Autonomous Complex Knowledge Mining and Graph Representation through Natural Language Processing and Transfer Learning

Xiaofeng Zhu, Haijiang Li*, Tengxiang Su
Cardiff University, UK
LiH@cardiff.ac.uk

Abstract:

Regulatory documents play a significant role in securing engineering project quality, standard process management and long-term sustainable developments. With the digitisation of knowledge in the AEC industry, the demand for automated knowledge mining has emerged when confronted with substantial regulations. However, the current interpretation approaches for regulatory documents are still mostly labour-intensive and flawed in complex knowledge. Based on transfer learning (BERT) and natural language processing (e.g., NLP-Syntactic Parsing), this paper proposes a fully automated knowledge mining framework to convert complex knowledge in textual regulations to graph-based knowledge representations. The framework uses a BERT-based engine to extract clauses from regulation documents through fine-tuning with the self-developed domain dataset. A constituent extractor is developed to process the provisions with complex knowledge and extract constituents. A knowledge modelling engine integrates the extracted constituents into a graph-based regulation knowledge model, which can be queried, visualised, and directly applied to downstream applications. The outcome has demonstrated promising performance in complex knowledge mining and knowledge graph modelling based on ISO 19650 case study. This research can effectively convert textual regulation documents to their counterpart regulatory knowledge base, contributing to automated knowledge acquisition and multi-domain knowledge fusion toward regulation digitalization.

Keywords:

Knowledge mining; Natural language processing (NLP); Transfer learning; Knowledge modelling; Regulation document.

1. Introduction

As a summary of human knowledge and experience, regulatory documents, including guidelines, codes, standards, specifications, and manuals of practice, play a significant role in an engineering project's quality control, process management, safety, and sustainability assurance [1]. With the continuous development of the architecture, engineering, and construction (AEC) industry, many public industrial regulation documents have been released, and correlations between knowledge in different regulation documents are gradually emerging [2]. In addition, an increasing number of governments and enterprises in the AEC industry are eager for knowledge-driven automated quality control and process management [3]. Thus, a comprehensive knowledge management system is indispensable to manage, converge, utilise, and maintain the knowledge embedded in regulation documents more effectively. Given that extensive human knowledge is required in the interpretation of the regulation document, the current knowledge mining process still mainly relies on the manual work of domain experts. This time-consuming and labour-intensive approach falls far behind the demand for large-scale knowledge mining [4].

Knowledge in regulation documents can be broadly divided into two categories, simple and complex [5]. Simple knowledge is a requirement that consists of a single logic or relationship (e.g., “the fire separation distance shall be not less than 10 feet.”). Complex knowledge represents requirements that encompass multiple entities, logic, and relationships (e.g., “the appointing party shall establish the requirements that tendering organizations shall meet within their tender response”.), which are more commonly found in qualitative regulations. In the past decades, significant efforts have been made to automate the interpretation of regulation documents in the AEC domain. Many rule-based methods [6–11], machine learning (ML) based algorithms [12,13] and deep learning (DL) based extraction models [4,14–16] have been developed to automatically extract information from technical standards. Furthermore, ontologies-driven techniques [17–19] are introduced to improve the extraction accuracy of entities and relations. However, the above approaches can only extract simple knowledge in regulation documents and are not capable of complex regulation knowledge [20]. To interpret the complex knowledge with multiple entities and relations, several studies combined NLP-based parsing techniques (e.g., Part of Speech, Parsing Tree) with conventional rule-based or ML-based approaches and successfully managed to extract some complex regulatory knowledge from the target standards [21,22]. Nevertheless, these studies mainly focus on quantitive requirements in technical standards, the qualitative or management requirements are ignored. In addition, existing studies mainly rely on regulation-oriented ontologies to identify clauses. Due to the differences in the content, extensive labour is still required on constructing ontologies if different regulation documents are to be processed. Moreover, existing knowledge mining approaches proposed in the AEC domain represent the extracted knowledge in poorly compatible formats, making it difficult to be accumulated, managed and reused for other tasks. Hence, there are still some gaps in the AEC domain for an automated complex knowledge mining approach, which can transfer textual regulation documents to serviceable domain RDF graphs.

To fill the research gap, this research proposes an autonomous knowledge mining framework for regulatory documents in the AEC domain, which utilises natural language processing, transfer learning, and graph modelling to achieve the fully automated knowledge transformation from textual regulation documents to graph-based representation. In the proposed framework, the authors develop a BERT-based extractor to recognise the clauses in different regulation documents through transfer learning. An NLP-based constituent extractor is developed with linguistic knowledge to parse complex logical relations in the clauses and mine the knowledge constituents, including entities, relations, constraints, and attributes. Finally, all extracted regulatory constituents are automatically integrated as RDF triples and assembled as a semantic web-based knowledge model by a knowledge modelling engine. This research makes the following three major contributions. Firstly, the proposed framework achieves fully automated knowledge mining for regulation documents, which greatly improves the efficiency of regulatory knowledge transformation. Secondly, the complex knowledge extraction of regulation documents is achieved through linguistics-supported NLP techniques. Thirdly, this framework automatically assembles the extracted regulatory knowledge into an operational graph representation, which can be directly used in downstream applications such as knowledge query and reasoning, holistic decision-making, etc.

The rest of this paper is organised as follows: Section 2 lists the related studies and highlights the research gaps. Section 3 presents the overarching framework of autonomous knowledge mining and explains the four core components of the developed framework. Section 4 explains the specific process of domain dataset creation, fine-tuning of the BERT model, and the development of constituent extractor and knowledge modelling engine. Section 5 demonstrates the validations of the clause extractor and constituent extractor in the proposed framework. A case study of the practical engineering standard (ISO 19650-2) is also conducted to validate the practical performance of the whole framework. The limitations and contributions of this research are illustrated in Section 6 and Section 7 respectively.

2. Related work

Knowledge mining is a technique that utilises sophisticated methods, such as natural language processing (NLP), language modelling, and machine learning (ML), to extract valuable information from structured data (relational databases, XML) and unstructured data (text, documents, images) sources [23]. The objective is to create a structured representation that allows researchers to better understand data and use it to build applications [24]. A consensus has been reached that the two critical components of knowledge mining for regulation documents are knowledge extraction and knowledge modelling [25,26]. Knowledge extraction aims to recognise and extract valuable information (e.g., entities, relations, attributes, and constraints) from textual clauses [27]. Knowledge modelling concentrates on reorganising extracted information and transforming it into usable knowledge representation [28]. The review of related studies mainly focuses on regulