and techniques. For instance, many studies intend to propose or improve engineering techniques of rehabilitation, such as the confinement technique that

restores concrete components after an earthquake (Ma et al., 2017) and the grouted splice sleeve that repairs damaged pre-cast concrete columns (Parks et al., 2016). Second, some studies invent or apply materials for rehabilitation tasks. For instance, carbon fibre reinforced polymer is used to repair concrete corrosion (Xie & Hu, 2013), and ultra-high-performance fibre reinforced concrete is applied to strength bridge decks (Bastien-Masse & Bruhwiler, 2014). Finally, there are also some studies aiming to improve the working procedures of bridge rehabilitation. For instance, Phares and Cronin (2015) proposed the accelerated bridge construction approach to reduce rehabilitation duration.

2.1.2 Information management in bridge maintenance

As introduced, information management includes four important steps: data collection and conversion, information integration, information analysis, and decision-making. Accordingly, many DDBM studies are conducted to improve these steps, which can in turn improve the four bridge maintenance stages.

2.1.2.1 Data collection

DDBM studies that fall in this aspect focus on improving bridge inspection by collecting data of bridge components and the surrounding environment in a real-time manner. These studies rely on applying SN (both wired and wireless) and NDTs. SN can accommodate various types of sensors (e.g., strain gauges, accelerators, global position system (GPS), thermometers, temperature/humidity meters, and weigh-in-motion systems), which can continuously collect data whereas no inspectors need to be sent onsite. On the other hand, NDTs (e.g., terrestrial laser scanner, light detection and ranging, infrared cameras, underwater sonars, and ground penetration radar) can accurately detect surface and sub-surface damage. Both SN and NDTs can perform bridge inspection without closing the bridge and disturbing the traffic. Some modern NDTs, e.g., unmanned aerial vehicles (UAV), can collect data at blind points which are difficult for human inspectors to reach (TxDOT, 2020).

There are also studies extracting text data from bridge maintenance documents. IE methods proposed in those studies are often applied to inspection reports thus can improve condition evaluation. For instance, Liu and El-Gohary (2017a) extracted bridge components and their condition ratings from inspection reports. The same researchers proposed the IE approaches 1) to identify dependency paths of sentences in inspection reports (Liu & El-Gohary, 2017b) and 2) to extract bridge component

deficiencies, deficiency causes, and maintenance actions through an ontology-based sequence labelling method (Liu & El-Gohary, 2016). However, as mentioned, these IE methods can only extract entities rather than semantic-rich relations. Thus, they cannot be used for automating AWP modelling that relies on both constraint entities and relations.

2.1.2.2 Information integration

Information integration aims to solve the 'data island' problem. Current studies of information integration for bridge maintenance focus on expanding data schemas and developing collaboration platforms. The most widely studied schema is the eXtensible mark-up language (XML). XML can describe structured and unstructured data in a standard and interoperable way and can be extended to specific tasks (Zhu et al., 2020). For instance, Jeong et al. (2016) developed the sensorML schema to improve sensor data storage, and Jeong et al. (2017) applied the openBrIM schema to record bridge conditions in inspection reports. In the last decade, Industrial Foundation Class (IFC) which is developed for vertical buildings, has been borrowed in the bridge sector. Many efforts are made to extend the IFC schema for bridges. Current IFC can encode various bridge maintenance information (e.g., bridge alignments, geometries, and structure conditions) while can be converted to XML (ifcXML) (Huthwohl et al., 2018; Zhang et al., 2016). Data encoded by a data schema must be stored in databases for retrieval and exchange. Most databases for bridge maintenance are relational databases, e.g., Oracle and DB2. To manage the massive bridge data collected by various ICTs, some studies adopt distributed databases, where separated databases are used to handle different data formats (Miyamoto & Asano, 2017; Zhang et al., 2016). Other studies employ NoSQL (Not Only Structured Query Language) databases, e.g., MongoDB and Apache Cassandra, which have better scalability when handling big data (Jeong et al., 2017, 2019). However, current work for developing data schemas and databases focus on structured data (e.g., geometries and sensor readings) but cannot effectively handle unstructured data (e.g., knowledge in texts) (Morgenthal et al., 2019). As such, a few studies implement graph databases (especially ontologies) (Costin et al., 2018; Wu et al., 2020a). For instance, Liu and El-Gohary (2017a) built the BridgeOnto to increase the efficiency of searching for maintenance history in BMS; Wu et al. (2020a) created an ontology to manage constraints in bridge rehabilitation projects. The DoTs in Netherland adopted an ontology-based management system to integrate asset data

and disambiguate information among participants (Luiten et al., 2018; NRA, 2018). However, as mentioned in Chapter one, the ontologies do not cover knowledge of bridge rehabilitation and cannot handle dynamic constraint information, which limit their usefulness in AWP for bridge rehabilitation.

To better enable collaboration among participants, many information collaboration platforms are developed to provide easy-to-use user interfaces for all participants to access data in the databases without the knowledge of database operation (Kuckartz & Collier, 2016). Many collaboration platforms also encapsulate the functions of information analysis and decision-making. In bridge maintenance, a BMS is the most common type of collaboration platform. In a BMS, separated modules are created for different functions, such as damage estimation, decision-making, and visualisation, meanwhile, the data needs, levels of detail, and data flows are defined among the modules to enable data exchange within the BMS (Feltrin et al., 2010). Typical BMS examples include the PONTIS, BRIDGIT, and BrM (Hawk & Small, 1998). The J-BMS developed by Japan DoTs, which could integrate real-time monitoring data and support decision-making using an expert system, is also widely recognised (Miyamoto & Asano, 2018). Moreover, BIM and geographic information system (GIS) have also been adopted for managing bridge maintenance projects, where BIM is referred as BrIM. For instance, Shire et al. (2017) built a BrIM platform to model, store, and visualise modal information of cables in cable-stayed bridges; and Javadnejad et al. (2017) developed a GIS system to handle multi-layer image information for damage detection. Nevertheless, the bottom-level databases are the basis of collaboration platforms, whereas the above efforts are built upon normal relational databases which are only good at managing structured data.

2.1.2.3 Information analysis and decision-making

As for structure condition evaluation, binary and discrete indexes can be estimated by mapping the severity of damage to different ratings (e.g., severe damage is mapped to a low rating). Such simple indexes are automatically estimated using certain programs (e.g., expert systems), where the mapping rules are developed based on engineers' experience and guidelines (Miyamoto & Asano, 2018). On the other hand, more complex indexes (e.g., reliability and failure probability) are often estimated using mechanical models, e.g., the Paris law for fatigue damage and Fick's second law for corrosion, whereas statistical distributions (e.g., the Weibull and Gamma distribution)

are applied to handle uncertainties of variables in these models (van Noortwijk & Frangopol, 2004). Many studies aim to improve condition evaluation, and mainstream methods include 1) investigating the relations between bridge conditions and more affecting factors, e.g., environmental factors and various maintenance tasks, and 2) improving the mechanical models by modifying equations (e.g., embedding newly discovered factor-structure relations in previous equations) or applying more sophisticated statistical distributions (Sabatino et al., 2016). Finally, ML models and genetic algorithms (GA) are also increasingly applied to estimate condition indexes, which can either directly predict condition indexes based on damage information or predict key parameters for mechanical models (Wu et al., 2020a).

When it comes to decision-making, bridge maintenance goals are often contradictory (e.g., reducing costs and improving structure conditions). Therefore, many DDBM studies regard decision-making as a multi-objective optimisation problem. A common solution is to integrate multiple objective functions into a single index using weighted summing (e.g., sustainability) for decision-making (Lounis & McAllister, 2016). Besides, many optimisation techniques, e.g., grid searching (Gong & Frangopol, 2020), decision-tree and event tree (Orcesi & Frangopol, 2011), linear, and dynamic programming (Liu & Madanat, 2015), GA algorithms (Okasha & Frangopol, 2009), and DL models (Wei et al., 2020), can be applied to find optimal decisions.

To this end, two limitations of existing bridge maintenance studies are summarised. From the perspective of bridge maintenance steps, current studies focus on the three pre-rehabilitation stages (i.e., inspection, condition evaluation, and decision-making). Although managing bridge rehabilitation projects can be challenging owing to the complex constraints, strict schedule requirements, and scattered information caused by the involvement of multiple project parties, few studies address the management aspect of bridge rehabilitation. From an information management perspective, ICTs have great potential to improve bridge maintenance. However, DDBM studies are limited to the pre-rehabilitation stages and collection of structured data, information analysis, and decision-making. Research efforts for extracting and integrating data from unstructured texts are inadequate. Meanwhile, modern constraint management approaches (e.g., AWP) can assist in managing bridge rehabilitation projects (more details are introduced in the next section), but they require effective extraction and integration of constraint information so that engineers can timely remove constraints

or adjust working plans to handle unremoved constraints. Such constraint information 1) takes the form of triples hence is unstructured, 2) can be scattered in documents in different project stages. Therefore, such information cannot be managed by current information management approaches in the industry. Despite that SWT approaches, e.g., ontologies, are gradually applied for bridge maintenance, for one thing, they do not cover domain knowledge of bridge rehabilitation thus cannot be directly applied. For another, ontologies in the sector focus on static information (e.g., geometries of components). Due to syntax limitations in terms of performing complex computation and dynamic updating, e.g., iteration, enumeration, and temporal computation, it is difficult for them to handle dynamic constraint information in ongoing projects.

This research proposes a hybrid IE approach for unstructured information extraction and develops an enhanced ontological knowledge base for integrating constraint information. The approach can to-some-extent address the information management problem and facilitate research and applications of modern constraint management approaches (e.g., AWP) in bridge rehabilitation projects which are selected as the context to implement and demonstrate the proposed approach.

2.2 Constraint management and AWP

2.2.1 Constraint definition

A constraint has different definitions in different sectors. For instance, in mathematic research, a constraint is a condition that a solution of an optimisation problem must satisfy; in the information theory, a constraint reflects the degree of statistical dependence among variables; studies of classical mechanics regard constraints as relationships between coordinates and momenta; and business managers believe that constraints are anything that prevents a system from achieving its objectives (Watson et al., 2007). The key concept of constraints in this research is derived from the lean concept. There are three dominant definitions of constraints in the AEC industry, which are based on three main constraint management approaches, i.e., LPS, WFP, and AWP, respectively. In LPS, a constraint is defined as anything that stands in the way of a task being executable or sound (LCI, 2007). In WFP, constraints are things that a foreman or supervisor needs to execute onsite construction work (Fayek & Peng, 2013). In AWP, constraints are any prerequisite items that can prevent or delay the smooth execution of work (CII, 2013a). The three definitions share similar meanings.

Given this research focuses on AWP, constraints are defined as anything that prevents work packages from being successfully or smoothly executed in the construction field.

2.2.2 Constraint classification

In the AEC industry, there is currently no standard classification of constraints, and many informal classifications exist. For instance, Ballard (2000) grouped constraints into eight types in LPS: information, preceding work, labour, space, material, funds, equipment, and external factors. Choo et al. (1998) defined six constraint types: contracts, engineering items, materials, human resources, equipment, and prerequisite tasks. Dawood and Sriprasert (2006) developed four classes of constraints: physical constraints (e.g., space, safety, technologies, and environment), contract constraints (e.g., progress, costs, and quality), resource constraints (i.e., availability, continuity, capacity, and perfection); and information constraints (e.g. accuracy and clarity). Chua et al. (2003) proposed three constraint types: precedence constraints that decide the starting/ending and sequences of work, resource constraints (e.g., materials, labour, and equipment), and information constraints (i.e., the information needed in tasks, e.g., shop drawings, specifications, and approvals).

The above studies may not cover all constraint types (e.g., weather and authority permits are not included in the study of (Choo et al., 1998; Chua et al., 2003)). Besides, some studies do not have well-structured hierarchies to organise constraints, e.g., the hierarchies can be vague or ambiguous (Dawood & Sriprasert, 2006). On the other hand, WFP and AWP break down construction work into different types of work packages, identify constraint entities of packages, and propose clear and flexible hierarchies to organise packages and constraints. WFP adopts three package types with an increasing level of detail: construction work areas (e.g., high-level sequences of areas to be built), construction work packages (mid-level tasks and constraints), and field installation packages (bottom-level procedures and constraints). AWP also has three package types, i.e., construction work packages (CWP) (e.g., the general logic of construction work), engineering work packages (EWP) (engineering requirements), and installation work packages (IWP) which are backlogs that one crew can finish in a safe, measurable, and efficient manner (CII, 2013a, 2020). A CWP can include multiple EWPs and is the basis to develop IWPs. CWP and EWP can constrain the release of IWP, while IWPs can constrain one another (Halala, 2018). Based on the ideas of AWP, Wang et al. (2016) proposed a constraint management framework for

maintenance projects of LNG plants, including three constraint types: engineering constraints (e.g., drawings and permits), supply chain constraints (i.e., resources that must be procured and delivered offsite), and site constraints (e.g., labour, temporary facilities, weather, and preceding tasks). Each type can be divided into more detailed types so that constraints at different levels can be organised in hierarchies. Results in that research can serve as an effective approach to identify and organise constraints. Specifically, when the general work sequence (i.e., CWP) is determined, engineering constraints and site constraints can be organised using EWP and IWP, respectively. Figure 2-1 presents a simple example of AWP graph which only includes one work package of each type.

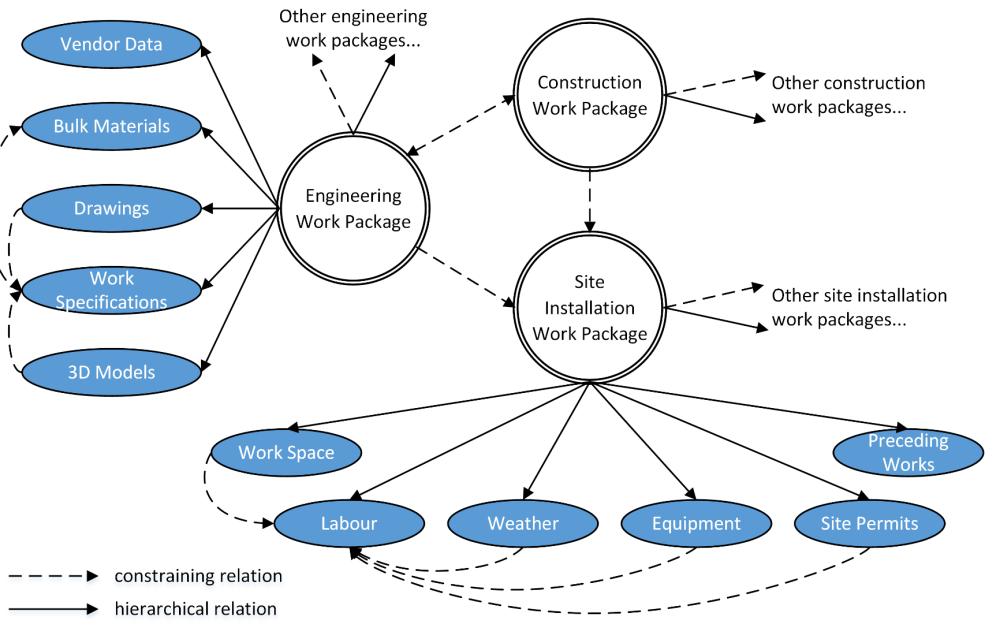


Figure 2-1 An example of AWP graph

2.2.3 Constraint management steps

Constraint management approaches, such as LPS, WFP, and AWP, aim to ensure all constraints of a construction task are removed before the task is carried out. There are three constraint management steps: constraint identification and modelling, constraint monitoring and analysis, and constraint removal (LCI, 2007; Wang, 2018).

(1) Constraint identification and modelling

Constraint identification discovers constraints and organises them in well-defined hierarchies. LPS identifies constraints in look-ahead plans which usually cover 3-12 weeks of work. Package-based approaches (e.g., WFP and AWP) first develop work packages and then identify constraints of each package. Work packages can also be

constraints, as constraints in one package can affect constraints in other packages, and packages should be released in a sequence. Constraint modelling aims to understand and describe the relations among constraints. Seven types of relations are investigated: 1) Relations among constraints, e.g., the delayed removal of a constraint (e.g., safety belts) can delay the removal of another constraint (e.g., labour). 2) Relations between constraints and tasks/procedures. 3) Relations between constraints and attributes (e.g., the amount and price of material constraints). 4) Work dependencies, including the task or procedure sequences and 'part-of' relations between tasks and procedures. Specifically, a bridge rehabilitation task can consist of several procedures which can be adopted in other projects. For instance, a deck paving task includes paving and rolling procedures, while rolling and paving are common procedures in road projects. 5) Relations that connect constraints and tasks/procedures to packages. 6) Relations among work packages (e.g., the releasing sequences and hierarchies of packages). 7) Relations between tasks/procedures/constraints and participants of the project (e.g., contractors and suppliers) who are responsible for removing constraints or managing tasks/procedures. The last three types of relations are only considered in WFP and AWP (Wang et al., 2016).

In package-based approaches (e.g., WFP and AWP). Constraint modelling produces a graph that captures interconnections among constraints. The graph works as a graph database (i.e., a KB), where critical information (e.g., the required amount of material constraints and removal progress of certain constraints) can be timely retrieved by graph queries and navigation. Constraint entities are described using nodes and are connected using the seven types of relations introduced before. Then, each node can be linked to its original data source out of the AWP graph (e.g., a drawing and a 3D model), regardless of the original data format. Therefore, an AWP graph provides a format neutral approach for information searching and retrieval, which can to-some-extent link the scattered information in bridge maintenance projects thus addressing the 'data island' problem (Halala, 2018; Hamdi, 2013).

(2) Constraint monitoring and analysis

Constraint monitoring tracks constraint statuses (e.g., if the removal of a constraint is delayed). There are two main types of monitoring in literature: monitoring resources (e.g., tracking material delivery) and monitoring task progress (e.g., if preceding tasks are delayed). Both monitoring tasks can be finished manually, which however can be

time-consuming and subjective, and the suboptimal results can affect management decision-making. Therefore, many ICTs are adopted to assist constraint monitoring. For tracking resource constraints, sensing technologies are extensively applied, such as barcode, Radio Frequency Identification (RFID), Bluetooth, Ultra-wide Bandwidth, wireless local area network, and GPS. The technologies have unique strengths and weaknesses. For instance, GPS is only applicable in an outdoor environment but can track targets in a medium- and long-range area. Other technologies, e.g., RFID and barcode are cheap and easy to use, however, they can only work in a restricted area and can be affected by the line-of-sight problem and environmental factors, e.g., the presence of metal that can affect the reading of radio waves. Thus, the technologies can be combined to supplement each other. Sensing technologies have been used in many projects to track the location and availability of resources in real-time (Wang, 2018; Wang et al., 2016). Meanwhile, laser scanning (e.g., LiDAR) and photogrammetry are emerging technologies to track task progress by quickly generating 3D models of construction products (e.g., building structures) and comparing as-built models with as-plan models (Turkan et al., 2018). Such image-based technologies are less applied compared to sensing technologies owing to the expensive investment in devices, strict requirements about environmental factors (e.g., light), and long time for training monitoring personnel (Puri & Turkan, 2020).

Studies of constraint analysis can be divided into three groups: mathematical analysis, pull-driven constraint analysis, as well as network (graph) based constraint analysis. Mathematical models focus on estimating the impact of constraints on a project, e.g., work progress, costs, task dependencies, and resources. Typical mathematical models include the Critical Path Method (Ottesen & Martin, 2019), Program Evaluation and Review Technique (Karabulut, 2017), and Line of Balance (Damci et al., 2013). Many studies regard constraint analysis as a multi-objective optimisation problem. Thus, linear/non-linear programming, configuration space optimisation, and GA algorithms are applied to reach a balance between constraints (Al Haj & El-Sayegh, 2015; Koo et al., 2015; Li et al., 2020). However, mathematical models cannot effectively handle constraints that are difficult to be quantitatively modelled, such as work quality and detailed requirement descriptions. In addition, mathematical models cannot describe hierarchies and complex interconnections among constraints.

Pull-driven methods intend to satisfy customer demands by producing the finished construction products as optimally as possible in terms of time, costs, quality. Pull-driven constraint management is extensively applied in practical LPS owing to its simplicity. In many cases, a spreadsheet is a popular tool for pull-driven methods, which records constraints for each task in a list (Nieto-Morote & Ruz-Vila, 2012). Most constraint types can be covered by pull-driven methods. However, the method (i.e., the list) is still too simple. It only models constraints in a tree structure (i.e., a look-head plan severs as the root node which has many tasks, and each task is connected to several constraints as bottom-level leaf-nodes) whereas cannot capture the interconnections among constraints (LCI, 2007). Finally, recent studies begin to manage constraints in a network or graph which can capture and analyse the complex constraint interconnections. In the AEC industry, two typical graph analysis methods are the Social Network Analysis (SNA) and Dynamic Network Analysis (DNA).

SNA is an effective and simple approach to characterise the structures and topology of a graph using certain indicators, e.g., betweenness, centrality and density (Streeter & Gillespie, 1993). SNA studies in the sector mainly investigate the roles of and relationships among participants. Some studies identify important participants and distinguish the networks formed by different participants (Wong et al., 2010). Other studies aim to optimise information and knowledge sharing among participants, such as analysing whether information can be effectively shared based on the connectivity of the network and identifying critical factors that affect collaboration (Farshchi & Brown, 2011). DNA to-some-extent covers SNA, however, is more sophisticated. Specifically, DNA can model different types of nodes (entities), relations, properties of nodes, and changes of the network over time. Thus, DNA can be used to analyse various types of network, such as social network (e.g., people to people), knowledge network (e.g., people to knowledge and resources), attendance network (e.g., people to events and tasks), information network (e.g., information sources to information sources), membership network (e.g., people to organisations), network of needs (e.g., resources to events/tasks), organisational capability (e.g., resources to organisations), temporal network (e.g., sequences of events/tasks), institutional support (e.g., events and tasks to organisations), and inter-organisational network (e.g., organisations to organisations). Compared to SNA, DNA is the more suitable tool for package-based constraint management, as it can capture all types of relations used in constraint

management. Thus, in some studies, DNA is adopted to develop the AWP graph and identify constraints having a large impact on projects (Wang, 2018; Wang et al., 2016).

(3) Constraint removal

A constraint can be regarded as removed when all constraints connected to it are removed. Hence, constraint removal depends on the results of constraint monitoring (i.e., current statuses of constraints) and is performed hierarchically. At the most general level, constraint removal concerns the main construction areas and sequences of construction. Meanwhile, key constraints (e.g., important engineering deliverables, general resource requirements, and long-head constraints that must be transported from remote areas) are determined and confirmed, whereas commitments of their removal are obtained from corresponding main participants (e.g., the owner, general contractors, and main design company). At the second level, all project participants (i.e., the main participants and specific suppliers, sub-contractors, and authorities) should be involved to identify new, more detailed, and short-ahead constraints based on the general plans and general constraint removal progress, identify issues, and make more detailed commitments. At the third level, detailed work assignments (e.g., IWPs in AWP) are developed for onsite tasks. Detailed constraint removal plans are created through collaboration with the site foremen. In this stage, it is critical to monitor the onsite progress of work and constraint removal and then provide timely feedback to engineers and managers of any delay. In addition, it is helpful to summarise issues and develop best practices to continuously improve constraint management in following tasks and projects (Wang et al., 2016).

Constraint removal should be performed periodically, where the frequency depends on the removal stage and project type. For example, the frequency can be quarterly (for large projects) or monthly (for smaller projects) at the early stage (the general level), which can be changed to weekly at the construction stage (the detailed levels). At any stage, different actions should be performed if a delay of constraint removal happens, e.g., adding a buffer to tasks/procedures and changing material usage (Wang et al., 2020). Finally, it is important to have a quantitative indicator to reflect constraint removal progress, so that engineers can easily compare actual constraint statuses with the plans. One option is to define a constraint maturity index that can range from 0-1. For instance, as a constraint, an EWP can be constrained by ten constraints, and removing each of them increases the maturity index of the EWP by 0.1. The increasing

amount of the index can be adjusted based on the importance of specific constraints. When the index reaches 1, the EWP can be regarded as removed (Wang, 2018).

Compared to other modern constraint management methods such as LPS and WFP, AWP is more complete. It covers both the construction stage and initial stages (e.g., design and procurement) hence can better control a project in its lifecycle and align more project participants responsible for different stages to remove constraints with joint efforts before starting work (CII, 2013b). Hence, this research adopts AWP to manage bridge rehabilitation projects while proposing novel IE methods to improve AWP. Based on the above review, it can be argued that current studies of AWP are restricted to the theory of constraints and work packages as well as the steps of constraint monitoring and removal (with the help of ICTs). However, as stated in Section 1.2.2, constraint modelling is still manually performed, which is inefficient and cannot meet the demands of practical AWP. As such, it is necessary to realise automatic constraint modelling to fully reap the benefits of AWP.

Automatic constraint modelling heavily relies on automatically extracting constraint information from documents. However, existing studies in this field are either not necessarily related to AWP or are restricted to certain types of constraints, such as quality (Zhang & El-Gohary, 2016), work dependencies (Zhong et al., 2020b), and spatial links (i.e., spatial constraints) of components (Xu & Cai, 2020). Some tools can model constraint entities and relations, e.g., BIM, Enterprise Resource Planning (ERP), and Supply Chain Management (SCM). However, ERP and SCM are applied at the organisational level and are difficult to cover bottom-level constraint entities (Spathis & Constantinides, 2003; Wei et al., 2005). These tools can manage information of separated constraint entities and relations between constraints and tasks/procedures (i.e., which constraints affect which tasks/procedures). However, they cannot capture complex interconnections among constraints (e.g., which constraints are affected by other constraints) (Gupta & Boyd, 2008). Moreover, initial information in the tools is still manually identified and inserted.

2.3 Information extraction in construction projects

IE tasks mainly concern entity and relation extraction, which can be realised by rule-based and ML-based approaches. This section introduces theories and/or background knowledge of logic and machine learning and reviews their applications for IE tasks.

2.3.1 Logic rules and reasoning

Logic is the foundation to represent knowledge, and the most common type of logic is the predicate logic (i.e., first-order logic (FoL)). Logic holds some distinct features:

- Logic provides a high-level language where knowledge can be expressed transparently.
- Logic features formal semantics therefore can assign unambiguous meanings of statements. Meanwhile, logic language can be well-understood by a human.
- There exist proof systems that can derive logic consequences syntactically from a set of premises. As such, one can trace the proof process that leads to a logical consequence. In this case, logic can provide explanations for answers, which is very important for knowledge acquisition and enrichment (Antoniou & Van Harmelen, 2012).

Logic is a well-studied area, and there is a family of logic languages. Some logic languages, e.g., high-order logic, can express very complex facts. However, there is a trade-off between the expressive power and computational complexity of logic. The more expressive the language, the more computation power it takes to infer results while in some cases the results cannot be derived or proofed (Shi et al., 2005). In the IE domain, the extensively adopted logic is the Horn logic, a subset of predicate logic and has sound proof systems. The Horn logic derives from the Horn clause which is defined as a disjunction of literals with at most one positive. An example of the Horn clause is demonstrated in Eq. 2-1, where $A_1, A_2 \ldots B$ are called atomic formulas. In most cases, an atomic formula can be unary, binary, or a constant. Unary atoms only involve one variable (e.g., 'Material(x)', indicating the variable x belongs to the class 'Material'). Binary atoms involve two variables to indicate the relationship between them, e.g., 'removes(supplier_1, concrete_mixture)'. Constants are static numbers or attributes, e.g., the number of days by which a task is delayed.

A Horn clause with exactly one positive literal is a definite clause, a definite clause with no negative literals is called a unit clause, and a unit clause without variables is called a fact. Besides, a Horn clause without a positive literal is called a goal clause (Gupta, 1999). The Horn logic is the implication form of the Horn clause, which is designed for logical programming and reasoning. A typical Horn logic is shown in Eq. 2-2, where the left part of the arrow is called rule body while the right part is called

rule head. From a deductive perspective, the reasoning process is interpreted as 'if-then', namely, if $A_1 \ldots A_n$ are known to be true, then B is also true.

$$\neg A_1 \vee \neg A_2 \cdots \vee \neg A_n \vee B \quad \text{Eq. 2-1}$$

$$A_1 \wedge A_2 \ldots \wedge A_n \rightarrow B \quad \text{Eq. 2-2}$$

2.3.2 Rule-based entity and relation extraction

Rules for information extraction are constructed based on syntactic and semantic features. Syntactic features concern the syntax of a language, and syntactic features can include frequency features (e.g., the word/phrase frequency, inverse document frequency, and bag of words), tokens of words/characters, part-of-speech tags, and phrase-structure grammars. On the other hand, semantic features concern the meanings of words/characters, while the meanings can change in different domains. As such, constructing semantic features usually requires domain knowledge of the intended applications (Le et al., 2020; Zhang & El-Gohary, 2016). For instance, to extract quality checking codes, a common rule using only syntactic features is: 'if a noun is followed by a modal verb and a basic verb, the sentence is a clause whereas the noun is extracted as the subject of the clause' (Zhang & El-Gohary, 2016). When semantic features are included, more information can be utilised, such as extracting the types of the clause subject (e.g., a component or a task) and then enriching the rule with the types. Rules can also be employed to extract relations. For instance, Le and David (2017) extracted domain terms of highway projects and created a terminology by setting-up synonymy and hypernym relations among them. Liu and El-Gohary (2017b) and Liu and El-Gohary (2016) extracted dependency paths of sentences in bridge inspection reports based on computing the similarity of sentence-level configurations and part-of-speech tags. The proposed relation extractor can link bridge deficiencies to maintenance actions. Wu et al. (2021b) extracted constraint relations using rules defined based on domain ontologies and experts' opinions, where constraint entities can be automatically connected if a relation has been defined between their classes in ontologies. Rule-based approaches can achieve high accuracy. However, developing rules needs much time and effort. In addition, rules suffer from subjectivity as most rules are manually developed by researchers. Finally, most rule-based approaches can only achieve good performance in a limited domain, because their ability of matching entities and relations heavily relies on the features of the domain where the rules are derived (Zhong et al., 2020a).

2.3.3 Foundation of machine learning

The working mechanism of logic rules is transparent because 1) rules are explicitly defined by a human (e.g., domain experts), and 2) every step of rule reasoning can be traced to obtain explanations (Gupta, 1999). In contrast, ML models, including DL models that gain much attention recently, are regarded as black boxes. They take in training data (i.e., inputs and labelled outputs) and learn the rules (or called functions) by themselves. The rules are represented by numerical parameters in the models (e.g., parameters for dependent variables in logistic regression) (Shrestha & Mahmood, 2019). In literature, the computational learning theory concerns quantifying learning problems using formal mathematical methods. The computational learning theory is recognised as a basis of ML models, where the most discussed areas are the probably approximately correct (PAC) learning and Vapnik–Chervonenkis (VC) dimension (Murphy, 2012).

PAC learning relies on two critical hypotheses: 1) some functions can map data to correct results, and 2) sub-optimal functions will be found according to predictions they make on unseen data (i.e., based on the generalisation errors in the testing dataset). Hence, a model which has the most or a large number of correct predictions in the testing data is adopted to approximate the unknown functions. In other words, instead of finding the best functions, ML seeks to find those probably good ones. The PAC learning is a process to estimate ML model parameters, where the most common approach is called forward-backward propagation (Figure 2-2). For each input data, the process: 1) makes predictions by passing the data through the model using current parameters, where activation functions (e.g., Relu, Sigmoid, and tanh) are adopted to generate non-linearity; 2) compares the predictions with true labels and computes the difference (i.e., loss) using a pre-defined loss function (e.g., the mean-square error function for linear ML models); 3) applies the chain rule to compute the so-called 'gradients' of model parameters as the partial derivatives of the loss function and all components which involve model parameters; and 4) updates the model by subtracting the gradients multiplied with a learning rate from the original parameters, which is controlled by an optimisation function. The propagation process can be performed many times, and one epoch is completed when all training data are processed for one time (Bolucu et al., 2019).

The VC dimension aims to quantify the complexity of an ML model based on the number of distinct data entities which can be discriminated completely by the model. For instance, given three points in a 2D plane, each of which has a label 0 or 1, they can be correctly distinguished using straight lines, where the lines are the model. In contrast, there are situations that the simple line model cannot correctly split the points, for instance, if four points are placed at the four corners of a square in the same 2D plane. In that case, a curve model can be needed. A large VC dimension indicates that an ML model is flexible, although it can come at the cost of overfitting (i.e., it can be difficult to generalise the model to unseen data) (Goodfellow et al., 2016). Thus, an ML model is considered efficient when it has a balanced VC dimension and can learn proper functions (i.e., rules) in polynomial time.

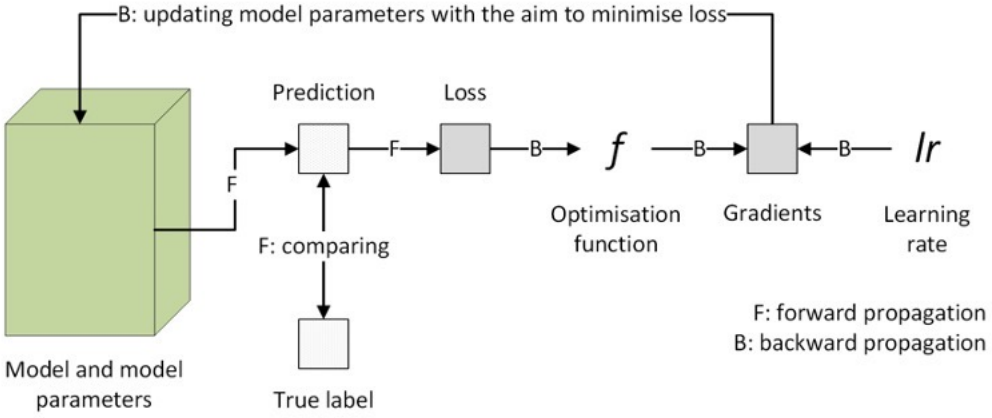


Figure 2-2 Forward and backward propagation

Although the PAC learning and VC dimension to-some-extent uncover the learning process and working mechanism in ML models with simple structures. It is recognised that ML/DL models still more or less lack interpretability. DL models are very complex, as they are commonly formed by stacking simple ML structures (e.g., NN, CNN, and LSTM), and emerged phenomena can happen, which cannot be explained by aggregating the results of simple structures. Hence, more theoretical research is needed to better interpret ML and DL models (Goodfellow et al., 2016).

2.3.4 ML-based entity and relation extraction

(1) ML-based entity extraction

Entity extraction in essence is a named entity recognition (NER) task, which assigns tags (e.g., constraint and task) to words/characters in texts. Most NER approaches apply supervised ML models, e.g., the HMM and CRF model. These models rely on syntactic and semantic features. ML-based entity extraction is in its infancy in the

construction sector. Several distinct studies include: Le and David (2017) used a minimal-supervised ML model to extract transportation entities and recognise if two entities are related; Liu and El-Gohary (2017a) combined a CRF model and ontologies to extract defects and conditions of bridge structures in inspection reports. Chi et al. (2019) applied a semi-supervised model to retrieve topics and detect hierarchical relations among these topics from seismic reports. Conventional ML models require manually constructing and tuning features, which is subjective and time-consuming. On the other hand, text mining methods can be applied to pre-process text data and generate syntactic features for downstream training (Le & David, 2017). Moreover, in a recent study, Zhong et al. (2020b) applied a DL model to extract tasks and their dependencies from quality checking standards, where the Bi-LSTM model was used to extract features. Although these models can extract entities, it is constraint triples that form AWP KBs, which are not well studied.

(2) ML-based relation (triple) extraction

State-of-the-art relation or triple extraction models include dependency-based models and KRL models (Zhou et al., 2018). Dependency-based models take sentences as inputs, obtain the dependency structure (a tree-structure) of words through syntactic parsing, then treat the words as nodes and dependency paths as edges. These models apply sequence models (e.g., LSTM) and graph-based models (e.g., graph-based NN) to extract and aggregate features of words and their neighbourhood. The aggregated features represent the connections among words hence can be fed into downstream ML classifiers to recognise relations in the input sentences (Yin et al., 2018; Zhou et al., 2018).

Dependency-based models can train entity and relation extraction simultaneously. However, they need to perform syntactic parsing which is error-prone. Errors in the parsing process can be propagated to downstream relation extraction and damage model performance (Peng et al., 2017). Besides, dependency-based models usually require the words representing the relations to be explicitly mentioned in texts (Miwa & Bansal, 2016; Zhang et al., 2018a). However, given project documents are freely written, it is common that knowledge triples are expressed by head and tail entities only while the relations remain implicit. For instance, in the sentence ‘supervisors must check safety belts of workers’, there is an implicit ‘constrains’ relation between ‘safe belts’ and ‘workers’, as workers cannot start work until safety belts are in place.

As such, applying dependency-based models to extract triples for AWP can cause many errors and is not adopted in this research.

KRL models are trained on independent triples therefore can extract triples without sentence parsing and explicit mentions of words. KRL models learn to assign a score to a candidate triple *(h, r, t)* to classify it as valid or invalid (Nguyen, 2020). KRL models can be divided into translation models and neural network (NN) models. Translation models train embeddings to represent triple elements. These models assume that a relation *r* can transfer the head entity so that it has a similar embedding to the tail entity, i.e., $h + r \approx t$ for valid triples (Bordes et al., 2013). The assumption draws upon the word embedding translation theory proposed in the *word2vce* model (Rong, 2014). Specifically, when words are represented by low-dimensional dense embeddings, they feature linear properties so that word analogies can often be solved with vector arithmetic, e.g., $\overrightarrow{king} - \overrightarrow{man} + \overrightarrow{woman} \approx \overrightarrow{queen}$. Therefore, translation models can be regarded as the extension of the theory from words to triples.

The first translation model is called TransE (i.e., translating embeddings) (Bordes et al., 2013). Since then, many translation models have been developed to extract more complex relations (e.g., many-to-many relations), such as the TransH (i.e., translating embeddings in hyperplanes) (Wang et al., 2014) and TransR model (i.e., translating embeddings in the space of relation elements) (Lin et al., 2015). Instead of using real-valued embeddings, a few translation models use complex embeddings to represent triple elements (Trouillon et al., 2016). In these models, a relation is defined as a rotation from the head entity to the tail entity in the complex space. Translation models can achieve good accuracy despite simple structures. Besides, embeddings produced by translation models can be fed into more sophisticated models (e.g., deep NN models) as initial inputs (Nguyen, 2020).

NN models often concatenate embeddings of triple elements as a matrix and then take it as the input. For instance, a triple *(h, r, t)* can be represented as a 3-column matrix, and each column refers to the embedding of one triple element (Dettmers et al., 2017; Nguyen et al., 2018). The matrix is fed into a multi-layer NN structure to compute a score of the input triple (Nguyen, 2020). Some studies apply bilinear tensors to replace linear neurons in conventional NN layers (Shi & Weninger, 2017). Popular feature extraction methods (e.g., CNN) have been employed to capture triple features. For instance, the ConvE (i.e., convolutional triple embeddings) and ConvKB model (i.e.,

convolutional KBs) adopt the classic 2D CNN to scan the input matrix (Dettmers et al., 2017; Nguyen et al., 2018). On the other hand, Vashishth et al. (2019) applied a circular CNN to extract triple features; and Nguyen et al. (2019) improved ConvKB by stacking a capsule network layer on top of the convolution layer.

However, most KRL models have a limitation, i.e., they do not consider entities that do not exist in the original KBs where the models are trained. Hence, a KRL model does not have trained embeddings for out-of-KBs entities to accurately extract triples containing them (Nguyen et al., 2018; Nguyen et al., 2019). Recent studies propose to retrain the model whenever out-of-KBs entities appear or estimate their embeddings using additional information (e.g., text descriptions) or complex graph-based DL models. These methods require much training time and computation power, which is impractical to be used for AWP modelling (Bi et al., 2020; Zhao et al., 2020). However, most KRL models still work well in practice, as they are designed for online general world knowledge searching (e.g., 'Jobs founder Apple'). Such knowledge is rather static, and the models are often trained on large and general databases (e.g., DBpedia and Freebase) containing billions of triples. Hence, these models can to-some-extent ignore out-of-KBs entities given their low possibility of appearing (Zhang et al., 2018b).

Based on the review, it can be concluded that classical rule-based matching, classical ML-based models, and DL models have been applied to extract entities and relations in construction documents. Most of these approaches can only extract entities of certain types (e.g., quality criteria and construction tasks). However, AWP modelling can involve more types of constraints from different disciplines (e.g., design, supply chain, site management, and external authorities) and project stages. Therefore, more capable models should be developed. Moreover, few efforts in the AEC industry address triple extraction, which can only identify relations with very basic semantics (Chi et al., 2019; Le & David, 2017). AWP involves relations with complex and rich semantics, which cannot be handled by current IE methods. KRL models can extract semantic-rich relations, but they cannot be directly applied for AWP modelling. For one thing, most KRL models are trained on general KBs. Unlike static and general knowledge in such KBs, constraint information is domain-specific and can regularly change. Thus, existing KBs cannot provide training data for constraint information extraction. The lack of domain-specific triple data also means that out-of-KB entities

can commonly appear. Hence, applying KRL models for AWP modelling requires a method to generate training data while handling out-of-KBs entities. Current KRL models also do not utilise domain-specific information in construction projects (e.g., type/class of constraints) therefore cannot achieve high accuracy when processing construction documents.

2.4 Project information integration in ontological knowledge bases

The ontologies and LPG database are both typical tools to store unstructured triple data. As mentioned, this research adopts ontologies to build project KBs to reap the advantages of the reasoning capacity of ontologies. On the other hand, a KBC model is developed in this research to continuously enrich the KBs. This section introduces 1) theories and relevant concepts of ontologies and 2) the graph theory for developing the KBC model. Existing studies of ontologies in the AEC industry and state-of-the-art KBC approaches are also reviewed.

2.4.1 Ontological databases

2.4.1.1 Ontologies

(1) Definition of ontologies

An ontology is defined as an explicit and formal specification of a conceptualization. This definition is different from the philosophy concept mentioned in Section 3.1 and has specific technical meanings in the sector of computer science and the semantic web. Ontologies intend to provide an unambiguous understanding of knowledge by mapping different descriptions of things into standard, formal, and interconnected concepts in ontologies (Gruber, 1995).

In general, ontologies formally describe a domain of discourse and typically include a finite list of conceptual terms (i.e., classes) in the domain and relations among them. A class can have multiple instances (i.e., entities). In the AEC domain, typical classes can include tasks, materials, equipment, people, and documents. The most common type of relation is the hierarchy of classes (i.e., the subclass relation). A hierarchical relation specifies a class C to be a subclass of another class C' if every instance in C is also included in C'. In ontologies, an instance can belong to more than one class. Ontologies also include other typical relations, such as properties (relations defined among instances) (e.g., 'supervisor reviews drawings') and disjointness statements (e.g., the class 'Material' and 'People' are disjoint, i.e., there is no instance belonging

to the two classes simultaneously) (Antoniou & Van Harmelen, 2012). An ontology consists of two main parts: the terminology part (TBox) and assertion part (ABox). The TBox includes all classes and relations among classes, and the ABox includes assertional knowledge (i.e., ground facts) of instances (instance names and relations among instances). Finally, ontologies follow the open-world assumption (OWA), namely, one cannot suppose that a thing does not exist in an ontology unless this is explicitly specified (Shi et al., 2005).

Developing ontologies requires certain languages. Modern ontologies are developed based on the RDF data model. RDF applies unique resource links (URLs) and the 'turtle' language to identify and describe things using subject-relation-object triples (elements in a triple are expressed by URL). To increase the expressiveness of RDF, RDF schema (RDFs) has been developed to describe rich semantics, e.g., subclasses and domain and range restrictions of subjects/objects. In recent years, the Ontology Web Language (OWL) is extensively applied for building ontologies, which is an extension of RDFs and can support even more complex semantics, e.g., existence and universal quantification (Hitzler et al., 2009).

(2) Logic rules in ontologies

Logic rules, which have been reviewed in Section 2.3.1, are important supporting techniques for ontologies. These rules are widely used to uncover implicit knowledge in ontologies and make them explicit. Rules are developed at the class level, whereas instances of ontological classes will populate the rules after inheriting the classes. Thus, the rule box (RBox) can be created as another component of the terminology part of an ontology. The essence of ontologies is to provide precise and unambiguous information, and the system needs to prove these properties. Therefore, logic rules in ontologies should have a sound proof system meanwhile holding a balance between expressiveness and complexity. In general, ontologies apply two types of logic: description logic (DL) and Horn logic (Hitzler et al., 2009).

Horn logic has been introduced before. As for DL, it is closely related to ontologies and consists of the DL for TBox and DL for ABox. The syntax (e.g., the binary and unary atomic formulas) of ABox DL coincides with FoL (or the Horn logic). On the other hand, TBox DL is derived from the set theory and does not involve variables. An example of TBox DL is illustrated in Eq. 2-3, which implies a subclass relation,

i.e., anything that is a 'Concrete' is also a 'Material'. The FoL translation is shown in Eq. 2-4. A specific feature of DL is that the basic classes (concepts) can be combined into more sophisticated classes by Boolean operators and quantification over relations. Eq. 2-5 provides an example which states that the tasks with delayed constraints can also be delayed. Both DL and Horn logic are subsets of the predicate logic but are orthogonal (neither of them is a subset of the other). In other words, some knowledge facts can be expressed using DL which however are impossible to be expressed using the Horn logic and vice versa. For instance, Horn logic rules cannot (in the general case) express the negation of classes, union information, or existential quantification (e.g., constraints that at least have a planned removal date). However, DL and Horn logic can be to-some-extent combined into complex rules to express rich semantics which are not supported by the original OWL syntax. For instance, each component $A_1, A_2 \dots A_n$ in the Horn logic can be expressed using DL axioms (logic statements are called axioms in ontologies).

The logic language realising the combination is the Semantic Web Rule Language (SWRL). Besides, the SPARQL query, which is designed for extracting information from ontologies, can be embedded in SWRL to turn it into a powerful language, the Semantic Query-enhanced Rule Language (SQWRL) (Fudholi et al., 2009; Wu et al., 2020b). The SWRL and SQWRL have become the mainstream languages to develop logic rules for ontology applications, which are adopted in this research as well. An example of the SQWRL rule is demonstrated in Eq. 2-6, which can identify the delayed constraints based on their removal progress while extracting these constraints by the 'select' query.

$$Concrete \sqsubseteq Material \quad \text{Eq. 2-3}$$

$$\forall x[Concrete(x) \rightarrow Material(x)] \quad \text{Eq. 2-4}$$

$$\exists is\_constrained\_by.Delayed\_Constraint \sqsubseteq Task\_Potential\_Delay \quad \text{Eq. 2-5}$$

$$Constraint(?\,c) \wedge (has\_removal\_delay\ some\ xsd{:}\ integer[\geq 0])(?\,c)$$

$$\rightarrow is\_timely\_removed(?\,c, true) \wedge sqwrl{:}\ select(?\,c) \quad \text{Eq. 2-6}$$

2.4.1.2 Ontological KBs and applications

Ontologies developed in the AEC sector aim to solve engineering problems hence belong to domain ontologies. Thus, all ontologies mentioned in the following contents

are domain ontologies. An ontological KB is a graph database that is built based on domain ontologies and includes nodes (i.e., the classes and instances that represent domain entities) and relations that specify the attributes of and connections among the entities. Many studies of ontological KBs have been conducted in the AEC sector, which focus on vertical buildings and bridge inspection and evaluation. The studies could be divided into three groups: 1) general knowledge modelling, 2) information sharing, and 3) reasoning and conformance checking.

The first group summarises general domain knowledge, which is often independent of application contexts thus can serve as a basis for developing specific ontologies. For instance, there are a few general ontologies that formalise the knowledge for building and infrastructure construction, covering essential concepts like processes, products, and stakeholders (El-Diraby, 2013; El-Gohary & El-Diraby, 2010). Studies in the second and third groups are context-dependent. Studies related to information sharing focus on building semantic relations among information sources and using queries (e.g., SPARQL) to extract information. In this way, one can not only find information that matches keywords textually but semantically related information as well. For instance, a bridge beam can be semantically linked to its design drawings. Hence, when searching for information of the beam, information of the design drawings can be easily identified and retrieved by navigating the semantic link between the two ontological instances. A data link can also be set up following the semantic link, and then accessing the drawings can be realised by directly querying the ontology rather than manually searching the designer's database (Wang, 2018). The third group of studies is related to reasoning and conformance checking. In these studies, reasoning rules are often expressed by formal logic, e.g., SWRL and SQWRL (Ren et al., 2019). Thus, if an entity (e.g., a building component) has certain attributes or is related to other entities through certain relations, the logic rules can derive implicit knowledge regarding the entity. For instance, Zhang et al. (2015) embedded safety rules in an ontology to identify building elements and areas violating the safety rules, considering geometries and topological relations among these elements. Ren et al. (2019) grouped semantically related bridge components and then embedded condition evaluation rules to compute structure conditions of bridge components and the entire bridge.

Ontologies can be object or process-oriented. The former is based on taxonomies of objects, such as building components like walls and windows; the latter is based on

the sequences and constraints of construction tasks (Dong et al., 2011). The object-oriented ontology is the dominant form partially because of the existence of many taxonomies that such ontologies can draw upon, e.g., the UNIFORMA for building components (Niknam & Karshenas, 2017). Object-oriented ontologies are mainly employed to manage static information in documents (e.g., inspection reports) and information systems (e.g., BIM and BMS) (Liu & El-Gohary, 2017a), including the material properties and geometries (Niknam & Karshenas, 2017), defects (Park et al., 2013), quantities and costs (Liu et al., 2016), risks (Zhang et al., 2015), and structure conditions (Ren et al., 2019). Some studies are developing simple process-oriented ontologies which often serve as auxiliary parts to integrate information of work progress. For instance, Dong et al. (2011) developed an ontology to monitor project progress using simple qualitative metrics (e.g., not started, behind schedule, and ahead of schedule). Zhang et al. (2015) employed an ontology to model the activities of masonry work, which could search for risk information. Wang (2018) created an ontology to store constraint information in energy plant projects.

2.4.2 Knowledge base completion models

2.4.2.1 Graph definition and the related theory

A graph can be denoted by $\boldsymbol{G = (V, E)}$, where $\boldsymbol{V}$ is the set of nodes (or called vertices) and $\boldsymbol{E}$ is the set of edges (or called relations). An edge $e = \{u, v\}$ has two endpoints u and v, which are joined (i.e., linked) by e. In this case, u is a neighbour of v. In other words, the two nodes are adjacent. Thus, data taking the form 'node-edge-node' are triples. An edge can be directed or undirected, and a graph is directed if all edges are directed or undirected if all edges are undirected. Besides, the degree of a node v, denoted by $d(v)$, is defined as the number of edges linked to v (Ji et al., 2020).

Observing a graph is an intuitive way to understand the interconnections among data entities. However, to quantitatively analyse graph data, a graph should be represented mathematically. Accordingly, there are some algebra representations for graphs that are widely adopted in existing literature, which are introduced below (Ji et al., 2020; Zhang et al., 2018b).

- **Adjacency matrix:** for a graph $\boldsymbol{G = (V, E)}$ that has **n** nodes, it can be described by an adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ which indicates if any two nodes are connected by any edge (see Eq. 2-7).

$$A_{i,j}\begin{cases}1\ if\ \{v_i, v_j\} \in E\ and\ i \neq j \\ 0\ otherwise\end{cases} \quad \text{Eq. 2-7}$$

- **Degree matrix:** for a graph ***G = (V, E)***, it has a degree matrix $\boldsymbol{D} \in \mathbb{R}^{n\times n}$ which is a diagonal matrix and reflects the degree of each node in ***G*** (see Eq. 2-8).

$$D_{ii} = d(v_i) \quad \text{Eq. 2-8}$$

- **Laplacian matrix:** for a graph ***G = (V, E)*** that has **n** nodes, if all edges in ***G*** are undirected, then the Laplacian matrix $\boldsymbol{L} \in \mathbb{R}^{n\times n}$ of ***G*** can be computed using the adjacency and degree matrix (see Eq. 2-9), while the matrix ***L*** is defined by Eq. 2-10.

$$L = D - A \quad \text{Eq. 2-9}$$

$$L_{i,j}\begin{cases}d(v_i)\ if\ j = j \\ -1\ if\ \{v_i, v_j\} \in E\ and\ i \neq j \\ 0\ otherwise\end{cases} \quad \text{Eq. 2-10}$$

Current KBC models are built based on the above basic definitions and concepts. Similar to the IE task, a KBC task can also be realised by rule-based and ML-based approaches. In addition, state-of-the-art ML models for KBC tasks can be further divided into triple-based and GNN-based models.

2.4.2.2 Rule-based KBC

Rule-based KBC constructs logic rules to infer missing triples using existing ones in a KB. The rules take the form of Horn logic, for instance, the rule '(manager manages foreman), (foreman manages crew) ⇒ (manager manages crew)' implies the constraining relation between 'manager and 'crew' (i.e., the rule head) according to triples at the left side (i.e., rule body) of ⇒. The logic rules can compactly and intuitively encode knowledge facts, which are widely applied for reasoning in early studies of KBs, e.g., those related to expert systems (Ren et al., 2019). However, as mentioned in Section 2.3.2. handcrafted rules can be subjective and incorrect (Qu & Tang, 2019; Yang et al., 2014; Yang et al., 2017). As such, some studies adopt the Markov logic network to transform rules into a graph and then apply Markov models to handle the uncertainty during reasoning (Zhang et al., 2020). On the other hand, reasoning is a sequence (i.e., multiple steps) of applying rules. Therefore, instead of manually developing rules, some studies use reinforcement learning (Lin et al., 2018) or sequential models (e.g., the LSTM model and gated recurrent network (GRU))

(Yang et al., 2017) to mine rules automatically using information in KBs. However, when it comes to predicting missing triples, the capacity of rule-based methods can be further limited, as many triples cannot be discovered by rules (Qu & Tang, 2019).

2.4.2.3 Triple-based KBC

The KRL models reviewed in Section 2.3.4 can extract triples from texts, thus, they can also be trained to complete KBs which in essence are formed by triples. To avoid redundancy, such models are not introduced again in this section. Despite the wide application of triple-based models for KBC, they suffer a distinct limitation, i.e., they only consider independent triples. Hence, the models can neither capture structure features (i.e., topology) of KBs nor leverage logic rules to infer information (Nathani et al., 2019; Velickovic et al., 2017).

2.4.2.4 GNN-based KBC

Most traditional ML models are designed for structured data. For instance, in image processing tasks, each image is structured in 2D or 3D tensors, whereas CNN filters can operate on each pixel node and scan information from its neighbourhood which has a fixed size and order of pixel nodes. However, nodes in a KB have different neighbourhood sizes and there is no order of nodes. Thus, conventional CNN filters cannot be directly applied to such unstructured data. On the other hand, the distinct advantage of graph-based models is that they enable CNN to extract features of KB data (Zhang et al., 2018b).

There are two main types of GNN in literature: spectral models and spatial models. The key difference relies on the way they process graphs before extracting features using relevant tools (e.g., CNN). In spectral models, the graph Laplacian matrix is generated first. Then, eigen-decomposition of the graph is performed based on the matrix, which projects the graph into the Fourier domain and enables CNN operations (Kazemi et al., 2020). Spectral models are more adopted in early studies. The models must perform eigen-decomposition which requires additional computation power. In addition, they depend on the graph's Laplacian matrix, which means a model trained on a specific graph could not be applied to another graph with different structures. In contrast, spatial models avoid the decomposition process and can directly apply CNN to a node's neighbourhood. They also do not require the Laplacian matrix hence are more flexible (Ji et al., 2020). Thus, the spatial GNN is adopted in this research, and the term GNN refers to spatial models in the following contents.

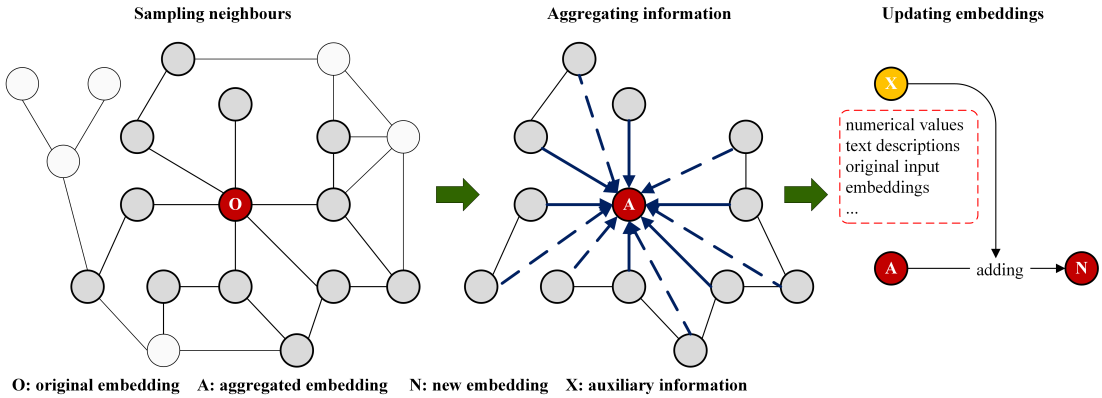


Figure 2-3 The overall process of GNN-based graph encoding

Spatial GNN generally relies on an iterative computation process, and each iteration has three activities: sampling, aggregation, and updating. The overall process is shown in Figure 2-3. Sampling selects nodes in each node's neighbourhood. For instance, a one-hop neighbourhood includes nodes inward and outward connected to the central node. Aggregation extracts and aggregates information (e.g., embeddings) from nodes in the neighbourhood of the central node for which a new embedding is computed using the aggregated information. Finally, updating replaces the previous embedding with the newly computed one (Hamilton et al., 2017). Different GNN models are proposed to improve the activities. For example, the attention mechanism and various statistical sampling methods can refine the neighbourhood generation by sampling critical nodes rather than taking all nodes for computation (Zhang et al., 2020). There are also multiple ways to aggregate node embeddings, e.g., summing, averaging, and pooling. Recent studies use the LSTM and GRU model to aggregate information (Hamilton et al., 2017; Schlichtkrull et al., 2018). When it comes to updating, additional information, such as numerical features (e.g., the coordinates of nodes representing physical locations), textual descriptions of nodes, and original input embeddings, can be added to incorporate more information for model training (Srivastava et al., 2014; Zhang et al., 2018b). Some studies combine logic rules and ML models. A common approach is to run logic rules to infer triples and adopt ML models to predict entities when the rules encounter missing information in rule bodies or heads (Qu & Tang, 2019; Zhang et al., 2020). However, similar to the KRL models, existing KBC models do not utilise specific knowledge in the AEC domain, which can largely affect their capacity when completing AWP graphs.

Chapter 3: Research methodology

In this chapter, the research methodology is introduced, which includes four sections for the four objectives summarised in Section 1.3. First, in Section 3.1, the research philosophy is introduced. The overall research design and mapping between research methods and objectives are demonstrated in Section 3.2. Section 3.3 – Section 3.6 introduce the research methods for realising Objective 1-4, respectively. Section 3.7 summarises this chapter.

3.1 Research philosophy

Paradigms are roots and stances of researches and can be defined as several basic beliefs that guide how things are understood or done by him/her during research (Killam, 2013). According to Guba and Lincoln (1994), paradigms are "*basic belief systems based on ontological, epistemological, and methodological assumptions*". These elements are interdependent under a paradigm, and different types of research are guided by different paradigms (Wilson, 2001). Thus, knowing the philosophical beliefs behind is essential for research.

(1) Ontology

Ontology is a theory of being or existence, dealing with the nature of reality (Aliyu et al., 2015). Ontology concerns what exists, what it looks like, what units make it up and how these units or things interact. There are two contrasting types of ontologies, namely, realism and relativism. Realists believe there is only one reality that can be discovered and objectively measured by different observers and researchers. On the contrary, relativists believe that the 'truth' relies on people who are observing, hence, multiple realities can be constructed based on individual opinions and experience, while each of them is true to the observer or researcher (Killam, 2013). The ontology discussed here is different from the ontologies developed for integrating information in this research, which is a technical concept in the computer science field (Studer et al., 1998).

(2) Epistemology

Epistemology is a theory about knowledge and the relationships between researchers and things that are researched. In other words, epistemology concerns how knowledge can be acquired (Aliyu et al., 2015). There are contrasting epistemological positions

of researchers, i.e., objectivism and subjectivism, which are determined by the ontology of researchers (Wilson, 2001). Objectivism is based on realism. As such, a realist will apply objective methods to observe things and discover the only be 'truth' that exists independently from researchers. However, a relativist who believes in relativism will interpret facts using his/her opinions, experience, and feelings, because they think the truth varies from different people and contexts.

(3) Methodology

Methodology refers to ways to systematically discover knowledge and is driven by ontology and epistemology (Killam, 2013). Based on objectivism or subjectivism a researcher believes, main research methodologies include deductive and inductive methodologies and quantitative and qualitative methodologies (Aliyu et al., 2015).

- *Deductive and inductive research*

The two methodologies refer to research logic, which concerns the role of the current knowledge body and the way to utilise data collection and subsequent data analysis methods. The logic of deductive research is based on objectivism, which proposes a hypothesis using current knowledge then tests the hypothesis by data collection and analysis (often quantitative). In contrast, the logic of inductive research is based on subjectivism. Inductive research first performs data collection and analysis to obtain findings (often qualitative), where the existing knowledge can be applied to inform data analysis when researchers see appropriate (Simon, 1996).

- *Qualitative and quantitative research*

The quantitative methodology follows objectivism thus prefers quantitative inquiry with measurable methods such as controlled experiments to minimise bias. Results of quantitative research are viewed as generalisable and replicable. Research adopting quantitative methods usually aims to test theories deductively through quantified and objective explanations. The qualitative methodology follows subjectivism and favours qualitative methods which can adequately consider interactions between reality and researchers and explain phenomena from viewpoints of participants, e.g., in-depth interview and focus group study. Qualitative methods are discovery-oriented, hence, research results are less concerned with generalisability and replicability. Qualitative research is commonly applied to suggest possible relationships, effects and dynamic processes (Gelo et al., 2008).

Since both methodologies have weaknesses while they can supplement each other, they can be adopted in a mixed manner (i.e., concurrently or subsequently) to gain more reliable research results.

(4) Axiology

Axiology mainly considers ethical issues, which is a theory on the nature, types and standards of value and value judgments, especially in morality (Wilson, 2001).

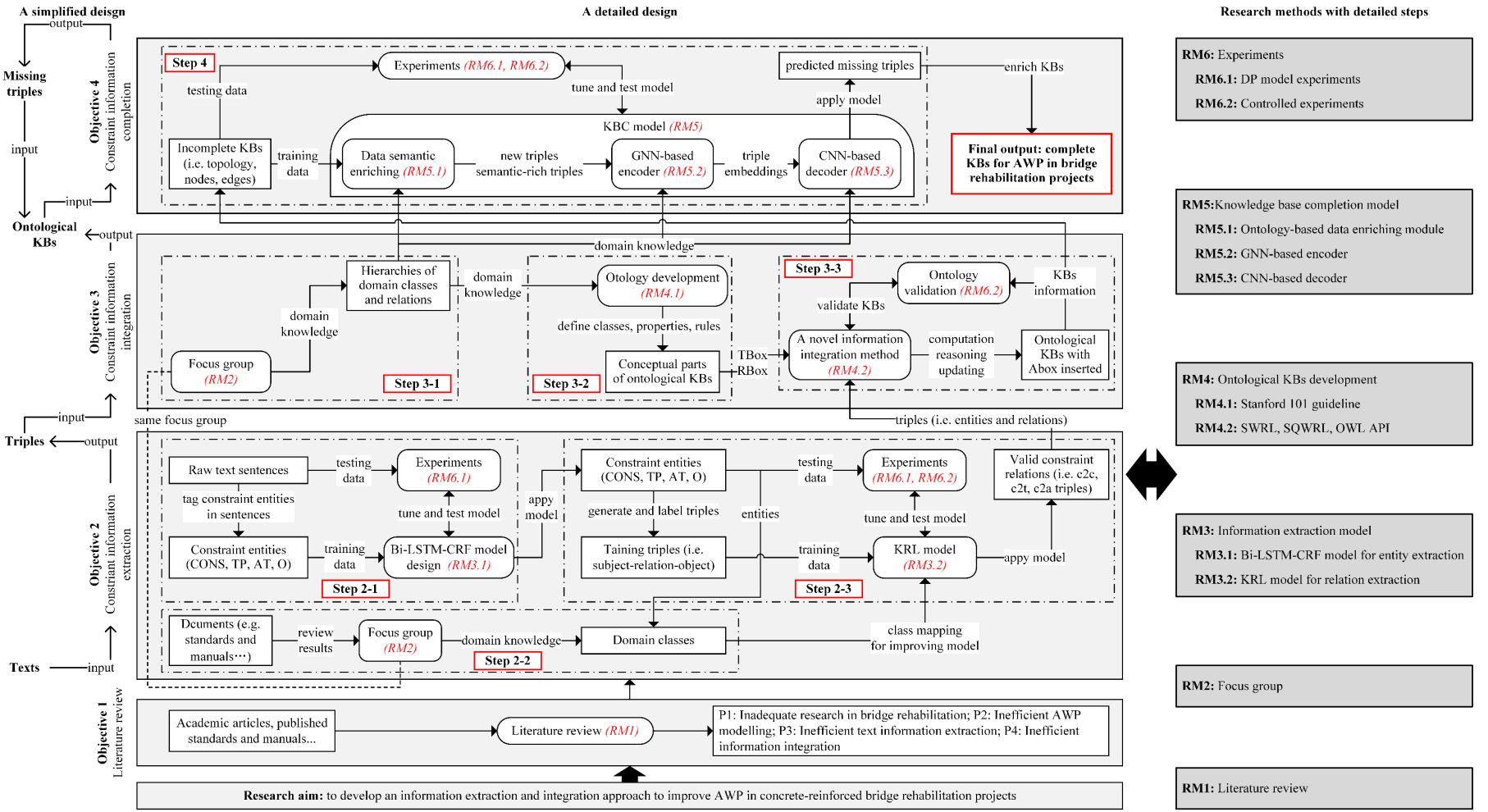
Based on different beliefs in the key components of paradigm, different paradigms emerge, and four dominant ones are positivism, post-positivism, critical theory and constructivism. Positivism is the most conventional paradigm and strictly follows realism (Koschmann, 1996). Constructivism is the contrasting paradigm to positivism and is fully based on relativism. Post-positivism and critical theory are in-between, which are closer to positivism and constructivism, respectively. Different paradigms do not necessarily work in isolation; they may work together (Killam, 2013).

This research aims to improve information extraction and integration to automate AWP and then verifies the proposed approaches in bridge rehabilitation projects. The DL models (i.e., the Bi-LSTM-CRF, KRL, and KBC model) are all quantitative models. Besides, the research proposes the hypothesis that adding domain information can improve the performance of DL models, which should be tested in experiments. Thus, the research is closer to deductive and quantitative research and is based on objectivism epistemology and realism ontology. Meanwhile, subjective knowledge of domain experts is also utilised to develop the ontological KBs and label training data of DL models. Thus, it can be argued that the research is a mixed study and belongs to the post-positivism paradigm.

3.2 Overview of the proposed method

The overview of adopted research methods is illustrated in Figure 3-1. Each method can be adopted to realise one or more objectives. The mapping between methods and objectives are also shown in the figure. As shown in the left part of Figure 3-1, the proposed information management approach has three key parts, where the outputs of the previous part are taken as the input of the subsequent part. The first part refers to the information extraction model developed in Objective 2, where the inputs are text sentences and the outputs are constraint entities and relations (i.e., triples). The second part corresponds to the ontological KBs developed in Objective 3, which integrate the

automatically extracted entities and relations. The third part refers to the KBC model developed in Objective 4, where the inputs are ontological KBs (including all the triples and graph topology), and the outputs are those identified triples which do not exist in the original KBs.



Abbreviations in the figure: CONS: constraints, TP: tasks/procedures; AT: attributes; O: other entities; c2c: constraint-constraint relations; c2a: constraint-attribute relations; c2t: constraint-task relations; KBs: knowledge bases; AWP: advanced work packaging; Bi-LSTM-CRF: bi-directional long-short term memory and conditional random field; KRL: knowledge representation learning; GNN: graph neural network; CNN: convolutional neural network;

Figure 3-1 Overall research design

3.3 Literature review method (Objective 1)

A literature review is the research method to summarise previous efforts and identify gaps. In this research, the review was conducted following the strategy proposed by Thome et al. (2016), which are introduced below.

3.3.1 Step-1 scope determination

Given the research aim and objectives, the review scope includes three aspects: 1) information management in bridge maintenance (i.e., DDBM studies), 2) constraint management approaches in the AEC industry, and 3) current information extraction and integration approaches. To cover state-of-the-art literature, the review of the first two aspects focuses on the AEC industry, while the review of the last aspect further covers efforts in both the AEC and NLP domains.

3.3.2 Step-2 data collection

The next step is to determine the databases and keywords for searching for review materials. The Web of Science database was selected due to its wide coverage and recognised quality (Bradley et al., 2016). Given the small number of academic articles of WFP and AWP, the databases of the CII and COAA were also searched, and standards, reports, and case reports were collected to provide more information. The advanced searching function was applied in the Web of Science database to increase the coverage of articles. The keywords used for restricting the topics in the advanced searching are as follows:

- **DDBM studies:** The query is based on the key information management steps: *(bridge*) AND (data NEAR/5 manage* OR information NEAR/5 manage* OR data NEAR/2 collect* OR information NEAR/2 shar* OR data NEAR/2 exchang* OR decision* OR decision* NEAR/2 make* OR optimi* OR multi* NEAR/2 object* OR multi* NEAR/2 criter*) AND (planning* OR monitoring* OR maintenance* OR inspect* OR repair* OR rehabilitat*).*
- **Studies of constraint management:** the query can cover mainstream advanced constraint management methods in the AEC industry: *(construction NEAR/2 project*) AND (constraint NEAR/2 manage* OR work NEAR/2 face NEAR/2 plan* OR work NAER/2 packag* OR last NEAR/2 plan* OR advanced NEAR/2 work NEAR/2 packag*).*

- **State-of-the-art information extraction and integration:** The query is based on purposes of information extraction and integration in this research: *(information NEAR/3 extract* OR data NAER/3 extract* OR ontolog* OR knowledge NAER/2 base* NEAR/2 complet* OR knowledge NAER/2 graph* NEAR/2 complet*) AND (construction NEAR/2 project*)*.

The queries consider words with similar meanings, e.g., information and data, as well as common variations and combinations of words, e.g., the term exchang* can include exchange and exchanging, while the term *NEAR/n* can specify the number of words between two terms.

Using the above keywords, the search returned 1052 documents, where 766, 35, and 251 of them are related to DDBM studies, constraint management in the industry, and information extraction and integration, respectively. The initially collected documents were filtered. For academic articles, only peer-reviewed journal articles were retained to ensure quality. Besides, the abstract (or executive summary) and keywords of each document were screened to ensure that they could comply with the review scope. As a result, 737 documents were left for reviewing, where 485, 135, and 117 of them fall in the three above-mentioned aspects, respectively. The 737 documents include 56 and 40 non-academic documents related to AWP and WFP, respectively.

3.3.3 Step-3 content analysis

The in-depth review was conducted using content analysis, an extensively recognised method for investigating texts in a document. To conduct content analysis, the texts were broken down by coding. Specifically, general categories were first established to roughly group the texts, then detailed codes were proposed to further classify the texts in each category. Finally, the concepts, themes, and patterns and trends in the domain of interest were extracted by interpreting the coded texts (Elo & Kyngäs, 2010). The categories and codes for content analysis are listed in Table 3-1.

Table 3-1 Categories and codes for content analysis

Categories	Codes
Bridge maintenance	(1) inspection, (2) condition evaluation, (3) decision-making, (4) repair and rehabilitation
DDBM studies	(1) sensor-based data collection, (2) NDT-based data collection (3) maintenance information analysis, (4) maintenance decision-making, (5) information integration and sharing

Categories	Codes
Constraint management	(1) constraint identification and constraint modelling, (2) constraint monitoring and analysis, (3) constraint removal
Information extraction	(1) rule-based entity extraction approaches, (2) rule-based relation extraction approaches, (3) ML-based entity extraction models, (4) ML-based relation extraction models
Information integration	(1) data formats and schemas, (2) relational databases, (3) graph databases (ontological KBs), (4) collaboration platforms, (5) rule-based KBC approaches, (6) triple-based KBC models, (7) GNN-based KBC models

The review addresses the questions that guide the research: 1) by reviewing bridge maintenance and DDBM studies, one can understand what ICTs have been applied in which bridge maintenance stage, and which stages can still be improved using what ICTs? 2) by reviewing modern constraint management approaches, one can understand what are the critical steps and advantages/disadvantages of AWP, and what hinders the implementation of AWP? 3) by reviewing information extraction and integration approaches, one can understand what are the mainstream IE methods (for entity and relation extraction) and information integration methods in the AEC area, can they meet the demands of practical AWP modelling, what are the state-of-the-art approaches in the literature, and can they be directly applied for AWP modelling?

3.4 Information extraction model design (Objective 2)

Constraint information extraction is achieved by a hybrid DL model. Specifically, the Bi-LSTM-CRF and CNN-based KRL models are developed for entity and relation extraction, respectively. The Bi-LSTM model extract entities, then the KRL model extract entity-relation-entity triples by identifying valid triples among candidate triples generated using the extracted entities.

3.4.1 Bi-LSTM-CRF model (Step 2-1)

3.4.1.1 Data inputs and outputs

The proposed Bi-LSTM-CRF model is used to extract constraint entities, including a Bi-LSTM model at the bottom layer and a CRF layer at the top layer. A Bi-LSTM-CRF model takes sentences as inputs and extracts entities by tagging words in the sentences. A sentence is a sequence of words/characters, i.e., $x = (x_1, x_2 \dots x_m)$. The tagging process assigns a tag to each word/character and produces $\hat{y} = (\hat{y}_1, \hat{y}_2 \dots \hat{y}_m)$. The proposed model concerns four entity tags, i.e., CONS, TP, AT, and O, indicating if the word is a constraint, a task or procedure, an attribute, or an irrelevant entity, respectively. It should be noted that the advanced DL model, bidirectional encoder

representations transformer (BERT), is proposed by Google recently and can reach better performances than more traditional models (e.g., LSTM) in many NLP tasks, e.g., machine translation, sentiment analysis, and NER. However, BERT is a heavy model consisting of over 20 building blocks, and each includes at least three neural network components and 10 attention heads (similar to filters in CNN). Thus, training a BERT model is extremely data-demanding (Vaswani et al., 2017). An alternative is transfer learning, i.e., using most parameters pre-trained by Google while fine-tuning the rest at the output end of the model. However, application research for the AEC industry should concern more on how much a proposed approach can increase the management efficiency and the practicality of the approach, rather than the small amount of accuracy improvement. Transfer learning of BERT still takes much time and computation power, making it impractical for project teams. More importantly, the original BERT model does not support relation extraction, hence, it can only be used to extract constraint entities. As shown in Section 4.3, the proposed Bi-LSTM-CRF model can already achieve 93% F1 when extracting entities, which is sufficient for constraint management. Thus, it is not cost-effective to implement a BERT model which cannot bring significant improvements for the entire proposed information management approach.

Nevertheless, two types of entities, i.e., the work packages and project participants, cannot be extracted by the proposed model. The number of training samples is small, as most documents that the researchers can access do not explicitly mention work packages (CII, 2013a, 2013b, 2020). The documents only have a few mentions of participants, as participant entities appear more frequently in contracts that are difficult to collect. Fortunately, the number of the two types of entities in a project KB is much less than that of constraints, attributes, and tasks/procedures, making it still practical to insert such information manually. Hence, in the experiments, the participant and work package entities were inserted manually.

3.4.1.2 Overall design of the entity extraction model (RM3.1)

DL models can only recognise numbers. Thus, each word/character must be converted to a numerical vector (either a one-hot vector or word embedding, see Section 4.2.1 for details) to be fed into the entity extraction model. However, sentences can have long dependencies between two words, e.g., two semantically related words can be separated by multiple irrelevant words. Hence, applying traditional DL models (e.g.,

recurrent networks) to capture text features is often challenging and can result in gradient exploding and vanishing when passing information in a long sentence. In contrast, owing to the three gates introduced below, the Bi-LSTM model can extract text features effectively meanwhile minimising the possibility of gradient vanishing and exploding (Hochreiter & Schmidhuber, 1997; Miwa & Bansal, 2016). However, the Bi-LSTM model can ignore the features of entity tags. As such, it can make some simple mistakes, e.g., tagging two consecutive words/characters as the beginning word or character of an entity at the same time. On the contrary, the CRF model is good at capturing such tag features, as it is designed to predict the entire sequence of tags of an input sentence. Therefore, a common practice is to stack a CRF layer on top of a Bi-LSTM model (Baker et al., 2019). In this case, the tags predicted by the Bi-LSTM model are fed into the CRF layer where the predictions are refined and outputted. The detailed model design is introduced in Section 4.2.

3.4.1.3 DL model experiments (RM6.1)

(1) Experiment data collection, pre-processing, and labelling

To verify the Bi-LSTM-CRF model, the experiment data (i.e., text sentences) were extracted from various documents, e.g., manuals, standards, technical specifications, working plans, case reports, and meeting records of both conventional construction and concrete-reinforced bridge rehabilitation projects. All the documents were pre-processed, including text normalisation and sentence splitting. The raw documents contained many tables, figures, and formulas. Text normalisation removed figures and formulas and then converted tables to sentences by extracting the texts in table cells. Irrelevant texts (e.g., the organisation structure of a project) were also removed. Then, sentence splitting recognised different sentences based on typical boundaries (e.g., periods), which produced sentences suitable for data labelling. Data labelling for the model should concern two things, i.e., the level of labelling and types of tags. The four entity tags can work as word-level tags, and another two character-level tags (i.e., B and I) can be used to indicate if a character is at the beginning or intermediate place of a word (Zhong et al., 2020a). Therefore, all sentences were labelled by supplementary usage of the six tags (see Figure 4-2 as an example).

(2) Training, validation, and testing protocols

Different metrics should be selected to evaluate different DL models. A Bi-LSTM-CRF model aims to classify if a word/character is an entity of interest (i.e., binary classification) and then assign a tag to the word/character (e.g., a constraint, task or procedure, or attribute), which is a multi-classification task. Thus, in experiments, the Bi-LSTM-CRF model used three common metrics for classification tasks: precision (Pr), recall (Re), and F1 score (F1), which can be computed by Eq. 3-1 - 3.3, where β is the weight between Pr and Re (Goodfellow et al., 2016).

$$Pr = \frac{TF}{TP+FP} \quad \text{Eq. 3-1}$$

$$Re = \frac{TF}{TP+FN} \quad \text{Eq. 3-2}$$

$$F1 = \frac{(\beta^2+1)\times P\times R}{\beta^2\times P+R} \quad \text{Eq. 3-3}$$

(3) Data splitting and experiment process

Experiments of DL models should generally include training, validation, and testing. The raw dataset should be divided into the training, validation, and testing datasets. In experiments, the model should be trained and evaluated using the training dataset to ensure the model can be trained (i.e., the model loss can decrease over time and become stable when the model converges). Then, the validation dataset should be employed to fine-tune the model and increase its performance. The main purpose is to obtain optimal hyper-parameters (more details are introduced below). The model can be finally tested in the testing dataset. The model never encounters data in the testing set, hence, the performance can objectively reflect the model capacity (Baker et al., 2019). The general proportion between the three datasets can be 7:2:1 if the number of training samples is small (e.g., less than 100000), otherwise, more data can be allocated to the training dataset (Shrestha & Mahmood, 2019). Given the data volume is not large in this research, 9:1 data splitting was adopted in all model experiments, i.e., 10% data for testing and 90% data for training and *k*-fold cross-validation (more details are introduced below).

It should be noted that in the experiments, the proposed Bi-LSTM-CRF model was compared with several classical ML models in terms of extracting constraint entities. All the models took the same inputs and went through the same training and tuning procedures. Their performance metrics were compared in the testing set to reflect if the Bi-LSTM-CRF model could outperform the classical ML models.

(4) Model hyper-parameter tuning methods

Hyper-parameters can affect DL model performance and efficiency, which can be divided into two groups. The first group of hyper-parameters mainly affects loss computation, such as the choice of activation and loss functions. Hyper-parameters in the second group affect the updating of model parameters, such as the learning rate and optimisation functions. Besides, training data are usually separated into batches while the model takes one batch at a time. The batch size and the number of epochs are also hyper-parameters because they affect training time and computation power (Goodfellow et al., 2016). Hyper-parameters cannot be trained but should be tuned manually to suit the input data and model structure. For each hyper-parameter, a list of values is created (e.g., 0.001, 0.01, 0.1 for the learning rate). The lists form a discrete space of hyper-parameters, and a combination of hyper-parameters is a point in that space. During tunning, different combinations are selected through grid-searching or random searching. Grid-searching can find the best hyper-parameters, as it can cover all combinations. However, the strategy can require much time and computation power thus is suitable when the number of hyper-parameters is small. In contrast, random searching selects some points in that space using statistical sampling. Despite that random searching can miss the optimal combination, it can save much time and still return a rather optimal set of hyper-parameters with properly designed sampling. This strategy is more applied when the model contains many hyper-parameters. The grid-searching strategy was employed in experiments of this research given the relatively small number of hyper-parameters (Probst et al., 2019).

For each selected combination of hyper-parameters, the model should be evaluated in the validation set. In the experiments, five-fold cross-validation was adopted. Except for 10% data in the testing set, 20% of the remaining data were randomly sampled as the validation set, roughly complying with the 7:2:1 proportion. The hyper-parameters gaining the best average performance in the five validation sets were chosen as the optimal hyperparameters. In this way, the model could encounter more data during training and validation (Shrestha & Mahmood, 2019). The hyper-parameters for the Bi-LSTM-CRF model and the tuning results are introduced in Section 4.3.1.

3.4.2 Focus group for domain knowledge collection (Step 2-2 & RM2)

A focus group is a topic-based, in-depth group interview method. It intends to obtain data from a purposely selected group of individuals (called participants) rather than a

sample of a broader population. Participants of a focus group should have similar socio-characteristics and adequate experience on certain topics of a study. Thus, the participants are often domain experts (O. Nyumba et al., 2018). A focus group can discover and integrate the opinions of different participants. Therefore, it is a useful qualitative tool to gain an in-depth understanding of and solutions to the problems without common agreement. Besides, a focus group can save much time and costs compared to conducting a large-scale survey and individual interview (Ho, 2006).

In the AEC industry, the focus group method has been adopted to investigate various topics, such as the stressors of construction staff (Leung & Chan, 2012), factors that affect public engagement in early project stages (Leung et al., 2014), risks in high-rise building projects (Kim et al., 2016). In addition, Wang (2018) applied the method to identify constraints in maintenance projects of LNG plants. As mentioned, domain knowledge (e.g., detailed constraint types) is important for DL models to reach high performance. The tasks/procedures and constraints in bridge rehabilitation projects can be different from those in other projects, and complex relations exist among project entities. Currently, there is no common understanding of such topics. As such, a focus group was conducted to identify typical classes of constraints and tasks/procedures, organise them in hierarchies (called domain taxonomies), and identify main relations and relation hierarchies for AWP modelling.

The results of the focus group can assist in realising not only Objective 2 but also Objectives 3-4. Specifically, the hybrid IE model (Objective 2) and KBC model (Objective 4) require domain class hierarchies to improve the model structures, and the KBC model additionally requires domain relation hierarchies to enrich input data semantics (see Section 4.4 and Section 6.2 for more details). On the other hand, the ontological KBs (Objective 3) require both class and relation hierarchies to build the skeleton (i.e., the TBox) of ontologies (see Section 5.2 for more details).

3.4.2.1 Focus group topic determination

The focus group topic covers domain classes of constraint entities and main relations in concrete-reinforced bridge rehabilitation projects. Four types of constraint entities are considered: constraints, constraints' attributes, tasks/procedures, and participants of a project. Tasks/procedures are also constraints, e.g., preceding tasks can constrain succeeding tasks. However, to comply with common sense and minimise confusion during focus group discussion, tasks/procedures are separated from other constraints

in this research. The first task of the focus group is to identify domain classes of constraint entities and build hierarchies of the classes. On the other hand, seven common types of relations for AWP are covered: 1) relations of the form 'constraint entity constrains constraint entity' (*c2c*), 2) relations of the form 'constraint entity constrains task/procedure' (*c2t*), 3) relations of the form 'constraint entity has-attribute attribute' (*c2a*), 4) dependencies of tasks or procedures (*t2t*), 5) relations between constraint entities and work packages (*c2p*), 6) relations organising packages in sequences/hierarchies (*p2p*), and 7) relations between constraints/tasks/procedures and project participants (*ct2pp*).

It should be noted that the *t2t* and *p2p* (releasing sequences) relations often have unambiguous meanings. For instance, the relations is-succeeded-by, is-preceded-by, and proceed-concurrently can model task/procedure dependencies in most projects while can be well understood by all participants. Besides, the *c2p* and *p2p* (package hierarchies) relations are straightforward, which can be modelled by 'is-constraint-of' and 'sub-package-of', respectively. On the contrary, the *c2c*, *c2a*, *c2t*, and *ct2pp* relations often do not have common expressions. Different names can be used to describe the same relation. It is very important to obtain unique and unambiguous descriptions and definitions of relations when integrating information in ontologies. Hence, this is another task in the focus group. Relations can contain simple or rich semantics. For example, *c2a* relations can all be modelled by 'has-attribute' or can be divided into more detailed relations (e.g., 'has-geometry' and 'has-price'). Thus, relations can form hierarchies which are important for developing the ontologies (Chapter 5) and the KBC model (Chapter 6). The relation hierarchies are an important topic of the focus group as well.

Accordingly, the topic of the focus group could be divided into three sub-topics: the development of hierarchies of classes of constraint entities, main relations for AWP modelling and their descriptions, and development of relation hierarchies.

3.4.2.2 Focus group participants

In the next step, the number of participants and their selection criteria should be determined. It is suggested that a focus group of 5-12 participants can keep a balance between depth and breadth of data collection (El-Sabek & McCabe, 2018). Hence, this research invited ten domain experts who were selected based on the following criteria: 1) they should have rich work experience (i.e., more than 8 years) of bridge

maintenance, 2) they should be involved in at least one bridge major rehabilitation project in the last five years, 3) they should come from different backgrounds to cover more project stages and represent the views of different participants. Table 3-2 lists the profile of the ten participants. The group involves project-level participants (e.g., the owner, contractor, designer, maintenance team, and supply company) as well as external participants which mainly refer to relevant authorities such as DoTs and municipal bureaus. Thus, it can be argued that the experts can provide comprehensive and useful advice on the topics of interest.

Table 3-2 Profile of domain experts in the focus group

	Organisation	Years of experience	Area of expertise
1	Southwest Jiaotong University	8	Applications of ICTs in infrastructure projects
2		8	
3	Qingzheng Road Ltd.	10	Bridge hazards treating
4		11	Bridge inspection
5		13	Bridge design and construction
6	Bureau of Public Roads of Chengdu	15	Bridge maintenance scheduling
7		10	
8		11	
9	Chongqing University	10	Project planning and management
10	Industrial Technology Research Institute	10	

3.4.2.3 Focus group data collection and analysis

The focus group lasted 120 minutes including three stages. In the first stage (25 minutes), the moderator (i.e., the researcher) introduced the purpose, topic and sub-topics, and ground rules of discussion, such as the equal status of each participant, allowance to provide suggestions and doubts, and confidentiality of the discussion results. These rules could help the participants stay relaxed, making the discussion proceed naturally, maximising the will of sharing ideas, and facilitating the collection of in-depth domain knowledge (Ho, 2006).

For each sub-topic, the researcher prepared initial results, e.g., the lists of typical classes of constraint entities and preliminary hierarchies. These initial results were obtained by reviewing 11 bridge rehabilitation manuals and 52 bridge maintenance reports in China, North America, and Australia because these countries have a large number of bridges as well as rich experience of bridge maintenance (Frangopol & Bocchini, 2012). In the second stage (60 minutes), hard copies of the initial results were presented to the experts who were then asked to provide advice on adding, deleting, and modifying classes and expanding or narrowing down hierarchies (i.e.,

increasing or reducing the levels in the initial hierarchies). Experts' advice was written down in the hardcopies. Any participant can raise questions and concerns about the initial results. Free discussion among the researcher and participants was encouraged and moderated by the researcher, where any participant could express opinions about questions raised by other participants. In the last stage (40 minutes), the researcher summarised the findings in the previous discussion and collected the views from all participants. When contradictory views appeared, the researcher asked for opinions from all experts to achieve an agreement.

Three ways were adopted to collect data in the focus group. First, the hardcopies were collected, which recorded specific ideas of each expert. Second, the researcher used memories to quickly capture important quotes and ideas of each speaker during free discussion. Such notes could significantly help the researcher to summarise the ideas of different experts and gain an overall understanding. Finally, an audio recorder was adopted to record the 120 minutes' focus group, which were translated into texts and saved in .doc files (O. Nyumba et al., 2018). The initial classes of constraint entities, relation descriptions, and hierarchies of classes and relations were refined based on summarising all data obtained in the focus group.

3.4.3 KRL model (Step 2-3)

The KRL model is responsible for extracting relations (i.e., triples) among constraint entities. The model includes four key parts, i.e., a class mapping module, a synonym mapping module, a TransE initialisation model, and a CNN-based KRL model.

3.4.3.1 Data inputs and outputs

A KRL model often takes pairs of entities as inputs and then extracts valid triples as outputs. Candidate triples can be formed by establishing different types of relations between an entity pair. As mentioned, the research concerns seven relation types for AWP modelling, while the KRL model can extract three of them: *c2c*, *c2t*, and *c2a*. On the other hand, task/procedure dependencies (*t2t*) and entity-package relations (*c2p*) are set up using rules. For one thing, the number of training samples of *t2t* and *c2p* relations (i.e., sentences implying the relations) is often much less than that of *c2c*, *c2a*, and *c2t* relations. For another, task/procedure entities are usually mentioned in separated sentences but still hold dependencies. Current DL models are not good at extracting relations from such separated data (Goodfellow et al., 2016). Finally, the package-package (i.e., *p2p*) and constraint/task/procedure-participant (i.e., *ct2pp*)

relations cannot be extracted automatically thus need to be manually inserted. Such relations generally rely on project properties (e.g., the project scale and type) hence are difficult for DL models to capture common patterns (Wang, 2018). The classes of constraint entities (obtained through the focus group) are also important inputs for the KRL model. The classes can provide the model with additional domain information thus facilitating training (see the next section and Section 4.4 for more details).

3.4.3.2 Overall design of the relation extraction model (RM3.2)

To achieve higher triple extraction performance in the bridge rehabilitation domain, specific information (e.g., domain classes of entities) is utilised in the KRL model. Besides, there are always entities that do not exist in the training dataset. The issue should be addressed, otherwise, it can largely damage model performance when the model encounters entities never seen during training. Finally, it is essential to have a good initialisation method to generate initial embeddings of triple elements (h, r, and t), which can also affect model performance. As such, three supplementary modules are created for the proposed KRL model.

First, a class mapping model is developed. For entities in an entity-pair, the model maps them to domain classes (e.g., mapping 'crane' to the class 'Equipment'). Second, a synonym mapping module is created. This module is employed during model testing only, which maps an unseen entity to an existing one in the training set according to the cosine similarity between the names of the two entities. Finally, a TransE model, a simple but effective model in the early studies of relation extraction, is applied to initialise embeddings of triple elements, which can accelerate KRL model training (Dettmers et al., 2017; Wang et al., 2014). Specifically, when a candidate triple is inputted, the classes and synonyms (when necessary) of entities in the triple are identified. The embeddings of entities, classes, and relations are extracted, which are initialised by the TransE model. The core of relation extraction, the KRL model, is based on a CNN structure. Above embeddings are concatenated as an input matrix so that CNN filters can extract features/patterns of the input triple by convolution and pooling operations. The operations produce a vector representing the validity of the triple, which is compared with the true triple label to train the model (i.e., updating model parameters and triple element embeddings) using backward propagation. The detailed working mechanism of the model is introduced in Section 4.4.

3.4.3.3 DL model experiments (RM6.1)

The DL experiment process of the KRL model is similar to that of the Bi-LSTM-CRF model introduced before, while the differences are summarised below.

(1) Experiment data collection, pre-processing, and labelling

As mentioned, the AEC industry does not have large KBs that contain triples to train the KRL model. Thus, training triples were generated based on extracted entities. It should be noted that the triple generation process only produced *c2c*, *c2a*, and *c2t* triples because the KRL model only supports automatic extraction of the three types of relations. During training and validating the KRL model, triples were generated using entities manually tagged in sentences to ensure the correctness of data. After manual tagging, vocabularies (e.g., ***V1*** and ***V2*** introduced in Table 4-1) were created to record unique entities and relations, respectively. During model testing, the triples were created using entities extracted by the Bi-LSTM-CRF model automatically. This could reflect the situation when implementing the model in practice. Nevertheless, triples for training and testing the KRL model were created by traversing entities and setting-up relations among them. Specifically, each CONS entity should set up all three types of relations with all other entities. As such, $n^c(n^c-1)+n^cn^a+n^cn^t$ triples can be generated, where the n^c, n^a, and n^t are the number of unique CONS, TP, and AT entities, respectively.

The generated training triples were manually labelled as valid or invalid. However, not all triples were included for training. The traversal process can generate a lot of triples, and manually labelling all of them is very time-consuming. Fortunately, some triples are too simple for the model to learn valuable patterns, as they are against common sense in construction projects (e.g., 'paving constrains asphalt') (Zhang et al., 2018b). In addition, the validity of some triples can be affected by construction stages thus can be ambiguous. For instance, the triple 'steel constrains scaffolding' is valid during temporary facility construction, as erecting scaffolding requires steel, however, the inverse (i.e., 'scaffolding constrains steel') is valid in subsequent project stages, as scaffolding can vertically transport materials. Correctly recognising working stages requires interpreting the graph topology (e.g., if a 'scaffolding' entity is linked to the entity 'building temporary facilities' in the graph) (Nguyen, 2020; Zhou et al., 2018). Such ambiguity is considered and addressed in the KBC model introduced in Section 3.6.2, where the entire graph has been built and the topology is available. However,

for relation extraction, such issues are left for the future. In experiments, unrealistic and ambiguous triples were deleted, and a proportion of the remaining triples were randomly sampled and manually labelled for training the KRL model.

(2) Training, validation, and testing protocols

The KRL model extracts relations through binary classification, i.e., by classifying triples as valid and invalid. Thus, the model again adopted the Pr, Re, and F1 scores for evaluation in experiments. Other model training, validation, and testing protocols are similar to those developed for the Bi-LSTM-CRF model.

(3) Data splitting and experiment process

The data splitting process is the same as that for the Bi-LSTM-CRF model. However, the KRL model utilises domain information to improve performance. Thus, several rounds of experiment were carried out to compare the performance metrics under different model settings in the testing dataset, including 1) whether the domain class information was added to the model structure, and 2) the domain class information was stacked horizontally or vertically. More details can be found in Section 4.5.2.

(4) Model hyper-parameter tuning methods

The hyperparameter tuning process is again similar to that for the Bi-LSTM-CRF model, except that some specific hyper-parameters can be different. More details are introduced in Section 4.5.1.

3.4.3.4 Controlled experiments (RM6.2)

To verify the proposed information management approach in practice, including the hybrid IE model (the Bi-LSTM-CRF and KRL model), the ontological KBs (i.e., BRMO), and the KBC model. Two concrete-reinforced bridge rehabilitation projects were selected to conduct controlled experiments. The first project was carried out on a bridge located in Zhejiang, China. It is a beam bridge (415m long and 42m wide) and three tasks were carried out: 1) fixing concrete cracks, 2) replacing deck pavement with modified asphalt, and 3) reinforcing bridge piers using concrete wrapping. The second project was carried out on a cable-stayed bridge (160m long and 19m wide) in Luohe, China. The main task was to replace damaged cables.

To demonstrate the usefulness of the hybrid IE model, the experiments compared the time to draw AWP graphs using the traditional manual approach and the Bi-LSTM-

CRF and KRL model. The input data were meeting records of the deck pavement replacement task in the first case project. To obtain fair results, the experiment was conducted by the researcher and one of his colleagues. The colleague is also a PhD student in the field of infrastructure engineering and has relevant domain knowledge. All the experiment data, activities or tasks, equipment (i.e., computer hardware and software), and environment (i.e., location and time) were the same for the researcher and the colleague, except that the researcher carried out AWP modelling using the hybrid IE model while the colleague relied on the manual approach. More details of the experiments are introduced in Section 4.5.3.

3.5 Ontology development (Objective 3)

3.5.1 Domain knowledge collection (Step 3-1 & RM2)

The ontologies should be built on a comprehensive collection of domain knowledge of bridge rehabilitation, i.e., domain classes of constraint entities and relations (called properties in ontologies) as well as the hierarchies (i.e., taxonomies) to organise the classes and relations. In this research, the domain classes, relations, and hierarchies were first extracted by reviewing relevant documents of bridge rehabilitation, e.g., working plans, meeting records, academic articles, standards, manuals, and case reports. The initial knowledge was refined through the focus group introduced in Section 3.4.2.

3.5.2 Ontology development steps (Step 3-2 & RM4.1)

The ontologies were constructed following the ontology development 101 guideline proposed by Stanford University owing to its wide recognition and clear instructions (El-Diraby, 2013; El-Gohary & El-Diraby, 2010; Ren et al., 2019; Stanford, 2002). Figure 3-2 shows the steps suggested in the guideline.

(1) Step 1 determine ontology domain and scope

The first step should define the domain and scope of the ontologies (i.e., BRMO), which can be achieved by answering the following fundamental questions:

Q1: What domain does the BRMO cover?

A1: The domain is bridge rehabilitation. The BRMO should cover rehabilitation tasks and procedures (a task includes several procedures whereas the same procedure can

be required in different tasks), constraints, constraints' attributes, as well as project participants.

Q2: For what purpose the BRMO is used?

A2: The BRMO intends to integrate constraint information in bridge rehabilitation projects, which also enables essential project management functions using semantic reasoning, e.g., evaluation of work progress, constraint statuses, and performance of participants.

Q3: Who can use and maintain the BRMO?

A3: The main user is the management team of the bridge rehabilitation project, and other participants, e.g., the bridge owner and maintenance team, also have access.

Q4: What are the sources for developing the BRMO?

A4: Bridge rehabilitation standards and manuals, case reports, project documents (e.g., work plans, schedules, and meeting records), and experts' opinions obtained by the focus group are the main sources.

Q5: What types of questions can the BRMO answer?

A5: The BRMO can answer questions that a bridge rehabilitation project manager may ask, such as the progress of tasks/procedures, reasons of delay, constraint statuses, critical constraints, participants and their performance, and relevant knowledge to address constraints that are not timely removed.

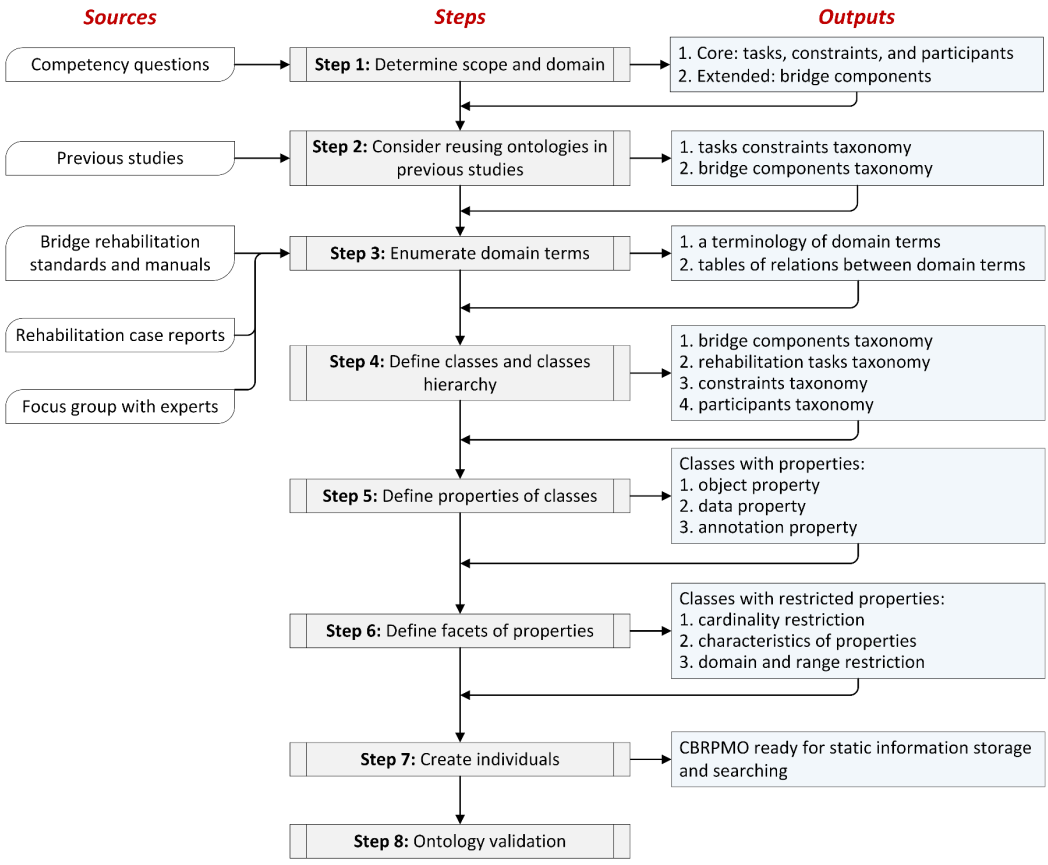


Figure 3-2 Steps to develop the BRMO

(2) Step 2 consider reusing existing ontologies

Reusing existing ontologies can save much time to develop the BRMO from scratch. During the ontology development process, several ontology libraries were searched, e.g., the Ontolingua, DAML, and DMOZ, while no relevant ontologies were found. Current bridge maintenance ontologies focus on static information of components rather than dynamic constraint information in rehabilitation projects. These ontologies were not taken as reference in this thesis (Liu & El-Gohary, 2017a; Ren et al., 2019). In contrast, some ontologies include common taxonomies of constraints and tasks or procedures in conventional construction projects (El-Diraby, 2013; El-Gohary & El-Diraby, 2010; Wang, 2018; Wang et al., 2016), which could fit the scope of BRMO hence were adopted for developing the ontologies.

(3) Step 3 enumerate domain terms

In this step, critical terms, i.e., domain concepts of bridge rehabilitation, should be identified, including tasks/procedures, constraints, constraints' attributes, and project

participants. In this research, the focus group results were taken as a reference during the process.

(4) Step 4 define a class hierarchy

In this step, domain classes are extracted from the above critical terms and class hierarchies should be developed using a mixed extraction approach. The most salient classes are extracted first, which can be generalised or specialised. For instance, when the class 'Deck System Replacement' is extracted, the term 'Replacement' is extracted as its super-class, while the terms 'Pavement Replacement' and 'Auxiliary System Replacement' are extracted as its sub-classes. It should be noted that when developing the hybrid IE model, the constraints' attributes were treated as domain classes (see Chapters 4 and 6). This is different from conventional ontologies where attributes are treated as simple literals but still does not violate the OWL syntax.

(5) Step 5 define properties

Relations are also called properties in the sector of ontologies, including two main types: the object and datatype property. Object properties refer to the relations among ontological instances (entities), such as the 'is-constrained-by' relation setup between two constraint entities. On the other hand, datatype properties link instances to their quantitative or qualitative attributes, for example, all constraint entities have a 'has-planned-removal-date' property. Definitions and descriptions of properties should be determined in this step. When developing the BRMO, the refined relation hierarchies obtained in the focus group were utilised for defining properties.

(6) Step 6 define facets of properties

Facets (e.g., characteristics and domain and range restrictions) should be defined to enrich property semantics. There are many characteristic types, in general, properties can be normal (i.e., no characteristics), functional, symmetric, asymmetric, transitive, and irreflexive. Functional properties have and only have one value as the object. For instance, the 'has-actual-removal-date' property is functional, because a constraint can only be removed once. Symmetric properties are equivalent to their inverse, whereas asymmetric properties do not have this feature. For instance, the property 'is-close-to' can model the spatial relations among components and is a symmetric property. If a subject 'is-close-to' an object, the object 'is-close-to' the subject as well. Transitive properties can propagate among instances and form a chain. One typical transitive

property is the 'sub-class-of', i.e., if *A* is the sub-class of *B* while *B* is the sub-class of *C*, then *A* is the sub-class of *C*. Reflexive properties allow an instance to be linked to itself through the relation, whereas this is forbidden in irreflexive properties. Object properties should consider all the characteristics whereas datatype properties mainly concern functionality. On the other hand, domain and range restrictions specify the allowed classes of the subject and object in a relation, respectively. The values of domain and range are classes and data types (e.g., integer and double) for object and datatype properties, respectively. For instance, the domain of 'has-planned-duration' and 'is-constrained-by' properties can be class 'Date' and 'Constraint', respectively. With such restrictions, properties (i.e., relations) can be only valid when they are set up among certain classes. For instance, a material cannot be linked to a procedure using the 'manage' property if the domain of the property is the 'People' class. The restrictions also enhance reasoning. For instance, given the domain of the property 'manage', it is easy to infer that an entity belongs to the 'People' class if it manages other entities.

(7) Step 7 create instances

Instances can be generated by mapping ontological classes to real-world things. For AWP constraint modelling, instances include constraints, attributes of constraints, tasks/procedures, as well as project participants. The number, name, and properties of instances should be determined before instance creation. The number of instances often relies on the complexity and scale of the project. The naming convention is flexible but should be consistent. Finally, properties among instances should comply with the definitions of their belonging classes (e.g., the domain/range restrictions).

3.5.3 Ontology information encoding and updating (Step 3-3 & RM4.2)

The triples extracted by the hybrid IE model are initially stored in .csv files. Thus, a method should be developed to encode the triples in ontologies. On the other hand, conventional ontologies do not support complex computation and updating which are however important to manage construction projects. The proposed ontological KBs combine the OWL API with SWRL and SQWRL rules to address the issue. The OWL API is an interface supporting manipulation of ontologies, i.e., exporting, importing, adding, deleting, and modifying ontology information. The OWL API features two advantages compared to other APIs, e.g., the Apache Jena. First, the development of the API is closely related to the OWL syntax hence is more compatible with current

ontologies. Second, the OWL API is applied at the axiom level whereas other APIs are applied at the triple level. In ontologies, an axiom is a logic statement and can include multiple triples. Therefore, the axiom-centric design can better isolate users from bottom-level operations, e.g., serialisation and parsing of triples (Horridge & Bechhofer, 2011).

3.5.3.1 Information encoding

When encoding constraint information in the ontological KBs, the OWL API iterates rows in the .csv files. The general working mechanism of the API is as follows. For each row, the API creates an ontological instance (i.e., an entity) with the name of the first element (i.e., the subject) in the row and then reads the second element (i.e., the relation). If the relation is 'is-a', the OWL API finds the domain class using the name of the third element in the TBox. Then, it creates a class assertion axiom to assign the instance to that class. If the relation is not 'is-a', there are two situations. If the third element (i.e., the object) is not an attribute, the API creates another entity with the name of the object element then creates an object relation assertion axiom, specifying the two entities are linked through the relation element. Otherwise, a datatype relation axiom is generated to link the first element to the third element (i.e., its attribute). The encoding is enabled by built-in functions of the OWL API (Horridge & Bechhofer, 2011). Figure 3-3 shows an example of the information encoding process.

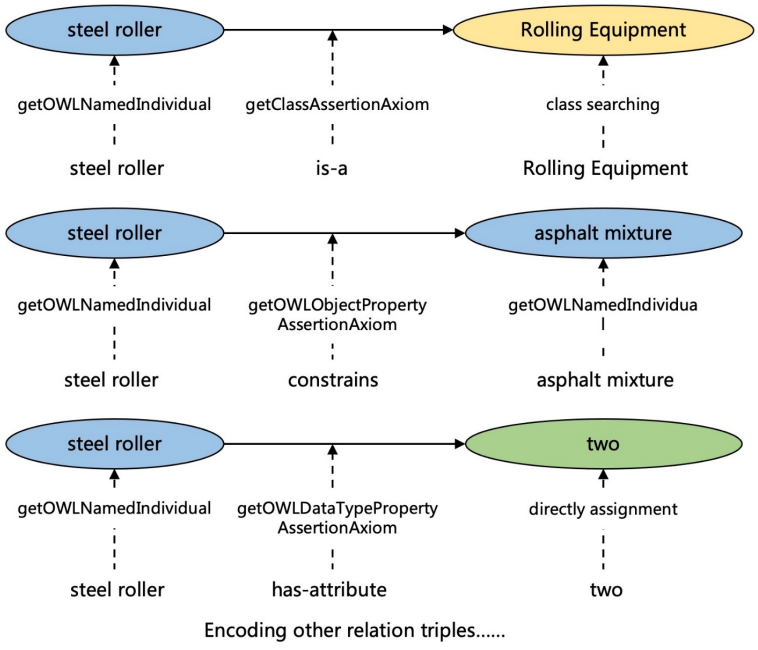


Figure 3-3 An example of ontology information encoding

3.5.3.2 Ontology computation and dynamic updating

To support complex computation and updating, the OWL API exports information (i.e., axioms) out of the ontologies (i.e., BRMO) for programmatical modification. On the other hand, SWRL and SQWRL can express complex and rich semantics using ontological rules. Compared to the OWL API working well out of ontologies, SWRL and SQWRL are effective in terms of inferring information within ontologies. Once new information is computed, the OWL API imports the information back into the ontologies. The reasoning rules are then enabled to infer new and implicit knowledge based on the updated information. The overall mechanism is shown in Figure 3-4. Thus, the BRMO can be continuously updated, enriched, and reasoned to integrate static and dynamic project information. The BRMO also supports three management functions: 1) evaluation of work progress, 2) evaluation of constraint statuses, and 3) evaluation of the performance of participants. Detailed workflow and reasoning rules in the computation and updating process can be found in Section 5.3.

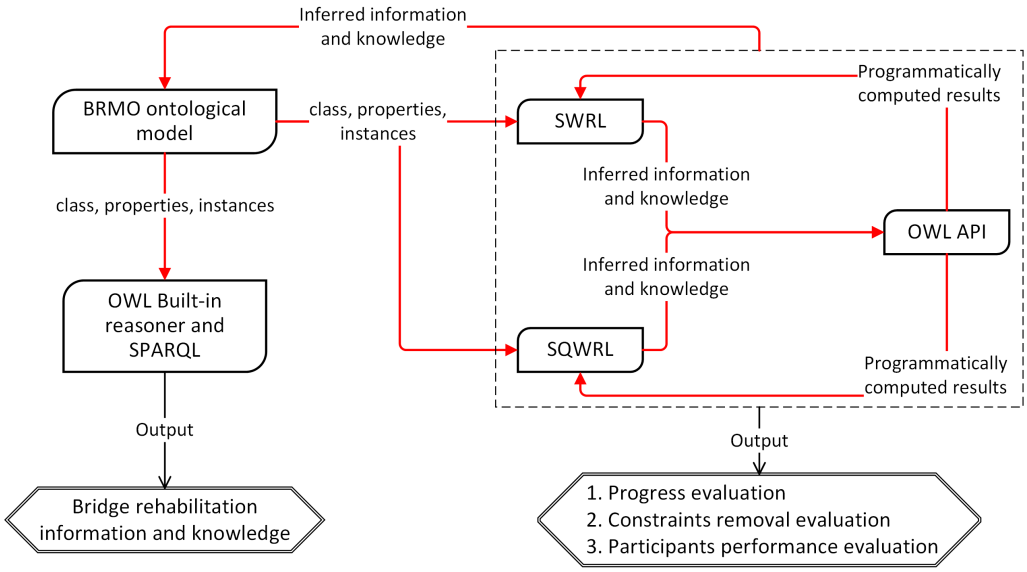


Figure 3-4 Overall workflow of BRMO updating

3.5.4 Ontology validation and controlled experiments (Step 3-3 & RM6.2)

Ontology validation should concern the semantic and syntactical correctness of an ontology. Semantic validation is often conducted by asking competency questions, consulting experts, and merging and comparing the newly developed ontologies with existing ones. For BRMO, there are no similar ontologies, hence, only the first two types of validation were conducted. Asking competency questions is a simple way to self-check semantics in ontologies (Stanford, 2002). The questions should echo the

questions in A5 of Step 1 of ontology development while covering ontological classes, instances, and relations, such as 1) how many sub-classes of a certain constraint class, 2) what are the relations among certain constraint entities, 3) what are the constraints of a certain task/procedure, 4) which tasks/procedures have been delayed and how severe the delay is, and 5) who is the participant with the best performance in terms of removing constraints? Artificial instances can be created to check if the ontologies contain enough information to answer the questions. During ontology development, self-checking was performed periodically to maximise the semantic validity of the BRMO. On the other hand, syntactical validation evaluates the ontologies against the underlying OWL syntax, e.g., subsumption, equivalence, and consistency. Syntactical validation can be conducted with specialised reasoners which can detect syntactic errors automatically. The introduced OWL API also has such reasoners for ontology checking. Thus, whenever the BRMO was modified in development and validation, the reasoners were launched to ensure the BRMO could pass the syntactical validation and was ready for further modifications (Antoniou & Van Harmelen, 2012).

The BRMO is used to integrate constraint triples extracted by the hybrid IE model, and its usefulness was demonstrated in controlled experiments. The experiments were also conducted by the researcher and colleague, including two aspects. First, the time to search for constraint information was compared. The researcher searched for the information by navigating the BRMO using SPARQL queries, while the colleague performed the same task by reviewing text documents manually. Second, the three management functions of the BRMO were tested, i.e., evaluation of work progress, constraint statuses, and project participants' performance. The experiments can prove the usefulness of the ontological KBs in terms of effectively exporting, importing, computing, reasoning, and updating constraint information in ongoing projects. More details of the experiments are introduced in Section 5.4.

3.6 Knowledge base completion model design (Objective 4)

The KBC model can predict missing triples in AWP KBs, which includes three key parts, i.e., an ontology-based data enriching module, a GNN-based encoder, and a KRL-based decoder.

3.6.1 Data inputs and outputs

The proposed KBC model takes the entire project KB generated by the hybrid IE model (i.e., the Bi-LSTM-CRF and KRL model) as the input. The outputs are missing information (i.e., missing triples) in the KB. In other words, the outputs form a more complete KB. Figure 3-5 presents a simple example of applying the KBC model, where the triple 'asphalt is-required-by paving' does not exist in the original KB but is predicted by the model.

3.6.2 Overall design of the KBC model (Step 4 & RM5)

3.6.2.1 Semantic data enriching (RM5.1)

The initial KBs generated by the hybrid IE model often lack adequate semantics for training the KBC model, which can largely hurt KBC performance. Therefore, a data enriching module based on the ontological KBs is created to enrich data semantics. The main method adopted by the module is to develop SWRL rules to 1) infer new triples using existing ones in the KBs, and 2) enrich semantics of triple relations, i.e., inferring relations with rich semantics, such as inferring relations 'supply-power-to' and 'works-in' rather than using the simple relation 'constrains'. The enriched KB (i.e., a knowledge graph) is then fed into the GNN-based encoder.

3.6.2.2 Encoder-decoder design (RM5.2-5.3)

(1) Encoder design (RM5.2)

The encoder carries out three tasks: sampling, aggregation, and updating (Figure 2-3 shows the process). It takes the information of constraint entities, connections among entities (i.e., the graph topology), and domain class information of entities as inputs and learns new embeddings of constraint entities and relations. During sampling, the encoder considers the nodes and edges in the one-hop neighbourhood of each node. Then, the aggregation process summarises the embeddings from the nodes and edges linked to the central node, where the attention mechanism is employed to weight the embeddings and integrate them as a new embedding for the central node. During the updating process, the new embedding replaces the node's original embedding. In the encoding process, 1) domain classes of entities are included as additional nodes so that the aggregation process can consider class information; 2) the original embedding of a node is added to its new embedding during updating, which can keep the original meaning of the node (e.g., the semantic meaning of the node's name).

(2) Decoder design (RM5.3)

The decoder can predict a missing entity or relation in a triple given the other two elements are known (e.g., predicting *t* given *h* and *r*). The decoder is based on CNN and its structure is similar to the KRL model for triple extraction. The model 1) takes the embeddings of the two known triple elements produced by the GNN encoder, 2) enumerates all other entities/relations in the KB to replace the missing element which is represented by a symbol '?', 3) computes the validity of all candidate triples (a candidate triple is formed by the entity/relation replacing the '?' and two known elements) and selects the triple that is most likely to be valid to complete the KB. As such, the KBC model can produce a descending list of triples, and triples with high validity are ranked at the top of the list. To further improve the model performance, another type of domain information, i.e., the working contexts of entities, are added to the CNN structure. Working contexts are the tasks/procedures that a constraint entity constrains. Working contexts are considered as they can affect the relation direction in a triple, which is introduced in Section 3.4.3.3. For instance, the working context of the entity 'asphalt' in bridge rehabilitation projects is paving and rolling. More details can be found in Section 6.2.

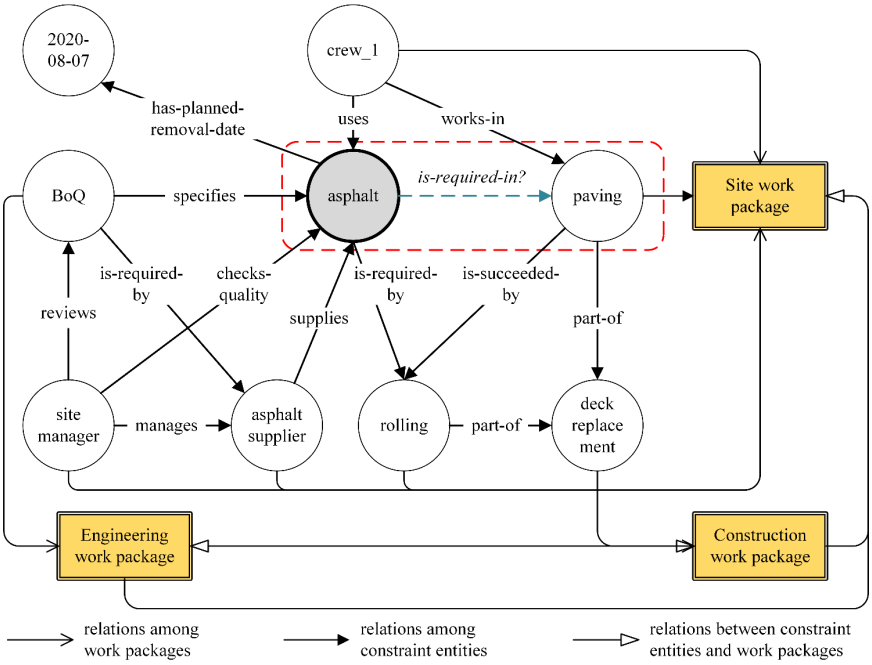


Figure 3-5 A simple example of KBC

3.6.3 DL model experiments (Step 4 & RM6.1)

(1) Experiment data collection, pre-processing, and labelling

Triples stored in the ontological KBs were gathered for training and testing the KBC model, which were all assumed to be valid. However, training needs both valid and invalid data. Thus, invalid training triples were generated by replacing the head or tail element in a valid triple with a randomly sampled entity in the KBs. As such, unlike training the KRL model, training the KBC model does not require data labelling, as labelling can be automated in the process of generating invalid triples.

(2) Training, validation, and testing protocols

In experiments, the KBC model was also trained and tuned using the training and validation datasets (the cross-validation was applied) before it was evaluated in the testing dataset. The KBC model produced a list of candidate triples with the true missing triples being ranked higher. Hence, performance metrics for evaluating the KBC model were different from those for conventional classification tasks, and the Hit@1, Hit@3, Hit@10, and mean rank (MR) were adopted. The three Hit@ metrics can indicate the percentage that the ground true missing triple is the first, within the first three, and within the first ten triples in the candidate list, respectively, while the MR indicates the rank of the ground true triple in the candidate list.

(3) Data splitting and experiment process

The 7:2:1proportion was again used to split the training, validation, and testing sets. Different settings of the KBC model were compared through experiments to reveal the effect of adding domain information (i.e., classes and working context information), which covered three situations: 1) whether data semantic enriching was applied, 2) whether domain class information was used in the encoder, and 3) whether working context information was used in the decoder.

(4) Model hyper-parameter tuning methods

The hyperparameter tuning process is similar to those for the Bi-LSTM-CRF and KRL models, except that some hyperparameters can be different. Specific hyperparameters and the tuning results are introduced in Section 6.3.1.

3.6.4 Controlled experiments (Step 4 & RM6.2)

The experiments compared the time and accuracy to check and complete incomplete KBs. The KBs of the two case projects (i.e., the deck replacement task and cable replacement task) were developed. Then, some triples were artificially deleted to make the KBs incomplete. The researcher and the colleague were informed of the types of

missing information (e.g., finding missing constraints of a constraint). The researcher and the colleague checked and completed the KBs using the KBC model and manual approach, respectively. The colleague could review the task working plans to obtain additional information. The time to find out all missing information was recorded for comparison. In addition, Pr, Re, and F1 score were used to evaluate the identified missing triples. The three metrics were computed by Eq. 3-4 - Eq. 3-6, where CIT, WIT, and MT are the correctly identified triples, wrongly identified triples (i.e., false positives), and missed triples (i.e., false negatives), respectively. More experiment details are introduced in Section 6.3.3.

$$Precision = {CIT}/{CIT + WIT} \quad \text{Eq. 3-4}$$

$$Recall = {CIT}/{CIT + MT} \quad \text{Eq. 3-5}$$

$$F1 = {2 \times Precision \times Recall}/{Precision + Recall} \quad \text{Eq. 3-6}$$

3.7 Chapter summary

This chapter summarises the research methodology. First, the research philosophy is introduced as the foundation of the thesis. The research belongs to the post-positivism paradigm. The research methodology is mainly deductive and quantitative, which is based on objectivism epistemology and realism ontology. However, subjective and qualitative methods (i.e., focus group) are also employed to obtain relevant domain knowledge. Sections 3.4-3.6 introduce specific research methods. In summary, the Bi-LSTM-CRF and KRL model are utilised to extract constraint entities and relations, which produce constraint triples (Section 3.4). Then, the ontological KBs are built to integrate these triples, which also support information computation, reasoning, and updating of constraint information (Section 3.5). Finally, a KBC model is developed to handle the incompleteness of ontological KBs, which can predict missing triples and continuously enrich the KBs (Section 3.6). By applying the three components, AWP modelling in bridge rehabilitation projects can be largely automated, and useful information can be timely integrated to assist project management.

Chapter 4: Developing automatic methods for constraint information extraction

4.1 Chapter introduction

This chapter presents the detailed model design and experiment results of the hybrid IE model for entity extraction (Sections 4.2-4.3) and triple extraction (Sections 4.4-4.5). Cross-comparison results are demonstrated to prove the usefulness of the model and effect of adding class information to the KRL model. Contributions of the hybrid IE model are discussed in Section 4.6. All the models were developed with Python 3.7, Tensorflow (1.14.0), and Keras (2.2.0) on a Mac machine equipped with a 2.3 GHz Intel core processor and 64 GB random-access memory.

4.2 Detailed design of the Bi-LSTM-CRF model

Extracting constraint entities is realised by the Bi-LSTM-CRF model. This section describes the detailed model design.

4.2.1 Word/character embeddings

In early NLP studies, words/characters are usually encoded in one-hot vectors. For instance, if the number of all unique words/characters in the entire dataset is ***N***, then a matrix $\in \mathbb{R}^{N\times N}$ is created. Each column in the matrix is a one-hot vector $o \in \mathbb{R}^{N\times 1}$ of a word/character, where all elements are 0 except the one denoting the position of the word/character in the dataset (Figure 4-1). However, one-hot vectors are too sparse to train DL models. Hence, modern NLP models adopt word/character embeddings, and the Bi-LSTM-CRF and KRL models are no exceptions.

Word/character embeddings are low-dimension (e.g., 50-300) real-valued numerical vectors which can represent the semantics (i.e., meanings) of words/characters. Based on the embeddings, a matrix $\boldsymbol{E} \in \mathbb{R}^{N\times n}$ is created (**n** is the embedding dimension). A one-hot vector o of any character/word can be transformed to its embedding vector e by Eq. 4-1. A sentence can be represented by a matrix $\boldsymbol{I} \in \mathbb{R}^{m\times n}$ by concatenating word/character embeddings (m is the number of characters/words in that sentence). Then, the ***I*** matrix can be fed into downstream DL models (e.g., the Bi-LSTM-CRF model). An example of embedding transformation is shown in Figure 4-1.

$$e_t = (o_t)^T E \ (t = 1,2 \dots m) \quad \text{Eq. 4-1}$$

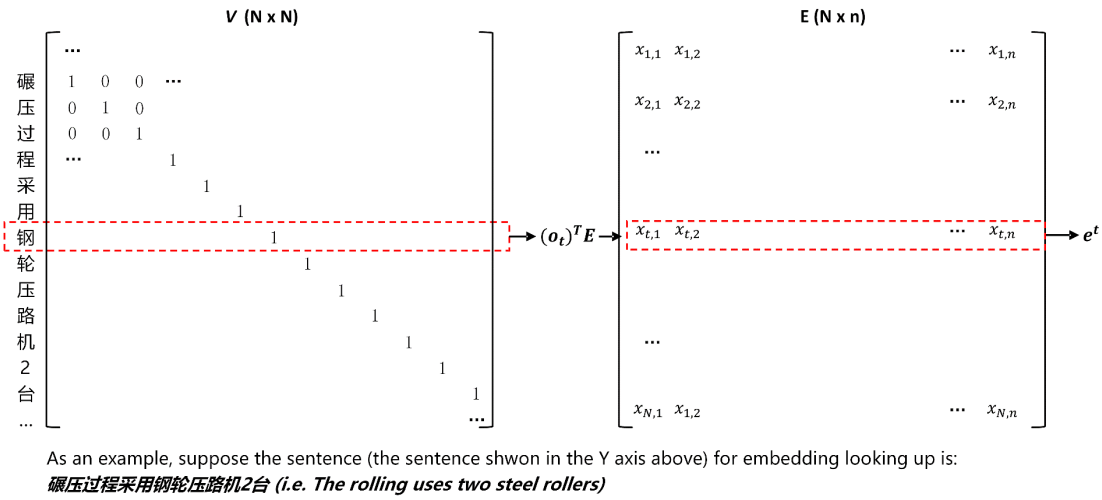


Figure 4-1 An example of embeddings transformation

Word/character embeddings need to be learned by ML models, such as the widely recognised *word2vec* model (Rong, 2014). However, training such models requires numerous data and has excessive demands on computation power. Therefore, it is common to use embeddings trained by others (Baker et al., 2019). In this research, the pre-trained embeddings were collected from the studies conducted by Li et al. (2018) and Pennington et al. (2014). The embeddings (n=300) were trained on well-known databases (e.g., Baidu encyclopedia and Wikipedia), covering over 653473 Chinese characters and 400000 English words, respectively.

To assist in understanding the following contents, Table 4-1 lists important embedding matrices and vocabularies in the hybrid model. N^{wc}, N^E, and N^R are the number of unique characters or words, entities, and relations, respectively. K^{wc} and K^t are the dimensions of embeddings of words/characters and entities/relations, respectively.

Table 4-1 Vocabularies and embedding matrices for model training

Name	Shape	Embedding	Usage	Development methods
V1	$(N^E, 1)$	n/a	Store the indices of unique entities and relations	Map entities and relations to unique indices
V2	$(N^R, 1)$	n/a		
WCE	(N^{wc}, K^{wc})	e^{wc}	Store the embeddings of characters and words	Employ embeddings pre-trained in other studies
EE	(N^E, K^t)	e^t	Store the embeddings of head and tail entities	Train the TransE model
RE	(N^R, K^t)		Store the embeddings of relation entities	

4.2.2 Bi-LSTM-CRF layer

The mechanism of the Bi-LSTM-CRF model is demonstrated in Figure 4-2. Given a sentence, the model reads words/characters from left and right directions, then it feeds

their embeddings into LSTM cells. Each cell corresponds to a word/character and has three gates: forget gate, update gate, and output gate. The gates compute how much information from the last cell is discarded, how much information in the current cell is added, and how much information is used for predicting entity tags, respectively. Despite the long distance, the gates can keep useful information and discard useless information. The structure of an LSTM cell is shown in Figure 4-3. A cell needs three inputs: embedding $\boldsymbol{e^t}$ of the current word/character and $\boldsymbol{a^{t-1}}$ and $\boldsymbol{c^{t-1}}$ from the last cell. The $\boldsymbol{a^{t-1}}$ and $\boldsymbol{e^t}$ are concatenated vertically to form a long vector (concatenation is presented by square brackets in equations) to be fed into the gates. The gates are also numerical vectors which are computed using Eq. 4-2 – 4-4.

Accordingly, $\boldsymbol{a^t}$ and $\boldsymbol{c^t}$ are computed by Eq. 4-5 – 4-7 which can be applied to both directions. In the equations, vectors from different directions are denoted as $\overrightarrow{\boldsymbol{a}}$ and $\overleftarrow{\boldsymbol{a}}$; the $\boldsymbol{\sigma}$ and ***tanh*** stand for ***sigmoid*** and ***tanh*** function, taking the form of $\frac{1}{1+e^{-x}}$ and $\frac{e^x-e^{-x}}{e^x+e^{-x}}$, respectively; and * and + mean element-wise multiplication and summing, respectively. All $\boldsymbol{W}$ and $\boldsymbol{b}$ are randomly initialised and continuously updated during training. For each word/character, the $\overrightarrow{\boldsymbol{a}}$ and $\overleftarrow{\boldsymbol{a}}$ are concatenated as a long vector $\boldsymbol{a}^*$ at the hidden layer in Figure 4-2. Finally, the $\boldsymbol{a}^*$ is fed into the softmax layer to predict the vector $\boldsymbol{p} \in \mathbb{R}^{1\times K}$ ($\boldsymbol{K}$ is the number of entity tags) using softmax $\frac{\exp(p_i)}{\sum_i^K \exp(p_i)}$. Thus, the output of a sentence is a matrix $\boldsymbol{P} \in \mathbb{R}^{m\times K}$, and $\boldsymbol{P_{i,j}}$ is the probability of the $\boldsymbol{i^{th}}$ word/character being predicted as the $\boldsymbol{j^{th}}$ tag.

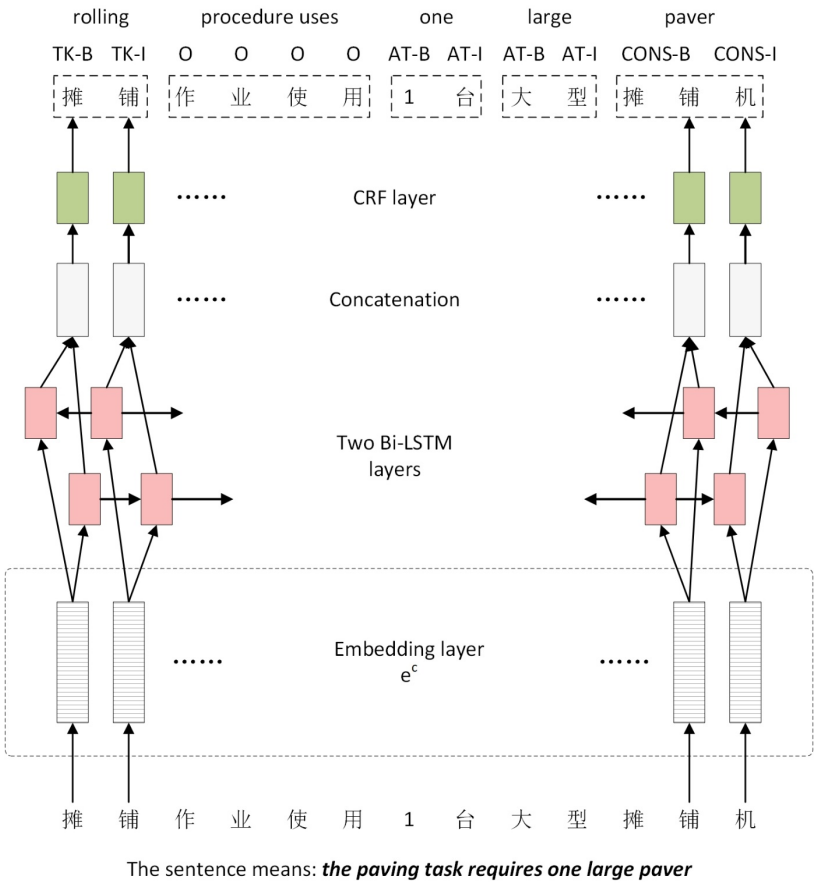


Figure 4-2 Bi-LSTM-CRF model mechanism

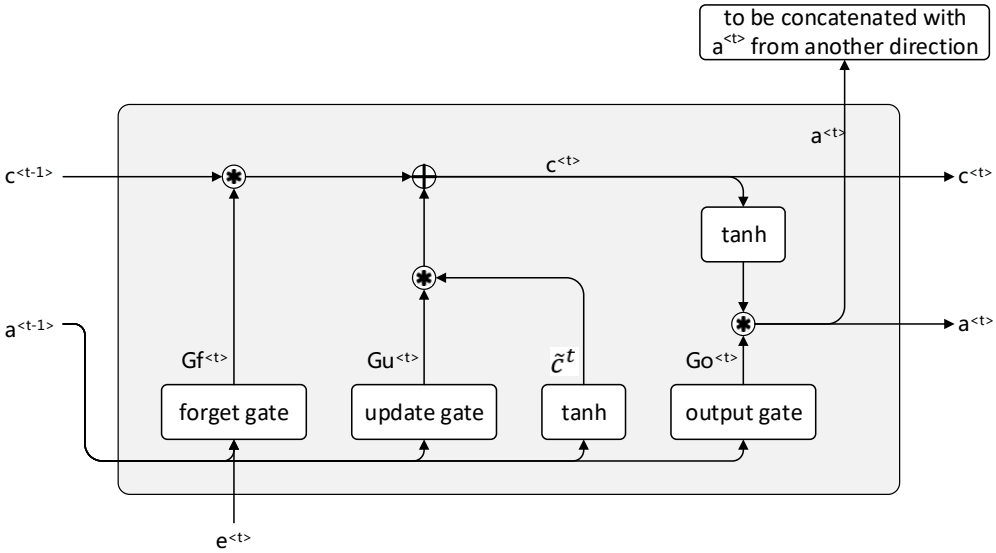


Figure 4-3 The internal structure of the LSTM cell

4.2.3 CRF layer

The CRF layer takes the ***P*** matrix and makes final predictions considering different combinations of tags. The score of a tag sequence $y = (y_1, y_2 \ldots y_m)$ can be computed using Eq. 4-8. The ***A*** in the equation is a transition matrix indicating the impacts of

adjacent tags, which is also learnable. The final tag sequence is the one gaining the highest score among all possible sequences, i.e., $\hat{y} = \boldsymbol{argmax}(s)$.

$$G_f = \sigma\left(W_f[a^{t-1}, e^t] + b_f\right) \quad \text{Eq. 4-2}$$

$$G_u = \sigma\left(W_u[a^{t-1}, e^t] + b_u\right) \quad \text{Eq. 4-3}$$

$$G_o = \sigma\left(W_o[a^{t-1}, e^t] + b_o\right) \quad \text{Eq. 4-4}$$

$$\tilde{c}^t = tanh(W_c[a^{t-1}, e^t] + b_c) \quad \text{Eq. 4-5}$$

$$c^t = G_u * \tilde{c}^t + G_f * c^{t-1} \quad \text{Eq. 4-6}$$

$$a^t = G_o * tanh(c^t) \quad \text{Eq. 4-7}$$

$$s(X, y) = \sum_{i=1}^{m} P_{i,y_i} + \sum_{i=0}^{m} A_{y_i y_{i+1}} \quad \text{Eq. 4-8}$$

4.3 Entity extraction experiment results

4.3.1 Data preparation and hyper-parameter tuning

To train the Bi-LSTM-CRF model, three standards/manuals and 31 working plans of concrete-reinforced bridge rehabilitation and general construction (in Chinese) were collected. The documents could cover common bridge rehabilitation tasks, e.g., crack fixing, deck pavement, cable replacement, and structure reinforcement. Non-relevant texts (e.g., the organisation structures of project teams) were removed, leaving 11283 sentences. Sentences including common constraint entities were filtered. This resulted in 1100 positive training samples, and the remaining sentences were treated as negative samples. For each positive sample, true tags were labelled manually as shown in Figure 4-2. To prove the generalisation ability of the model, another 3100 English sentences were collected from four bridge rehabilitation manuals. In total, 550 positive samples were tagged in these English sentences. Pre-trained word/character embeddings were collected in research conducted by Li et al. (2018) and Pennington et al. (2014). The embeddings (n=300) cover 653473 Chinese characters and 400000 English words, respectively, which are comprehensive for training the model.

It should be noted that mainstream Bi-LSTM-CRF models intend to extract general information (e.g., companies) in general texts (e.g., webpage news) (Bolucu et al., 2019; Hochreiter & Schmidhuber, 1997), where the data are often sparse, i.e., entities only appear a few times in all texts. In addition, entities can have different meanings in different contexts, e.g., ‘Apple’ can represent a fruit or a company. Thus, training

requires massive data to learn the rules for matching entities in texts. In contrast, the proposed model is designed for the construction domain. The data used are much denser, with most entities having distinct meanings. Hence, the model can effectively learn the matching rules despite the small amount of data.

The hyperparameters were tuned following the methods introduced in Section 3.4.1. The following optimal hyper-parameters were obtained: 1) for Chinese data, learning rate=0.01, batch size=64, epochs=10, and $\boldsymbol{a^t} \in \mathbb{R}^{256\times1}$; 2) for English data, learning rate=0.01, batch size=64, epochs=15, and the shape of $\boldsymbol{a^t} \in \mathbb{R}^{128\times1}$.

4.3.2 Model results and analysis

Table 4-3 lists performance metrics of the Bi-LSTM-CRF model, and Figure 4-4 and Figure-5 illustrate the confusion matrix and several examples of extracted entities, respectively. The experiment results indicate that the model can accurately extract constraint entities from texts in both languages. The 'F1-diff' indicates differences of F1 scores between the training and validation sets, reflecting the degree of overfitting (DOF). Currently, the 'F1-diff' is 6%-7%. Although there are no strict requirements for the indicator, it means the model can suffer slight overfitting. However, the F1 score in the testing dataset reaches 0.936 for Chinese data and 0.912 for English data, respectively, meaning the model can reach high accuracy and to-some-extent handle unseen texts. Moreover, overfitting can be addressed by adding more data. To further demonstrate the advantages of the Bi-LSTM-CRF model in terms of extracting entities, two classical ML models (i.e., the HMM and CRF model) were additionally trained for cross-comparison, where the tokens, part-of-speech tags, and frequencies of characters/words were extracted as features. It turns out that the Bi-LSTM-CRF model outperforms the two ML models because it achieves the highest F1 scores and lowest DOF while requiring no manual feature engineering.

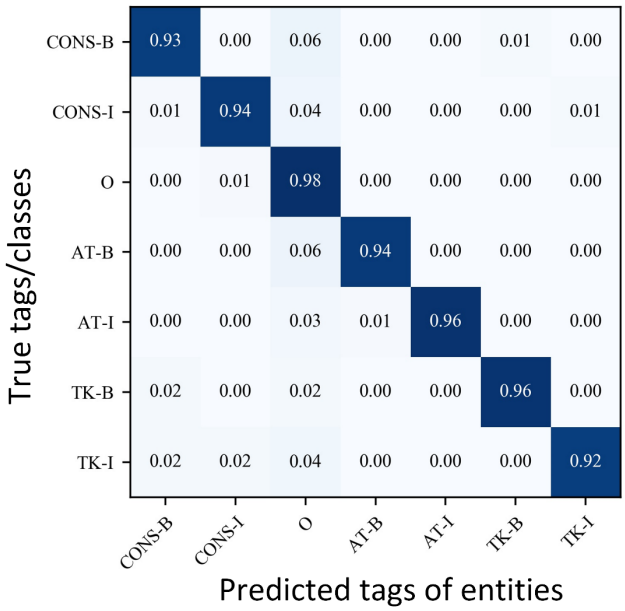


Figure 4-4 The confusion matrix of NER results

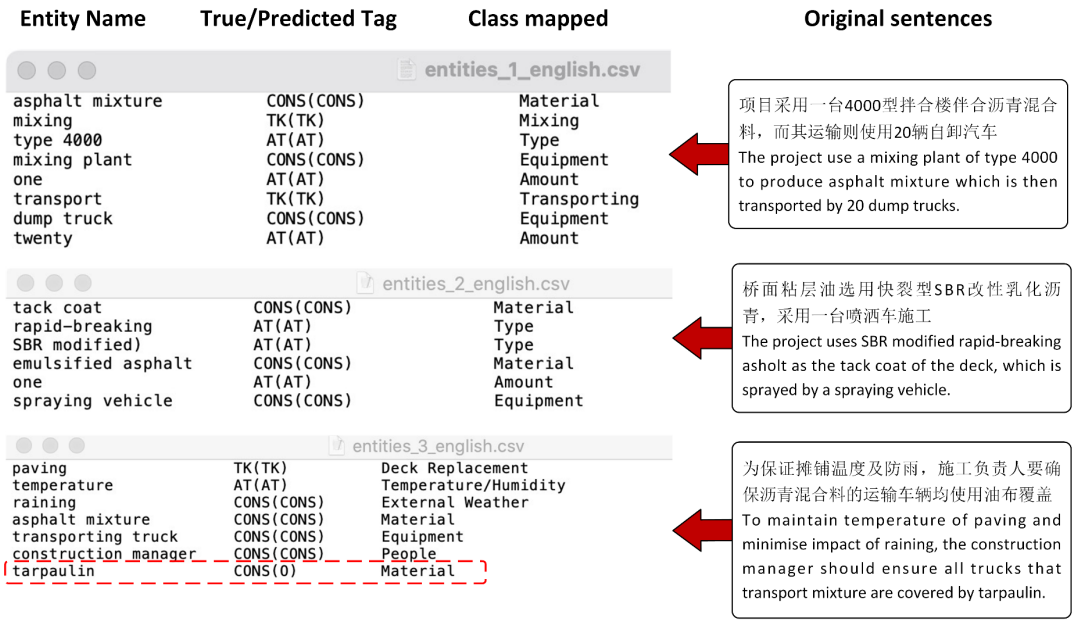


Figure 4-5 Examples of NER results

It is found that most errors of the entity extraction task are caused by classifying the three tags (i.e., CONS, TP, AT) as the 'O' tag. The wrongly recognised entities only have a few samples in the training data (for instance, 'tarpaulin' in Figure 4-5). Thus, the model cannot see enough samples to distinguish them from 'O' entities.

4.4 Detailed design of the KRL model

Relation extraction is realised by the KRL model. This section describes the detailed design of each component in that model.

4.4.1 Class mapping model

Class mapping can link head and tail entities to domain classes shown in Figure 4-6. There are usually distinct patterns when Chinese words are representing the CONS, TP, and AT entities. Hence, class mapping is realised by rule-based matching and regular expression. For CONS and TP entities, the ending characters of their names can largely determine their classes. For instance, the character '机' in '压路机' (roller) indicates an equipment entity, and the character '料' in '混合料' (mixture) indicates a material entity. As for AT entities, their units (e.g., meters) are extracted for class mapping. For instance, '8m' is mapped to 'Geometry' while '5Mpa' is mapped to 'Pressure/Stress'. The remaining AT entities containing alphabets and/or numbers are mapped to 'Type/Property' (e.g., 'C50' for concrete materials and 'SBS' for asphalt materials). To support accurate mapping in experiments, based on common words in the industry, 179, 79, and 65 ending characters were identified to map CONS, TP, and AT entities, respectively.

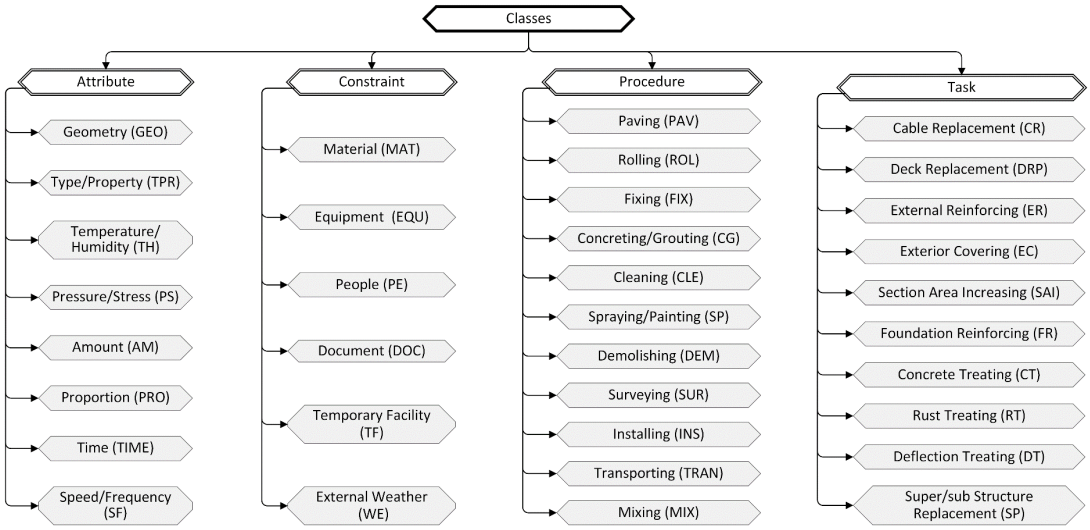


Figure 4-6 Domain classes for class mapping

Moreover, to allow the hybrid IE model to handle multiple languages, a translation mechanism is adopted. Specifically, based on civil engineering dictionaries, entities extracted by the Bi-LSTM-CRF model in other languages are translated to Chinese to make use of the distinct patterns for class mapping. Then, the entities and their classes are translated into English to keep language consistency for training the KRL model. This mechanism is applicable, as: 1) most entities have unambiguous meanings in the construction domain and can be precisely mapped to counterparts in other languages;

2) it is simple but effective (see experiment results) and does not need any additional training.

4.4.2 Synonym mapping module

Training the KRL model only requires the ***V1*** and ***V2*** in Table 4-1. Thus, synonym mapping is only implemented during model testing, where the model can encounter entities out of ***V1***. When the model meets an out-of-KB entity, it: 1) iterates existing entities in ***V1***, 2) extracts e^{wc} of each entity pair (i.e., the unseen entity and entity in ***V1***), 3) computes the cosine similarity *s* between the embeddings using Eq. 4-9, 3) sorts *s* values in a descending list. Then, e^t of the unseen entity can be filled by the average e^t of the first three entities in the list. The unseen entity is also added into ***V1*** to continuously enrich the vocabulary. However, an entity's name might not be found in the ***WCE***. For instance, '沥青混合料' (asphalt concrete) is a single entity, but the ***WCE*** only includes e^{wc} of 'asphalt' and 'concrete' separately. In this case, the entity name is divided into parts through tokenization. Then, the ***WCE*** is searched to find embeddings of all the parts. The dividing continues until a match is found for each part. Therefore, an entity can be divided into single characters, and its embedding is the average of all the found e^{wc}.

$$s = \frac{\sum_{i=1}^{K} e_{si}^{wc} \times e_{ei}^{wc}}{\sqrt{\sum_{i=1}^{K} \left(e_{si}^{wc}\right)^2} \times \sqrt{\sum_{i=1}^{K} \left(e_{ei}^{wc}\right)^2}} \quad \text{Eq. 4-9}$$

4.4.3 TransE model

TransE assumes $h + r \approx t$ for valid triples in Θ while $h + r$ is far away from t for invalid triples in Θ' (Bordes et al., 2013). The model loss can be computed by Eq. 4-10, where *d* is a dissimilarity measure (e.g., *L2*-norm) between embeddings, γ is a margin greater than 0, and $[x]_+$ denotes the positive part of x. In the model, $e^t$of all triple elements can be initialised using the uniform distribution $(-6/\sqrt{K^t}, 6/\sqrt{K^t})$ (Bordes et al., 2013). The TransE model is usually trained by comparing the true and predicted $e^t$of *t* given e^t of *h* and *r* while updating all e^t using the gradient descent approach. The TransE model is applied to initialise parameters in the proposed KRL model. The outputs of the TransE model are the ***EE*** and ***RE*** in Table 4-1, recording e^t of each entity and relation, respectively.

$$\mathcal{L} = \sum_{(h,r,t)\in\Theta\cup\Theta'} [\gamma + d(h + r, t) - d(h' + r, t')]_+ \quad \text{Eq. 4-10}$$

4.4.4 CNN-based KRL model

In existing KRL models, a triple is represented by a matrix $\boldsymbol{A} \in \mathbb{R}^{K^t \times 3}$ formed by e^t of the tree triple elements. Then, CNN filters can effectively scan $\boldsymbol{A}$ and extract triple features (Dettmers et al., 2017; Nguyen, 2020). Two options are proposed to enhance the KRL model's structure by adding domain class information (i.e., e^{wc} of the class names of *h* and *t*). First, class information is vertically stacked at the ends of e^t of head and tail entities, which reshapes $\boldsymbol{A}$ to $\mathbb{R}^{2K^t \times 3}$. Second, class information is stacked at the left and right sides of $\boldsymbol{A}$, which reshapes $\boldsymbol{A}$ to $\mathbb{R}^{K^t \times 5}$. In both cases, the enhanced model can learn features of relations among specific entities and classes. For instance, given the wrong triple 'concrete has-attribute 5km/h', the model can learn that the class 'Material' may not have attributes of the class 'Speed'. Thus, when the model encounters another material entity (e.g., asphalt), it is more likely to classify the triple 'asphalt has-attribute 3km/h' as invalid.

Following the two options, CNN filters $\in \mathbb{R}^{1\times 3}$ or $\in \mathbb{R}^{1\times 5}$ are created to extract triple features. Specifically, a filter τ is applied to every row of $\boldsymbol{A}$ using Eq. 4-11. In the equation, $*$ means the convolution operation, g is the activation function (e.g., Relu), and b is a bias (Dettmers et al., 2017). When all rows are scanned by a filter, a feature map of shape $\mathbb{R}^{K^t \times 1}$ or $\mathbb{R}^{2K^t \times 1}$ is generated. To capture more features, multiple filters (i.e., $\boldsymbol{L}$) can be applied. The feature maps are concatenated to form a feature vector *v* of shape $\mathbb{R}^{LK^t \times 1}$ or $\mathbb{R}^{2LK^t \times 1}$. Then, the vector *v* is transformed to a single score using inner product (Eq. 4-12). The mechanism is shown in Figure 4-7 ($\boldsymbol{L}$ is set as two for demonstration).

As triples with different relations can have different features, three KRL models are developed for the three relation types (i.e., *c2c*, *c2a*, and *c2t*). For training, the triple scores are compared with a threshold. A triple is valid if its score $s(h,r,t)$ is below the threshold. The model loss is computed using Eq. 4-13, based on which the model parameters are updated using the forward-backward propagation. Dropout is applied to randomly reset a proportion of model parameters to zero to reduce overfitting (Yin et al., 2018).

$$v_i = g(\tau * A + b) \quad \text{Eq. 4-11}$$

$$s(h,r,t) = concat(v_i) \cdot w \quad \text{Eq. 4-12}$$

$$\mathcal{L} = \sum_{(h,r,t)\in\Theta} \log\left(1 + \exp\left(l_{(h,r,t)} \cdot s_{(h,r,t)}\right)\right)$$

$$\text{Where } l_{(h,r,t)} = \begin{cases} 1 \; if \; (h,r,t) \in \Theta \\ -1 \; otherwise \end{cases} \quad \text{Eq. 4-13}$$

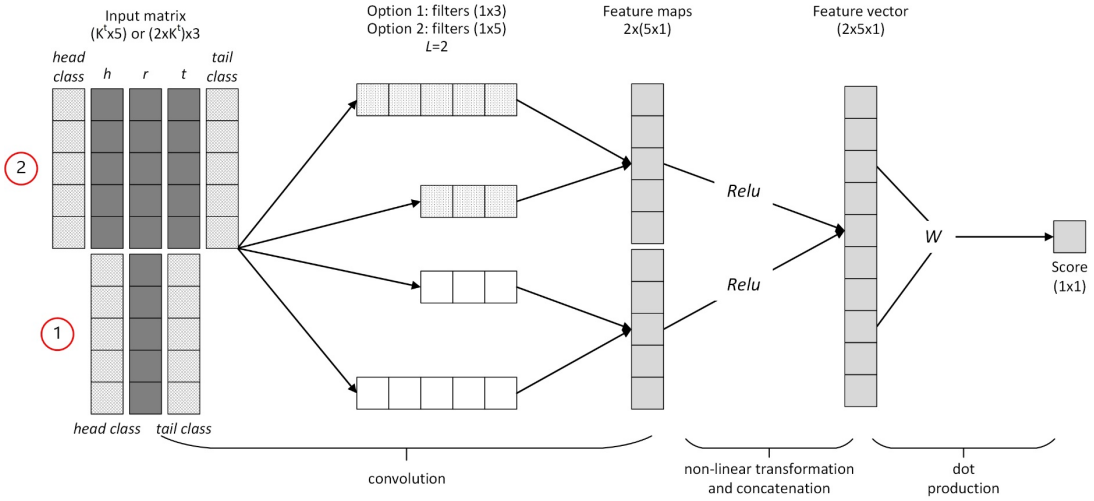


Figure 4-7 KRL model mechanism

4.4.5 Extracting other relation types

The KRL model above can only automatically extract three types of relations. On the other hand, *t2t* and *c2p* relations need to be extracted by pre-defined rules. For *c2p* relations, these rules link CONS entities to engineering packages or site installation packages based on the domain classes and the tasks of the CONS entities affect. For instance, if a CONS entity belongs to the 'Document' class and constrains the entity representing a pavement rolling procedure through a *c2t* relation, the CONS entity should be linked to the engineering package of the rolling procedure.

To extract *t2t* relations, the rules encode common work dependencies in construction projects (e.g., paving should be preceded by crack fixing in most deck rehabilitation tasks). The rules are created among domain classes, which are inherited by task or procedure entities. For instance, if a 'is-succeeded-by' relation is set up between the 'Paving' and 'Rolling' classes, two TP entities belonging to the classes respectively should be linked by the relation. As work dependencies include work sequences and the 'part-of' relation, the rules for extracting *t2t* relations are as follows: 1) each TP entity checks the existence of work dependencies with other TP entities; 2) each TP entity checks the existence of 'part-of' relation with other TP entities.

The rules to extract *t2t* relations are applicable in most construction projects and do not need significant modifications in different situations. However, TP entities that should be linked can be scattered in different sentences. As such, a scope should be defined to apply the rules. It should be noted that if the scope is too large (e.g., it includes too many sentences), there can be many false positives when extracting *t2t* relations, as the rules are not as smart as human engineers and can connect two TP entities which even do not describe the same task. Fortunately, it is found that TP entities that should be linked are usually clustered in a small scope, i.e., the sentence that is investigated currently plus the preceding and succeeding one sentence. Thus, the proposed rules make use of the three-sentences scope to extract *t2t* relations (an example is shown in Figure 4-8).

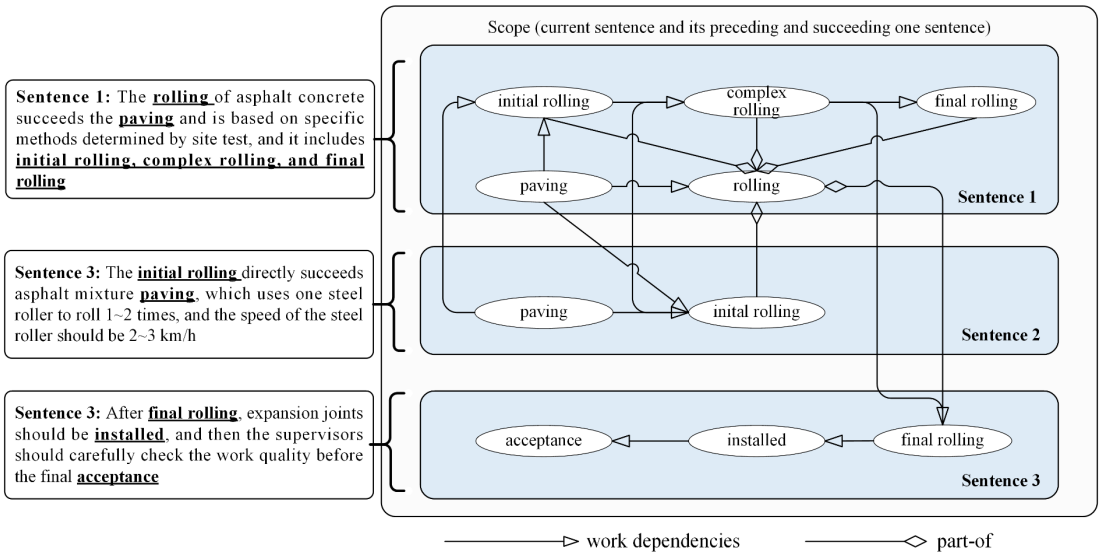


Figure 4-8 An example of rule-based *t2t* relation extraction

4.5 Relation extraction experiment results

4.5.1 Data preparation and hyper-parameter tuning

The same text corpus for training the Bi-LSTM-CRF model was used. The triples to train the KRL model were generated using manually tagged entities in raw sentences. Based on 452 unique constraint entities, 16555 *c2c*, 11067 *c2t*, and 9908 *c2a* triples (in Chinese) were generated using the method introduced in Section 3.4.3. It should be noted that the English documents were not considered for KRL model training, as the generalisation ability of the model only relies on the Bi-LSTM-CRF model due to the translation mechanism mentioned before. The unrealistic and ambiguous training triples were deleted. Then, 50% of the remaining triples were randomly sampled. As

such, 4356 *c2c*, 2924 *c2a*, and 2615 *c2t* triples (in total 9895 triples) were left for labelling, where 2012, 1133, and 1264 were manually labelled as valid, respectively. The triples then were divided into the training and testing datasets following the 9:1 proportion, and cross-validation was again applied. Finally, another 120 sentences in the meeting records of the deck pavement replacement task in the first case project were collected, where 51 unique entities and 451 valid triples were manually extracted for the controlled experiments (Section 4.5.3).

Table 4-2 lists the hyper-parameter tuning results for developing the KRL model. It should be noted that the KRL models for extracting *c2c*, *c2a*, and *c2t* relations can require different optimal hyper-parameters. Therefore, the 'optimal values' column can include three values for extracting the three types of relation, respectively. To control variables and demonstrate the effect of adding domain class information, in experiments, the hyper-parameters were tuned for the preliminary model setting (i.e., initialising model parameters with random e^t).

Table 4-2 Results of hyper-parameters tuning

Hyper-parameters	Potential values	Optimal values
Embedding dimension K^t	{50, 100, 300}	300
Margin γ	{1, 2, 5, 10}	1
Classification threshold	{0.01-2} (interval 0.05)	(0.5, 0.45, 0.4)
The number of filters	{1, 5, 10, 15, 20}	(10, 15, 10)
Learning rate	{0.001, 0.01, 0.1}	(0.01, 0.01, 0.005)
Batch size	$max(2^n, FULL)$	128
Dropout proportion	{0.1, 0.2, 0.3, 0.4, 0.5}	(0.3, 0.2, 0.3)
Optimisation function	{Adam, RMSProp, Momentum}	Adam
Non-linearity activation	{Relu, Elu, sigmoid, tanh}	Relu

4.5.2 Model results and analysis

4.5.2.1 Performance metrics of extracting relations

Using the optimal hyper-parameters, Table 4-3 presents the metrics of different KRL model settings. The results are averaged values obtained by running a model ten times. The terms 'random', 'TransE', and 'v_expanded' or 'h_expanded' indicate a KRL model 1) initialised using random e^t, 2) initialised using e^t produced by the TransE model, and 3) initialised using the TransE e^t and enhanced by vertically or horizontally stacking domain class information, respectively.

According to Table 4-3, the KRL model can accurately extract constraint triples. First, the class mapping module is very effective, where the average accuracy is 97.07%.

The class mapping makes use of distinct language (i.e., Chinese) patterns of domain entities' names hence can correctly identify most classes. The accuracy is higher when mapping attribute entities, as many attributes have distinct units which significantly facilitate classification. Errors of mapping CONS and TP entities are mainly caused by the fact that some entities have different semantics but share ending characters, which misleads the model. For instance, the character '装' is commonly used in the words expressing 'installing', however, it can also express 'transporting' in a few cases. Alternatively, one can also train ML models to map classes when more data (i.e., entity-class pairs) are labelled. However, given the high mapping accuracy, this research leaves such topics in the future. The good class mapping performance is critical for the downstream KRL model, as it can largely prevent errors of wrongly identified classes propagating to the relation extraction process. Figure 4-9 shows the confusion matrices of class mapping in the testing set (Figure 4-6 shows the meanings of axis labels). Figures 4-10 presents some examples of extracted triples.

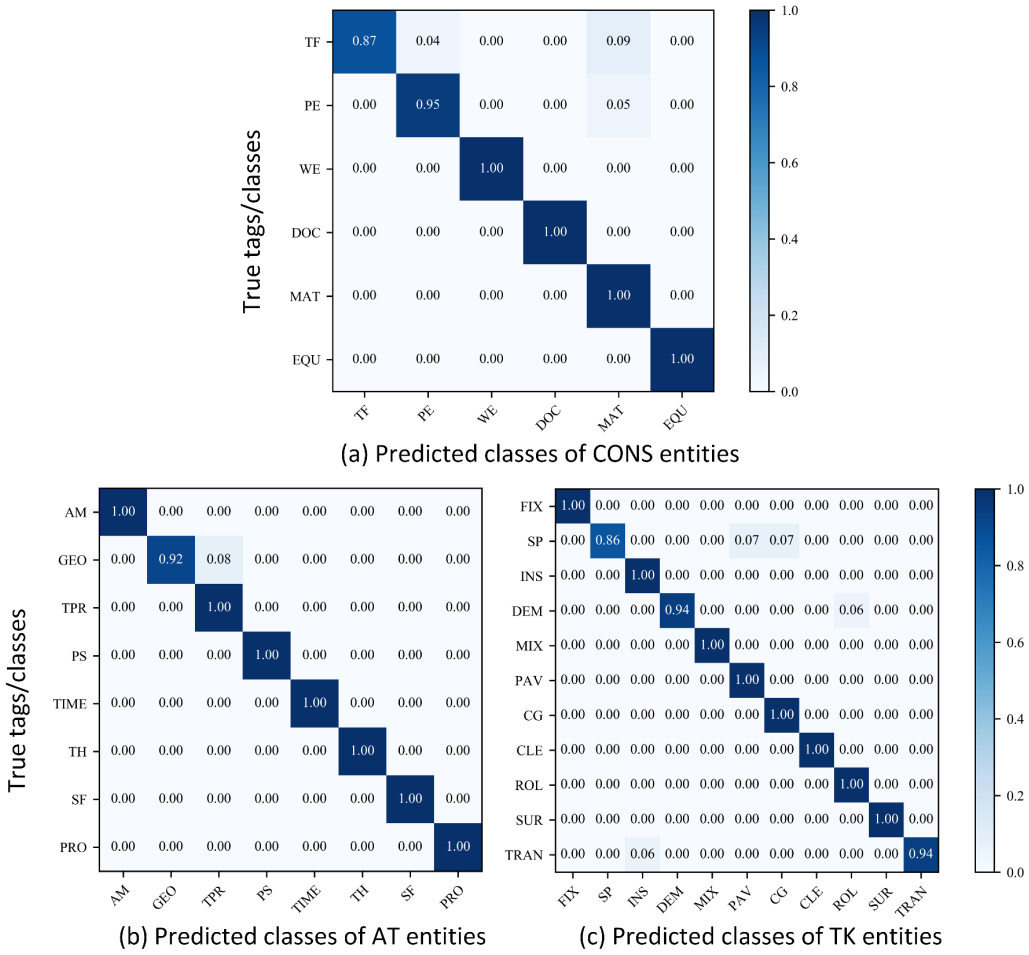


Figure 4-9 Confusion matrices in the testing dataset

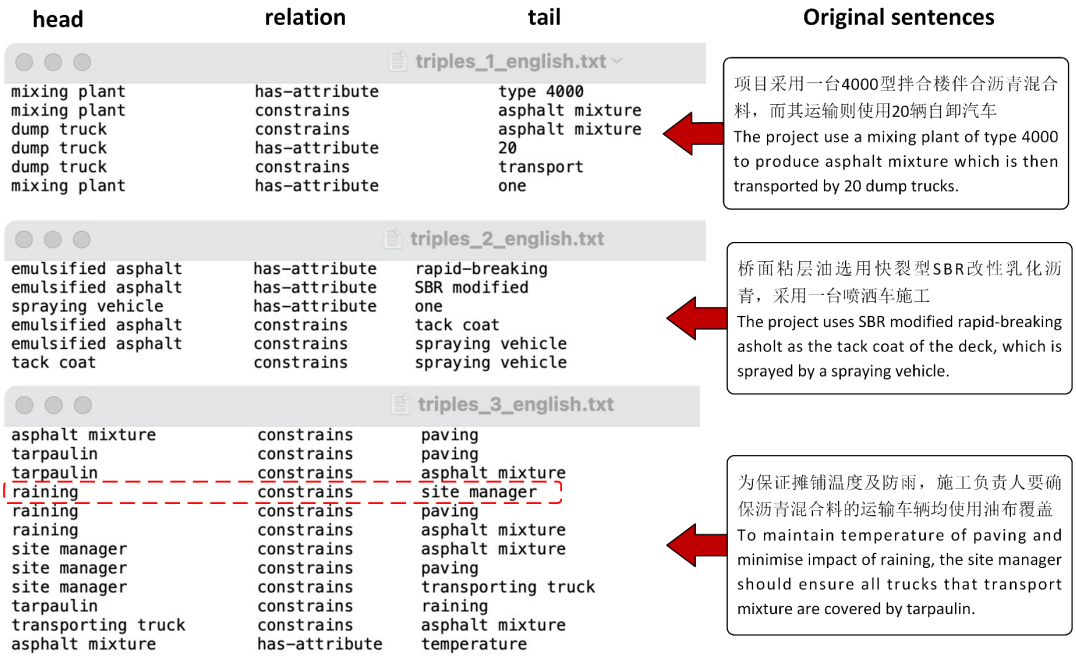


Figure 4-10 Examples of extracted triples (wrong predictions are highlighted)

The performance metrics of relation extraction are lower. The highest F1 score when extracting the *c2c*, *c2a*, *c2t*, *t2t*, an *c2p* triples in the testing set is 0.859, 0.885, 0.908, 0.912, and 0.890, respectively. For extracting *c2c*, *c2a*, and *c2t* triples (only the three types of relations can be extracted by the KRL model), the training was finished in a very short period (5.15 minutes). State-of-the-art KRL models can reach 85-90% F1, which however are often trained on millions/billions of triples while training can take hours/days (Lin et al., 2015; Nguyen et al., 2019). Hence, the proposed KRL model can achieve competitive performance in literature with much less time. Most errors of triple extraction are caused by the fact that some entities have similar names but different semantics. For instance, in Figures 4-10, 'crew' and 'manager' share some characters in Chinese texts. If the model learns that 'raining' constrains the work of 'crew', it can wrongly infer that 'raining' also constrains 'manager' as both entities belong to the 'People' class.

4.5.2.2 The effect of stacking domain class information

To demonstrate the effect of stacking domain class information in the model structure, Figure 4-11 compares the loss curves in the training and validation datasets. It turns out that the model randomly initialised has the worst performance, while the models initialised using the TransE model and enhanced with domain class information have higher metrics and lower loss. The model stacking class information horizontally in the input matrix has the most obvious performance increase compared to metrics of

the ‘TransE’ setting, where the F1 score can be increased by 1.9%, 12.0%, and 6.0% when extracting *c2c*, *c2a*, *c2t* triples in the testing dataset, respectively. The model stacking class information vertically has similar performance to the model adopting the ‘TransE’ setting, except the additional 5.1% F1 score when extracting *c2t* triples. It seems that the $\boldsymbol{A}$ matrix expanded horizontally is easier for CNN filters to capture the features of entity-class relations, relations among classes, and relations between head/tail entities. Another reason is that the vertically expanded KRL models have more parameters hence requiring more training data. Moreover, when domain class information is added, the model loss declines more quickly and smoothly. This means the model converges in a shorter time and has less oscillation. One reason is that class information can cluster entities based on their classes so that the model is less likely to be distracted by heterogenous entities’ names. However, the models stacking class information horizontally have larger DOF, i.e., they can cause more overfitting. This is reasonable, as these enhanced models have more complex structures which often cause additional overfitting. Nevertheless, overfitting can be alleviated by feeding the model with more data (Goodfellow et al., 2016).

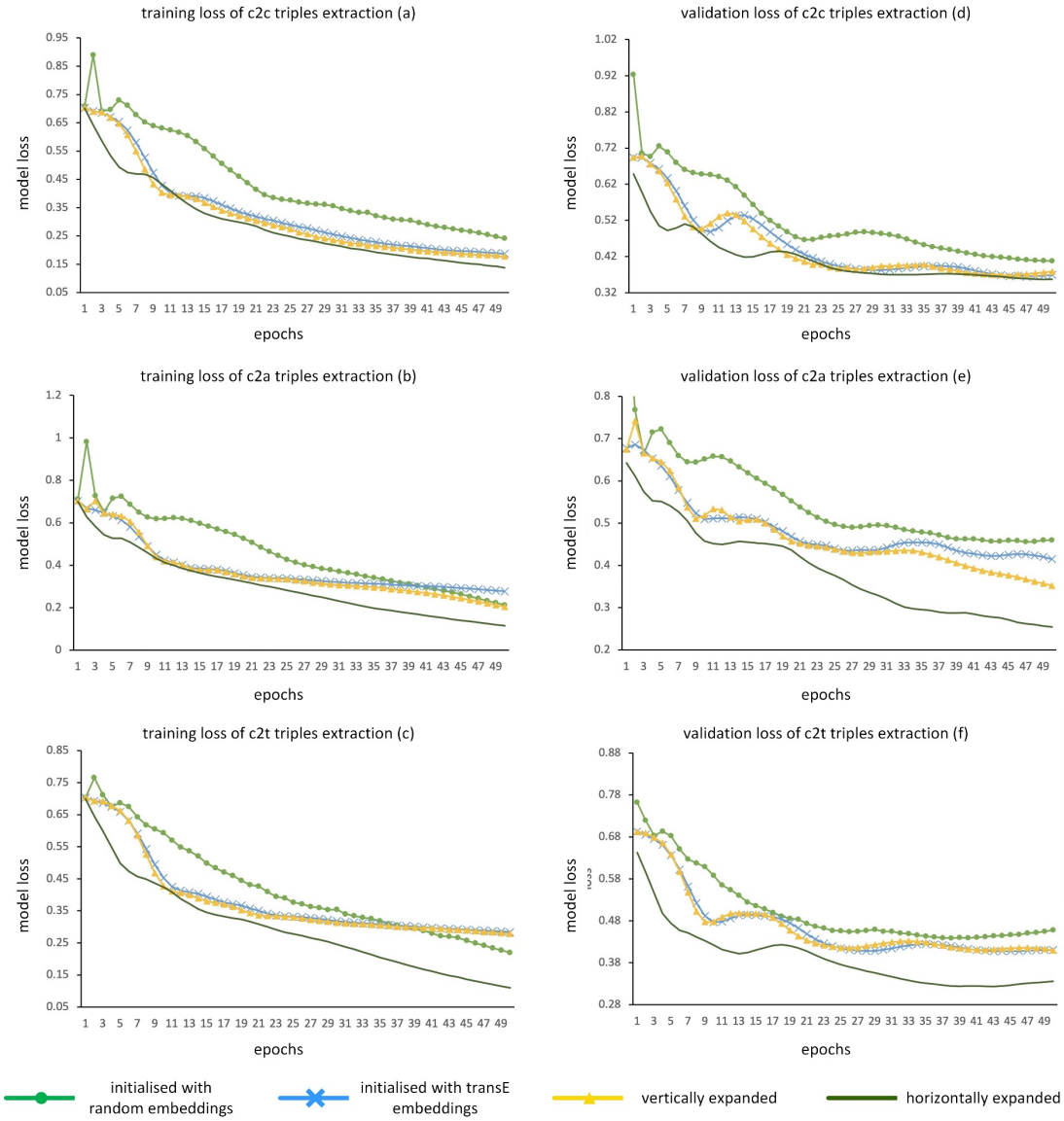
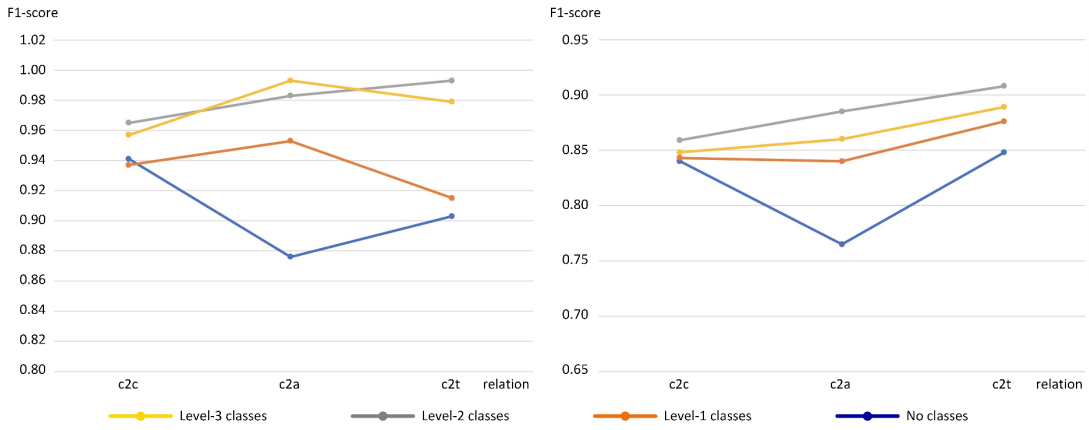


Figure 4-11 Loss curves of different KRL model configurations

Table 4-3 Model performance metrics

Model	Pr-train	Pr-val	Pr-test	Re-train	Re-val	Re-test	F1-train	F1-val	F1-test	F1-diff
NER (Chinese)^{B}	0.969	0.909	0.944	0.966	0.895	0.928	0.968	0.902	**0.936**	*0.066*
NER (Chinese)^{H}	0.968	0.902	0.881	0.957	0.886	0.873	0.962	0.894	0.877	0.068
NER (Chinese)^{C}	0.976	0.885	0.910	0.981	0.901	0.907	0.978	0.893	0.908	0.085
NER (English)^{B}	0.959	0.892	0.925	0.945	0.872	0.900	0.952	0.882	**0.912**	0.070
NER (English)^{H}	0.943	0.872	0.863	0.960	0.869	0.851	0.951	0.870	0.857	0.081
NER (English)^{C}	0.970	0.870	0.896	0.947	0.894	0.844	0.958	0.882	0.869	0.076
c2c_random	0.941	0.806	0.803	0.926	0.866	0.840	0.933	0.835	0.821	0.098
c2c_TransE	0.948	0.808	0.809	0.935	0.905	0.874	0.941	0.854	0.840	*0.087*
c2c_v_expanded	0.954	0.824	0.823	0.930	0.881	0.849	0.942	0.851	0.838	0.091
c2c_h_expanded	0.966	0.889	0.856	0.964	0.854	0.862	0.965	0.871	**0.859**	0.094
c2a_random	0.965	0.816	0.764	0.941	0.770	0.759	0.953	0.792	0.761	0.160
c2a_TransE	0.918	0.837	0.781	0.838	0.768	0.756	0.876	0.799	0.765	0.077
c2a_v_expanded	0.949	0.928	0.883	0.928	0.785	0.759	0.938	0.850	0.816	0.088
c2a_h_expanded	0.990	0.945	0.931	0.977	0.881	0.842	0.983	0.912	**0.885**	*0.071*
c2t_random	0.966	0.844	0.862	0.938	0.815	0.788	0.952	0.829	0.824	0.123
c2t_TransE	0.947	0.857	0.887	0.863	0.829	0.812	0.903	0.843	0.848	0.060
c2t_v_expanded	0.943	0.864	0.903	0.868	0.828	0.806	0.904	0.846	0.852	*0.059*
c2t_h_expanded	0.993	0.920	0.934	0.993	0.865	0.883	0.993	0.892	**0.908**	0.101

P.S. B, H, C indicate the Bi-LSTM-CRF, HMM and CRF model, respectively; the best metrics and the lowest DOF are highlighted in the bold and underlined font, respectively



(a) (b)

Figure 4-12 Effect of using different class levels (a) training (b) testing

The effect of adding domain classes of different levels of detail was also tested. The model performance was evaluated using only the three main classes (i.e., Constraint, Attribute, and Task in Figure 5-3), level-two classes in Figure 4-6 (i.e., the default setting), and more detailed level-three classes (e.g., 'Labour' and 'Manager' can be created as the subclasses of 'People'). Figures 4-12 presents the results. The model using level-two classes gains the best performance of triple extraction (especially in the testing dataset), which is followed by the performance of using level-three and level-one classes, respectively. The performance metrics of the TransE setting (i.e., no class information) are also presented in the figure as the baseline. Particularly, if the class hierarchy is too simple, the effect of clustering entities is weak. On the contrary, if there are too many classes, the number of entities belonging to each class is small. This can cause underfitting, i.e., making it difficult for the model to learn patterns of entity-class and class-class relations.

4.5.3 Controlled experiments (AWP KBs development)

To demonstrate the usefulness of the hybrid model, AWP constraint modelling was carried out based on three meeting records of the deck replacement task in the first case project introduced in Section 3.4.3. The task was selected as it was the most time-consuming and labour-intensive task in that project, involving more constraints than other tasks. Triples were extracted using the hybrid IE model and rules (for extracting *t2t* and *c2p* relations). The triples were encoded into the Neo4j graph database for visualisation and information searching (Gong et al., 2018). As mentioned, the model cannot automatically identify working package entities. Hence, package entities and relations among them (i.e., *p2p* relations) were manually inserted at the beginning. In

the experiments, the initial AWP graph only included work packages (Figure 4-13(a)). However, the constraint modelling was partially automated, where constraint entities and triples were extracted and the AWP graph was automatically enriched by these triples (Figure 4-13(a)-(d)), To demonstrate the interconnections among constraints, Figure 4-13(d) shows the constraints (red circles) that constrain the central constraint 'asphalt mixture' (the red dashed circle), where only one relation is highlighted for clarity. The constraint modelling was completed in 78s, where the time to manually insert work package entities and setup links was included. It took 38 minutes for the colleague of the researcher to construct the graph manually.

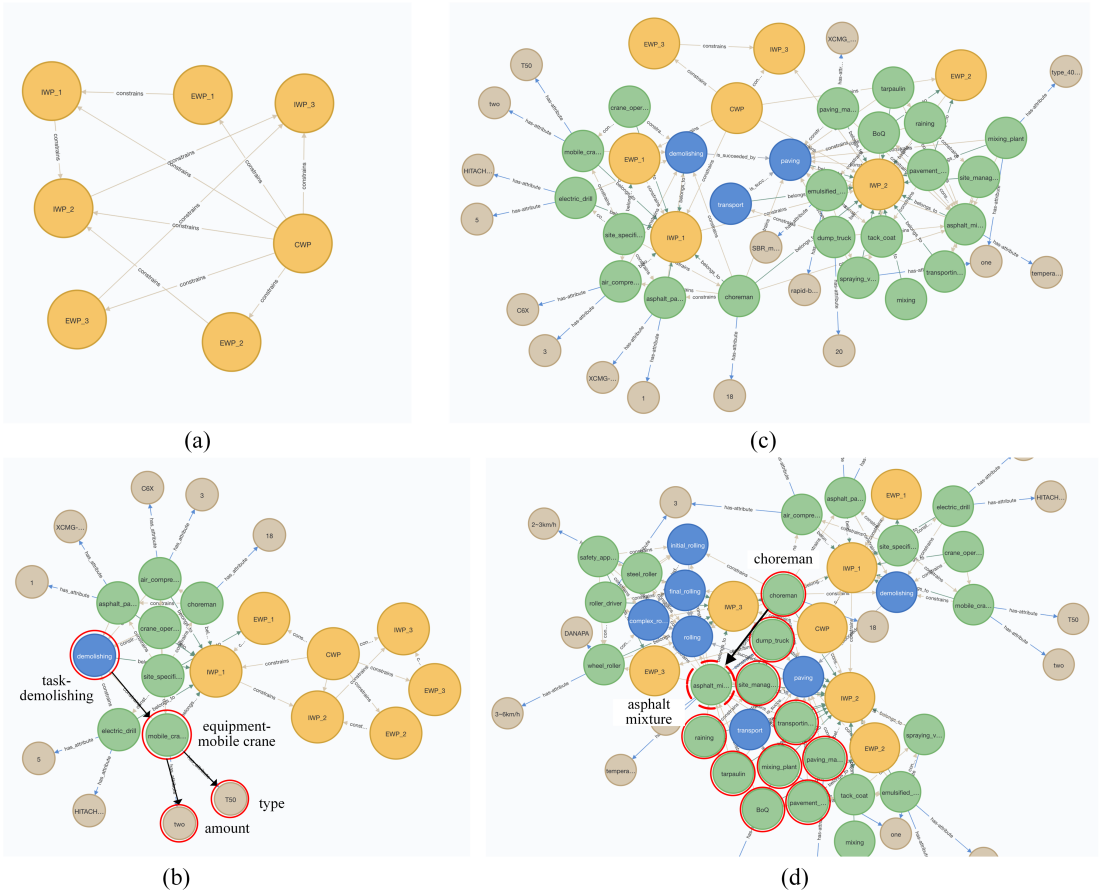


Figure 4-13 (a) Initial graph, (b)-(d) AWP modelling in three weeks, where the yellow, green, blue, and yellow-grey nodes refer to the work packages, constraints, tasks, and attributes, respectively

The automatically generated graph can facilitate project management by integrating unstructured constraint information, which enables efficient information searching using graph-based queries (Gong et al., 2018). For instance, a foreman can quickly retrieve the required amount and type of equipment for the demolishing procedure (Figure 4-13(b)); he/she can also easily identify the constraining relations among

project entities and prepare constraint removal. For instance, the entity 'emulsion asphalt' constitutes 'tack coat' therefore should be delivered first, and the constraint 'choreman' should be removed before material constraints (e.g., asphalt mixture) for onsite material storage and transportation.

The AWP graphs are often incomplete. In other words, using the hybrid IE model cannot extract all needed information from texts and there are always some triples missing. For instance, in Figure 4-13, the entity 'site manager' should constrain all materials and equipment, whereas the model can only identify some of them. This is because the hybrid IE model is designed for extracting triples from single sentences hence cannot capture triples hidden in multiple sentences. If the graph is manually drawn, some missing triples can be avoided, as a human can reason the existence of them. However, the time spent is 29 times that of the hybrid IE model. In addition, a human can also miss triples when he/she gradually loses concentration, even finding such triples is very simple for an experienced engineer. Incompleteness is a common issue for KBs (Dettmers et al., 2017; Shi & Weninger, 2017; Trouillon et al., 2016). To address the problem, the KBC model is developed to automatically complete KBs (Chapter 6). Finally, the controlled experiments only included brief texts in meeting records for demonstration. As such, the graph can be easily enriched when more texts, e.g., detailed plans, are processed by the hybrid IE model. In that case, the advantages of automated modelling will be more evident.

4.6 Discussion

It can be argued that the hybrid IE model has three contributions. First, the model can improve information extraction in the AEC industry. Existing IE methods in the area focus on extracting entities. A few efforts attempt relation extraction. However, most of them can only extract simple relations, e.g., the existence of relations, synonyms, and hypernyms (Chi et al., 2019; Le & David, 2017). As such, semantic rich relations cannot be extracted from texts using current methods. A recent study could extract complex constraint relations, which however heavily relied on handcrafted rules (Wu et al., 2021b). Creating and updating rules not only require much time and effort but also make the results (i.e., extracted triples) subjective. On the contrary, the hybrid IE model combines the Bi-LSTM-CRF and KRL model, which automatically identifies constraints, their attributes, and tasks/procedures and establishes relations among the

entities. Another study used multi-layer NN models to extract task dependencies from quality codes, reaching an average F1 score of 0.74 (Zhong et al., 2020b), Despite that the performance metrics cannot be directly compared with those of the hybrid IE model, the model in this research can effectively extract five types of relations and achieve a high average F1 score (0.891). It can be intuitively argued that the hybrid IE model outperforms the previous studies. As such, a distinct feature of the proposed model is that it is an early exploration in the area that can extract both entities and semantic rich relations using DL models.

Second, the hybrid IE model contributes to the implementation of AWP. Current AWP is inefficient, as constraint modelling still depends on manually extracting constraint information from documents (Li et al., 2019; Wang et al., 2016). The hybrid IE model partially automates constraint modelling, as it can automatically extract entities and relations, based on which the AWP graph can be automatically developed. However, some human intervention is still inevitable, e.g., inserting the *p2p* relations which are project-specific (Halala, 2018; Halala & Fayek, 2019). Full automation is left in the future. However, the hybrid IE model can still significantly reduce the constraint modelling time to 1/29 of the manual approach. Thus, the model can save much time for constraint monitoring and removal. The model training relies on multiple types of documents (e.g., manuals, standards, and plans), therefore, it can handle both static data (e.g., imperative requirements stated in standards) and dynamic data (e.g., the changing interconnections among constraints), meeting the demands of intensive and repetitive AWP modelling. The controlled experiments only included brief texts in three weekly meeting records. There can be thousands of constraints in practice, and it is exhausting and error-prone for engineers to extract all constraint information merely based on experience, especially for young engineers who have less experience (CII, 2020; Hamdi, 2013). Thus, the effect of the hybrid model is more significant in practice. In recent years, construction management approaches are shifting from an experience-driven to a data-driven manner (Cao et al., 2019; Wu et al., 2020a). Hence, the hybrid IE model is an attempt to make AWP data-driven, which can help (not replace) engineers (especially new engineers) to quickly understand interconnections among constraints and improve management decision-making.

Third, the hybrid model contributes by improving current KRL models so that they can be better applied to construction documents. Most KRL models (e.g., ConvKB

and ConvE) are trained on general KBs which model general knowledge and can contain billions of triples. However, such KBs do not have enough triples for AWP modelling. Thus, the training triples for the proposed KRL model were generated by traversing entities extracted by the Bi-LSTM-CRF model. Although the dataset is small (452 entities and 9895 triples), the experiment results prove that the model can achieve high relation extraction performance (i.e., 0.891 F1 on average) while the training can be very fast (5.15 minutes). The main reason is that the number of unique entities and triples is much smaller than those in general KBs. For instance, the deck rehabilitation domain can be covered by the 452 unique entities while the five types of relations can support general AWP modelling (Halala & Fayek, 2019; Wang et al., 2016). In such a dense dataset, triple patterns appear frequently, which can alleviate the data sparsity problem encountered in general KBs and facilitate model training (Zhang et al., 2018b). Besides, current KRL models do not adequately consider out-of-KBs entities (Bi et al., 2020; Zhao et al., 2020). To address the issue, the hybrid IE model adopts synonym mapping to map them to existing ones in KBs. Existing studies handle out-of-KBs entities by adding information (e.g., text descriptions) or applying complex graph-based DL models to estimate unknown embeddings of these entities, which requires much computation power and training data. This is because many entities in general KBs are ambiguous (e.g., Apple can be a fruit or a company), which cannot be recognised by naive synonym mapping (Bi et al., 2020; Zhao et al., 2020). In contrast, as most entities in construction projects have distinct meanings, synonym mapping is efficient to handle out-of-KBs entities, which does not need additional data and training.

This hybrid IE model also enhances state-of-the-art KRL models by adding domain information. Current CNN-based KRL models take input matrix $\boldsymbol{A}$ of shape (K^t, 3) (Nguyen, 2020; Nguyen et al., 2018). However, the proposed model maps the head and tail entities to their domain classes and then expands $\boldsymbol{A}$ by stacking character/word embeddings of the classes in $\boldsymbol{A}$. Model performance using different structures was compared in experiments. The results prove that although enhanced models can cause more overfitting due to more complex structures, relation extraction performance is increased by 6.63% on average with domain class information stacked horizontally. Moreover, the loss curves during training and validation decline more quickly and smoothly, which can save computation power and accelerate model convergence. As

such, the hybrid IE model improves existing KRL models by proposing a novel way to utilise domain information in the CNN-based structure.

The hybrid IE model was mainly trained for Chinese documents. However, it can be generalised to process texts in other languages. For one thing, the experiments have proved that the Bi-LSTM-CRF model can achieve high entity extraction accuracy (0.912 F1) when processing English data. This is because DL models support transfer learning, i.e., the model can be trained using different data while the model structure remains unchanged (Pan & Yang, 2009). Thus, the Bi-LSTM-CRF model can extract entities in different languages as long as the text data are provided (Peng et al., 2017). For another, the KRL model was trained using separated triples, hence, it does not rely on syntactic features of certain languages (Zhao et al., 2020). Due to the translation mechanism introduced in Section 4.4.1, the proposed KRL model can be trained on English data regardless of input languages. Therefore, the entire hybrid IE model is to-some-extent language independent.

4.7 Chapter summary

This chapter introduces the detailed design and experiment results of the hybrid IE model. The model includes a Bi-LSTM-CRF model to extract constraint entities, a class mapping module to identify classes of the entities, a synonym mapping module to handle out-of-KBs entities, and a CNN-based KRL model to extract triples among the entities, where the KRL model structure is improved by adding domain class information. The hybrid model can extract CONS, AT, and TP entities and five types of relations (i.e., *c2c*, *c2t*, *c2a*, *c2p*, and *t2t*), which can achieve high performance. Besides, adding domain class information (especially when the information is stacked horizontally in the input matrix) can increase model performance and accelerate model convergence. In practical AWP modelling, the hybrid IE model can largely automate the modelling and updating of AWP graphs when the project proceeds. Thus, much time to develop the graphs can be saved for constraint monitoring and removal.

Chapter 5: Developing ontological KBs for AWP-based bridge rehabilitation projects

5.1 Chapter introduction

This chapter introduces the development of the TBox/RBox of the ontological KBs (i.e., BRMO) following the steps introduced in Section 3.5. The information encoding process is then introduced to integrate constraint information. Finally, the results of the information searching experiments are presented to show the usefulness of the KBs.

5.2 Ontology taxonomy

The proposed ontological KBs include four taxonomies, i.e., 'Rehabilitation Task', 'Procedure', 'Constraint', and 'Project Participant'. A taxonomy can be expanded up to the fifth level. An overview of relationships among the taxonomies is shown in Figure 5-1.

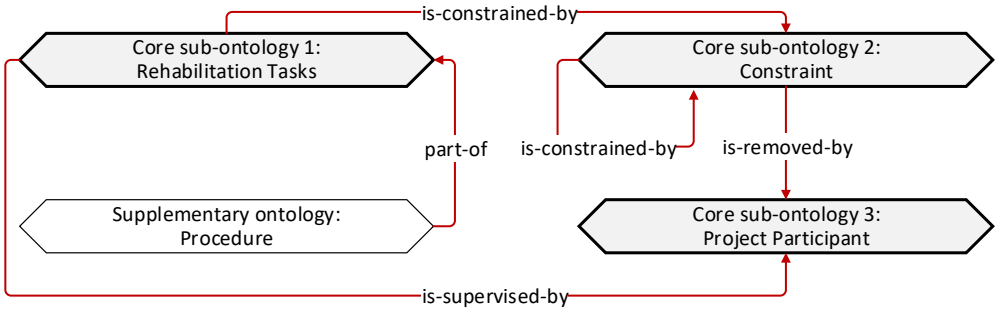


Figure 5-1 High-level overview of the BRMO

5.2.1 Taxonomy of bridge rehabilitation tasks and procedures

The taxonomy of bridge rehabilitation tasks is shown in Figure 5-2 (it is not fully expanded). The three top-level classes are 'Hazards Treating', 'Reinforcement', and 'Replacement'. The 'Replacement' class is divided by bridge components to be replaced. Different engineering approaches are applicable for hazard treatment and reinforcement. Thus, the 'Hazard Treating' and 'Reinforcement' classes are divided by dominant engineering approaches, while the 'Hazard Treating' class is divided by main hazard types first. To model relations between procedures and tasks (i.e., the part-of relations), a procedure taxonomy is developed, including three basic classes: 'Preparation', 'Execution', and 'Acceptance'. A task could have some or all these procedures, while a procedure class can also be detailed and expanded.

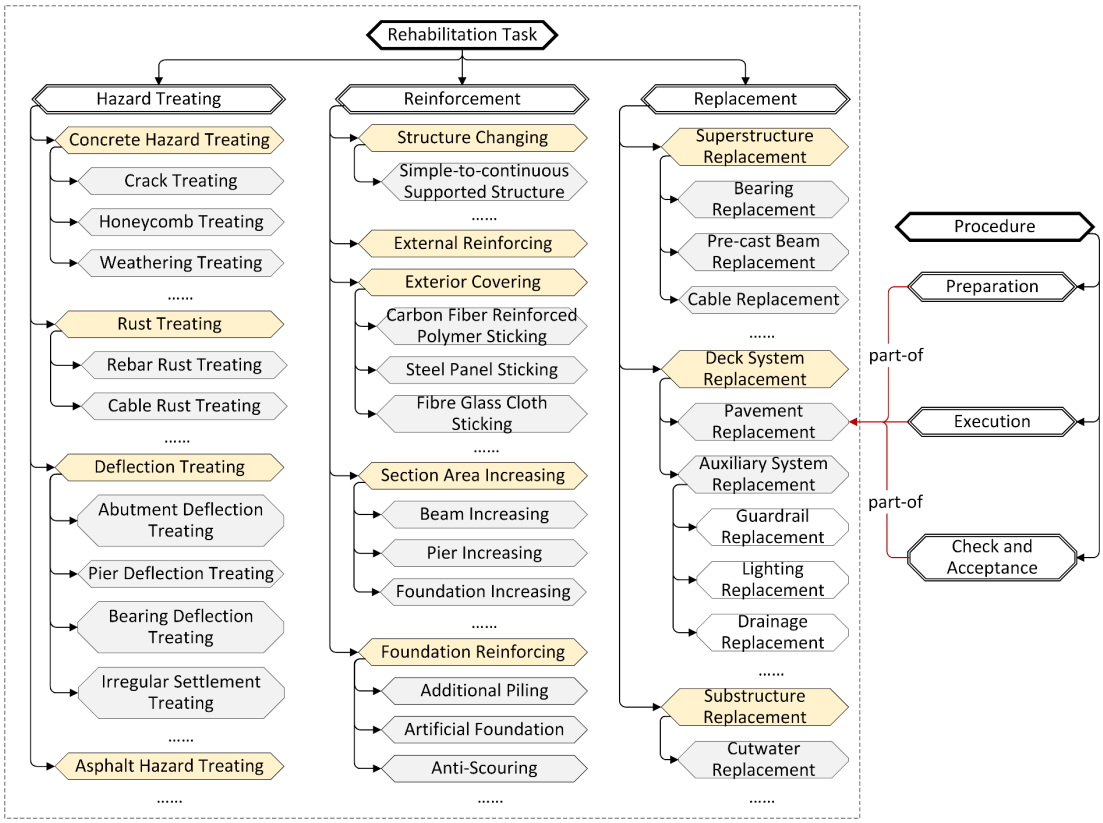


Figure 5-2 Overview of task/procedure taxonomy

5.2.2 Taxonomy of constraints

As shown in Figure 5-3 (not fully expanded), the constraint taxonomy includes three first-level classes. Specifically, the engineering constraints mainly cover engineering documents, e.g., drawings and approvals, the supply chain constraints refer to the delivery of materials and equipment, and the site constraints refer to constraints that can hinder the work of site crews. The classification between the special and general labour relies on whether the job requires considerable training and can pose danger to others (SAWS, 2010; Wu et al., 2019).

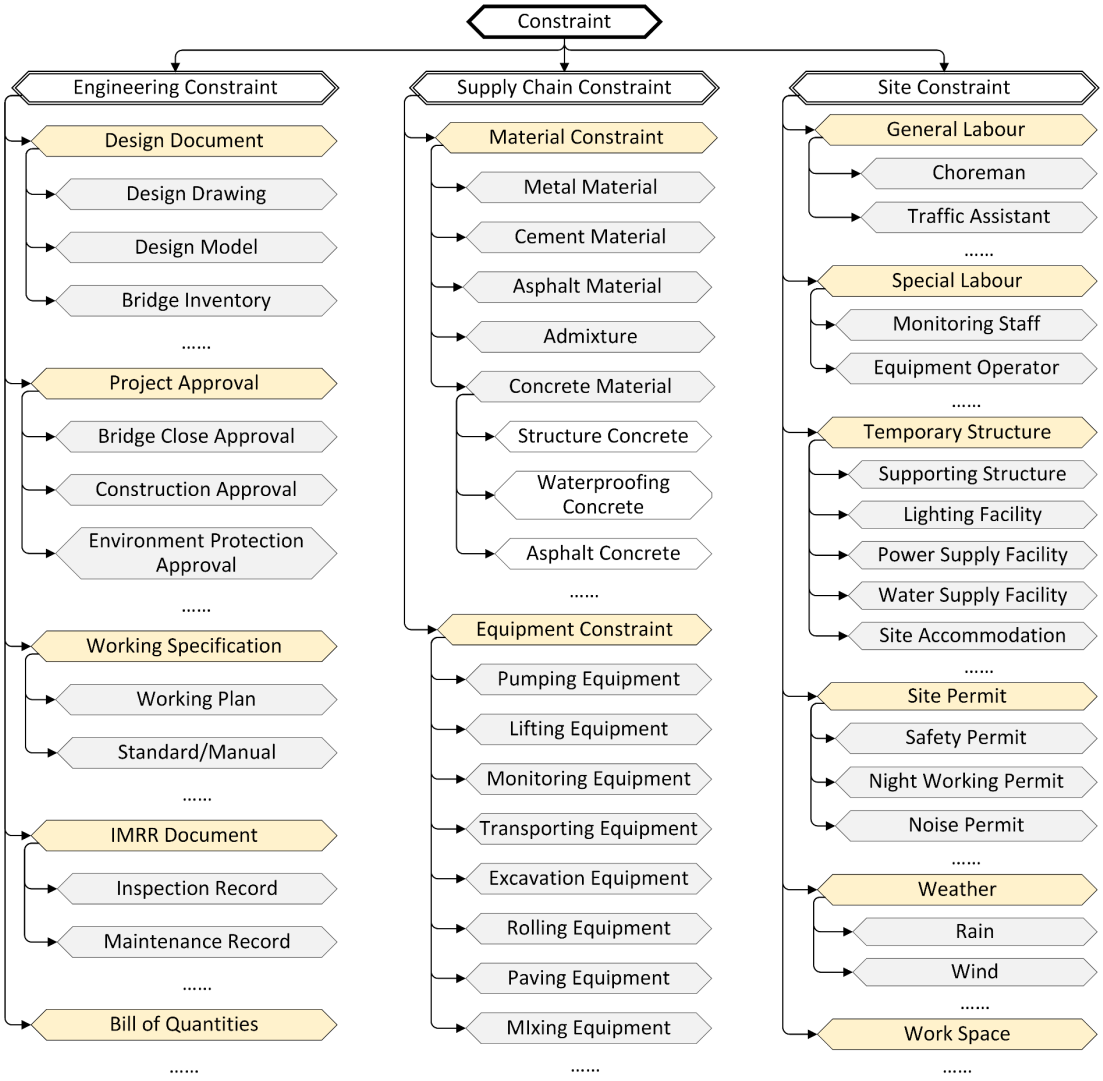


Figure 5-3 Overview of constraint taxonomy

5.2.3 Taxonomy of project participants

As shown in Figure 5-4 (not fully expanded), the taxonomy of participants covers project-level and external-level participants as the top classes. The two classes are divided based on the roles and responsibilities as well as project stages related to the participants. It should be noted that although the ontological KBs (i.e., BRMO) have three taxonomies, the intention is not to cover all terms but to include common and critical terms in the domain of bridge rehabilitation. The taxonomies can be expanded when necessary. Finally, the class hierarchy for class mapping in the KRL model (Section 4.4.1) is also developed based on the class taxonomies.

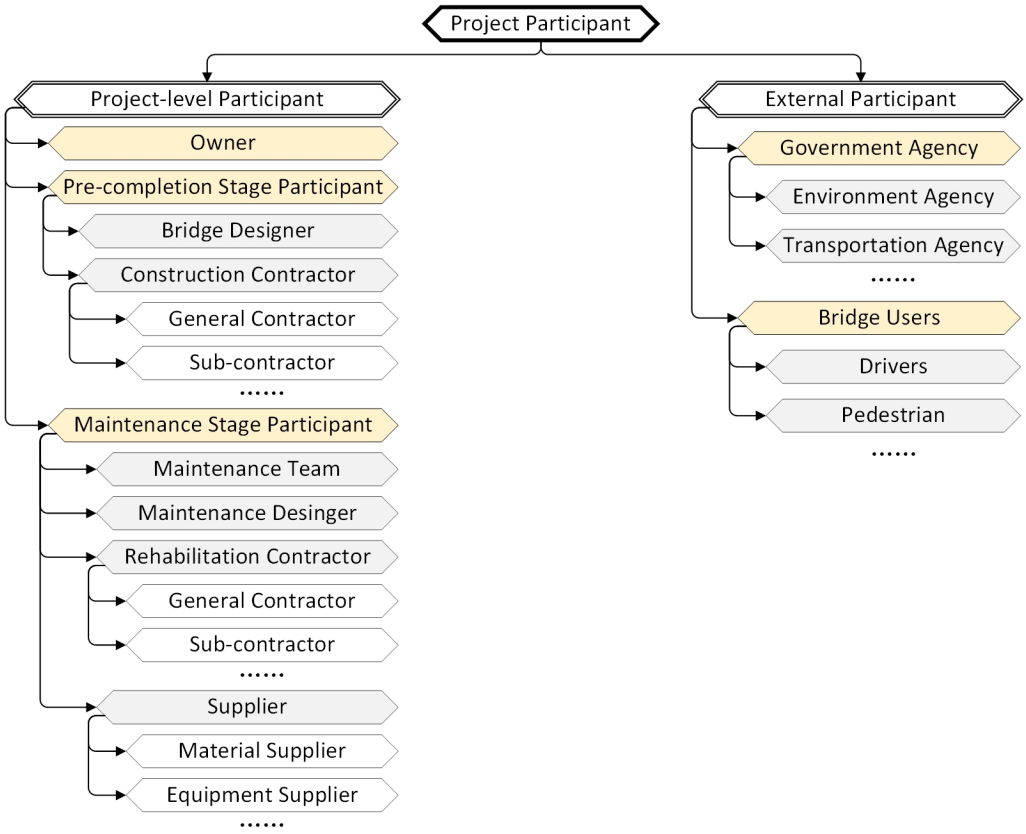


Figure 5-4 Overview of participant taxonomy

5.2.4 Relation hierarchies

Based on the domain knowledge obtained in the focus group, descriptions of object and datatype ontological relations have been summarised in Table 5-1 and Table 5-2, respectively, providing unambiguous meanings for relations in the BRMO. Besides, the object and datatype relation hierarchies are shown in Table 5-3 and Table 5-4, respectively. In the tables, 'Trans', 'Sym', and 'Func' indicate transitive, symmetric, and functional characteristics, respectively, 'PP' indicates project participants, and the meanings for other abbreviations can be found in Figure 4-6. The relations highlighted by the bold font are first-level relations, the others are second-level relations.

Table 5-1 Object relation descriptions

Relation name	Description
constrains	
accommodate	Temporary facilities provide accommodation for workers and engineers
check-quality	Engineers check the quality of tasks/procedures and constraints
constitute	A material constraint can consist of multiple materials (e.g., concrete consists of sand, stone, cement, and water)
vertically-transport	Some constraints vertically deliver other constraints off-site or onsite
horizontally-transport	Some constraints horizontally deliver other constraints off-site or onsite

grant-permission-to	Some project participants provide approvals for other constraints (e.g., materials, equipment, and tasks/procedures)
manage	Engineers supervise constraints and tasks/procedures
monitor	Some equipment can monitor statuses of other constraints
needs-material	Some equipment can require certain materials
use	Some people (e.g., labour) can require certain materials or equipment
build	Some people (e.g., labour) construct facilities/structures
pre-requisite-doc-of	Approval of one document can require another document
produce	Some constraints can produce certain constraints (e.g., some equipment can produce certain materials)
protect	Some constraints can protect onsite engineers and workers
prevent-harm-from	Some constraints can prevent onsite engineers and workers from injury caused by certain constraints
provide-space-for	Temporary facilities provide space for constraints and tasks/procedures
work-in	Onsite workers work for certain tasks/procedures
is-required-by	A material/equipment is required by certain tasks/procedures
remove	Project participants remove certain constraints
deliver-work	Project participants are responsible for certain tasks/procedures
review	Engineers check and review documents
specify	Documents specify requirements of constraints
supply-power-to	Some equipment provides power to other equipment
supply-water-to	Some equipment provides water to other equipment
supply-gas-to	Some equipment provides gas to other equipment
has-unremoved-constraint	Link a constraint or task/procedure to its unremoved constraints
part-of	Link a detailed procedure to a more general procedure or task
work-dependencies	
finish-procedure-of	The final procedure of a task/procedure
is-preceded-by	Preceding relations
is-succeeded-by	Succeeding relations
latest-procedure-of	The procedure that is currently carried out
proceed-concurrently	Two tasks/procedures proceed concurrently
start-procedure-of	The first procedure of a task/procedure
is-a	Link an entity to its domain class
subclass-of	Subclass relations between domain classes

Table 5-2 Datatype relation descriptions

Relation name	Description/examples
has-attribute	
has-amount	number, m2, m3, kg, t, etc.
has-geometry	m, cm, mm, etc.
has-pressure/stress	Pa, KPa, N, KN, etc.
has-price	dollar, yuan, etc.
has-proportion	The proportion of each part when mixing materials (e.g., concrete mixture)
has-speed/frequency	m/s, km/h, Hz, etc.
has-temperature/humidity	°F, °C, etc.
has-time	day, hour, second, etc.
has-type/property	C50, C20, etc. for concrete, and SBR, SBS, etc. for asphalt
has-constraint-status	
has-actual-removal-date has-planned-removal-date has-removal-delay is-timely-removed is-removal-potentially-delayed is-removed	Indicating the actual/planned removal dates of a constraint and whether the removal is delayed or potentially delayed
has-reason	Specifying the reasons for delay (if any)

can-be-delayed-by	Specifying the fact that a constraint entity might be delayed by another entity
has-performance	
has-constraint-removal-performance	The ratio of the number of timely removed constraints to the number of all constraints a participant removes
has-task-performance	The ratio of the number of timely finished tasks or procedures to the number of all tasks or procedures a participant performs
has-progress-information	
has-actual-duration has-actual-duration-from-start has-actual-finish-date has-actual-start-date has-current-duration has-current-progress is-finished has-planned-duration has-planned-finish-date has-planned-start-date has-started has-total-progress is-work-delayed is-work-potentially-delayed	Indicating the actual/planned duration and starting and finishing dates of a task/procedure as well as whether a task/procedure is started, finished, and delayed (or potentially delayed). The current and total progress indicates the current schedule progress performance compared to the plan in terms of specific days.

P.S. the meanings of 'has-attribute' relations are very clear in their names, as such, only common attribute units are listed to help one understand.

Table 5-3 Object relation hierarchy and properties

Relation name	Trans	Sym	Func	Domain	Range
constrains					
accommodate	×	×	×	TF	PE
check-quality	×	×	×	PE	CONS, TP
constitute	×	×	×	MAT	MAT
vertically-transport	×	×	×	CONS	CONS
horizontally-transport	×	×	×	CONS	CONS
grant-permission-to	×	×	×	PP	CONS, TP
manage	×	×	×	PE	CONS, TP
monitor	×	×	×	EQU	CONS
needs-material	×	×	×	EQU	MAT
use	×	×	×	PE	MAT, EQU
build	×	×	×	PE	TF
pre-requisite-doc-of	×	×	×	DOC	DOC
produce	×	×	×	CONS, TP	MAT
protect	×	×	×	CONS	PE
prevent-harm-from	×	×	×	CONS	CONS
provide-space-for	×	×	×	TE	CONS
work-in	×	×	×	PE	TP
is-required-by	×	×	×	MAT, EQU	TP
remove	×	×	×	PP	CONS
deliver-work	×	×	×	PP	TP
review	×	×	×	PE	DOC
specify	×	×	×	DOC	CONS
supply-power-to	×	×	×	EQU	EQU
supply-water-to	×	×	×	EQU	EQU
supply-gas-to	×	×	×	EQU	EQU
has-unremoved-constraint	×	×	×	CONS	CONS
is-constrained-by	×	×	×	CONS	CONS
part-of	√	×	×	TP	TP

work-dependencies					
finish-procedure-of	√	×	×	TP	TP
is-preceded-by	√	×	×	TP	TP
is-succeeded-by	√	×	×	TP	TP
latest-procedure-of	√	×	×	TP	TP
proceed-concurrently	√	√	×	TP	TP
start-procedure-of	√	×	×	TP	TP
is-a	√	×	√	CONS, TP, PP, AT	CONS, TP, PP, AT
subclass-of	√	×	√	CONS, TP, PP, AT	CONS, TP, PP, AT

Table 5-4 Data relation hierarchy and properties

Relation name	Func	Domain	Range
has-attribute			
has-amount	×	CONS	String
has-geometry	×	CONS	String
has-pressure/stress	×	CONS	String
has-price	×	CONS	String
has-proportion	×	CONS	String
has-speed/frequency	×	CONS	String
has-temperature/ humidity	×	CONS	String
has-time	×	CONS	String
has-type/property	×	CONS	String
has-constraint-status			
has-actual-removal-date	√	CONS	Date
has-planned-removal-date	×	CONS	Date
has-removal-delay	√	CONS	Integer
is-removal-delayed	√	CONS	Boolean
is-removal-potentially-delayed	√	CONS	Boolean
is-removed	√	CONS	Boolean
is-timely-removed	√	CONS	Boolean
has-reason	×	CONS	String
can-be-delayed-by	×	CONS	CONS
has-performance			
has-constraint-removal-performance	√	PP	Double
has-task-performance	√	PP	Double
has-progress-information			
has-actual-duration	√	TP	Integer
has-actual-duration-from-start	√	TP	Integer
has-actual-finish-date	√	TP	Date
has-actual-start-date	√	TP	Date
has-current-duration	√	TP	Integer
has-current-progress	√	TP	Integer
is-finished	√	TP	Boolean
has-planned-duration	×	TP	Date
has-planned-finish-date	×	TP	Date
has-planned-start-date	×	TP	Date
has-started	√	TP	Boolean
has-total-progress	√	TP	Integer
is-work-delayed	√	TP	Boolean
is-work-potentially-delayed	√	TP	Boolean

5.3 OWL API workflow and ontological reasoning rules

The OWL API and ontological reasoning rules are combined to realise three critical management functions, namely, evaluation of work progress, evaluation of constraint

statuses, and evaluation of participants' performance. The API exports information in ontologies to carry out the complex computation not supported by conventional OWL syntax, then, it imports the results into the ontologies to enable SWRL and SQWRL rules. Figure 5-5 presents the detailed workflow in the OWL API, and the following sections introduce the details of realising the management functions. It should be noted that most rules can be applied to both procedures and tasks. However, for clarity, the examples of rules in the following contents are procedure-level rules.

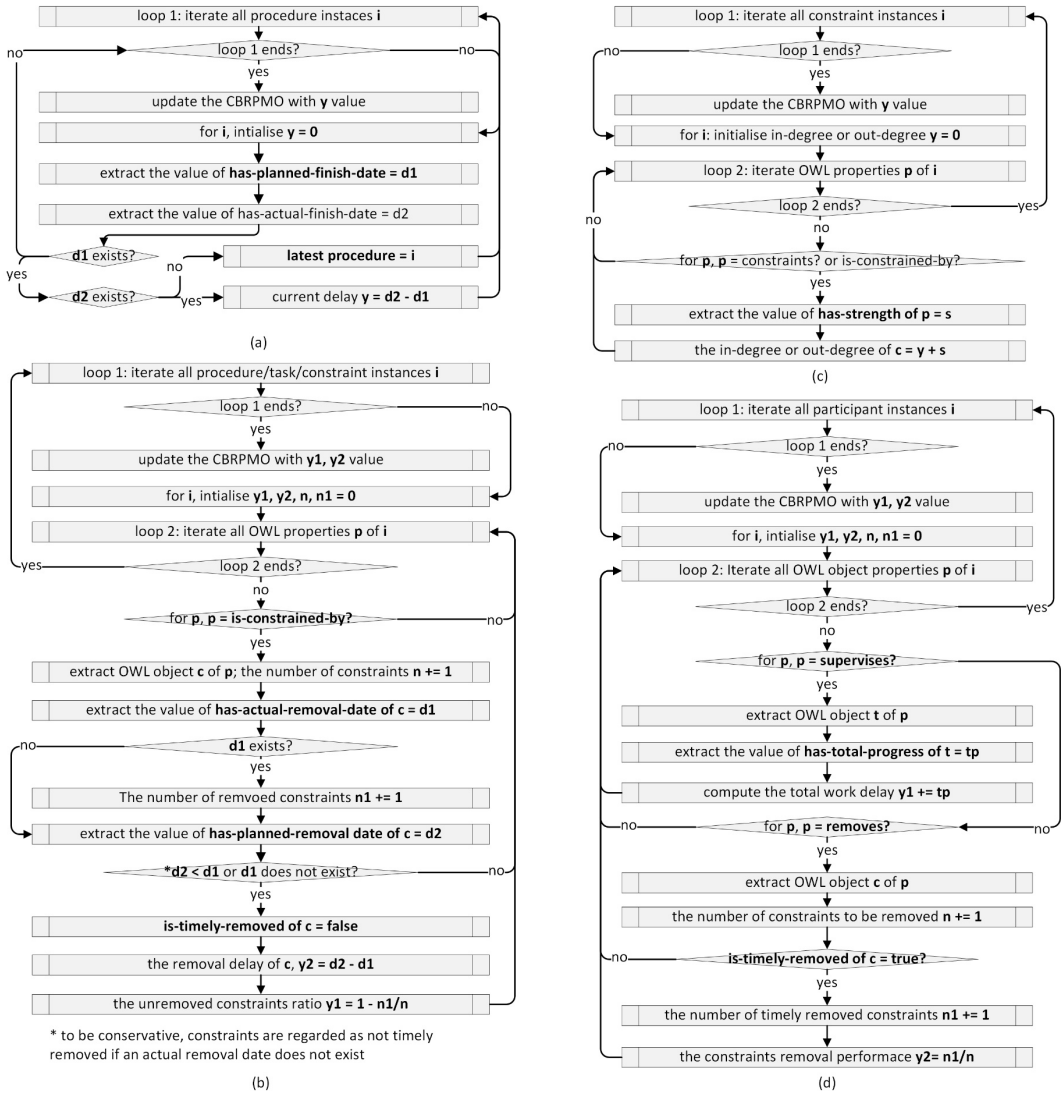


Figure 5-5 Workflow in OWL API

5.3.1 Evaluation of work progress

This function evaluates the progress of a procedure, a task (multiple procedures), and a project (multiple tasks). In practice, task durations and progress are often recorded by starting and ending dates. Given temporal computation is not supported by SWRL and SQWRL, OWL API extracts the date information from the datatype properties of

task/procedure entities, identifies the latest task/procedure that is ongoing, and then computes actual and planned durations and current progress performance (i.e., the specific days behind or ahead of the planned schedule) of each task/procedure. The information is then imported back into the BRMO to enable rules to infer additional schedule information such as identifying the potentially delayed work and evaluating the total delay of a task or project. A flowchart in the API is shown in Figure 5-5(a). Critical rules to realise the function are listed in Table 5-5. It should be noted that during reasoning in the ontologies, the two relations 'has-total-progress' and 'has-current-progress' indicate the progress performance, where the values can be positive (ahead of schedule) or negative (delay).

Table 5-5 Rules for progress evaluation

Rule	Rule body	Explanation
1	*has-actual-duration(?p, ?ad) ^ has-current-progress(?p, ?cp) ^ start-procedure-of(?p, ?t) -> has-total-progress(?p, ?cp) ^ has-actual-duration-from-start(?p, ?ad)*	This rule computes the duration and delay of the starting procedure of a task as its total duration and delay.
2-1	*Procedure(?p1) ^ Procedure(?p2) ^ is-succeeded-by(?p1, ?p2) ^ has-actual-duration-from-start(?p1, ?adfs) ^ has-actual-duration(?p2, ?ad) ^ swrlb:add(?y, ?adfs, ?ad) -> has-actual-duration-from-start(?p2, ?y)*	The procedures of a task are sequential. The rules traverse them to sum the duration and progress values, which enable Rules 3-1 and 3-2 to evaluate the total duration and delay of that task.
2-2	*Procedure(?p1) ^ Procedure(?p2) ^ is-succeeded-by(?p1, ?p2) ^ has-total-progress(?p1, ?tp1) ^ has-current- progress (?p2, ?cp) ^ swrlb:add(?y, ?tp1, ?cp) -> has-total-progress (?p2, ?y)*	
3-1	*latest-procedure-of(?p, ?t) ^ is-preceded-by(?p, ?pp) ^ has-actual-duration-from-start(?pp, ?adfs) ^ has-current-duration(?p, ?cd) ^ swrlb:add(?y, ?adfs, ?cd) -> has-actual-duration-from-start(?t, ?y)*	The rules evaluate the total duration and delay of the latest procedure of a task and then assign the values to the task.
3-2	*latest-procedure-of(?p, ?t) ^ is-preceded-by(?p, ?pp) ^ has-total-progress(?pp, ?tp) ^ has-current-progress(?p, ?cp) swrlb:add(?y, ?tp, ?cp) -> has-total-progress(?t, ?y)*	
4	*Rehabilitation_Task(?t1) ^ Rehabilitation_Task(?t2) ^ is-constrained-by(?t1, ?t2) ^ has-actual-duration-from-start(?t1, ?ad1) ^ has-actual-duration-from-start(?t2, ?ad2) ^ swrlb:add(?y, ?ad1, ?ad2) -> sqwrl:max(?y)*	Task dependencies can be sequential or parallel. The rule enumerates all paths of tasks and computes the maximum duration as the duration of the project.
5-1	*Procedure(?p) ^ is-finished(?p, true) ^ (has-current-progress some xsd:integer[<0])(?p) -> Delayed_Procedure(?p)*	The rules identify delayed procedures and tasks based on the progress values.
5-2	*Rehabilitation_Task(?t) ^ is-finished(?t, true) ^ (has-total-progress some xsd:integer[<0])(?t) -> Delayed_Task(?t)*	

5.3.2 Evaluation of constraint removal

This function covers four aspects: 1) finding constraint removal statuses, i.e., if a constraint is removed and if the removal is delayed, 2) warning delay and evaluating

the reasons of delay, 3) evaluating constraint removal performance, and 4) identifying critical constraints.

The planned and actual removal dates of a constraint can determine if the constraint removal is delayed. The OWL API extracts the date information and computes the removal delay. Then, it updates the information in the BRMO. Delayed constraints (i.e., constraints not timely removed) can warn of potential delay of ongoing tasks or procedures while helping identify reasons for delayed work. In contrast, unremoved constraints can diagnose work before it starts. Based on the number of unremoved and total constraints, the ratio of unremoved constraints is computed (Figure 5-5(b)). The ratio is taken by the reasoning rules to order tasks/procedures and constraints so that constraint entities needing more attention (i.e., they have a high unremoved constraint ratio) are found. The rules can also identify critical constraints. The ontologies KBs in essence is a network. As such, network measures (i.e., in-degree and out-degree) can be computed for each constraint. The former reflects a constraint's vulnerability, i.e., how many constraints can affect it, while the latter reflects its impact, i.e., how many constraints it can affect (Figure 5-5(c)). Those constraints with high degree values are regarded as critical. Some rules to realise the function are summarised in Table 5-6.

Table 5-6 Rules for constraint-removal evaluation

Rule	Rule body	Explanation
6-1	*Constraint(?c) ^ is-constrained-by(?p, ?c) ^ (has-removal-delay some xsd:integer[>= 0])(?c) -> is-timely-removed(?c, true)*	The rules find the delayed constraints for certain procedures according to removal delay.
6-2	*Constraint(?c) ^ is-constrained-by(?p, ?c) ^ (has-removal-delay some xsd:integer[< 0])(?c) -> is-timely-removed(?c, false)*	
6-3	*Procedure(?p) ^ has-started(?p, true) ^ is-finished(?p, false) ^ is-constrained-by(?p, ?c) ^ is-timely-removed(?c, false) -> Potentially_Delayed_Procedure(?p)*	The rule warns delay of ongoing procedures that have delayed constraints.
6-4	*Delayed_Procedure(?p) ^ is-constrained-by(?p, ?c) ^ is-timely-removed(?c, false) -> can-be-delayed-by(?p, ?c)*	The rules find the delayed constraints as the causes of delayed procedures.
6-5	*Potentially_Delayed_Procedure (?p) ^ is-constrained-by(?p, ?c) ^ is-timely-removed(?c, false) -> can-be-delayed-by(?p, ?c)*	
6-6	*Constraint(?c) ^ Procedure(?p) ^ is-constrained-by(?p, ?c) ^ is-timely-removed(?c, false) ^ has-reason(?c, ?r) -> sqwrl:select(?p, ?c, ?r)*	The rule extracts the reasons for delay (if any).
7	*Constraint(?c) ^ Procedure(?p) ^ is-constrained-by(?p, ?c) ^ has-started(?p, false) ^ has-unremoved-constraints-ratio(?c, ?r) -> sqwrl:select(?p, ?c, ?r) ^ sqwrl:orderBy(?r)*	The rule finds the unremoved constraints of procedures not started and then orders them by the unremoved constraint ratio.

8-1	*Constraint(?c) ^ has-out-degree(?c, ?l) -> sqwrl:select(?c, ?l) ^ sqwrl:orderBy(?l)*	The rules rank constraints of a project by their criticality.
8-2	*Constraint(?c) ^ has-in-degree(?c, ?l) -> sqwrl:select(?c, ?l) ^ sqwrl:orderBy(?l)*	

5.3.3 Evaluation of the performance of project participants

Participants' performance is mainly evaluated based on the ability to timely remove constraints and deliver tasks/procedures. Rules created for realising the function can identify responsible participants of delayed tasks/procedures and constraint removal. The rules can also rank participants by performance. To enable these rules, the API traverses the delay of tasks/procedures and constraint removal related to each project participant and then computes its performance following the process shown in Figure 5-5(d), where the delay is computed using the previous two functions. Finally, rules are created so that one can select participants based on certain performance criteria. Critical rules to realise the function are summarised in Table 5-7.

Table 5-7 Rules for participant performance evaluation

Rule	Rule body	Explanation
9	*is-supervised-by(?p, ?pp) ^ Delayed_Procedure (?p) -> Participant_With_Delayed_Procedure(?pp) ^ sqwrl:select(?pp, ?p)*	The rules find participants who fail to complete work or remove constraints on time, respectively.
10	*Constraint(?c) ^ to-be-removed-by(?c, ?pp) ^ is-timely-removed(?c, false) -> Participant_With_Delayed_Constraints(?pp)*	
11	*has-constraints-removal-performance(?pp, ?cp) -> sqwrl:select(?pp, ?cp) ^ sqwrl:orderBy(?cp)*	The rules compare the delay of participants in terms of constraint removal and delivering work then rank participants based on the performance.
12	*has-work-performance(?pp, ?wp) -> sqwrl:select(?pp, ?wp) ^ sqwrl:orderBy(?wp)*	
13	*has-constraints-removal-performance(?pp, ?cp) ^ swrlb:largerThan(?cp, 0.9) -> Good_Participant(?pp)*	The rules select participants based on their performance and certain thresholds.
14	*has-work-performance(?pp, ?wp) ^ swrlb:largerThan(?wp, 0) -> Good_Participant(?pp)*	

5.4 Controlled experiments (information integration and searching)

5.4.1 Ontology preparation

To verify the BRMO, five components must be in place: 1) The TBox, RBox, and ABox, which are built following the steps introduced in Section 3.5 2) An ontology management tool (Protégé 5.50 in this research) that can edit ontologies using state-of-the-art syntax and interact with information in the ontologies using queries. 3) A rule engine that can edit the SWRL and SQWRL rules, and such engines are also supported by the Protégé 5.50. 4) The OWL API (version 5.50) that exports, modifies,

and imports ontology information. 5) A built-in reasoner, i.e., Pellet, that executes rules and infers implicit knowledge. The workflow is shown in Figures 5-6. It should be noted that although the hybrid IE model can automatically extract constraint information, in the experiments, the TBox and RBox were still manually constructed as the skeleton of the KBs, and Figure 5-7 shows the overview of them in Protégé.

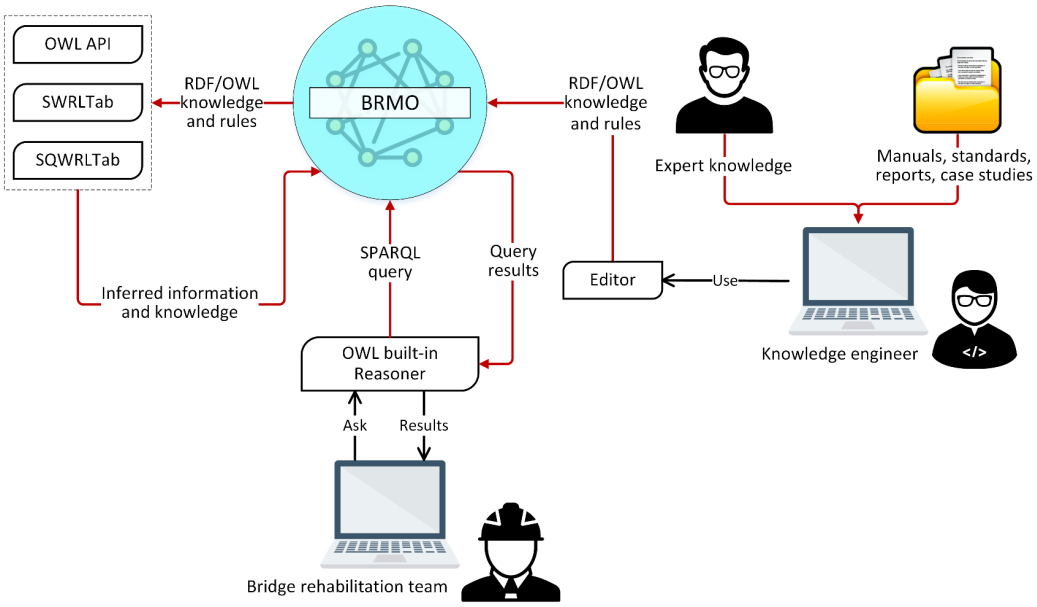


Figure 5-6 Workflow among BRMO components

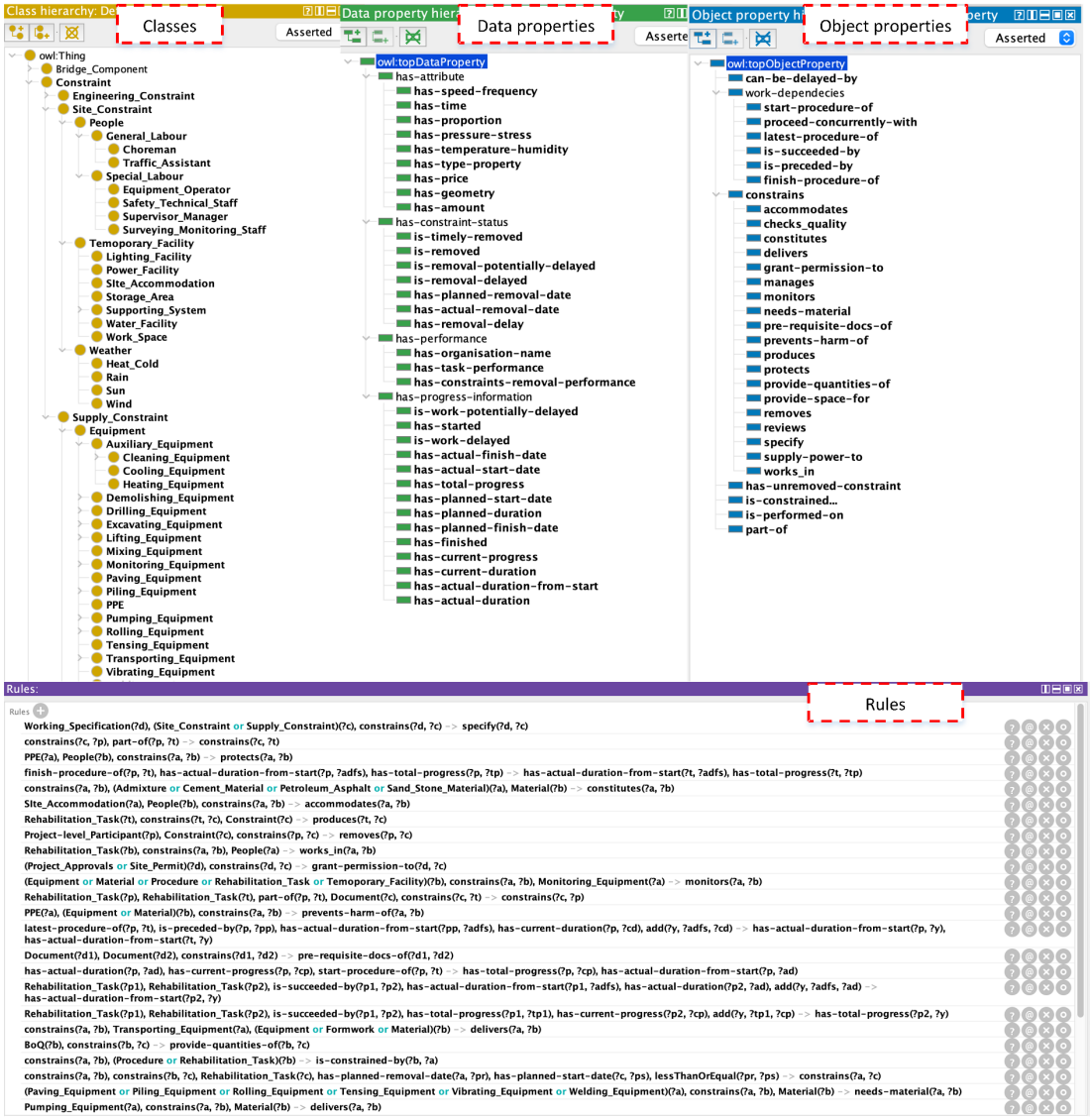


Figure 5-7 Overview of the TBox and RBox in Protégé (no instances)

5.4.2 Information encoding experiments

In the experiments, a large proportion of the ABox was developed by applying the hybrid IE model to automatically extract constraint triples from documents of the deck pavement replacement and concrete wrapping task in the first case project. Table 5-8 and Figure 5-8 compare the statistics and overview of the ontologies in Protégé before and after information encoding. Figure 5-9 shows a complete view of the change of the ontologies. In the experiments, the initial ontologies only included the TBox (class nodes are coloured in red, and the RBox is not visualised). After encoding, the TBox remained unchanged while entities (blue coloured nodes) and relations were added. The two entity clusters in Figure 5-9 represent the two main tasks (i.e., deck pavement replacement and concrete wrapping of bridge piers). Figure 5-10 shows a few encoded triples, which can reveal the positions of entities in the domain class hierarchies and

the relations among entities and classes. As mentioned, some entities and relations cannot be automatically extracted, i.e., the project participants, work packages, and *p2p* and *ct2pp* relations, which were manually inserted in the ontologies, considering the specific project conditions.

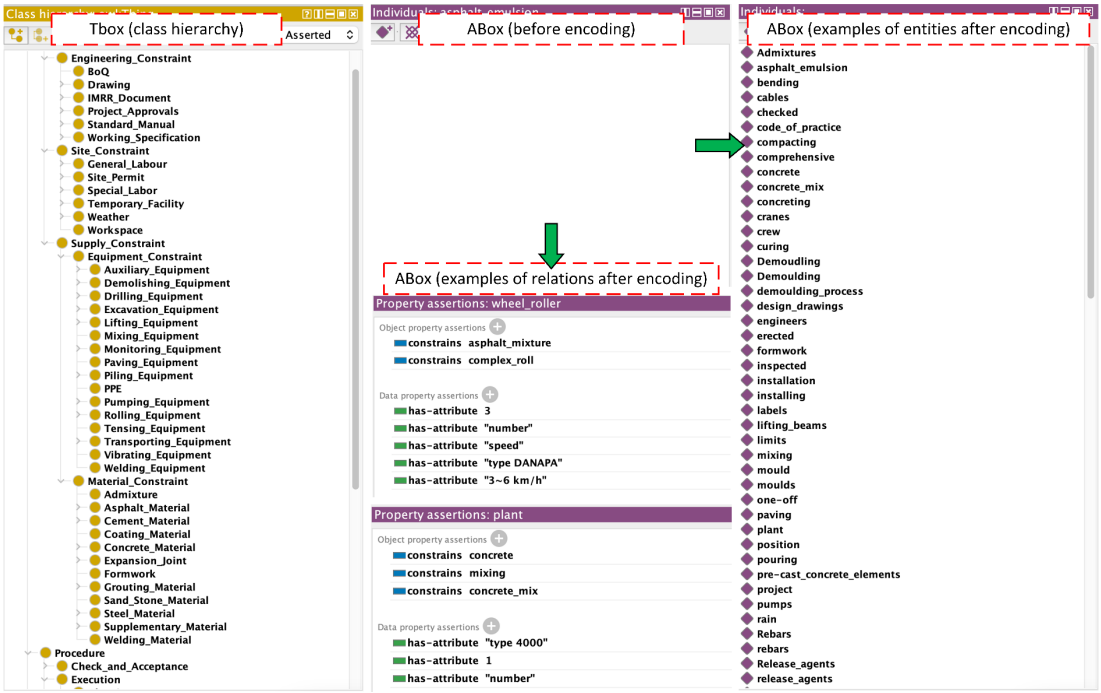
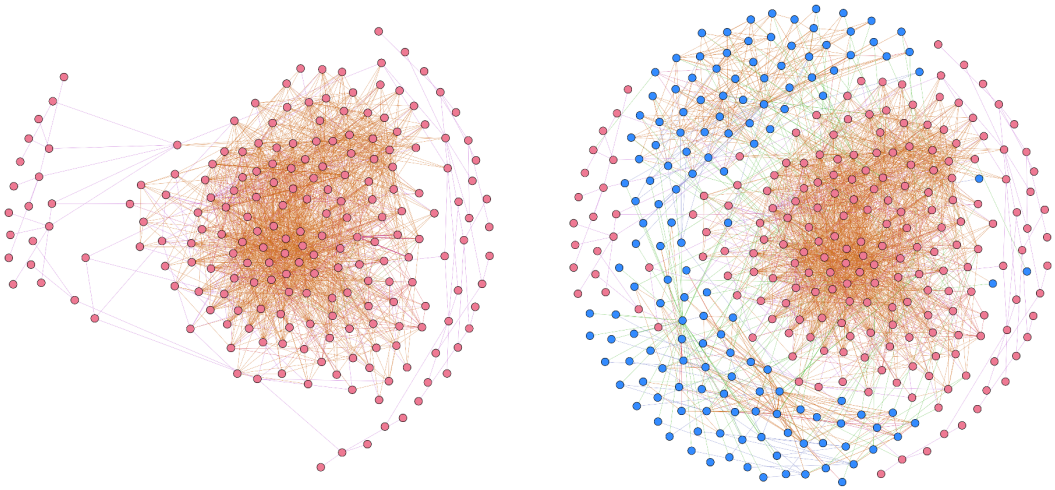


Figure 5-8 Overview of information encoding in Protégé

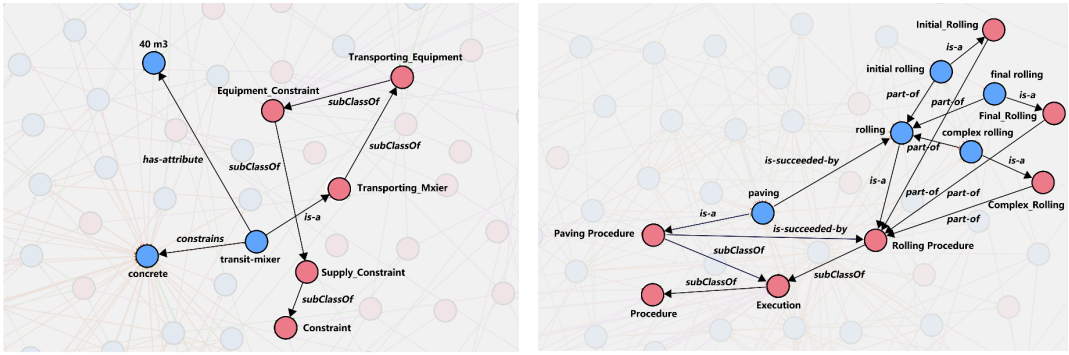
Table 5-8 Statistics of the ontological KBs

Part	Statistics	Before encoding	After encoding
TBox	The number of maximum levels of the class hierarchies	5	5
	The number of classes	207	207
ABox	The number of object relation assertion axioms	0	367
	The number of datatype relation assertion axioms	0	59
	The number of class assertion axioms	0	110
	The number of entities	0	110



(a) (b)

Figure 5-9 Change of ontologies (a) before encoding, (b) after encoding



(a) (b)

Figure 5-10 Examples of encoded triples of (a) the concrete wrapping task, (b) the pavement replacement task

5.4.3 Information searching experiments

After information encoding, the BRPMO was tested in four scenarios. Scenario 1 tested the functions of static information searching. In the scenario 2-4, the BRMO's ability in terms of supporting the three management functions (Section 5.3) based on integrating, inferring, and searching for dynamic constraint information were tested.

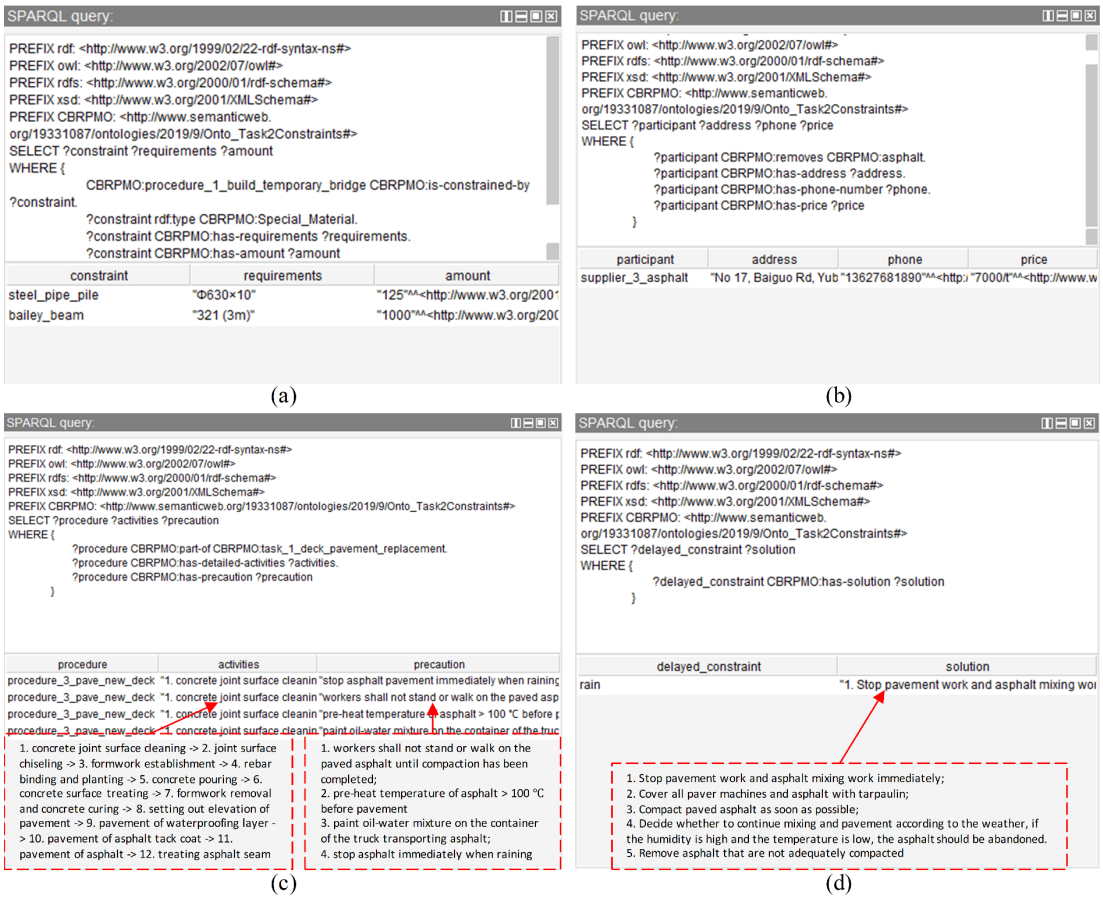


Figure 5-11 SPARQL queries and results

Scenario 1: Some project participants wanted to know more about the project. Instead of searching for information scattered in documents or systems manually, the BRMO encoded such information for efficient retrieval using the SPARQL query. The queries and information searching results are introduced below.

Query 1 (Figure 5-11 (a)) can not only return the specific constraints of procedures or tasks (steel materials for temporary bridge construction in this case) but also detailed requirements (e.g., the type and amount of material constraints) so that the engineer can arrange constraint removal in advance.

Query 2 (Figure 5-11 (b)) can retrieve information (e.g., contact information) of project participants (the asphalt supplier in this case) to facilitate communication among participants.

Query 3 (Figure 5-11 (c)) can show the detailed activities and precautions of a task or procedure (paving the new deck in this case), which are often required by the foreman and supervisors to control onsite work sequences and quality.

Query 4 (Figure 5-11 (d)) can find solutions to the unremoved constraints (rain in this case), serving as remedial actions after delay occurs.

Scenario 2: When the project ongoing, the engineer needed to check the progress of tasks/procedures against plans and identify delayed work. It was assumed that at Sep 15, 2018, the engineer checked the progress of deck paving which was the latest ongoing procedure. The original BRMO only included static information and could not support the checking. Thus, the OWL API extracted the date information of the tasks/procedures to compute duration and progress values. The BRMO was updated with the results, based on which Rules 1-4 in Table 5-5 were run to infer additional progress information of the tasks/procedures (yellow shaded). The process is shown in Figure 5-12, where information computed by the OWL API is highlighted in red boxes. The engineer found that the deck pavement replacement was delayed by 21 days with the first three procedures each contributing to seven days, while the total task duration was 95 days. The delayed tasks/procedures were automatically inferred by running Rule 5 in Table 5-5.

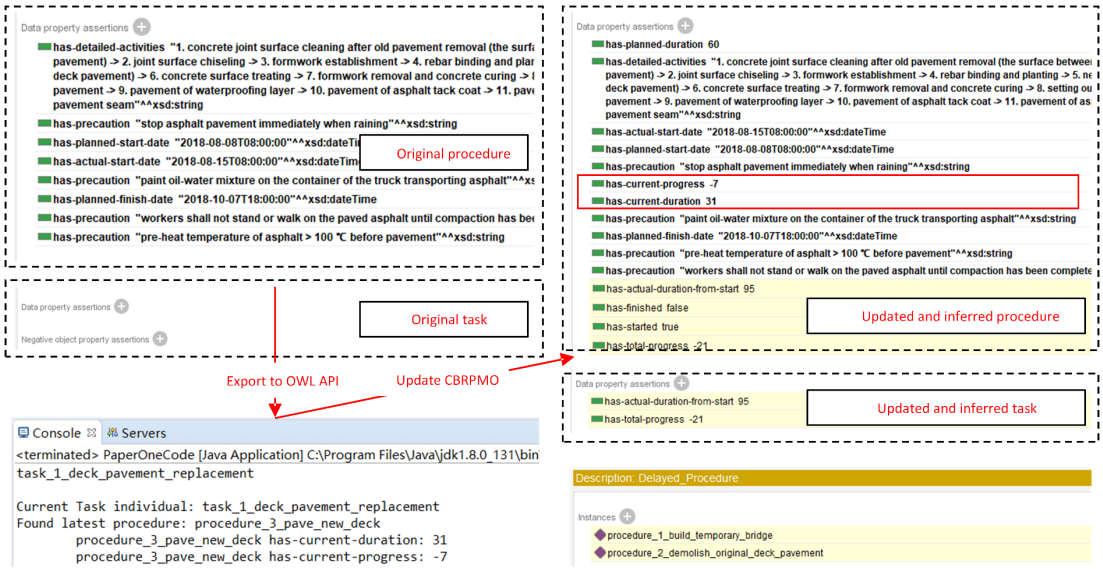


Figure 5-12 Evaluation and inferring of procedure progress

Scenario 3: The engineer wanted to minimise delay by better controlling constraints. For ongoing and finished work, the focus was on the constraints not timely removed. As such, the OWL API took the removal date information of constraints and then computed removal progress. The results enabled rules to identify delayed constraints (Rule 6-1 - 6-2 in Table 5-6), warn potential delay of tasks/procedures (Rule 6-3 in Table 5-6), and infer or extract the causes of delay (Rule 6-4 - 6-6 in Table 5-6). For instance, the engineer found that a warning was triggered for the new deck paving

procedure (Figure 5-13), because its constraints, such as materials (e.g., asphalt), equipment (e.g., the asphalt paver), and labour (e.g., choremen and operators), were not timely removed. In addition, the engineer could explore the reasons of delay following the 'can-be-delayed' and 'has-reason' relations. For instance, one cause of the delay of the asphalt paver was the delay of the temporary power generator, while the delay of the generator was caused by quality issues. For work not started, the focus was on the number of unremoved constraints so that they could be removed before work started. Therefore, the OWL API computed the unremoved constraint ratio of constraints and then enabled Rule 7 in Table 5-6. For instance, as shown in Figure 5-14, on 10 August 2018, when the new deck paving procedure had not started, the engineer checked its constraints and found that the procedure was likely to be delayed by the special labour, as half of constraints affecting the arrival of special labour were not removed yet.

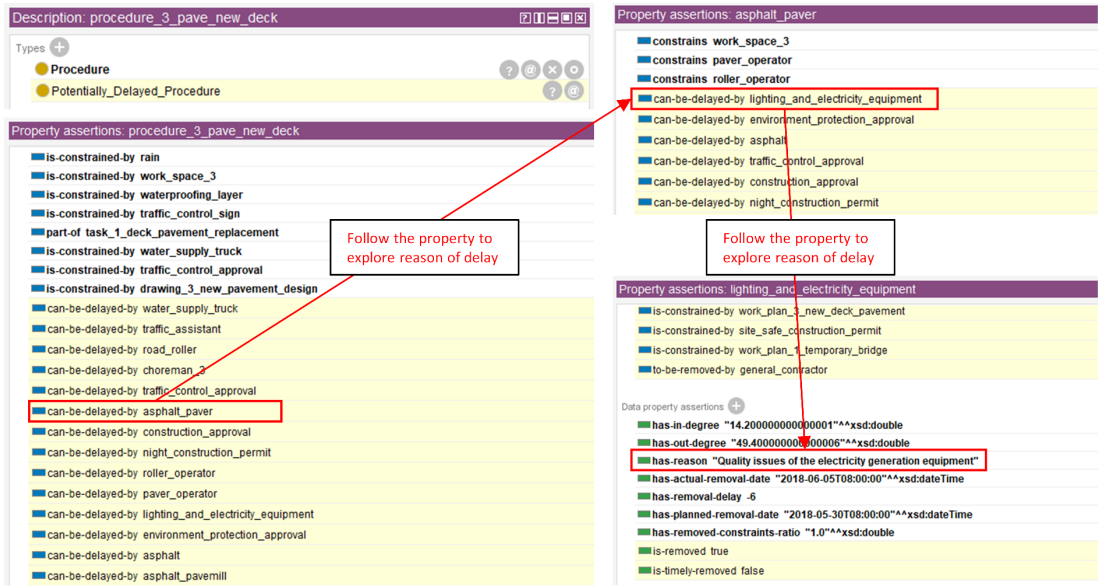


Figure 5-13 Exploration of delayed constraints

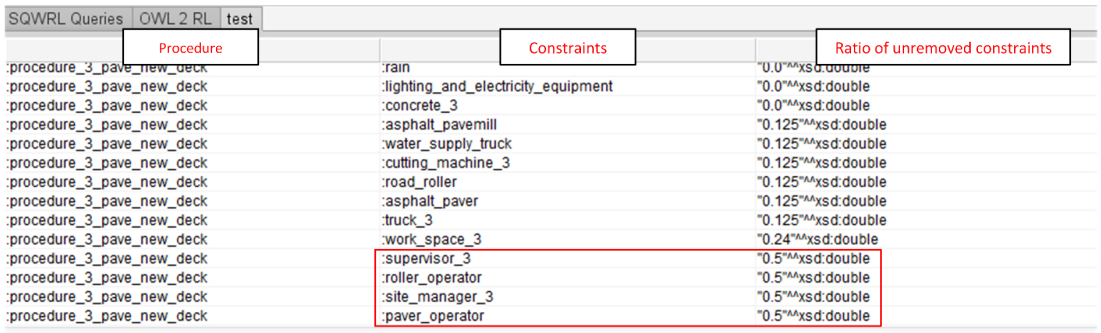
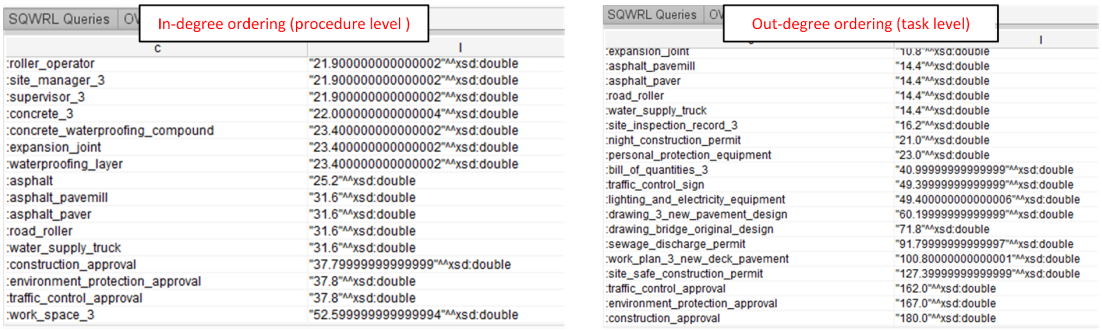


Figure 5-14 Evaluation of unremoved constraint ratio

The in- and out-degrees of constraints were also computed, allowing Rules 8-1 and 8-2 in Table 5-6 to identify critical constraints at different levels. For instance, the vulnerable constraints at the procedure (deck paving) level were identified (Figure 5-15(a)), e.g., workspace, approvals, and equipment. Hence, more attention should be paid to their constraints and related participants to avoid delay. On the contrary, the engineer also found the constraints with greater impact on others at the task (deck replacement) level (Figure 5-15(b)), e.g., approvals, permits, engineering drawings, and temporary facilities. These constraints should be closely monitored, buffer could be assigned to the procedures affected by them, and remedial solutions should be proposed to handle possible delay.



(a) (b)

Figure 5-15 Identification of critical constraints

Scenario 4: The bridge owner wanted to assess the performance of participants for future collaboration. For this purpose, the owner could execute Rules 9-10 in Table 5-7 to identify the participants with delayed tasks/procedures or constraint removal. In addition, to evaluate specific performance of participants, the OWL API computed the total delay when delivering tasks/procedures and ratio of timely removing constraints (i.e., the number of timely removed constraints to the number of total constraints for which the participant was responsible). The results enabled Rules 11 and 12 in Table 5-7, which compared and selected participants. For instance, the owner found that the sub-contractor of the old deck demolishment procedure (sub-contractor_2) had good performance, because it had less work delay (Figure 5-16(a)) and outperformed others in terms of removing constraints (Figure 5-16(b)-(d)). The owner also found that the government agencies, such as the building and construction authority granting the construction approval and DoTs granting the bridge closure approval, had poorer performance in terms of removing constraints (Figures 5-16(b)-(d)), indicating that additional buffer should be assigned to these participants.

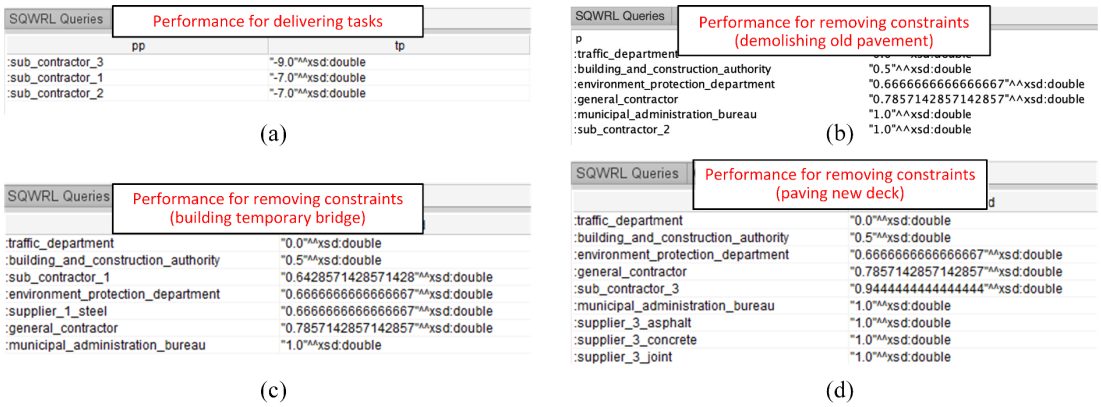


Figure 5-16 Comparison of participant performance

Information to answer the queries in Scenario 1 can be scattered in different sources, and the management functions in scenarios 2-4 could also generate important project information. To show the usefulness of information searching of the BRMO, Table 5-9 lists the time to search for information using the BRMO and manual approach. The average time to search for information by querying the BRMO (i.e., scenario 1) is about 0.1s. The BRMO can largely reduce the searching time from 215.8s to 50.5s, including the time for writing queries. The average time to perform the activities in scenarios 2-4 is 27.3s, only 1/50 of the manual approach (1003s). The time consumed using the manual approach also presents large variance and can dramatically increase when the information is scattered in multiple sources (e.g., scenarios 2 and 4). On the contrary, the time consumed by the BRMO is much more stable, as information has been integrated in the KBs. Besides, scenarios 2-4 involve semantic reasoning based on domain knowledge not explicitly mentioned in documents. Therefore, in some cases (e.g., scenario 3), it can be impossible to obtain the information merely using manual searching.

Table 5-9 Comparison of searching time

Query	Meeting records	Address books	Working plans	BoQ	Domain knowledge	BRMO	Manual
S1, Q1	√		√	√		0.13s (64s)	562s
S1, Q2		√				0.08s (56s)	116s
S1, Q3			√			0.09s (45s)	77s
S1, Q4			√			0.11s (37s)	108s
S2	√		√		√	23s	1172s
S3	√		√		√	44s	n/a
S4	√		√		√	15s	834s

P.S. ‘S’ indicates a scenario and ‘Q’ indicates a query; the time values in parentheses include the time for writing queries; the time values in S2-S4 include the time spent for performing all activities in that scenario, e.g., in S2, the activities include finding the cause of delay, computing the unremoved constraints ratio, and identifying critical constraints.

5.5 Discussion

The BRMO has three contributions. First, existing ontologies for bridge maintenance focus on integrating information at the inspection, evaluation, and decision-making stages (El-Gohary & El-Diraby, 2010; Liu & El-Gohary, 2017c; Ren et al., 2019; Zhou et al., 2016). However, bridge rehabilitation projects involve specific information, e.g., specialised constraints and tasks. Therefore, current ontologies cannot be used directly. The development of the BRMO relies on comprehensive collection of bridge rehabilitation knowledge from various sources, e.g., standards, manuals, case reports, and previous studies. The knowledge was further refined through the focus group. As such, the BRMO covers adequate domain knowledge of bridge rehabilitation and can integrate information of constraints, tasks/procedures, and project participants. Thus, the BRMO extends the coverage of domain ontologies to the bridge rehabilitation stage. Besides, extensibility and flexibility are important features of ontologies. The BRMO can be merged with current ontologies without significant modifications. For instance, the entities of the ‘Procedure’ class can be linked to bridge components in existing ontologies through the relation ‘is-performed-on’ (Liu & El-Gohary, 2017c; Ren et al., 2019). Therefore, the BRMO has unique contributions and is compatible with previous work in the field.

Second, the BRMO supports a novel information updating approach which improves the functions of conventional ontologies in the AEC industry. Most ontologies handle information of static objects (e.g., components) and facts (e.g., defects) (Niknam & Karshenas, 2017; Zhang et al., 2015). However, owing to the syntax limitations, these ontologies cannot perform complex computation and dynamic updating thus cannot integrate dynamic information in ongoing projects. Even in previous studies focusing on process-oriented ontologies, the functions are simple and only work as auxiliary parts. For instance, procedure entities are generated to merely store other information of a procedure (e.g., hazards and constraints), which however are not considered in

reasoning and computation (Wang, 2018; Zhang et al., 2015)). Progress information in these ontologies is also simple and qualitative (e.g., progress is recorded using 1-5 ratings rather than actual durations). As such, sophisticated management functions (e.g., detailed progress evaluation and delay analysis) are not supported (Dong et al., 2011). On the other hand, the BRMO combines the SWRL, SWRQL, and OWL API to address the limitations. Thus, the BRMO can manage dynamic and quantitative project information (e.g., constraint removal and task/procedure progress). When the information is imported, the BRMO can support various management functions, e.g., estimating delay of tasks, procedures, and constraint removal, identifying critical constraints, and evaluating participants' performance. Although these functions can also be realised in traditional tools (e.g., Microsoft Project), one can conveniently navigate in the BRMO to explore implicit information (e.g., finding causes of delay). The easy semantic exploration is a key benefit of ontological KBs, which is difficult to be realised in traditional tools, as information in these tools is not integrated in a data format neutral and unambiguous manner (Park et al., 2013; Ren et al., 2019; Woldesenbet, 2014).

Third, current constraint management approaches (e.g., AWP) suffer from manual information searching, which can delay information delivery and hinder constraint removal. Moreover, finding some information (e.g., the critical constraints) relies on knowledge reasoning (e.g., interpreting the relations among constraints). Finding such information is extremely difficult using the manual approach. On the contrary, the BRMO can search for, compute, reason, and update both static (e.g., the required material types and contact information of participants) and dynamic information (e.g., progress of tasks/procedures and constraint removal) in a much shorter time than the manual searching, especially when the information is scattered in multiple sources. Thus, the BRMO can improve AWP by automating the information searching step. Enormous time can be saved for proposing more effective constraint removal plans.

5.6 Chapter summary

This Chapter introduces the BRMO to improve information integration and searching in bridge rehabilitation projects. Development of the BRMO is based on adequate domain knowledge and follows a guideline. The BRMO has three class taxonomies (i.e., tasks/procedures, constraints, and participants) and two relation hierarchies. By

combining the SWRL, SQWRL, and OWL API to export, compute, import, and infer information, the BRMO can overcome the OWL syntax limitations in conventional ontologies. Thus, the BRMO supports integrating, inferring, and searching for both static and dynamic constraint information. The BRMO was validated in controlled experiments. The results prove that the BRMO can efficiently integrate constraint information in ongoing projects. Based on the continuously updated information, the BRMO can realise essential functions for project management, e.g., computing the delay of tasks/procedures and constraint removal, identifying critical constraints, and evaluating performance of participants. The BRMO extends the coverage of domain ontologies in the bridge sector to the rehabilitation stage. In practice, the BRMO can promotes AWP implementation by providing timely access to project information, which can facilitate constraint monitoring and removal.

Chapter 6: Developing automatic methods for constraint knowledge base completion

6.1 Chapter introduction

This chapter presents the detailed design of the KBC model for identifying missing information in ontological KBs. Cross-comparison experiment results are introduced to show the effect of enriching data semantics and adding domain information (class and working context information). Controlled experiment results are also introduced to show the usefulness of the KBC model in practice. The KBC model was developed with Python 3.7 and Pytroch 1.7.1. Model training, validation, and testing were carried out on the Google Colab cloud computing platform.

6.2 Detailed design of the KBC model

6.2.1 Ontology-based data enriching module

Data enriching has two key steps: mapping entities to domain classes and enriching data semantics using ontology rules. Class mapping mechanism has been introduced in Section 4.4.1, and this section introduces the rule-based data enriching in detail.

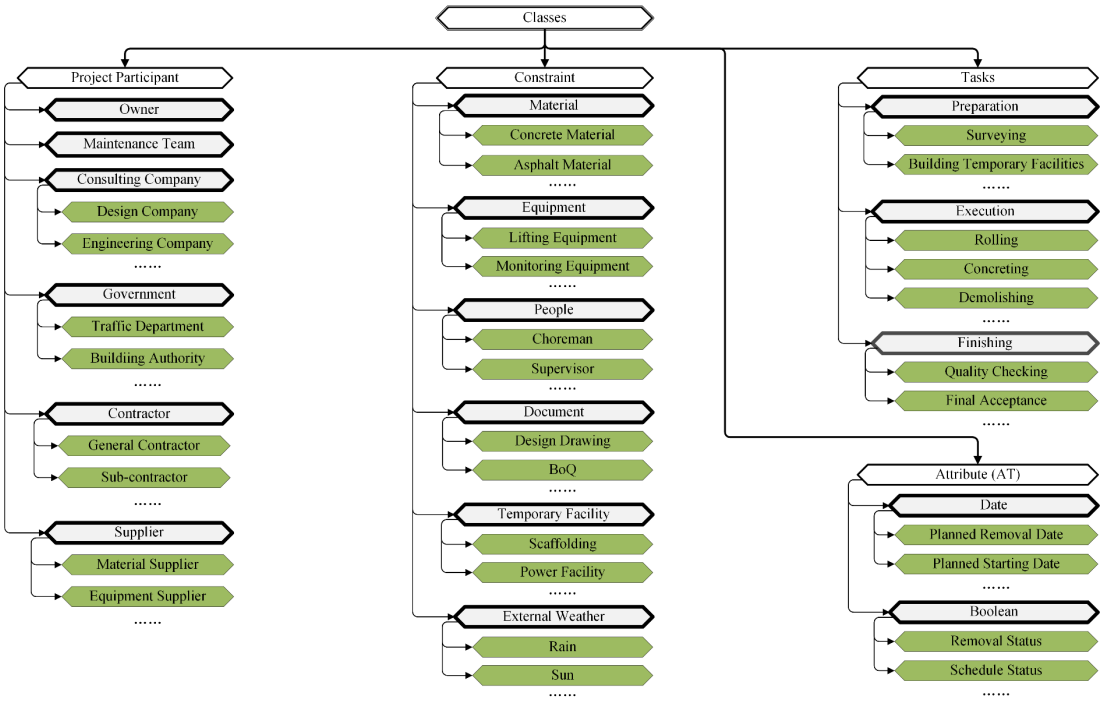


Figure 6-1 Domain classes for data enriching

The ontologies and relation hierarchies are presented in Chapter 5. It should be noted that the full ontologies are used to enrich data through rule reasoning (introduced below). However, when identifying and adding class information to the GNN (Section 6.2.2), not all classes are used. In that case, the number of entities belonging to each class is small, which can cause underfitting and require more data to allow the model to learn entity-class patterns. Moreover, it is commonly difficult to capture relations between constraints and attributes, as attributes vary significantly (e.g., the amount of concrete usage for different tasks). In most cases, such data are also very sparse (e.g., one type of equipment can only appear once in KBs). Therefore, adding all attribute data when training the KBC model can hurt its performance. To overcome the issue, only attributes of the 'Date' (e.g., the removal date of constraints) and 'Boolean' (e.g., if a constraint is removed and if a task is started) classes are considered. The classes used in the encoder are highlighted by the bold borders in Figure 6-1.

Based on the ontologies, semantic rules are constructed to enrich data in the KBs. All the rules follow the SWRL syntax introduced in Section 2.3.1 and Section 2.4.1. The rules only take the relations with basic semantics: 'constrains', 'part-of', 'is-a', 'is-succeeded-by', and 'has-attribute' and identify all axioms satisfying the conditions at rule bodies. The rules have two purposes: 1) adding triples (inferring new triples with basic relation semantics) and 2) enriching relation semantics (inferring more complex

relation expressions based on the relation hierarchies). Newly inferred triples with basic relation semantics are also fed into the semantic enriching process. There are in total 42 rules in the data enriching module, and some examples are summarised in Table 6-1.

Table 6-1 Examples of rules for adding data semantics

Purpose	Rule
Adding triples	constrains(c, t1), part-of(t1, t2) ⇒ constrains(c, t2)
	constrains(c1, c2), is-removed(c1, true) ⇒ is-removed(c2, true)
	constrains (a, b)), constrains(b, t), Task(t), has-planned-removal-date(a, pr), has-planned-start-date(t, ps), lessThanOrEqual(pr, ps) ⇒ constrains(a, t)
Enriching semantics	Task(t), constrains(p, t), People(p) ⇒ works-in(p, t)
	Manager(m), constrains(m, c), (Equipment or Material or Temoporary_Facility)(c) ⇒ checks-quality(m, c)
	Power_Facility(f), constrains(f, c) ⇒ supply-power-to(f, c)
	Work_Space(s), constrains(s, c) ⇒ provide-space-for(s, c)
	Lifting_Equipment(e), constrains(e, c), Material(c) ⇒ transport(e, c)
	BoQ(b), constrains(b, c), (Equipment or Material)(c) ⇒ specify(b, c)

6.2.2 GNN-based encoder

In general, a ***KB*** = ***(E, R, T),*** where ***E***, ***R***, and ***T*** are the set of entities (i.e., nodes), relations (i.e., edges), and valid triples **(h, r, t)**, respectively. ***N*** and ***M*** are the total number of entities and relations, respectively. Each node and relation are associated with an embedding $\in \mathbb{R}^{D_0}$, i.e., $\{h_1, h_2, \dots h_N\}$ for nodes and $\{r_1, r_2, \dots r_M\}$ for edges. $\mathcal{N}_i$ denotes the 2-hop neighbourhood of central node *i*. A triple between node *i* and *j* in $\mathcal{N}_i$ is denoted as t_{ijp}, (i.e., **h**=h_i, **t**=h_j, and **r**=r_p). All triples starting from node *i* in $\mathcal{N}_i$ is denoted as $\mathcal{T}_i$. Then, an adjacency matrix ***AD*** can be built, where each entry is 1 or 0, denoting if two nodes are linked. Thus, a KB has three basic matrices: $\boldsymbol{EM} \in \mathbb{R}^{N \times D_0}$ (each row refers to an entity), $\boldsymbol{RM} \in \mathbb{R}^{M \times D_0}$ (each row refers to a relation), and $\boldsymbol{AD} \in \mathbb{R}^{N \times N}$. It should be noted that in the GNN-encoder, if node *i* is not directly linked to node *j* in $\mathcal{N}_i$ (e.g., the node *j* is in the outer layer of $\mathcal{N}_i$), a virtual relation is setup between them. The embedding of the virtual relation is computed by summing the relation embeddings along the path (see Figure 6-2(b)) (Nathani et al., 2019). Given valid triples $\in \boldsymbol{T}$, the proposed KBC model can find missing triples by classifying potential triples in the hidden set ***H***, where $\boldsymbol{H} \cap \boldsymbol{T} = \emptyset$.

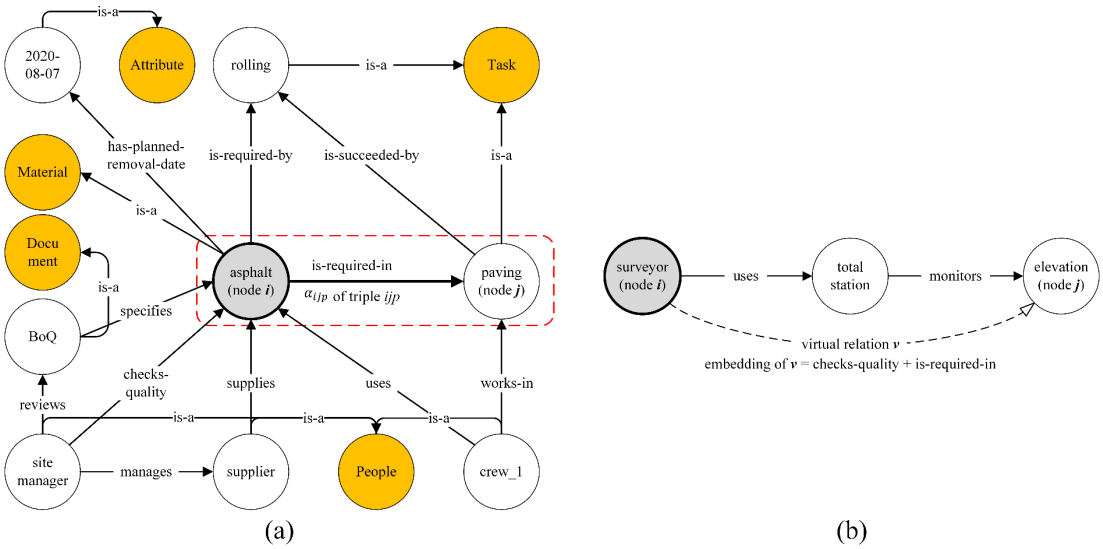


Figure 6-2 (a) Neighbourhood expanded by domain classes, (b) Virtual relations

The proposed KBC model concerns three types of nodes, i.e., CONS, AT, and TP entities. As for relations, to cover rich semantics, all relation types which can be reasoned by rules in Table 6-1 are included. At each iteration of the encoding, the GNN computes $\{h_1^k, h_2^k, \dots h_N^k\} \in \mathbb{R}^{D_k}$ for nodes and $\{r_1^k, r_2^k, \dots r_M^k\}$ for relations, where k=$\{0, 1 \dots \boldsymbol{K}\}$ is the k^{th} iteration (e.g., h_i^0 refers to the original embedding of a node). Algorithm 1 (Figure 6-3) shows the pseudo code of encoding, while Figure 2-3 illustrates the process. Each iteration can be realised by three functions introduced below, i.e., SAMPLING, AGGREGATING, and UPDATING. After each iteration, a transformation matrix $\boldsymbol{WR}$ is employed to update the relation embeddings ($\boldsymbol{RM}$) to adapt the change of node embeddings.

Algorithm 1 overall encoding process

Input: node set $\boldsymbol{E}$; original embeddings: $\boldsymbol{EM}$; $\boldsymbol{RM}$; $\boldsymbol{AD}$; transformation matrices $\boldsymbol{W0}$; $\boldsymbol{W1}$; $\boldsymbol{W2}$;$\boldsymbol{K}$; $\boldsymbol{S}$
Onput: new embeddings $\boldsymbol{EM}$; $\boldsymbol{RM}$

for k=1,2$\cdots \boldsymbol{K}$ **do**
  $\boldsymbol{EM}$, $\boldsymbol{RM}$ = normalise($\boldsymbol{EM}$, $\boldsymbol{RM}$)
  for node $i \in \boldsymbol{E}$ **do**
    $h_i^{k-1} \xleftarrow{looking\ up} \boldsymbol{EM}$
    if k=0 **then**
      $\mathrm{h}_i^{k-1} = \mathrm{h}_i^0$
    end if
    $\mathcal{T}_i \xleftarrow{get} \boldsymbol{AD}$
    $\mathcal{N}_i \xleftarrow{get} \boldsymbol{AD}$
    $\mathcal{A}_i = \textbf{SAMPLE}(\mathcal{T}_i, \mathcal{N}_i, \boldsymbol{EM}, \boldsymbol{RM}, \mathrm{h}_i^{k-1}, \boldsymbol{W1}, \boldsymbol{W2})$
    $\mathrm{h}_i^k = \textbf{AGGREGATE}(\mathcal{A}_i, \mathcal{T}_i, \mathcal{N}_i, \mathrm{h}_i^{k-1}, k, \boldsymbol{K})$
    h_i^k, working context dictionaries = $\textbf{UPDATE}(\mathcal{N}_i, \boldsymbol{AD}, \mathrm{h}_i^k, \mathrm{h}_i^0, \boldsymbol{W0})$
  end for
  $\boldsymbol{RM} = \boldsymbol{WR} \times \boldsymbol{RM}$
end for

Figure 6-3 Overall algorithm of GNN encoding

6.2.2.1 Attention-based neighbourhood sampling

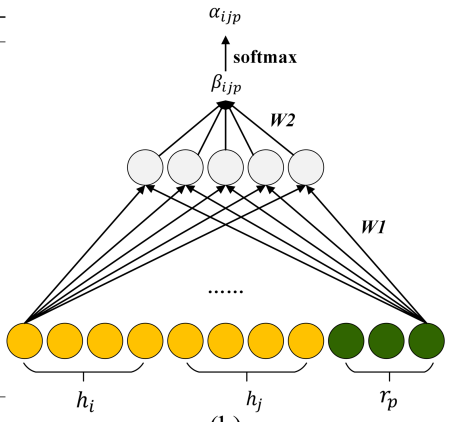
The 'SMAPLE' function computes the attention values (i.e., relative importance) to identify important nodes in $\mathcal{N}_i$ (Velickovic et al., 2017). A KB usually has different types of nodes and relations, and a node can play different roles. Therefore, attention values should consider both nodes and relations. For instance, in Figure 6-2(a), the node 'asphalt' appears in two triples: it is-required-by 'paving' while a 'supplier' is responsible for supplying it. The attention mechanism computes an attention value α_{ijp} for each triple in $\mathcal{T}_i$. The process for computing attention is visualised in Figure 6-4. Taking the node *i* in Figure 6-2(a) as an example, a triple embedding c_{ijp} for the node is created by stacking embeddings of the three triple elements into a matrix which is fed into two transformation matrices (i.e., ***W1*** and ***W2***) and a Relu non-linearity function. If the relation *p* is a virtual relation for multi-hop connections, it is computed as the sum of relation embeddings in the path of connections. Then, β_{ijp} is computed as the absolute attention value of c_{ijp}. Finally, *softmax* function is used to convert β_{ijp} into the relative attention value α_{ijp} (Nathani et al., 2019).

Algorithm 2 SAMPLE: computing attention values

Input: $\mathcal{T}_i$; $\mathcal{N}_i$; ***EM***; ***RM***; h_i^{k-1}; ***W1***; ***W2***
Onput: attention information $\mathcal{A}_i$ of node i

1: **for** node $j \in \mathcal{N}_i$ **do**
2: $h_j^{k-1} \xleftarrow{get} \mathcal{N}_i$, ***EM***
3: $r_p^{k-1} \xleftarrow{get} \mathcal{T}_i$, ***RM***
4: $\mathcal{A}_i \xleftarrow{initialise}$ *an empty list*
5: $\mathrm{c}_{ijp} = \boldsymbol{W1} \times (\mathrm{Concat}(\mathrm{h}_i^{k-1}, \mathrm{h}_j^{k-1}, \mathrm{r}_p^{k-1}))$
6: $\beta_{ijp} = \mathrm{Relu}(\boldsymbol{W2} \times \mathrm{c}_{ijp})$
7: $\alpha_{ijp} = \frac{exp(b_{ijp})}{\sum_{\mathcal{N}_i}\sum_{\mathcal{T}_i} exp(b_{ijm})}$
8: $\mathcal{A}_i \xleftarrow{append} (\alpha_{ijp},\ c_{ijp})$
9: **end for**
10: **return** $\mathcal{A}_i$

(a)



(b)

Figure 6-4 (a) Algorithm of SAMPLE function, (b) Attention mechanism

Furthermore, classes of entities in the ontological KBs are identified and inserted as additional nodes using class mapping. This expands the neighbourhood of entities by further including class nodes when computing attention values. An example is shown in Figure 6-2(a), which is the expansion of Figure 3-5 (work packages are omitted for clarity). With domain class information, the model can learn patterns among entities as well as between classes and entities. As suggested by (Lin et al., 2015), adding class

information can divide KB entities into 'clusters', and the model can better learn triple patterns in distinct clusters thus improve its learning efficiency and capacity.

6.2.2.2 Multi-head information aggregation

The 'AGGREGATE' process is illustrated in Algorithm 3 (Figure 6-5). The hidden embedding h_i^k of node *i* is computed by summing all triple embeddings c_{ijp} weighted by attention values in $\mathcal{N}_i$. In addition, the multi-head attention mechanism is applied to stabilise the process and gather more neighbourhood information. At each iteration, $\boldsymbol{S}$ attention heads compute the $\hat{h}_i^k$ values independently and simultaneously, which are concatenated as the final output h_i^k. An exception is the last encoding iteration (i.e., k=$\boldsymbol{K}$), where the h_i^K is computed by averaging the multi-head attention results to merge the information.

Algorithm 3 AGGREGATE: aggregate neighbourhood information

Input: $\mathcal{A}_i$; $\mathcal{T}_i$; $\mathcal{N}_i$; h_i^{k-1}; k; $\boldsymbol{K}$; $\boldsymbol{S}$;
Onput: hidden embeddings h_i^k

$sum \xleftarrow{initialise} 0$
for s=1,2$\cdots \boldsymbol{S}$ **do**
  $(\alpha_{ijp},\ c_{ijp}) \xleftarrow{get} \mathcal{A}_i$
  $\hat{h}_i^k = \sigma(\sum_{\mathcal{N}_i} \sum_{\mathcal{T}_i} \alpha_{ijp} \mathrm{c}_{ijp})$
  $\mathrm{h}_i^k = \mathrm{Concat}(\hat{h}_i^k)^s$
  $sum = sum + \hat{h}_i^k$
end for
if k=$\boldsymbol{K}$ **then**
  $\mathrm{h}_i^k = \frac{sum}{\boldsymbol{S}}$
end if
return h_i^k

Figure 6-5 Algorithm of AGGREGATE function

6.2.2.3 Graph information updating

After sampling and aggregation, the 'UPDATE' function replaces h_i^{k-1} with h_i^k that is returned from the 'AGGREGATE' function. At each iteration, the encoder gathers information from $\mathcal{N}_i$ (2-hop) of the node *i*. However, critical semantics in the node's original embedding can be lost when $\boldsymbol{K}$ is large (i.e., when the information has been passed through multiple iterations). The information loss can hurt the model especially when the initial embeddings (i.e., h_i^0) are not random values, e.g., they are initialised with word embeddings. To recover such information, the original embedding of each node is transformed by a matrix $\boldsymbol{W0}$ and then added to h_i^K after the last encoding iteration following Eq. 4-14.

$$h_i^K = h_i^K + W0 \times h_i^0 \quad \text{Eq. 4-14}$$

Another critical activity during information updating is to identify working contexts of nodes. Working contexts are determined based on tasks (i.e., TP entities). For the TP nodes, their working contexts are themselves. For CONS nodes, they follow the top-level relation 'constrains' to identify the TP entities they affect as their working contexts. For AT nodes, as one attribute can be only linked to one constraint, their working contexts are the same as the CONS nodes to which they are linked. All identified working contexts (i.e., TP entities) are mapped to their domain classes to avoid ambiguation and duplication. For instance, the working contexts of the entity 'asphalt' are identified as {'Paving', 'Rolling'}. A dictionary is built for each node to record its working contexts. Adding working context information can again help the model cluster entities and minimise false positive triples. For instance, if the model learns 'crew_1 constrains rolling', it is likely to predict the triple 'crew_2 constrains rolling' as a valid missing triple, as the two triples have similar head/tail entities and connection structures in the KB. However, the second triple should be invalid when crew_2 is not assigned to the rolling task (i.e., the two head entities do not belong to the same working context). This kind of information cannot be correctly recognised until working contexts are utilised. It should be noted that working contexts are only used during decoding process to improve the CNN-based decoder (see Section 6.2.3).

6.2.3 KRL-based decoder

The CNN-decoder takes the similar structure to the KRL model for triple extraction. However, the KRL model stacks domain class information while the decoder stacks working context information in the input matrix. Besides, an additional step is needed to improve the model structure in the CNN-decoder, i.e., encoding working context information. Based on different TP entities involved, the dictionaries built during the updating process can include many combinations of working contexts. For instance, entities 'asphalt_mixture' and 'asphalt_paver' have same working contexts {'Paving', 'Rolling'}, and the working contexts of the entity 'supervisor' have more TP entities {'Paving', 'Rolling', 'Acceptance'}. The word embeddings of classes of TP entities in a working context dictionary are extracted and averaged. Suppose the number of working context combinations is c, a matrix $\boldsymbol{C} \in \mathbb{R}^{c \times D_K}$ is created to encode all of them. The head or tail entities of any triple can look up the matrix $\boldsymbol{C}$ to retrieve the working context embeddings. The mechanism is shown in Figure 6-6. Then, working

context embeddings of the head/tail entities in a triple are stacked at the left and right sides of the input matrix, respectively, and the matrix is fed to model training.

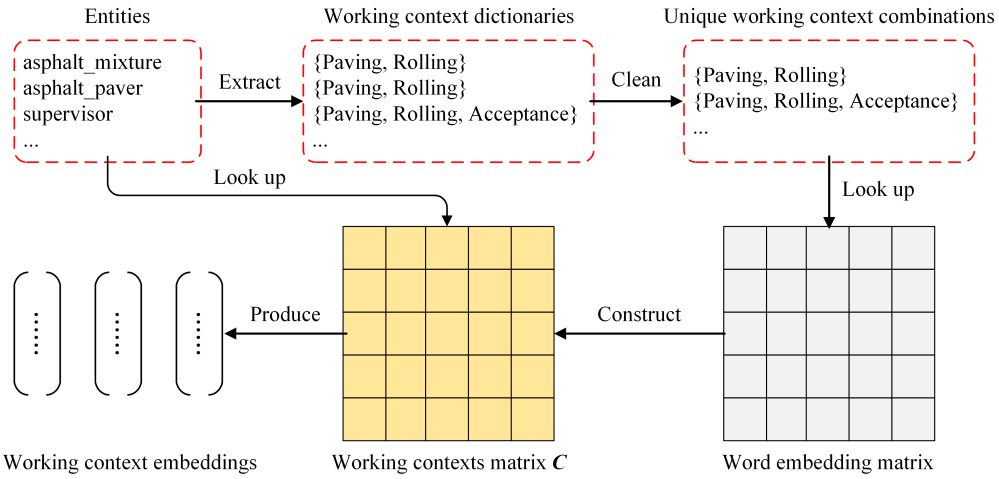


Figure 6-6 The decoding process

6.3 Knowledge base completion experiment results

6.3.1 Data preparation and hyper-parameter tuning

Experiments of the KBC model were performed based on the data for developing the hybrid IE model (Section 4.5). In total, 6049 triples were extracted to train the KBC model. The raw triples only included three types of relations with simple semantics: 'constrains', 'has-attribute', and work dependencies ('is-succeeded-by') and 'part-of'. The raw triples were fed into the data enriching module, where the number of relation types was increased from 39 to maximum 74, and 17587 triples (with simple and rich semantics) were generated. The protocols of training, validation, testing, and hyper-parameter tuning are similar to those for developing the Bi-LSTM-CRF and KRL model. The results of hyper-parameter tuning are listed in Table 6-2.

Table 6-2 Results of hyper-parameters tuning

Hyper-parameters	Explanation	Potential values	Optimal values
Learning rate	Control model parameters updating	$\{5^{-3}, 8^{-3}, 10^{-3}, 5^{-2}, 8^{-2}, 10^{-2}\}$	0.008
Batch size	Divide data into batches which are taken by the model separately	$\{2^k\ k$=1,2…12$\}$	1024
Epoch	Decides the number of times that the model processes all training data	{100, 200…2000}	600
K	The number of attention iterations	{1, 2, 3}	1
L	The number of CNN filters	{1 - 50}	12
Embedding size	The dimension of embeddings of nodes, relations, and words/characters	{100, 200, 300}	200

Activation function	Trigger non-linearity transformation in the model structure	{sigmoid, Relu, Elu}	Relu
Optimiser	Compute gradients to update model parameters	{Adam, RMSprop, Momentum, Stochastic gradient descent}	Adam

6.3.2 Model results and analysis

6.3.2.1 Overall results

Eight experiments were carried out, which could reveal model performance under eight model configurations with increasing complexity, i.e., with increasing semantics or domain specific information (Table 6-3). To minimise the impact of randomness in initialisation. To evaluate each model configuration, the model was ran for ten times, and the median values of model performance were computed as results, which are shown in Table 6-4. The performance metrics in and out of the parentheses indicate the metrics of training and testing, respectively. The best and the second-best metrics in the testing dataset are highlighted in the bold and italic font, respectively.

Table 6-3 Summary of model configurations

Config	Simple semantics	Rich semantics	Class	Context
SR	√			
SR+T	√			√
SR+C	√		√	
SR+C+T	√		√	√
R		√		
R+C		√	√	
R+T		√		√
R+C+T		√	√	√

Figure 6-7 plots the model loss curves during training. Although models using simple semantics have slightly higher loss during first 300 epochs, all models can converge after 600 epochs. On the other hand, according to Table 6-4 and Figure 6-8, the R+C+T configuration outperforms other configurations. The simplest configuration SR (i.e., the model without any enriching) produces the worst performance. Besides, based on Figure 6-8, it can be argued that: 1) the models with rich semantics outperform those with simple semantics in all situations; 2) with rich semantics, the variance of metrics is also largely reduced. This is because rich semantics can increase expressiveness of data, thus the model can better distinguish entities linked by different relations. For instance, if only the simple 'constrains' relation is used, the model can assign high scores to all triples of the form 'entity constrains entity/task', which can lead to many false positives.

On the other hand, Figure 6-9 illustrates the change of attention values of all nodes in the KBs under R, R+T and R+C configurations. During training, the R+T and R+C configurations have similar patterns of attention value changing, which are different from the patterns of the R configuration. In addition, when class or working context information is added, the number of nodes with high attention values is less than that of using R configuration. This indicates the models enhanced by domain information can better distinguish important nodes from irrelevant ones when nodes are clustered according to classes or working contexts.

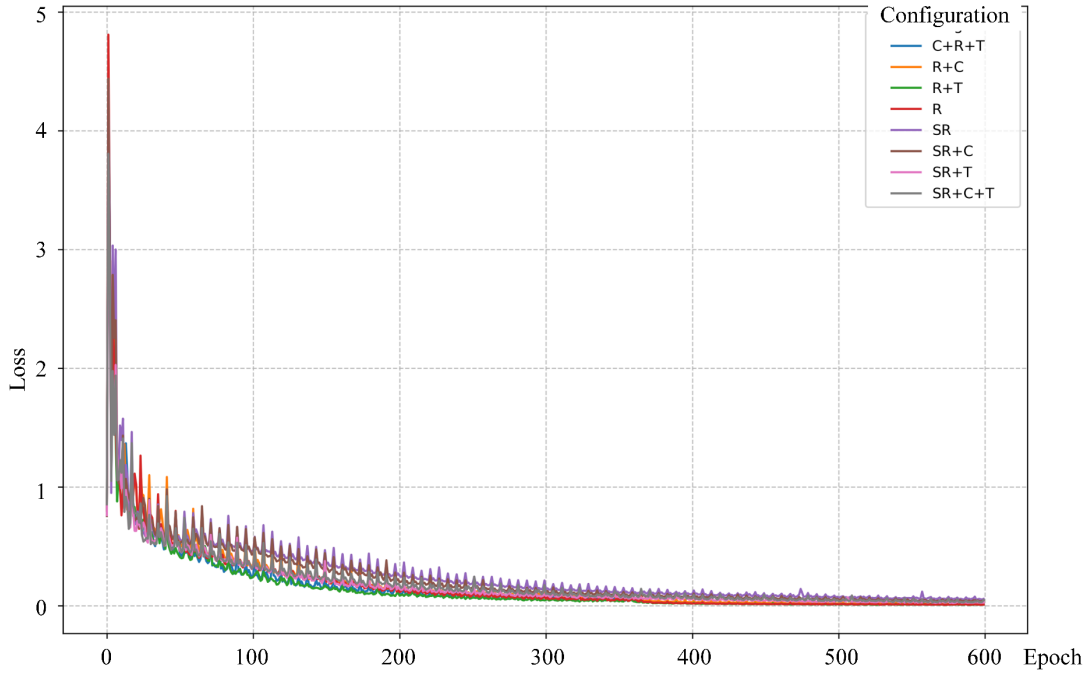


Figure 6-7 Training loss curves

Another finding is related to overfitting. The DOF values in Table 6-4 are averaged differences of hit@10, hit@3, and hit@1 between the training and testing datasets. According to the results, with rich semantics, the models have less overfitting (i.e., all DOF values are less than 0.1). In contrast, all DOF values are larger than 0.1 and can reach 0.23 when simple semantics are adopted. The R+C+T model features the least overfitting, indicating again that it is the best model configuration.

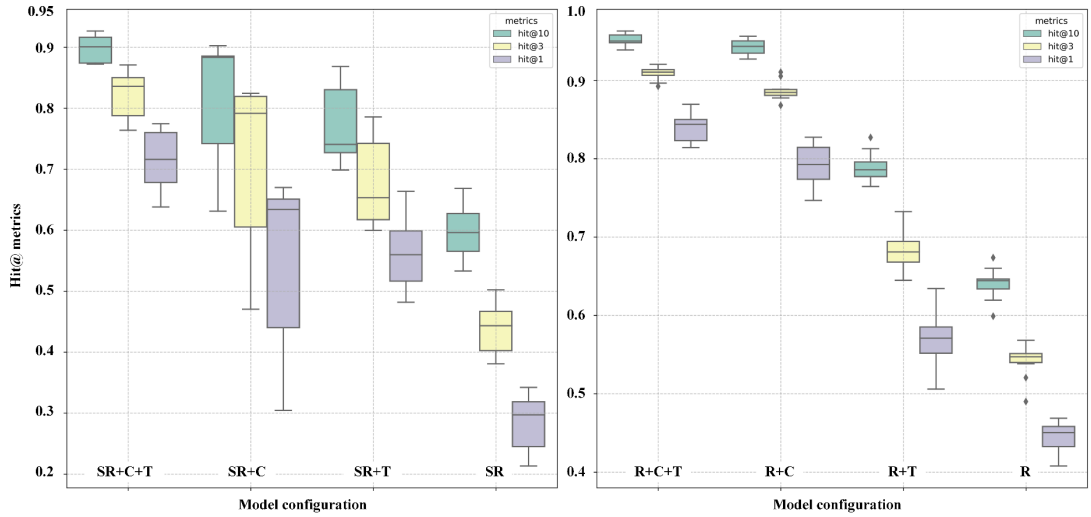


Figure 6-8 Comparison between different model configurations

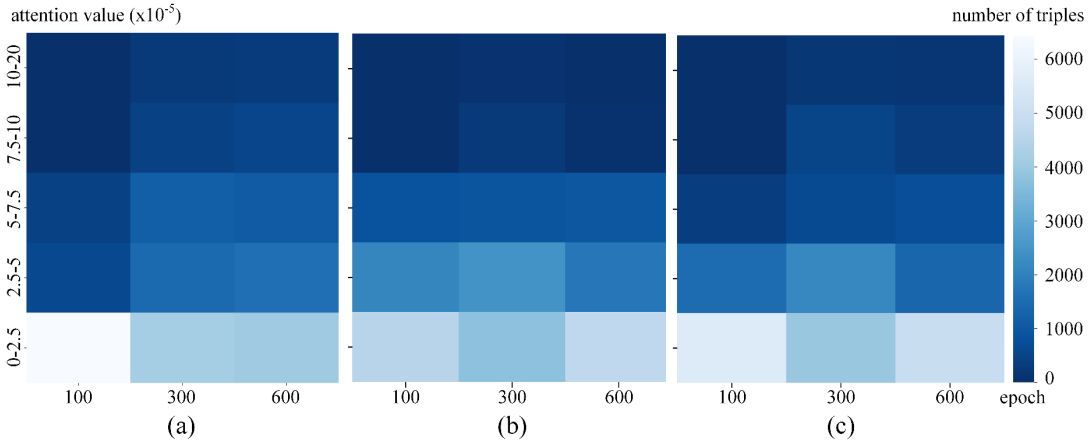


Figure 6-9 Change of attention values using configuration (a) R (b) R+C (c) R+T

6.3.2.2 Ablation study

To demonstrate the effectiveness of adding domain information to the original model proposed by Nathani et al. (2019). The amount of performance increase (hit@1, the strictest metric in this case) owing to each type of improvement strategy (i.e., adding semantics, adding class information, and adding working context information) is presented in a diagram (Figure 6-10). In the diagram, the starting point is the simplest configuration SR. The paths are extended following different improvement strategies until they reach the full configuration R+C+T. The added information and increased performance are shown at the paths.

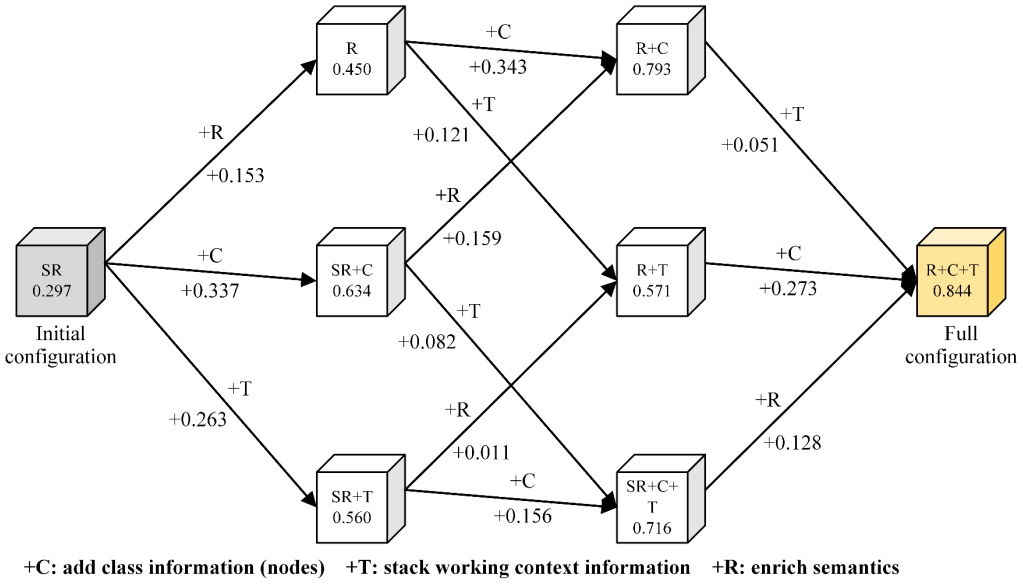


Figure 6-10 Effect of improvement strategies

There are three findings. First, all three strategies are effective, which increase model performance with different degrees (0.011-0.343). Second, adding class information causes the most significant performance improvement (0.277 in average). This is followed by adding working context information (0.129 in average), while enriching semantics has less effect (0.112 on average). Third, the amount of improvement due to inserting class nodes presents less variance when the paths extend (0.156-0.343). On the contrary, the amount of increase owing to adding working contexts (0.051-0.263) and semantics (0.011-0.153) can significantly change when the strategies are adopted in different ways (i.e., different places along the paths).

6.3.3 Controlled experiments (AWP KBs completion)

The controlled experiments were conducted to show the usefulness of the KBC model (i.e., R+C+T configuration) by comparing it with the manual KB checking approach. Constraint triples were extracted from the working plans of the second case project (the cable replacement project). Then, a full AWP was developed using these triples as ground truth, where some links and nodes were intentionally deleted to make it incomplete. Again, the entities and relations which cannot be handled by the current approach were manually inserted. As listed in Table 6-5, five activities were tested, covering four common relation types (*c2c*, *c2t*, *c2a*, *t2t)* and missing information in practical AWP (Hamdi, 2013; Li et al., 2019).

When performing the first four activities, the KBC model enumerated all entities or relations to replace the '?'. Then, the model produced a list of scores of triples formed

by the candidate entity/relation replacing the '?' and two known entities/relations. The candidate entities/relations of the first three triples in that list were added in another list based on which the human engineer (i.e., the researcher) made the final decisions. In other words, as the model cannot reach very high hit@1 (around 84%), one could rely on the more accurate hit@3 to filter irrelevant information and make decisions. When performing activity five, the model first identified entities constraining the task or procedure entity through reasoning. Then, it traversed the entities and tested the validity of triples with the form 'entity is-removed false'. If the score of a triple is below a threshold (e.g., 0.8), the triple was regarded as valid, and the constraint entity was added to a list recording unremoved constraint of the task/procedure. The researcher conducted all five activities using the KBC model, while the colleague took a manual approach relying on his experience. The working plan was available to the colleague to provide additional help.

Precision, Recall, and F1 scores were computed to evaluate the performance of the two approaches. Figures 6-10 - 6-12 show different triples predicted by the model. Table 6-6 lists the performance for completing the activities. The KBC model can gain a higher F1 score while reducing the time to 1/6-1/40 of the manual approach. The time saving is due to the automatic traversing and filtering. The effect of increasing accuracy of completion is smaller than that of saving time, as the validity of most triples can be determined by engineering experience. However, using the manual approach, one must navigate the KBs, find relevant nodes, and evaluate the validity of candidate triples. Thus, more errors can appear when the engineer losses focus. This is proved by the fact that the engineer has high precision but lower recall, i.e., most triples identified by the engineer are correct, but many correct triples are also missed. Nevertheless, higher F1 scores can be gained when applying the model to all activities except finding missing tasks/procedures (the third activity). One reason is that the number of tasks or procedures is much smaller than that of constraints, which makes it hard for the model to learn their patterns. In addition, missing work dependencies are easier to be identified than relations among constraints. This is mainly because such dependencies are often explicitly mentioned in working plans, thus, the engineer could check candidate triples thoroughly with reasonable efforts and time.

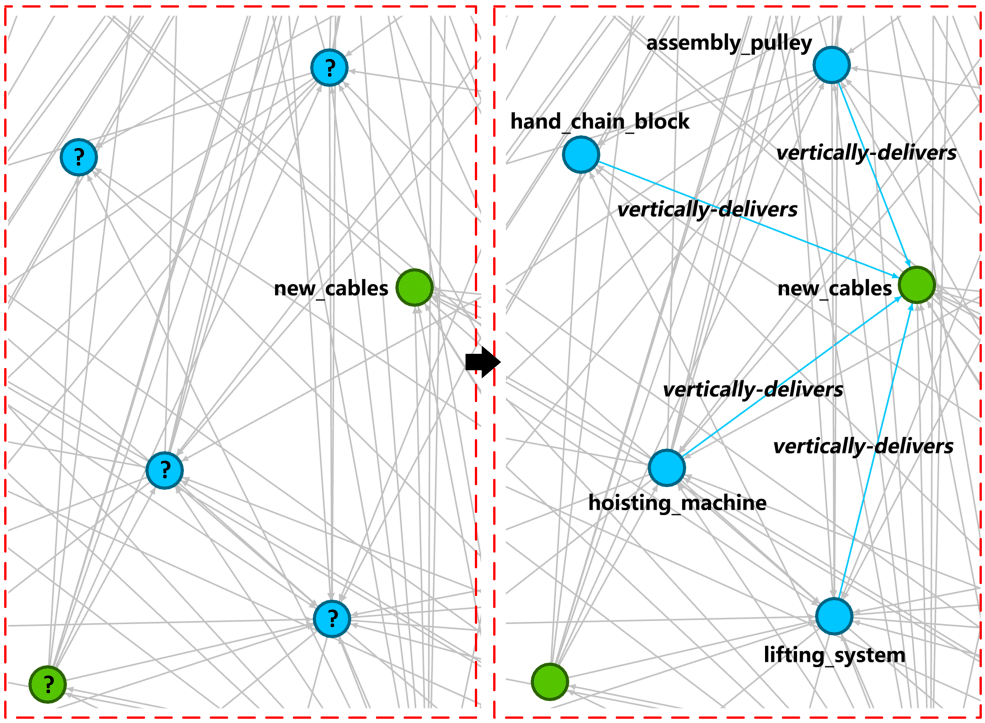


Figure 6-11 Examples of predicted triples (triple form '? relation entity')

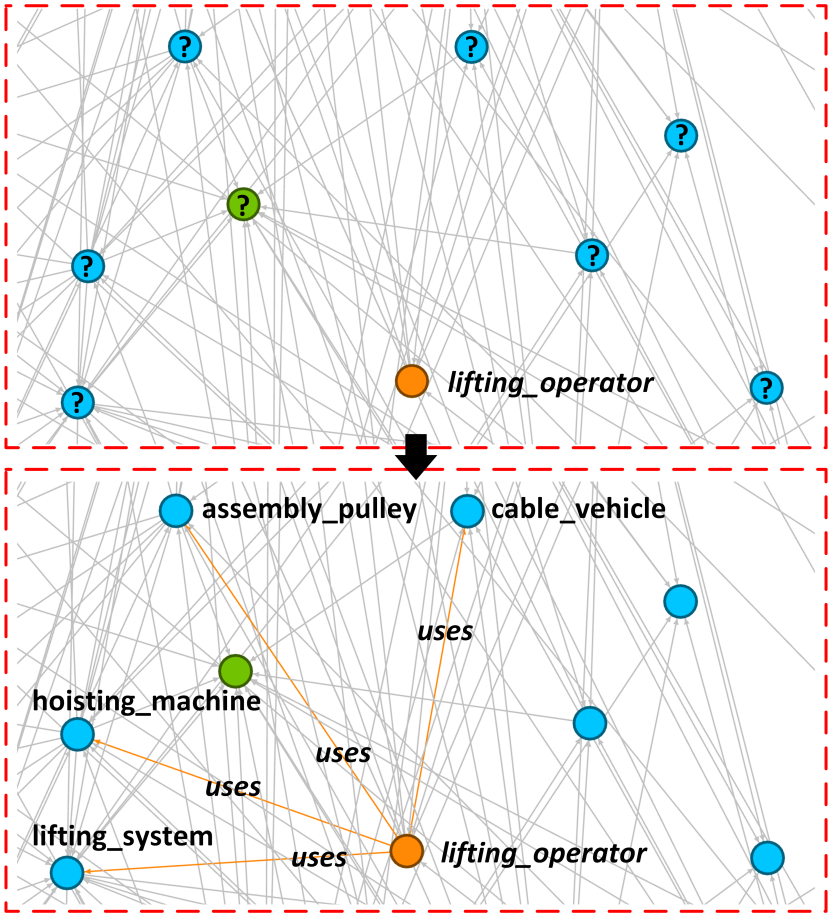


Figure 6-12 Examples of predicted triples (triple form 'entity relation ?)

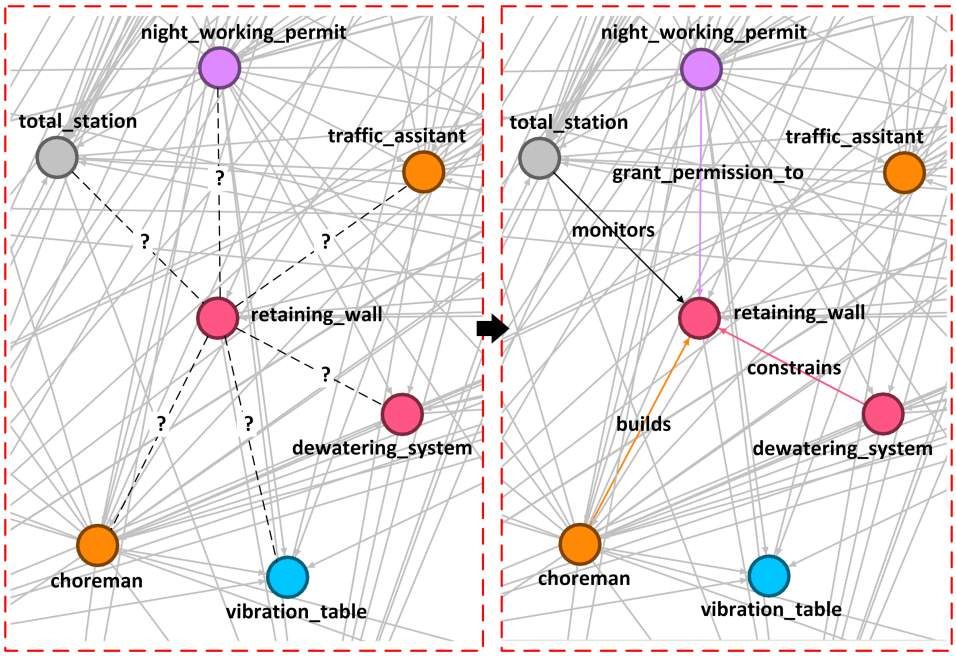


Figure 6-13 Examples of predicted triples (triple form 'entity ? entity')

6.4 Discussion

This KBC model has two main contributions. First, it improves AWP that relies on high-quality knowledge graphs (i.e., KBs). Current AWP KBs have two limitations: 1) They suffer incompleteness because information extraction methods in the industry cannot extract all needed information. 2) They lack rich semantics as they often only consider relations with simple semantics, e.g., existence of relations (i.e., no relation type is identified), synonyms, hypernyms (Chi et al., 2019; Xu & Cai, 2020), as well as basic constraint management relations ('constrains' and 'has-attribute') (Wu et al., 2021b; Zhong et al., 2020b). The two limitations can hinder AWP as follows. Given the large number of entities and complex relations in modern projects, it is difficult to complete KBs using manual checking or reasoning rules (many triples cannot be reasoned by rules) (Qu & Tang, 2019; Yang et al., 2017). Thus, missing information in KBs can affect management functions in AWP (e.g., information searching and graph analysis). On the other hand, current KBC models cannot be directly applied to AWP, because the lack of semantics in KBs can largely hurt model performance. The proposed KBC model can address both limitations. The ontology-based data enriching module includes 42 semantic rules to add constraint triples and enrich semantics of existing triples. The experiment results show that the number of triples and relation types can be increased from 6049 to 17587 and from 39 to 74, respectively. Using the enriched data, the KBC model can be effectively trained to identify missing triples in

KBs. The model achieves good performance (95.0%, 91.0%, 84.4%, and 3.84 for hit@10, hit@3, hit@1, and mean rank, respectively), and the hit@1 of the model can be improved by up to 15.9% due to data enriching.

Second, the main computational novelty of the KBC model is that it improves the original model in Nathani et al. (2019) by utilising domain information to increase model performance. The information of domain classes and working contexts are considered. Classes are identified as additional nodes and fed into the GNN encoder. Tasks/procedures are selected as working contexts of entities. A constraint entity can be linked to several tasks/procedures, and the information of different task/procedure combinations is integrated as the working context embedding of a constraint entity, which is stacked in the input matrix of the decoder. The two strategies have similar effect of clustering entities in two different aspects, i.e., domain classes and project stages. In this way, the model can learn triple patterns among entities, between entities and clusters, and among clusters. Therefore, the model is less likely to be affected by entities with heterogenous names. According to the ablation study, the two strategies can gain additional 27.7% and 12.9% hit@1 compared to using the original model structure, respectively.

It is difficult to find useful information in incomplete KBs. Thus, the main function of the KBC model is not to search for information, instead, it can help engineers identify missing information important for implementing AWP, e.g., statuses of constraints and tasks. As shown in Section 6.3.3, the KBC model can reduce the time to complete a KB to 1/6-1/40 of manual checking while remain higher accuracy. Thus, in practice, the KBC model can supplement existing information searching tools (e.g., SPARQL) to increase the comprehensiveness and accuracy of searching results.

6.5 Chapter summary

The AWP graphs (i.e., KBs) are often incomplete and can hinder the effectiveness of AWP. This Chapter introduces experiment results of the novel KBC model that can automatically identify missing triples and complete AWP KBs. The model has an encoder-decoder structure. The encoder applies the attention mechanism based GNN to learn embeddings of entities and relations. The decoder applies CNN to scan triple embeddings and compute scores for triples as their possibility of being valid. The proposed model features two improvements compared to existing studies. First, a data

enriching module is developed based on ontological reasoning rules to add semantics of triple data and facilitate KBC training. Second, the model utilises two types of domain information: i.e., domain classes and working contexts. Domain classes are inserted in KBs as additional nodes which are taken by the encoder, while working context embeddings are stacked in the decoder. The improvement strategies can significantly increase model performance. The KBC model can effectively identify different types of missing triples in KBs. As such, it can be supplementarily applied with the hybrid IE model. The hybrid IE model automatically develops the KBs while the KBC model continuously adds missing information to increase the KBs' quality. Completed KBs can help engineers identify constraints and tasks/procedures requiring more attention and support informed decision-making regarding constraint removal.

Table 6-4 Experiment results under different model configurations

Config	hit@10	hit@3	hit@1	MR	Overfitting (DOF)
SR	0.596 (0.846)	0.443 (0.709)	0.297 (0.482)	39.01 (12.79)	0.233
R	0.644 (0.728)	0.547 (0.640)	0.450 (0.550)	37.48 (22.78)	0.092
SR+T	0.740 (0.844)	0.653 (0.775)	0.560 (0.686)	21.83 (10.81)	0.117
R+T	0.786 (0.862)	0.681 (0.781)	0.571 (0.683)	14.75 (7.83)	0.096
SR+C	0.883 (0.979)	0.791 (0.946)	0.634 (0.831)	10.53 (3.26)	0.149
R+C	*0.943 (0.980)*	*0.884 (0.966)*	*0.793 (0.916)*	*5.48 (3.14)*	*0.081*
SR+C+T	0.900 (0.989)	0.836 (0.969)	0.716 (0.886)	7.55 (1.66)	0.130
R+C+T	**0.950 (0.989)**	**0.910 (0.979)**	**0.844 (0.939)**	**3.84 (1.84)**	**0.068**

P.S. performance metrics in above table are all median values

Table 6-5 KBC activities

No.	Missing information	Triple form	Examples
1	Participants who remove a constraint entity	(? removes constraint)	(? removes new_cable)
2	Entities that a constraint can constrains	(constraint constrains ?)	(labour uses ?)
3	Task entities of a particular task	(? work-dependencies task)	(intalling_cables is-succeeded-by ?)
4	Relations among entities	(constraint/task ? constraint/task)	(tensing_machine ? new_cable)
5	Unremoved constraints of a particular task/procedure	(constraint is-removed ?) where (constraint constrains task/procedure)	(crane is-removed?) where (crane is-required-by intalling_cables)

Table 6-6 Comparison between manual and KBC approaches

	Activity 1		Activity 2		Activity 3		Activity 4		Activity 5	
	M	K	M	K	M	K	M	K	M	K
Time (second)	71	6.3	121	9.4	141	10.5	33	5.2	240	6.8
P	0.917	0.897	0.920	0.871	0.952	0.885	0.962	0.862	1.000	0.920
R	0.815	0.963	0.767	0.900	0.800	0.920	0.893	0.893	0.800	0.920
F1	0.863	0.929	0.836	0.885	0.870	0.902	0.926	0.877	0.889	0.920

P.S. M and K indicate the manual and automatic KBC approach, respectively.

Chapter 7: Conclusions, contributions, implications, and future work

7.1 Conclusions

In this section, important findings of previous chapters are summarised to draw unambiguous conclusions of the research. This research develops an automatic and effective approach to improve AWP in bridge rehabilitation projects based on novel DL models for information extraction and completion and ontologies for information integration. This research is mainly deductive and quantitative based on objectivism epistemology and realism ontology, e.g., DL model development and experiments. On the other hand, subjective domain knowledge is employed in the proposed KBs and DL models, e.g., domain classes and relations collected through literature review and the focus group. Hence, it can be argued that this research is a mixed research and belongs to the post-positivism paradigm.

The proposed information management approach includes three key components: 1) a hybrid IE model to extract constraint entities and setup relations among entities; 2) ontological KBs (i.e., BRMO) to integrate constraint information and support project management functions; and 3) a KBC model to identify missing triples in KBs. The hybrid IE model extracts constraint information from documents; then, the BRMO integrates such information in ontological KBs; finally, the KBC model is used to enrich the KBs and improve the quality of KBs. Both DL model experiments and controlled experiments have been carried out to validate the capacity and usefulness of each component in the proposed approach. The results show that the approach can reach high performance in terms of entity/relation extraction and KB completion, it can also integrate dynamic project information, i.e., constraints, tasks, procedures, attributes of constraints, and project participants. The approach can largely automate AWP constraint modelling while enabling effective information integration. Hence, it can contribute to project success by saving much time for other AWP tasks, e.g., constraint monitoring and removal.

7.1.1 Research findings for Objective 1

Objective 1: To investigate topics, trends, and limitations of information management in bridge maintenance projects, implementation of AWP in the AEC industry, and information extraction and integration approaches.

Summary of findings: The critical review is carried out based on 485 articles of DDBM studies, 29 articles and 106 industry documents (e.g., standards and reports) of AWP, and 117 articles of information extraction and integration approaches. All documents come from the Web of Science and databases of AWP implementors. The review has clearly shown the research gaps below:

- Bridge rehabilitation projects are complex, and advanced package-based constraint management approaches (e.g., AWP) can contribute to the success of such projects. However, current efforts of bridge rehabilitation focus on engineering techniques and do not adequately consider the management aspect. Thus, modern constraint management approaches have not been implemented in bridge rehabilitation projects.
- Successful project management needs to integrate project information in KBs. The challenge is that the information is often scattered in isolated sources, buried in unstructured documents, and changes as a project proceeds. Existing information management approaches in the sector cannot effectively extract and integrate such unstructured and dynamic project information.
- As a result of the above two limitations, constraint modelling, the prerequisite of AWP, still depends on manually identifying constraint entities and relations, which is believed to be very inefficient at the time of this study. Besides, the generated KBs (i.e., AWP graphs) are often incomplete, while information is not integrated into a central environment to assist information access.

7.1.2 Research findings for Objective 2

Objective 2: To develop a novel deep-learning-based information extraction model to automate AWP constraint modelling by extracting constraint entities and relations from text documents.

Summary of findings: the hybrid IE model combines a Bi-LSTM-CRF model to extract constraint entities and a CNN-based KRL model to extract relations through identifying valid triples from candidate triples formed by traversing and connecting the extracted entities. Based on experiment results, the following findings are drawn.

- The Bi-LSTM-CRF model can accurately extract CONS, AT, and TP entities with the F1 score being 0.936 in the testing dataset.

- The CNN-based KRL model can effectively extract five types of relations, i.e., *c2c*, *c2t*, *c2a*, *t2t*, and *c2p* relations, with the F1 scores being 0.859, 0.885, 0.908, 0.912 and 0.890 in the testing dataset, respectively.
- In the KRL model, adding class information can significantly increase the model performance of relation extraction and accelerate model convergence. Particularly, when the information is horizontally stacked at the input matrix of the model structure, the most significant increase of F1 score (1.9%, 12.0%, and 6.0%) can be gained for *c2c*, *c2a*, and *c2t* relation extraction, respectively It should be noted that the other two types of relations (i.e., *t2t* and *c2p*) are set up by rules, thus, the performance cannot be improved using the strategies.
- Most data for training the hybrid IE model are Chinese documents, but the model can be generalised to other languages. The Bi-LSTM-CRF model can reach 0.912 F1 when extracting entities in the additionally collected English texts, and the KRL model is to-some-extent independent of languages due to the translation mechanism (Section 4.4.1).
- The hybrid IE model can partially automate AWP constraint modelling, and the time to develop the AWP KBs can be reduced to 1/29 of that using the manual approach.

7.1.3 Research findings for Objective 3

Objective 3: To develop ontological knowledge bases to integrate the constraint information in bridge rehabilitation projects.

Summary of findings: The development of the ontological KBs (i.e., the BRMO) follows a standard guideline and is based on a comprehensive collection of domain knowledge.

- The BRMO has three class taxonomies for tasks/procedures, constraints, and participants, respectively as well as two relation hierarchies for object and datatype relations, respectively.
- The BRMO overcomes the syntax limitations in conventional ontologies by combining the SWRL, SQWRL, and OWL API. Thus, the BRMO supports complex computation and updating therefore enabling integration, inferring, updating, and searching for both static and dynamic constraint information.

- In the information encoding experiments, the BRMO can integrate all triples automatically extracted by the hybrid IE model developed in Objective 2.
- In the information searching experiments, the BRMO can search for project information of AWP efficiently. The searching time can be reduced up to 1/50 of manual searching, where the time for writing queries has been considered.
- The BRMO can realise essential management functions, e.g., computing the progress and delay of tasks/procedures, evaluating constraint statuses (e.g., evaluating removal progress), identifying critical constraints, and evaluating the performance of participants.

7.1.4 Research findings for Objective 4

Objective 4: To develop a novel knowledge base completion model to automatically identity missing triples in AWP KBs.

Summary of findings: The KBC model has three essential parts: an ontology-based data enriching module, a GNN encoder to learn embeddings of entities/relations, and a CNN-based decoder to predict missing triples.

- This research proposes three strategies to improve model performance. The data enriching module infers new triples and enriches semantics of existing triples to facilitate training. Besides, two types of domain information: i.e., domain classes and working contexts, are utilised. Domain classes are inserted in KBs as additional nodes to be processed by the encoder, whereas working context embeddings are stacked in the decoder structure.
- The proposed KBC model can effectively identify different types of missing information in KBs. The maximum performance is 95.0%, 91.0%, 84.4%, and 3.84 for hit@10, hit@3, hit@1, and mean rank, respectively.
- In cross-comparison experiments, eight model configurations are tested. The full configuration applying all three strategies significantly outperforms other configurations. All three strategies increase model performance to different degrees. To be specific, adding domain class information, adding working context information, and enriching semantics can increase hit@1 (the strictest metric) by 0.277, 0.129, and 0.112 on average, respectively.

- In the controlled experiments, the KBC model can reduce the time to check and complete KBs to 1/6-1/40 of manual checking while obtaining higher F1 in terms of identifying missing information.

7.2 Contribution, implication, and future work

7.2.1 Summary of theoretical contributions

The main theoretical contributions of this research lie in three aspects, i.e., expansion of existing domain ontologies, a novel approach for integrating dynamic information in ontologies, and novel computational models for automatic information extraction and KB completion in the AEC industry.

(1) Expansion of domain ontologies

Existing ontologies for bridge maintenance are often developed for the inspection, evaluation, and decision-making stages. However, bridge rehabilitation has specific domain knowledge, e.g., specialised constraints and tasks as well as relations among these entities. Thus, previous ontologies cannot effectively integrate information of bridge rehabilitation due to lacking such domain knowledge. This research proposes the BRMO created based on the comprehensive collection of bridge rehabilitation knowledge. As such, the BRMO expands the coverage of ontologies to the bridge rehabilitation stage. The BRMO can effectively integrate information of rehabilitation tasks/procedures, project participants, and three types of constraints (i.e., engineering constraints, supply-chain constraints, and site constraints). Moreover, although the BRMO is designed specifically for bridge rehabilitation, it can be usefully integrated with other bridge ontologies (e.g., ontologies that model bridge components) without significant modifications to support informed maintenance decisions.

(2) A novel approach for integrating dynamic information in ontologies

Most previous ontologies in the AEC sector focus on integrating static information (e.g., geometries) and facts (e.g., reasons of defects and accidents). Such information often does not change regularly. However, these ontologies do not adequately take dynamic project information, e.g., the progress of tasks and constraint removal into consideration. The main reason is that conventional ontologies do not support critical computations required for updating such dynamic information. The BRMO addresses this issue by combining the SWRL, SQWRL, and OWL API to overcome the syntax

limitations in conventional ontologies and realise an effective information updating process. Specifically, the OWL API exports information out of the BRMO and then performs all required computation programmatically (e.g., computing the delay of tasks/procedures and ratio of unremoved constraints). The results are imported back into the BRMO through the API. On the other hand, SWRL and SQWRL are good at inferring new knowledge in ontologies. Based on the updated information, the rules are used to infer additional information (i.e., triples) to reflect the performance of a project in three aspects: work progress, constraint removal progress, and participant performance. It can be argued that the proposed method extends current information management approaches in ontologies so that the BRMO can be continuously updated to integrate both static and dynamic project information.

(3) Novel computational models for automatic information extraction and knowledge base completion

To automate constraint modelling and provide comprehensive information for AWP, this research proposes two critical DL models, the hybrid IE model (Section 3.4 and Chapter 4) and KBC model (Section 3.6 and Chapter 6). The models make use of cutting-edge NLP studies. However, NLP models in these studies focus on general knowledge. They do not consider specific information in the AEC domain therefore cannot reach good performance if being directly used for AWP. Thus, this research contributes by utilising domain-specific information to modify structures of state-of-the-art DL models and improve their performance. Two types of domain information are considered: domain classes and working contexts of project entities. Such domain information can cluster constraint entities so that the DL models are less likely to be distracted by heterogeneous entity names in training and testing. Based on detailed model experiments, the research proposes two ways to utilise domain information that can realise maximum performance improvement: 1) For the KRL model, embeddings of domain classes of a triple's head/tail entities are horizontally stacked at both sides of the input matrix. 2) For the GNN encoder, the domain classes are inserted in the KBs as additional nodes which are processed by the encoder. 3) For the CNN decoder, embeddings of working contexts of the head/tail entities in a triple are horizontally stacked at both sides of the input matrix. On average, the proposed KRL model can increase the F1 score of triple extraction by 6.63%, and the proposed KBC model can increase the hit@1 values by 11.2%-27.9%. Finally, the research is an early attempt to

extract both entities and semantic rich relations in the AEC sector, thus, the model training and validating protocols, optimal hyper-parameters, and model performance metrics are all valuable baselines for future IE or NLP studies in the sector.

7.2.2 Summary of implications

Package-based constraint management approaches, e.g., AWP, rely on three critical steps, constraint modelling, constraint monitoring/analysis, and constraint removal. However, as the first step, constraint modelling still relies on manually reviewing project documents in practice, owing to the lack of efficient IE approaches to extract both constraint entities and relations. Besides, project teams lack tools (e.g., graph-based KBs) to integrate extracted constraint information for reuse. Finally, even KBs are built, there are no practical methods to automatically check, update, and enrich these KBs. Practical AWP is an iterative and intensive process, thus, these challenges can damage AWP functions and hinder the remaining constraint management steps. Hence, the research proposes the hybrid IE model, ontological KBs, and KBC model to solve the issues for implementing AWP in bridge rehabilitation. Accordingly, the proposed information management approach has three practical implications.

(1) Automatic AWP modelling tools

The hybrid IE model is a useful tool to extract constraint information and automate constraint modelling. The model can extract three types of entities and five types of relations, which can cover typical routines in construction projects. In the controlled experiments, the model can reduce constraint modelling time to 1/29 of the manual approach. Thus, much time can be saved for constraint monitoring and removal. The model can handle both static data (e.g., imperative requirements in standards) and dynamic data (e.g., task progress and constraint statuses). In real projects, the number of constraint entities is much larger than that in the experiments. It is exhausting for engineers to manually extract all constraint information. The approach developed in this research can to-some-extent automate the AWP modelling process, which helps (not replaces) engineers to capture interconnections among constraints and improve management decision-making. Finally, some efforts propose advanced management frameworks for AWP, concentrating on efficient constraint monitoring and removal. However, information in such frameworks is still manually inserted (Wang et al., 2016). Therefore, as an effective and automatic IE tool, the hybrid IE model can supplement such efforts and better reap the benefits of AWP.

(2) Ontology-based project information integration platform

The BRMO can integrate, infer, and search for both static and dynamic information in ongoing projects in a much shorter time (1/50) compared to the manual approach, especially when the information is scattered in multiple sources. Thus, the BRMO is an effective and software neutral platform that allows different participants to access project information. Moreover, the BRMO supports several important management functions, e.g., evaluating work progress, constraint removal progress, and project participant performance, warning potential delay, and identifying critical constraints. Although the functions can be realised in traditional tools (e.g., Microsoft Project), using the BRMO, one can navigate the KBs to explore implicit information (e.g., finding root causes of delay). This is difficult to achieve in traditional tools.

(3) Automatic KBs completion tools

It can be difficult to find useful information (manually or automatically) to support AWP management functions if KBs are incomplete. The proposed KBC model can help engineers quickly identify critical missing information in ontological KBs, e.g., statuses of constraints/tasks. As shown in Section 6.3.3, the KBC model can reduce the time for checking and completing a KB to 1/6-1/40 of the manual approach while maintaining high completion accuracy. The main function of KBC in practice is not to extract or search for information, instead, it is more beneficial to apply the model as a supplement tool. For instance, when initial AWP KBs are created using either manual or automatic IE methods, the KBC model can improve the quality of KBs by adding missing information. It can work with information searching tools (e.g., the SPARQL for ontology querying) to increase the comprehensiveness and accuracy of searching.

7.2.3 Towards construction 4.0

Since the 1760s, the world has experienced three industrial revolutions which have made a significant advance in many sectors and greatly improved people's life. The first revolution focuses on mechanization, i.e., using machines to replace human labour. The second revolution focuses on the intensive use of electrical energy. The third revolution focuses on widespread digitalisation (modern computers and the Internet). During the last decade, the world is undergoing the fourth industrial revolution (i.e., industry 4.0) which focuses on establishing the connections among information, objects, and people using computers and cyber-physical systems. The main aim is to develop a decentralised connection between the real world and cyberspace so that

different scenarios are simulated in cyberspace (i.e., the virtual world) to derive optimised decisions before carrying out tasks in the real world.

Despite that the construction industry is notorious for inadequate applications of new ICTs, it also experiences a process similar to industry revolutions, i.e., construction 1.0 (from labour-intensive construction to the adoption of machines, e.g., cranes), construction 2.0 (from non-standard construction to standard construction, e.g., off-site construction), construction 3.0 (from document-based construction to computer-based construction, e.g., computer-aided design). In recent years, various intelligent techniques of industry 4.0 are increasingly adopted in the sector, e.g., BIM, artificial intelligence (AI), and IoT. They have brought many benefits to construction projects, including but not limited to improved productivity, safety, and quality. Therefore, domain experts are unanimous on the fact that the construction industry is shifting towards construction 4.0 or intelligent construction (Schönbeck et al., 2020).

The concept of construction 4.0 was first proposed in 2016, and there is currently no common agreement of its definition (Lasi et al., 2014). There are two pillars for the transformation towards construction 4.0, namely, industrialisation and digitalisation. Specifically, industrialisation refers to the new materials, industrialised construction methods (e.g., modular construction), and construction robotics. On the other hand, digitalisation includes the following parts: 1) big data (i.e., approaches for automatic data or information collection, storage, analysis, and exchanging), 2) AI techniques for information analysis and decision-making, 3) computer and BIM-based design, construction, and maintenance, and 4) virtual reality (VR) and augmented reality (AR). Accordingly, there are three essential features of construction 4.0, i.e., digitalised, automated, and connected (Forcael et al., 2020; Schönbeck et al., 2020).

As such, this research can facilitate the transformation towards construction 4.0. In particular, the research contributes to the development of the digitalisation pillar of construction 4.0 in the following aspects, which makes the industry more digitalised, automated, and connected. Figure 7-1 illustrates the architecture of construction 4.0 and highlights the improved areas through this research.

(1) Improving unstructured information extraction (more digitalised)

This improvement belongs to the data/information collection and applications of AI techniques of the digitalisation pillar. Construction projects involve different data

types, e.g., images, sensor readings, and plain texts. Construction 4.0 is information-driven, and useful information should be efficiently extracted from raw data. Current studies focus on processing data collected by sensors and imaging techniques (e.g., UAVs and LiDAR). For instance, many methods are developed to extract defect information from photos (Xu et al., 2020). However, when it comes to text data, useful information is buried in unstructured texts and cannot be efficiently digitalised. This is especially true when it comes to extracting semantic-rich relations and is against the characteristics of construction 4.0 (Wu et al., 2020b). The information management approach proposed in this research can extract entities and semantic-rich relations from texts and then integrate them in KBs. Thus, it can to-some-extent address the above problem and make construction more digitalised. Besides, in the last decade, owing to the fast development of CNN-based models, much more efforts of construction 4.0 are dedicated to applying computer vision techniques, e.g., detecting cracks (Yeum & Dyke, 2015) and unsafe behaviours of workers (Fang et al., 2020). However, the applications of NLP techniques largely lag. Along with the advances of NLP models (e.g., Google's BERT and GNNs), an important future direction of AI is human language understanding based on combinations of knowledge graphs with intelligent NLP algorithms (Vaswani et al., 2017). From this perspective, the research explores applications of state-of-the-art NLP techniques in the AEC sector (e.g., information extraction for AWP using the Bi-LSTM-CRF and KRL model as well as automatic knowledge completion using the KBC model). This can help the industry catch up the cutting-edge AI research and bridge the gap of unstructured information extraction and integration, which can make steps moving towards construction 4.0 more balanced in terms of applying AI techniques.

(2) Minimising human intervention for information modelling (more automated)

This improvement belongs to the computer and BIM-based construction management of the digitalisation pillar and can make the construction sector more automated. One key challenge of construction 4.0 is to set up and maintain the link between physical and cyber projects. In this case, a typical implementation is to create n-D BIM models to model different aspects of a project (e.g., schedule (4D) and costs (5D)) before the physical project commences. The link (i.e., the BIM environment) is maintained by IoT and computer vision systems which collect data of structures, labour, materials, and equipment in a real-time manner and upload the data into BIM (Dave et al., 2018).

However, such systems focus on structured data, e.g., sensor readings and geometries of defects measured in images, but they cannot automatically capture some important information for modern project management approaches (e.g., AWP), e.g., complex semantics of and interconnections among project entities (Wu et al., 2021b). As such, AWP heavily relies on inefficient manual approaches (e.g., manually extracting and updating interconnections among entities). In contrast, the proposed approach can handle unstructured data thus largely automating AWP modelling and KBs checking. Such automation frees engineers from strenuous and repeating manual work so that they can spend more time on essential management tasks (e.g., constraint monitoring and removal). This can also supplement BIM-based management. For instance, a data link can be set up between AWP KBs generated by the proposed approach and BIM systems, so that data from both sides can be automatically integrated to enable more sophisticated functions. As an example, some studies of historical building restoration propose to 1) store non-geometric information (e.g., historical events and complex material properties) in ontologies, 2) export geometric information from BIM to the ontologies for reasoning (e.g., detecting inconsistency between different inspection activities), 3) visualise the results in BIM for communication (Niknam & Karshenas, 2017; Simeone et al., 2019; Werbrouck et al., 2020). The approach can be adopted in bridge maintenance projects (e.g., storing geometries and defects of components in BrIM while storing condition evaluation rules in ontologies to assist structure health assessment) to take use of strengths of different information management tools.

(3) Improving unstructured information integration (more connected)

This improvement belongs to the data exchange part of the digitalisation pillar, which can make construction projects more connected. According to Schönbeck et al. (2020), the studies of construction 4.0 pay more attention to industrialisation and automated construction approaches (e.g., off-site construction and robotics) than information communication and integration. This complies with the findings that existing BIM studies focus on modelling capabilities and lack consideration of interoperability (Costin et al., 2018). Exchanging information in construction projects largely relies on relational databases and focuses on structured information. Thus, even unstructured information (e.g., knowledge triples) can be extracted, the information cannot be effectively stored and exchanged, which negatively affects the connectivity among stakeholders. On the other hand, the BRMO can to-some-extent tackle the problem.

The BRMO can integrate, store, and search for both static and dynamic information of constraints, tasks/procedures, and project participants, it can also compute and continuously infer new information by supplementary usage of the reasoning rules (SWRL and SQWRL) and OWL API. Furthermore, the BRMO can also be combined with relational databases to exchange both unstructured data (e.g., knowledge triples) and structured data (e.g., sensor readings).

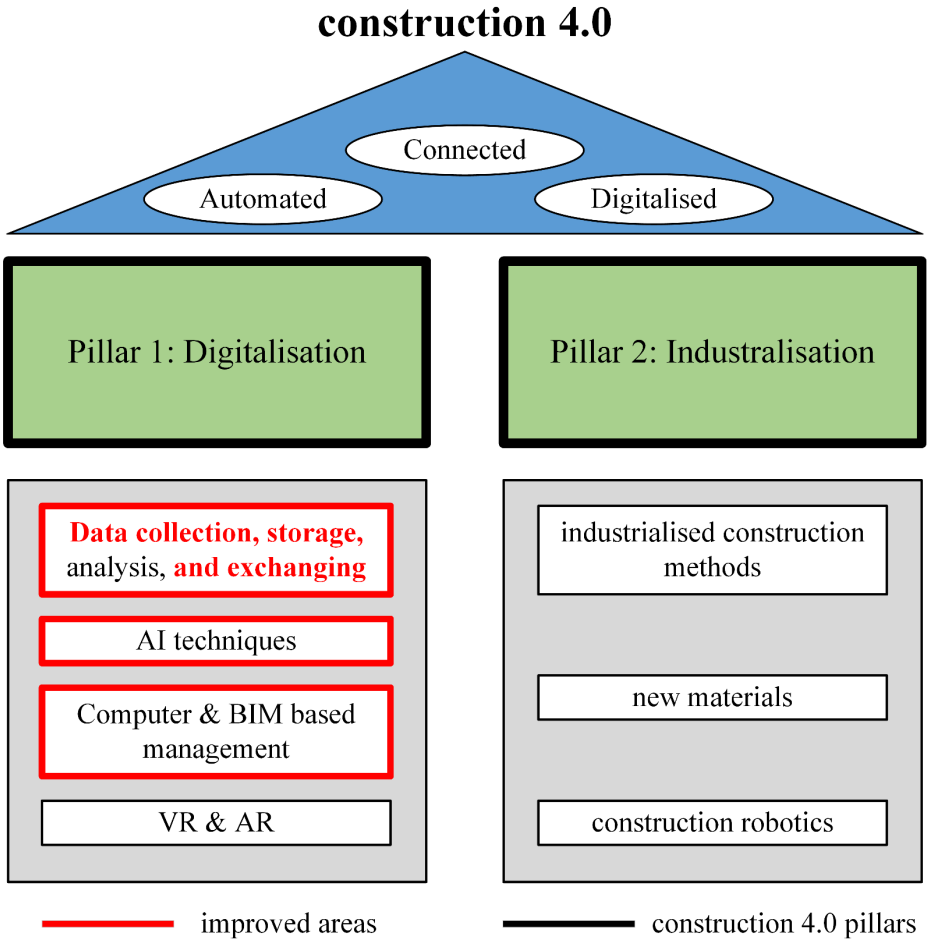


Figure 7-1 Pillars of construction 4.0 and improved areas of this research

In summary, the proposed information management approach focuses on improving three parts of the digitalisation pillar of construction 4.0, i.e., unstructured information extraction, automated project modelling, and unstructured information integration. Therefore, the research can make the AEC domain more digitalised, automated, and connected, which facilitates the transformation towards construction 4.0.

7.2.4 Limitations and future work

In this section, limitations of the proposed information extraction and integration approach for AWP modelling are identified. Accordingly, potential future research directions are proposed.

(1) Limitations of the hybrid IE model (Objective 2)

The hybrid IE model includes two parts, each of which has several limitations. The Bi-LSTM-CRF model still produces some errors, which can hurt downstream triple extraction. The model also suffers from slight overfitting. In the future, to improve model capacity and alleviate overfitting, more training data will be collected, the balance between data samples (i.e., the number of different tags for NER) will be improved using advanced sampling techniques, and novel methods to incorporate domain knowledge in the model structure will be proposed (in this research such knowledge is only utilised for KRL and KBC models) (Srivastava et al., 2014). In addition, in the automatically developed KBs, some extracted entities are duplicated, which can cause ambiguation during information sharing. Hence, ontology merging methods, e.g., description logical reasoning (Kumar & Harding, 2013), will be used to further improve the practicality of the hybrid model.

On the other hand, as mentioned, there are seven common relation types in AWP, but only five of them can be automatically extracted, i.e., the *c2c*, *c2a*, *c2t* relations are extracted by the KRL model, while the *t2t* and *ct2p* relations are extracted by rules. However, the other two relation types, i.e., *t2t* and *c2p*, need to be manually inserted. To fully automate the relation extraction process and maximise the model flexibility (i.e., minimising the use of rules), the model that can support triple extraction from multi-sentences will be tested. This can help extract *t2t* relations that are separated in texts. Additional training samples will be collected for the model to enable extracting *c2p* relations. Besides, although the original BERT model is not designed for relation extraction, it, in essence, provides an effective method to extract and integrate text features in a parallel manner (i.e., the attention mechanism) (Vaswani et al., 2017). The features, once extracted, can be used in almost any ML task (Murphy, 2012). As such, it is also worth trying to modify the BERT structure and make it applicable for extracting AWP relations. The final two relation types (i.e., *p2p* and *ct2pp*) are subject to project features (e.g., scale and type). Thus, future research will develop templates for typical project/task types and scales to automatically extract the relations.

(2) Limitations of the BRMO (Objective 3)

The BRMO has two main limitations. First, the BRMO only supports basic functions for computation and reasoning, which still leaves much room for improvement. For instance, the strength of different relation types can be added to better represent the degree of connection among entities. A common practice is to model such connection strength using fuzzy sets (Moufti et al., 2014). Various network analysis techniques, such as dynamic network analysis and social network analysis, can also be used to discover more project knowledge. For instance, these methods can identify critical constraints using more sophisticated metrics, such as betweenness, centrality, and PageRank values (Farshchi & Brown, 2011). Second, searching for information in the BRMO requires certain skills (e.g., writing SPARQL queries). Therefore, studies will be carried out to automatically generate queries from natural languages based on NLP methods (Tahery & Farzi, 2020), which can further reduce information searching time and make BRMO more practical in real projects.

(3) Limitations of the KBC model (Objective 4)

One limitation of the proposed KBC model is that not all data in KBs are used. Most triples related to constraints' attributes are not considered, because: 1) such data have high variance and are very sparse, and 2) GNN models are good at interpreting the connections among nodes rather than predicting specific values (e.g., attribute values of constraints). One solution that will be tested in the future is to train additional ML models to predict missing attributes, considering various factors. For instance, an ML model can predict the delay of a task based on the type, quantities, and its current constraint removal progress (Hashemi et al., 2020). The current performance (0.844 hit@1) is not very high, and human intervention is needed in the KBC tasks (e.g., selecting one entity from three candidates). As such, more data will be collected for training to increase the model performance.

Moreover, as mentioned in Section 2.4, some unsupervised methods can mine rules automatically (e.g., association rule mining), whereas a few DL models are proposed to even automatically create reasoning rules. The main idea is to create a few initial rules to infer implicit knowledge (e.g., triples) in KBs while combining DL models and Markov logic network to predict missing information encountered when reasoning (e.g., missing entities in the rule's body). The methods/models can improve both the information searching and reasoning capacity of the BRMO and KBC performance.

Many of them are still in an infancy stage and the studies applying in the AEC sector are limited. Therefore, it is still worth testing their performance for AWP in bridge rehabilitation projects.

Reference

AASHTO. (2010). Bridge element inspection guide manual American Association of State Highway and Transportation Officials, Washington, U.S., Retrieved from https://live.iplanevents.com/files/AASHTO2010/bridge_element_guide%20manual%20_05092010.pdf

R. A. Al Haj, & S. M. El-Sayegh. (2015). Time–cost optimization model considering float-consumption impact. *Journal of Construction Engineering and Management*, **141**(5), 04015001.

A. A. Aliyu, I. M. Singhry, H. Adamu, & M. Abubakar. (2015). Ontology, epistemology and axiology in quantitative and qualitative research: Elucidation of the research philophical misconception. *the Academic Conference: Mediterranean Publications & Research International on New Direction and Uncommon,* **2,**(1) 1-26, Ogun State, Nigeria.

M. B. Anoop, B. K. Raghuprasad, & K. B. Rao. (2012). A refined methodology for durability-based service life estimation of reinforced concrete structural elements considering fuzzy and random uncertainties. *Computer-Aided Civil and Infrastructure Engineering*, **27**(3), 170-186.

G. Antoniou, & F. Van Harmelen. (2012). A semantic web primer. Cambridge, MA, U.S., MIT press. 0262012103.

ASCE. (2017). The report card for America's infrastructure, American Society of Civil Engineers, Reston, Virginia, U.S. Retrieved from: https://www.infrastructurereportcard.org/wp-content/uploads/2016/10/2017-Infrastructure-Report-Card.pdf.

H. Baker, M. R. Hallowel, & A. J. Tixier. (2019). Automatically learning construction injury precursors from text. *Automation in Construction*, **118**, 103145.

H. G. Ballard. (2000). The last planner system of production control. (PhD). University of Birmingham, Birmingham, UK. Retrieved from https://etheses.bham.ac.uk/id/eprint/4789/1/Ballard00PhD.pdf

M. Bastien-Masse, & E. Bruhwiler. (2014). Ultra high performance fiber reinforced concrete for strengthening and protecting bridge deck slabs. *7th International Conference on Bridge Maintenance, Safety and Management (IABMAS),* 2176-2182, London, UK.

Z. Bi, T. Zhang, P. Zhou, & Y. Li. (2020). Knowledge transfer for out-of-knowledge-base entities: improving graph-neural-network-based embedding using convolutional layers. *IEEE Access*, **8**, 159039-159049.

P. Bocchini, & D. M. Frangopol. (2013). Connectivity-based optimal scheduling for maintenance of bridge networks. *Journal of Engineering Mechanics*, **139**(6), 760-769.

A. Bolar, S. Tesfamariam, & R. Sadiq. (2014). Management of civil infrastructure systems: QFD-based approach. *Journal of Infrastructure Systems*, **20**(1), 04013009.

N. Bolucu, D. Akgol, & S. Tuc. (2019). Bidirectional LSTM-CNNs with extended features for named entity recognition. *Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science,* 1-4, Istanbul, Turkey.

A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, & O. Yakhnenko. (2013). Translating embeddings for modeling multi-relational data. *26th International Conference on Neural Information Processing Systems,* **2,** 2787-2795, Nevada, US.

A. Bradley, H. Li, R. Lark, & S. Dunn. (2016). BIM for infrastructure: An overall review and constructor perspective. *Automation in Construction*, **71**, 139-152.

B. Cao, J. Zhao, Y. Gu, S. Fan, & P. Yang. (2019). Security-aware industrial wireless sensor network deployment optimization. *IEEE transactions on industrial informatics*, **16**(8), 5309-5316.

N. W. Chi, Y. H. Jin, & S. H. Hsieh. (2019). Developing base domain ontology from a reference collection to aid information retrieval. *Automation in Construction*, **100**, 180-189.

H. J. Choo, I. D. Tommelein, G. Ballard, & T. R. Zabelle. (1998). Constraint-based database for work package scheduling. *Computing in Civil Engineering*, **125**(3), 169-180.

D. K. Chua, L. Shen, & S. Bok. (2003). Constraint-based planning with integrated production scheduler over internet. *Journal of Construction Engineering and Management*, **129**(3), 293-301.

CII. (2013a). Advanced Work Packaging: design through workface execution, Construction Industry Institute, Alberta, Canada. Retrieved from: https://www.construction-institute.org/resources/knowledgebase/best-practices/advanced-work-packaging/topics/rt-272/pubs/rr272-12.

CII. (2013b). Advanced Work Packaging: implementation case studies and expert interviews, Construction Industry Institute, Retrieved from: https://www.construction-institute.org/resources/knowledgebase/best-practices/advanced-work-packaging/topics/rt-272.

CII. (2020). AWP-integrated practices for construction completions, commissioning, and startup, Construction Industry Institute, Retrieved from: https://www.construction-institute.org/resources/knowledgebase/best-practices/advanced-work-packaging/topics/rt-364.

A. Costin, A. Adibfar, H. J. Hu, & S. S. Chen. (2018). Building Information Modeling (BIM) for transportation infrastructure - Literature review, applications, challenges, and recommendations. *Automation in Construction*, **94**, 257-281.

A. Damci, D. Arditi, & G. Polat. (2013). Multiresource leveling in line-of-balance scheduling. *Journal of Construction Engineering and Management*, **139**(9), 1108-1116.

B. Dave, A. Buda, A. Nurminen, & K. Främling. (2018). A framework for integrating BIM and IoT through open standards. *Automation in Construction*, **95**, 35-45.

N. Dawood, & E. Sriprasert. (2006). Construction scheduling using multi‐constraint and genetic algorithms approach. *Construction management and Economics*, **24**(1), 19-30.

T. Dettmers, P. Minervini, P. Stenetorp, & S. Riedel. (2017). Convolutional 2D knowledge graph embeddings. *32nd AAAI Conference on Artificial Intelligence,* **1707,** 1811-1818, New Orleans, US.

H. Dong, F. K. Hussain, & E. Chang. (2011). ORPMS: an ontology-based real-time project monitoring system in the cloud. *Journal of Universal Computer Science*, **17**(8), 1161-1182.
T. E. El-Diraby. (2013). Domain ontology for construction knowledge. *Journal of Construction Engineering and Management*, **139**(7), 768-784.
N. M. El-Gohary, & T. E. El-Diraby. (2010). Domain ontology for processes in infrastructure and construction. *Journal of Construction Engineering and Management*, **136**(7), 730-744.
L. M. El-Sabek, & B. Y. McCabe. (2018). Framework for managing integration challenges of last planner system in IMPs. *Journal of Construction Engineering and Management*, **144**(5), 04018022.
S. Elo, & H. Kyngäs. (2010). The qualitative content analysis process. *Journal of Advanced Nursing*, **62**(1), 107-115.
W. Fang, L. Ding, P. E. Love, H. Luo, H. Li, F. Pena-Mora, B. Zhong, & C. Zhou. (2020). Computer vision applications in construction safety assurance. *Automation in Construction*, **110**, 103013.
M. A. Farshchi, & M. Brown. (2011). Social networks and knowledge creation in the built environment: a case study. *Structural Survey*.
A. R. Fayek, & J. Peng. (2013). Adaptation of workface planning for construction contexts. *Canadian Journal of Civil Engineering*, **40**(10), 980-987.
G. Feltrin, J. Meyer, R. Bischoff, & M. Motavalli. (2010). Long-term monitoring of cable stays with a wireless sensor network. *Structure and Infrastructure Engineering*, **6**(5), 535-548.
E. Forcael, I. Ferrari, A. Opazo-Vega, & J. A. Pulido-Arcas. (2020). Construction 4.0: A literature review. *Sustainability*, **12**(22), 9755.
D. M. Frangopol, & P. Bocchini. (2012). Bridge network performance, maintenance and optimisation under uncertainty: Accomplishments and challenges. *Structure and Infrastructure Engineering*, **8**(4), 341-356.
D. H. Fudholi, N. Maneerat, R. Varakulsiripunth, & Y. Kato. (2009). Application of Protégé, SWRL and SQWRL in fuzzy ontology-based menu recommendation. *2009 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS),* 631-634.
O. Gelo, D. Braakmann, & G. Benetka. (2008). Quantitative and qualitative research: Beyond the debate. *Integrative psychological and behavioral science*, **42**(3), 266-290.
C. Gong, & D. M. Frangopol. (2020). Condition-based multiobjective maintenance decision making for highway bridges considering risk perceptions. *Journal of Structural Engineering*, **146**(5), 04020051.
F. Gong, Y. Ma, W. Gong, X. Li, C. Li, & X. Yuan. (2018). Neo4j graph database realizes efficient storage performance of oilfield ontology. *PloS one*, **13**(11), e0207595.
I. Goodfellow, Y. Bengio, A. Courville, & Y. Bengio. (2016). Deep learning. Cambridge, MA, USA, MIT Press. 9780262035613.
T. R. Gruber. (1995). Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, **43**(5-6), 907-928.
E. G. Guba, & Y. S. Lincoln. (1994). Competing paradigms in qualitative research. 105-117. Urbana, Champaign, U.S., SAGE. 9781506382920.

M. Gul, F. N. Catbas, & H. Hattori. (2015). Image-based monitoring of open gears of movable bridges for condition assessment and maintenance decision making. *Journal of Computing in Civil Engineering*, **29**(2), 04014034.
G. Gupta. (1999). Horn logic denotations and their applications. 127-159. Berlin, Germany, Springer. 9783540654636.
M. C. Gupta, & L. H. Boyd. (2008). Theory of constraints: A theory for operations management. *International Journal of Operations & Production Management*, **28**(10), 4927-4954.
Y. Halala. (2018). A framework to assess the costs and benefits of utilizing Advanced Work Packaging (AWP) in industrial construction. (M.D). University of Alberta, Alberta, Canada. Retrieved from https://era.library.ualberta.ca/items/7d63c96b-5a56-47cd-803f-d9a88cc3a678
Y. S. Halala, & A. R. Fayek. (2019). A framework to assess the costs and benefits of advanced work packaging in industrial construction. *Canadian Journal of Civil Engineering*, **46**(3), 216-229.
O. Hamdi. (2013). Advanced work packaging from project definition through site execution: driving successful implementation of WorkFace planning. (PhD). The University of Texas at Austin, Texas, U.S. Retrieved from https://repositories.lib.utexas.edu/handle/2152/21384
W. L. Hamilton, Z. Ying, & J. Leskovec. (2017). Inductive representation learning on large graphs. *31st Conference on Neural Information Processing Systems,* 1024-1034, Long Beach, CA, USA.
S. T. Hashemi, O. M. Ebadati, & H. Kaur. (2020). Cost estimation and prediction in construction projects: A systematic review on machine learning techniques. *SN Applied Sciences*, **2**(10), 1-27.
H. Hawk, & E. P. Small. (1998). The BRIDGIT bridge management system. *Structural engineering international*, **8**(4), 309-314.
P. Hitzler, M. Krotzsch, & S. Rudolph. (2009). Foundations of semantic web technologies. New York, U.S., CRC. 9780429143472.
D. G. Ho. (2006). The focus group interview: Rising to the challenge in qualitative research methodology. *Australian review of applied linguistics*, **29**(1), 1-19.
S. Hochreiter, & J. Schmidhuber. (1997). Long short-term memory. *Neural computation*, **9**(8), 1735-1780.
M. Horridge, & S. Bechhofer. (2011). The OWL API: A java API for OWL ontologies. *Semantic-Web*, **2**(1), 11-21.
P. Huthwohl, I. Brilakis, A. Borrmann, & R. Sacks. (2018). Integrating RC bridge defect information into BIM models. *Journal of Computing in Civil Engineering*, **32**(3), 04018013.
F. Javadnejad, D. T. Gillins, C. C. Higgins, & M. N. Gillins. (2017). BridgeDex: proposed web GIS platform for managing and interrogating multiyear and multiscale bridge-inspection images. *Journal of Computing in Civil Engineering*, **31**(6), 04017061.
S. Jeong, R. Hou, J. P. Lynch, H. Sohn, & K. H. Law. (2017). An information modeling framework for bridge monitoring. *Advances in Engineering Software*, **114**, 11-31.
S. Jeong, R. Hou, J. P. Lynch, H. Sohn, & K. H. Law. (2019). A scalable cloud-based cyberinfrastructure platform for bridge monitoring. *Structure and Infrastructure Engineering*, **15**(1), 82-102.

S. Jeong, Y. L. Zhang, S. O'Connor, J. P. Lynch, H. Sohn, & K. H. Law. (2016). A NoSQL data management infrastructure for bridge monitoring. *Smart Structures and Systems*, **17**(4), 669-690.
S. Ji, S. Pan, E. Cambria, P. Marttinen, & P. S. Yu. (2020). A survey on knowledge graphs: Representation, acquisition and applications. *34th AAAI Conference on Artificial Intelligence,* 1-21, New Your, US.
H. Jiang, Q. Bao, Q. Cheng, D. Yang, L. Wang, & Y. Xiao. (2020). Complex relation extraction: Challenges and opportunities. *arXiv: Computation and Language*, **1**, 2012.04821.
G. Kabir, R. Sadiq, & S. Tesfamariam. (2014). A review of multi-criteria decision-making methods for infrastructure management. *Structure and Infrastructure Engineering*, **10**(9), 1176-1210.
M. Karabulut. (2017). Application of Monte Carlo simulation and PERT/CPM techniques in planning of construction projects: A case study. *Periodicals of Engineering and Natural Sciences*, **5**(3), 409-420.
S. M. Kazemi, R. Goel, K. Jain, I. Kobyzev, A. Sethi, P. Forsyth, & P. Poupart. (2020). Representation learning for dynamic graphs: A survey. *Journal of machine Learning research*, **21**(70), 1-73.
L. Killam. (2013). Research terminology simplified: Paradigms, axiology, ontology, epistemology and methodology. Ontario, Canada, Laura Killam. 0993622801.
S. Kim, S. Kim, & J. Yang. (2016). Extraction and analysis of construction phase risk factors in high-rise construction project. *Korean Journal of Construction Engineering and Management*, **17**(2), 90-98.
C. Koo, T. Hong, & S. Kim. (2015). An integrated multi-objective optimization model for solving the construction time-cost trade-off problem. *Journal of Civil Engineering and Management*, **21**(3), 323-333.
T. Koschmann. (1996). Paradigm shifts and instructional technology: An introduction. *CSCL: Theory and practice of an emerging paradigm*, **12**(4), 18-19.
J. Kuckartz, & P. Collier. (2016). Achieving user-centric structural health monitoring: An integrated development approach. *Journal of Civil Structural Health Monitoring*, **6**(5), 803-816.
R. K. Kumar, & J. A. Harding. (2013). Ontology mapping using description logic and bridging axiom. *Computers in Industry*, **64**(1), 19-28.
H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, & M. Hoffmann. (2014). Industry 4.0. *Business & information systems engineering*, **6**(4), 239-242.
LCI. (2007). The Last Planner Production system workbook: Improving reliability in planning and work flow, Lean Construction Institute, Birmingham, UK, Retrieved from https://www.leanconstruction.org/wp-content/uploads/2016/06/Last-Planner-Workbook
T. Le, & J. David. (2017). Nlp-based approach to semantic classification of heterogeneous transportation asset data terminology. *Journal of Computing in Civil Engineering*, **31**(6), 04017057.
T. Le, H. D. Jeong, & S. B. Gilbert. (2020). Generating partial civil information model views using a semantic information retrieval approach. *Journal of information technology in construction*, **25**(2), 41-54.
M.-y. Leung, & I. Y. S. Chan. (2012). Exploring stressors of Hong Kong expatriate construction professionals in Mainland China: Focus group study. *Journal of Construction Engineering and Management*, **138**(1), 78-88.

M.-y. Leung, J. Yu, & Y. S. Chan. (2014). Focus group study to explore critical factors of public engagement process for mega development projects. *Journal of Construction Engineering and Management*, **140**(3), 04013061.
S. Li, Z. Zhao, R. Hu, W. Li, T. Liu, & X. Du. (2018). Analogical reasoning on Chinese morphological and semantic relations. *56th Annual Meeting of the Association for Computational Linguistics,* 138-143, Melbourne, Australia.
X. Li, H. L. Chi, P. Wu, & G. Q. Shen. (2020). Smart work packaging-enabled constraint-free path re-planning for tower crane in prefabricated products assembly process. *Advanced Engineering Informatics*, **43**, 101008.
X. Li, C. K. Wu, P. Wu, L. Q. Xiang, G. Q. Shen, S. Vick, & C. Z. Li. (2019). SWP-enabled constraints modeling for on-site assembly process of prefabrication housing production. *Journal of Cleaner Production*, **239**, 117991.
X. V. Lin, R. Socher, & C. Xiong. (2018). Multi-hop knowledge graph reasoning with reward shaping. *arXiv: Artificial Intelligence*, **2**, 1808.10568.
Y. Lin, Z. Liu, M. Sun, Y. Liu, & X. Zhu. (2015). Learning entity and relation embeddings for knowledge resolution. *Procedia Computer Science*, **108**, 345-354.
H. Liu, & S. Madanat. (2015). Adaptive optimisation methods in system-level bridge management. *Structure and Infrastructure Engineering*, **11**(7), 884-896.
H. X. Liu, M. Lu, & M. Al-Hussein. (2016). Ontology-based semantic approach for construction-oriented quantity take-off from BIM models in the light-frame building industry. *Advanced Engineering Informatics*, **30**(2), 190-207.
K. Liu, & N. El-Gohary. (2016). Ontology-based sequence labelling for automated information extraction for supporting bridge data analytics. *Procedia Engineering,* **145,** 504-510, Arizona, U.S.
K. Liu, & N. El-Gohary. (2017a). Ontology-based semi-supervised conditional random fields for automated information extraction from bridge inspection reports. *Automation in Construction*, **81**, 313-327.
K. Liu, & N. El-Gohary. (2017b). Similarity-based dependency parsing for extracting dependency relations from bridge inspection reports. *ASCE International Workshop on Computing in Civil Engineering,* 316-323, Seattle, U.S.
K. J. Liu, & N. El-Gohary. (2017c). Ontology-based semi-supervised conditional random fields for automated information extraction from bridge inspection reports. *Automation in Construction*, **81**, 313-327.
Z. Lounis. (2007). Aging highway bridges. *Canadian Consulting Engineer*, **48**(1), 30-34.
Z. Lounis, & T. P. McAllister. (2016). Risk-based decision making for sustainable and resilient infrastructure systems. *Journal of Structural Engineering*, **142**(9).
B. Luiten, M. Böhms, D. Alsem, & A. O'Keeffe. (2018). Asset information management using linked data for the life-cycle of roads. *The 6th International Symposium on Life-Cycle Civil Engineering (IALCCE),* 1-7, Ghent, Belgium.
C. K. Ma, N. M. Apandi, S. C. S. Yung, N. J. Hau, L. W. Haur, A. Z. Awang, & W. Omar. (2017). Repair and rehabilitation of concrete structures using confinement: a review. *Construction and Building Materials*, **133**, 502-515.
S. Medhi, & H. K. Baruah. (2017). Relational database and graph database: A comparative analysis. *Journal of Process Management. New Technologies*, **5**(2), 1-9.

M. Miwa, & M. Bansal. (2016). End-to-end relation extraction using LSTMs on sequences and tree structures. *54th Annual Meeting of the Association for Computational Linguistics,* **1,** 1105-1116, Berlin, Germany.
A. Miyamoto, & H. Asano. (2017). Development and practical application of a lifetime management system for prestressed concrete bridges. *Civil Engineering Infrastructures Journal*, **50**(2), 395-410.
A. Miyamoto, & H. Asano. (2018). Application of J-BMS to performance evaluation and remaining life prediction of an existing RC bridge. *Civil Engineering Infrastructures Journal*, **51**(2), 311-337.
MLIT. (2015). Road maintenance in Japan: Problems and solutions, Ministry of Land, Infrastructure, Transport and Tourism, Tokyo, Japn. Retrieved from: https://www.mlit.go.jp/road/road_e/s3_maintenance.html.
G. Morgenthal, N. Hallermann, J. Kersten, J. Taraben, P. Debus, M. Helmrich, & V. Rodehorst. (2019). Framework for automated UAS-based structural condition assessment of bridges. *Automation in Construction*, **97**, 77-95.
S. A. Moufti, T. Zayed, & S. Abu Dabous. (2014). Defect-based condition assessment of concrete bridges fuzzy hierarchical evidential reasoning approach. *Transportation Research Record*, **243**(1), 88-96.
K. P. Murphy. (2012). Machine learning: A probabilistic perspective. Cambridge, MA, USA, MIT Press. 0262304325.
D. Nathani, J. Chauhan, C. Sharma, & M. Kaul. (2019). Learning attention-based embeddings for relation prediction in knowledge graphs. *arXiv: Machine Learning*, **1**, 1906.01195.
NCHRP. (2007). Bridge inspection practcies: A synthesis of highway practice, Colorado, U.S. Retrieved from: https://www.trb.org/Publications/Blurbs/159753.aspx.
D. Q. Nguyen. (2020). An overview of embedding models of entities and relationships for knowledge base completion. *14th Workshop on Graph-Based Natural Language Processing,* **1703,** 08098, Michigan, US.
D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, & D. Phung. (2018). A novel embedding model for knowledge base completion based on convolutional neural network. *Conference of the North American Chapter of the Association for Computational Linguistics,* **2,** 327-333, New Orleans, US.
D. Q. Nguyen, T. Vu, T. D. Nguyen, D. Q. Nguyen, & D. Phung. (2019). A capsule network-based embedding model for knowledge graph completion and search personalization. *17th Annual Conference of the North American Chapter of the Association for Computational Linguistics,* 2180-2189, Minneapolis, US.
A. Nieto-Morote, & F. Ruz-Vila. (2012). Last planner control system applied to a chemical plant construction. *Journal of Construction Engineering and Management*, **138**(2), 287-293.
M. Niknam, & S. Karshenas. (2017). A shared ontology approach to semantic representation of BIM data. *Automation in Construction*, **80**, 22-36.
NRA. (2018). CEDR-INTERLINK approach with basic European road OTL, European National Road Authorities, Brussels, Belgium. Retrieved from: https://roadotl.eu/static/ireport/index.html.
T. O. Nyumba, K. Wilson, C. J. Derrick, & N. Mukherjee. (2018). The use of focus group discussion methodology: Insights from two decades of application in conservation. *Methods in Ecology and evolution*, **9**(1), 20-32.
OCIA. (2015). Aging and failing infrastructure systems: Highway bridges, Office of Cyber and Infrastructure Analysis, Washington, U.S. Retrieved from:

http://cip.gmu.edu/wp-content/uploads/2015/09/OCIA-Aging-and-Failing-Infrastructure-Systems-Highway-Bridges.pdf.
N. M. Okasha, & D. M. Frangopol. (2009). Lifetime-oriented multi-objective optimization of structural maintenance considering system reliability, redundancy and life-cycle cost using GA. *Structural Safety*, **31**(6), 460-474.
N. M. Okasha, & D. M. Frangopol. (2010). Novel approach for multicriteria optimization of life-cycle preventive and essential maintenance of deteriorating structures. *Journal of Structural Engineering*, **136**(8), 1009-1022.
A. D. Orcesi, & D. M. Frangopol. (2011). Use of lifetime functions in the optimization of nondestructive inspection strategies for bridges. *Journal of Structural Engineering*, **137**(4), 531-539.
J. L. Ottesen, & G. A. Martin. (2019). Bare facts and benefits of resource-loaded CPM schedules. *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, **11**(3), 02519001.
S. J. Pan, & Q. Yang. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, **22**(10), 1345-1359.
C. S. Park, D. Y. Lee, O. S. Kwon, & X. Wang. (2013). A framework for proactive construction defect management using BIM, augmented reality and ontology-based data collection template. *Automation in Construction*, **33**, 61-71.
J. E. Parks, D. N. Brown, M. J. Ameli, & C. P. Pantelides. (2016). Seismic repair of severely damaged precast reinforced concrete bridge columns connected with grouted splice sleeves. *ACI Structural Journal*, **113**(3), 615-626.
N. Peng, H. Poon, C. Quirk, K. Toutanova, & W. Yih. (2017). Cross-sentence N-ary relation extraction with graph LSTMs. *Transactions of the Association for Computational Linguistics*, **5**(1), 101-115.
J. Pennington, R. Socher, & C. Manning. (2014). Glove: global vectors for word representation. *Conference on Empirical Methods in Natural Language Processing,* 1532-1543, Doha, Qatar.
B. M. Phares, & M. Cronin. (2015). Synthesis on the use of accelerated bridge construction approaches for bridge rehabilitation, U.S. Department of Transportation, Iowa, USA. Retrieved from: https://rosap.ntl.bts.gov/view/dot/41826.
P. Probst, M. N. Wright, & A. L. Boulesteix. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, **9**(3), e1301.
N. Puri, & Y. Turkan. (2020). Bridge construction progress monitoring using lidar and 4D design models. *Automation in Construction*, **109**, 102961.
M. Qu, & J. Tang. (2019). Probabilistic logic neural networks for reasoning. *33rd Conference on Neural Information Processing Systems,* 7712-7722, Vancouver, Canada.
G. Q. Ren, R. Ding, & H. J. Li. (2019). Building an ontological knowledgebase for bridge maintenance. *Advances in Engineering Software*, **130**, 24-40.
X. Rong. (2014). word2vec parameter learning explained. *arXiv: Computation and Language*, **4**, 1411.2738.
S. Sabatino, D. M. Frangopol, & Y. Dong. (2016). Life cycle utility-informed maintenance planning based on lifetime functions: optimum balancing of cost, failure consequences and performance benefit. *Structure and Infrastructure Engineering*, **12**(7), 830-847.

SAWS. (2010). Provisions on the administration of safety technology training and examination for special operation personnel, State Administration of Work Safety, Beijing, China, Retrieved from https://wenku.baidu.com/view/b95340255901020207409c92.html
M. S. Schlichtkrull, T. Kipf, P. Bloem, R. V. Den Berg, I. Titov, & M. Welling. (2018). Modeling relational data with graph convolutional networks. *European Semantic Web Conference,* 593-607, Heraklion, Greece.
P. Schönbeck, M. Löfsjögård, & A. Ansell. (2020). Quantitative review of construction 4.0 technology presence in construction project research. *Buildings*, **10**(10), 173.
B. Shi, & T. Weninger. (2017). ProjE: embedding projection for knowledge graph completion. *31st AAAI Conference on Artificial Intelligence,* 1236-1242, California, US.
Z. Shi, M. Dong, Y. Jiang, & H. Zhang. (2005). A logical foundation for the semantic Web. *Science in China Series F: Information Sciences*, **48**(2), 161-178.
C. S. Shire, H. Kang, N. S. Dang, & D. Lee. (2017). Development of BIM-based bridge maintenance system for cable-stayed bridges. *Smart Structures and Systems*, **20**(6), 697-708.
A. Shrestha, & A. Mahmood. (2019). Review of deep learning algorithms and architectures. *IEEE Access*, **7**, 53040-53065.
D. Simeone, S. Cursi, & M. Acierno. (2019). BIM semantic-enrichment for built heritage representation. *Automation in Construction*, **97**, 122-137.
M. A. Simon. (1996). Beyond inductive and deductive reasoning: The search for a sense of knowing. *Educational Studies in mathematics*, **30**(2), 197-210.
Z. T. Şimşit, N. S. Günay, & Ö. Vayvay. (2014). Theory of constraints: A literature review. *Procedia-Social and Behavioral Sciences*, **150**, 930-936.
C. Spathis, & S. Constantinides. (2003). The usefulness of ERP systems for effective management. *Industrial Management & Data Systems*, **103**(9), 677-685.
N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, & R. Salakhutdinov. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of machine Learning research*, **15**, 1929-1958.
J. Stallkamp, M. Schlipsing, J. Salmen, & C. Igel. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, **32**, 323-332.
Stanford. (2002). Ontology development 101: A guide to creating your first ontology, Standford University, San Francisco, U.S., Retrieved from https://protege.stanford.edu/publications/ontology_development/ontology101.pdf
C. L. Streeter, & D. F. Gillespie. (1993). Social network analysis. *Journal of Social Service Research*, **16**(1-2), 201-222.
R. Studer, V. R. Benjamins, & D. Fensel. (1998). Knowledge engineering: Principles and methods. *Data & knowledge engineering*, **25**(1-2), 161-197.
S. Tahery, & S. Farzi. (2020). Customized query auto-completion and suggestion: A review. *Information Systems*, **87**, 101415.
A. M. T. Thome, P. S. Ceryno, A. Scavarda, & A. Remmen. (2016). Sustainable infrastructure: A review and a research agenda. *Journal of Environmental Management*, **184**, 143-156.
P. D. Thompson. (2012). Estimating asset deterioration and life expectancy by using levels of service. *Transportation Research Record*, **2285**(1), 19-26.

T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, & G. Bouchard. (2016). Complex embeddings for simple link prediction. *33rd International Conference on Machine Learning,* **48,** 2071-2080, New York, USA.
Y. Turkan, J. Hong, S. Laflamme, & N. Puri. (2018). Adaptive wavelet neural network for terrestrial laser scanner-based crack detection. *Automation in Construction*, **94**, 191-202.
TxDOT. (2020). Bridge Inspection Manual, Texas Department of Transportation, Texas, U.S., Retrieved from http://onlinemanuals.txdot.gov/txdotmanuals/ins/ins.pdf
J. M. van Noortwijk, & D. M. Frangopol. (2004). Two probabilistic life-cycle maintenance models for deteriorating civil infrastructures. *Probabilistic Engineering Mechanics*, **19**(4), 345-359.
S. Vashishth, S. Sanyal, V. Nitin, N. Agrawal, & P. Talukdar. (2019). InteractE: improving convolution-based knowledge graph embeddings by increasing feature interactions. *34th AAAI Conference on Artificial Intelligence,* **1911,** 00219, New York, US.
A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, & I. Polosukhin. (2017). Attention is all you need. *Advances in neural information processing systems,* **30,** 5998-6008, Long Beach, CA, USA.
P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, & Y. Bengio. (2017). Graph attention networks. *arXiv: Machine Learning*, **1710**, 10903.
N. Venkatraman. (1997). Beyond outsourcing: Managing IT resources as a value center. *Sloan Management Review*, **38**(3), 51-64.
A. Vukotic, N. Watt, T. Abedrabbo, D. Fox, & J. Partner. (2014). Neo4j in action. New York, U.S., Manning. 9781617290763
J. Wang. (2018). Total constraint management for improving construction work flow in liquefied natural gas industry. (PhD). Curtin University, Perth, WA. Retrieved from https://espace.curtin.edu.au/bitstream/handle/20.500.11937/73516/Wang%20Jun%202018.pdf
J. Wang, W. C. Shou, X. Y. Wang, & P. Wu. (2016). Developing and evaluating a framework of total constraint management for improving workflow in liquefied natural gas construction. *Construction management and Economics*, **34**(12), 859-874.
P. Wang, P. Wu, H. Chi, & X. Li. (2020). Adopting lean thinking in virtual reality-based personalized operation training using value stream mapping. *Automation in Construction*, **119**, 103355.
Z. Wang, J. Zhang, J. Feng, & Z. Chen. (2014). Knowledge graph embedding by translating on hyperplanes. *28th AAAI Conference on Artificial Intelligence,* 1112-1119, California, US.
K. J. Watson, J. H. Blackstone, & S. C. Gardiner. (2007). The evolution of a management philosophy: The theory of constraints. *Journal of operations management*, **25**(2), 387-402.
C. Wei, C. Chien, & M. Wang. (2005). An AHP-based approach to ERP system selection. *International journal of production economics*, **96**(1), 47-62.
S. Wei, Y. Bao, & H. Li. (2020). Optimal policy for structure maintenance: A deep reinforcement learning framework. *Structural Safety*, **83**, 101906.
J. Werbrouck, P. Pauwels, M. Bonduel, J. Beetz, & W. Bekers. (2020). Scan-to-graph: Semantic enrichment of existing building geometry. *Automation in Construction*, **119**, 103286.

S. Wilson. (2001). What is an indigenous research methodology? *Canadian journal of native education*, **25**(2), 175-179.

A. K. Woldesenbet. (2014). Highway infrastructure data and information integration and assessment framework: A data driven decision-making approach. (PhD). Iowa State University, Iowa, USA. Retrieved from https://lib.dr.iastate.edu/cgi/viewcontent.cgireferer=&httpsredir=1&article=5024&context=etd

K. Wong, H. Unsal, J. E. Taylor, & R. E. Levitt. (2010). Global dimension of robust project network design. *Journal of Construction Engineering and Management*, **136**(4), 442-451.

WTW. (2018). Ageing Infrastructure: More than a bump in the road, Willis Towers Watson, London, UK. Retrieved from: https://mvvsp1.5gcdn.net/a2b6e8faeb024bee99941403f8eaac57.

C. Wu, C. Chen, R. Jiang, P. Wu, B. Xu, & J. Wang. (2019). Understanding laborers' behavioral diversities in multinational construction projects using integrated simulation approach. *Engineering construction & architectural management*, **26**(9), 2120-2146.

C. Wu, X. Wang, P. Wu, J. Wang, R. Jiang, M. Chen, & M. Swapan. (2021a). Hybrid deep learning model for automating constraint modelling in advanced working packaging. *Automation in Construction*, **127**, 103733.

C. Wu, P. Wu, J. Wang, R. Jiang, M. Chen, & X. Wang. (2020a). Critical review of data-driven decision-making in bridge operation and maintenance. *Structure and Infrastructure Engineering*, **12**, 1-24.

C. Wu, P. Wu, J. Wang, R. Jiang, M. Chen, & X. Wang. (2020b). Ontological knowledge base for concrete bridge rehabilitation project management. *Automation in Construction*, **121**, 103428.

C. Wu, P. Wu, J. Wang, R. Jiang, M. Chen, & X. Wang. (2021b). Developing a hybrid approach to extract constraints related information for constraint management. *Automation in Construction*, **124**, 103563.

J. H. Xie, & R. L. Hu. (2013). Experimental study on rehabilitation of corrosion-damaged reinforced concrete beams with carbon fiber reinforced polymer. *Construction and Building Materials*, **38**, 708-716.

S. Xu, J. Wang, W. Shou, T. Ngo, A.-M. Sadick, & X. Wang. (2020). Computer vision techniques in construction: A critical review. *Archives of Computational Methods in Engineering*, 1-15.

X. Xu, & H. Cai. (2020). Semantic approach to compliance checking of underground utilities. *Automation in Construction*, **109**, 103006.

B. Yang, W. T. Yih, X. He, J. Gao, & L. Deng. (2014). Embedding entities and relations for learning and inference in knowledge bases. *3rd International Conference on Learning Representations,* **4,** 1-12, San Diego, U.S.

F. Yang, Z. Yang, & W. W. Cohen. (2017). Differentiable learning of logical rules for knowledge base completion. *31st Conference on Neural Information Processing Systems,* **3,** 1702.08367, Long Beach, CA, USA.

C. M. Yeum, & S. J. Dyke. (2015). Vision-based automated crack detection for bridge inspection. *Computer-Aided Civil and Infrastructure Engineering*, **30**(10), 759-770.

W. Yin, Y. Yaghoobzadeh, & H. Schutze. (2018). Recurrent one-hop predictions for reasoning over knowledge graphs. *27th International Conference on Computational Linguistics,* **1,** 1806.04523, New Mexico, USA.

J. Zhang, & N. M. El-Gohary. (2016). Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking. *Journal of Computing in Civil Engineering*, **30**(2), 04015014.
S. Zhang, F. Boukamp, & J. Teizer. (2015). Ontology-based semantic modeling of construction safety knowledge: towards automated safety planning for job hazard analysis (JHA). *Automation in Construction*, **52**, 29-41.
Y. Zhang, X. Chen, Y. Yang, A. Ramamurthy, B. Li, Y. Qi, & L. Song. (2020). Efficient Probabilistic Logic Reasoning with Graph Neural Networks. *The International Conference on Learning Representations (ICLR),* **1,** 2001.11850, Ababa, Ethiopia.
Y. Zhang, P. Qi, & C. D. Manning. (2018a). Graph convolution over pruned dependency trees improves relation extraction. *Conference on Empirical Methods in Natural Language Processing,* 2205-2215, Brussels, Belgium.
Y. L. Zhang, S. M. O'Connor, G. W. van der Linden, A. Prakash, & J. P. Lynch. (2016). SenStore: A scalable cyberinfrastructure platform for implementation of data-to-decision frameworks for infrastructure health management. *Journal of Computing in Civil Engineering*, **30**(5), 04016012.
Z. Zhang, P. Cui, & W. Zhu. (2018b). Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, **14**(8), 1-24.
M. Zhao, W. Jia, & Y. Huang. (2020). Attention-based aggregation graph networks for knowledge graph information transfer. *Pacific-Asia Conference on Knowledge Discovery and Data Mining,* 542-554, Singapore.
B. Zhong, X. Pan, P. E. D. Love, J. Sun, & C. Tao. (2020a). Hazard analysis: A deep learning and text mining framework for accident prevention. *Advanced Engineering Informatics*, **46**, 101152.
B. Zhong, X. Xing, H. Luo, Q. Zhou, H. Li, T. Rose, & W. Fang. (2020b). Deep learning-based extraction of construction procedural constraints from construction regulations. *Advanced Engineering Informatics*, **43**, 101003.
J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, & M. Sun. (2018). Graph neural networks: A review of methods and applications. *AI Open*, **1**, 57-81.
Z. P. Zhou, Y. M. Goh, & L. J. Shen. (2016). Overview and analysis of ontology studies supporting development of the construction industry. *Journal of Computing in Civil Engineering*, **30**(6).
J. Zhu, P. Wu, M. Chen, M. J. Kim, X. Wang, & T. Fang. (2020). Automatically processing IFC clipping representation for BIM and GIS integration at the process level. *Applied Sciences*, **10**(6), 1-19.

Appendix

Appendix 1 List of publications

1. **Wu, C.**, Wu, P., Jiang, R., Wang, J., Chen, M., & Wang, X*. (2021). Hybrid deep learning model for automating constraint modelling in advanced working packaging. Automation in Construction, 127, 103733. https://doi.org/10.1016/j.autcon.2021.103733
2. **Wu, C.**, Wu, P., Wang, J., Jiang, R., Chen, M., & Wang, X*. (2021). Developing a hybrid approach to extract constraints related information for constraint management. Automation in Construction, 124, 103563. https://doi.org/10.1016/j.autcon.2021.103563
3. **Wu, C.**, Wu, P., Wang, J., Jiang, R., Chen, M., & Wang, X* (2020). Ontological knowledge base for concrete bridge rehabilitation project management. Automation in Construction, 121, 103428. https://doi.org/10.1016/j.autcon.2020.103428
4. **Wu, C.**, Wu, P., Wang, J., Jiang, R., Chen, M., & Wang, X*. (2020). Critical review of data-driven decision-making in bridge operation and maintenance. Structure and Infrastructure Engineering, 12, 1-24. https://doi.org/10.1080/15732479.2020.1833946
5. **Wu, C.**, Wu, P., Jiang, R., Wang, J., & Wan, M*. (2020). Evaluating the economic and social benefits of multiutility tunnels with an agent-based simulation approach. Engineering Construction & Architectural Management, (ahead-of-print). https://doi.org/10.1108/ECAM-07-2019-0399
6. Li, X., **Wu, C.***, Wu, P., Xiang, L., Shen, G. Q., Vick, S., & Li, C. Z. (2019). SWP-enabled constraints modeling for on-site assembly process of prefabrication housing production. Journal of Cleaner Production, 239, 117991. https://doi.org/10.1016/j.jclepro.2019.117991
7. **Wu, C.***, Chen, C., Jiang, R., Wu, P., Xu, B., & Wang, J. (2019). Understanding laborers' behavioral diversities in multinational construction projects using integrated simulation approach. Engineering, Construction and Architectural Management, 26(9), 2120-2146. https://doi.org/10.1108/ECAM-07-2018-0281
8. **Wu, C.***, Xu, B., Mao, C., & Li, X. (2017). Overview of BIM maturity measurement tools. Journal of Information Technology in Construction (ITcon), 22(3), 34-62. https://www.itcon.org/paper/2017/3

Appendix 2 Focus group questions (classes)

Initial domain concepts/classes of bridge rehabilitation have been identified by the researcher. These classes 1) cover three aspects: constraint, rehabilitation task, and project participant, 2) are organised in hierarchies which have maximum four levels currently. Please provide advice in terms of adding, deleting, and modifying classes and subclasses. Please note the maximum number of class level in this research is five.

Constraint (level-1)

1. Engineering constraint (level-2)

Initial sub-classes (level-3) of 'Engineering constraint' (level-2): BoQ, design document, inspection, maintenance, and rehabilitation (IMRR) reports, project approval, site permit, standard or manual, working specification

Suggestions for adding/deleting/modifying classes/sub-classes

1) BoQ (level-3)

Initial sub-classes (level-4) of 'BoQ': N/A

Suggestions for adding/deleting/modifying classes/sub-classes

2) Design document (level-3)

Initial sub-classes (level-4) of 'Design document' (level-3): original design drawing, original shop drawing, maintenance design drawing, maintenance shop drawing, original 3D model, maintenance 3D model

Suggestions for adding/deleting/modifying classes/sub-classes

3) IMRR report (level-3)

Initial sub-classes (level-4) of 'IMRR report' (level-3): maintenance/rehabilitation history report, inspection report

Suggestions for adding/deleting/modifying classes/sub-classes

4) Project approval (level-3)

Initial sub-classes (level-4) of 'Project approval' (level-3): bridge close approval, construction approval, environment approval

Suggestions for adding/deleting/modifying classes/sub-classes

5) Site permit (level-3)

Initial sub-classes (level-4) of 'Site permit' (level-3): site discharge permit, night working permit, noise permit, safety permit

Suggestions for adding/deleting/modifying classes/sub-classes

6) Standard and manual (level-3)

Initial sub-classes (level-4) of 'Standard and manual' (level-3): N/A

Suggestions for adding/deleting/modifying classes/sub-classes

7) Working specification (level-3)

Initial sub-classes (level-4) of 'Working specification' (level-3): N/A

Suggestions for adding/deleting/modifying classes/sub-classes

2. Site constraint (level-2)

Initial sub-classes (level-3) of 'Site constraint' (level-2): people, temporary facility, weather

1) People (level-3)

Initial sub-classes (level-4) of 'People' (level-3): general labour, special labour, engineer/manager

Suggestions for adding/deleting/modifying classes/sub-classes

2) Temporary facility (level-3)

Initial sub-classes (level-4) of 'Temporary facility' (level-3): lighting facility, power facility, site accommodation, storage area, supporting system, water facility, work space

Suggestions for adding/deleting/modifying classes/sub-classes

3) *Weather (level-3)*

Initial sub-classes (level-4) of 'Weather' (level-4): heat/cold, rain, sun, wind

Suggestions for adding/deleting/modifying classes/sub-classes

3. Supply constraint (level-2)

Initial sub-classes (level-3) of ‘Supply constraint’ (level-2): equipment, material

1) Equipment (level-3)

Initial sub-classes (level-4) of ‘Equipment’ (level-3): drilling equipment, excavating equipment, transporting equipment, mixing equipment, monitoring equipment, paving equipment, piling equipment, PPE, pumping equipment, rolling equipment, vibrating equipment, welding equipment, auxiliary equipment, demolishing equipment

Suggestions for adding/deleting/modifying classes/sub-classes

2) Material (level-3)

Initial sub-classes (level-4) of ‘Material’ (leve-3): asphalt material, cement material, coating material, admixture, concrete material, resin material, soil material, metal material, sand and stone material, supplementary material

Suggestions for adding/deleting/modifying classes/sub-classes

Rehabilitation task (level-1)

1. Hazard treating (level-2)

Initial sub-classes (level-3) of ‘Hazard treating’ (level-2): asphalt hazard treating, concrete hazard treating, rust treating, deflection treating

1) Asphalt hazard treating (level-3)

Initial sub-classes (level-4) of ‘Asphalt hazard treating’ (level-3): asphalt crack treating, asphalt aging treating, asphalt deformation treating

Suggestions for adding/deleting/modifying classes/sub-classes

2) *Concrete hazard treating (level-3)*

Initial sub-classes (level-4) of ‘Concrete hazard treating’ (level-3): concrete crack treating, concrete weathering treating, concrete spalling treating, honeycomb treating, pockmark treating

Suggestions for adding/deleting/modifying classes/sub-classes

3) *Rust treating (level-3)*

Initial sub-classes (level-4) of 'Rust treating' (level-3): N/A

Suggestions for adding/deleting/modifying classes/sub-classes

4) *Deflection treating (level-3)*

Initial sub-classes (level-4) of 'Deflection treating' (level-3): anchoring, jacking, retaining wall

Suggestions for adding/deleting/modifying classes/sub-classes

2. Reinforcement (level-2)

Initial sub-classes of 'Reinforcement' (level-2): exterior covering, external reinforcing, foundation reinforcing, section area increasing, structure changing

1) Exterior covering (level-3)

Initial sub-classes (level-4) of 'Exterior covering' (level-3): CFRPS sticking, fibre glass cloth sticking, panel sticking, concrete wrapping, steel wrapping

Suggestions for adding/deleting/modifying classes/sub-classes

2) *External reinforcing (level-3)*

Initial sub-classes (level-4) of 'External reinforcing' (level-3): N/A

Suggestions for adding/deleting/modifying classes/sub-classes

3) *Foundation reinforcing (level-3)*

Initial sub-classes (level-4) of 'Foundation reinforcing' (level-3): additional piling, artificial foundation, anti-scouring

Suggestions for adding/deleting/modifying classes/sub-classes

4) Section-area increasing (level-3)

Initial sub-classes (level-4) of 'Section-area increasing' (level-3): pier section-area increasing, beam section-area, foundation section-area increasing

Suggestions for adding/deleting/modifying classes/sub-classes

5) *Structure system transformation (level-3)*

Initial sub-classes (level-4) of 'Structure system transformation' (level-3): beam to beam-arch combination, simply-supported system to continuous-slab deck system, adding traversing beam

Suggestions for adding/deleting/modifying classes/sub-classes

3. Replacement (level-2)

Initial sub-classes (level-3) of 'Replacement' (level-2): deck system replacement, sub-structure replacement, super-structure replacement

1) *Deck system replacement (level-3)*

Initial sub-classes (level-4) of 'Deck system replacement' (level-3): auxiliary system replacement, deck pavement replacement

Suggestions for adding/deleting/modifying classes/sub-classes

2) *Sub-structure replacement (level-3)*

Initial sub-classes (level-4) of 'Sub-structure replacement' (level-3): pre-cast pier replacement, cutwater replacement, pre-cast abutment component replacement

Suggestions for adding/deleting/modifying classes/sub-classes

3) *Super-structure replacement (level-3)*

Initial sub-classes (level-4) of 'Super-structure replacement' (level-3): pre-cast beam replacement, bearing replacement, cable replacement

Suggestions for adding/deleting/modifying classes/sub-classes

General task (procedure) (level-1)

1. Preparation (level-2)

Initial sub-classes (level-3) of 'Preparation' (level-2): site surveying, site layout, bridge inspection, building temporary facility, project mobilisation

Suggestions for adding/deleting/modifying classes/sub-classes

2. Execution (level-2)

Initial sub-classes (level-3) of 'Execution' (level-2): cleaning, transporting, coating, pouring, curing, dismantling, drilling, excavating, grouting, fixing, installing, mixing, paving, piling, rebar engineering, rolling, spraying, sticking, vibrating, welding

Suggestions for adding/deleting/modifying classes/sub-classes

3. Check and acceptance (level-2)

Initial sub-classes (level-3) of 'Check and acceptance' (level-2): quality checking, intermediate acceptance, final acceptance

Suggestions for adding/deleting/modifying classes/sub-classes

Project participant (level-1)

1. Project-level participant (level-2)

Initial sub-classes (level-3) of 'Project participant' (level-2): owner, pre-completion stage participant, rehabilitation stage participant

1) *Owner (level-3)*

Initial sub-classes (level-4) of 'Owner' (level-3): N/A

Suggestions for adding/deleting/modifying classes/sub-classes

2) *Pre-completion stage participant (level-3)*

Initial sub-classes (level-4) of 'Pre-completion stage participant' (level-3): designer, contractor, supplier, consulting team

Suggestions for adding/deleting/modifying classes/sub-classes

3) Rehabilitation stage participant (level-3)

Initial sub-classes (level-4) of 'Rehabilitation stage participant' (level-3): designer, contractor, supplier, consulting team, maintenance team

Suggestions for adding/deleting/modifying classes/sub-classes

2. External participant (level-2)

Initial sub-classes (level-3) of 'External participant' (level-2): bridge user, government agency

1) Bridge user (level-3)

Initial sub-classes (level-4) of 'Bridge user' (level-3): driver, pedestrian

Suggestions for adding/deleting/modifying classes/sub-classes

2) Government agency (level-3)

Initial sub-classes (level-4) of 'Government agency' (level-3): municipal bureau, transportation department, environment department, construction department

Suggestions for adding/deleting/modifying classes/sub-classes

Appendix 3 Focus group questions (relations)

1. constrains (level-1)

Initial sub-relations (level-2) of ‘constrains’ (level-1): there are 14 initial relations listed below, please provide advice in terms of adding, deleting, and modifying these relations.

Relation name	Head/tail entity linked by the relation	Advice
accommodate	head: temporary facility entities; tail: people	
check-quality	head: engineer/manager; tail: constraint entities	
transport	head: equipment; tail: constraint entities	
grant-permission-to	head: project participant; tail: document	
supervise	head: engineer/manager; tail: constraint entities	
monitor	head: equipment entities; tail: constraint entities	
use	head: labour; tail: material entities	
produce	head: equipment entities or task/procedure entities; tail: material or temporary facility entities	
protect	head: equipment entities; tail: people entities	
provide-space-for	head: temporary facility entities; tail: constraint entities	
work-in	head: people entities; tail: task/procedure entities	
remove	head: participant entities; tail: constraint entities	
specify	head: document entities; tail: constraint entities	
has-unremoved-constraints	head: constraint entities; tail: constraint entities	
Suggestions for adding/deleting/modifying relations		

2. work dependency (level-1)

Initial sub-relations (level-2) of ‘work dependency’ (level-1): there are three initial relations listed below, please provide advice in terms of adding, deleting, and modifying these relations.

Relation name	Head/tail entity linked by the relation	Advice
is-preceded-of	head and tail: task/procedure entities	
is-succeeded-of		
proceed-concurrently		
Suggestions for adding/deleting/modifying relations		

3. has-attribute (level-1)

Initial sub-relations (level-2) of 'has-attribute' (level-1): there are eight initial relations listed below, please provide advice in terms of adding, deleting, and modifying these relations.

Relation name	Head/tail entity linked by the relation	Advice
has-amount	head: constraint entities; tail: numerical values	
has-geometry		
has-price		
has-speed		
has-temperature		
has-humidity		
has-time		
has-type	head: constraint entities; tail: String/numerical values	
Suggestions for adding/deleting/modifying relations		

4. has-constraint-status (level-1)

Initial relations (level-2): there are initially six relations listed below, please provide advice in terms of adding, deleting, and modifying these relations.

Relation name	Head/tail entity linked by the relation	Advice
has-actual-removal-date	head: constraint entities; tail: date values	
has-planned-removal-date		
has-removal-delay	head: constraint entities; tail: numerical values	
is-timely-removed	head: constraint entities; tail: Boolean values	
is-potentially-delayed		
is-removed		
Suggestions for adding/deleting/modifying relations		

5. has-progress-information (level-1)

Relation name	Head/tail entity linked by the relation	Advice

has-actual-duration	head: task/procedure entities; tail: numerical values	
has-progress		
has-actual-start-date	head: task/procedure entities; tail: date values	
has-actual-finish-date		
has-planned-start-date		
has-planned-finish-date		
is-finished	head: task/procedure entities; tail: Boolean values	
is-started		
is-delayed		
is-potentially-delayed		
Suggestions for adding/deleting/modifying relations		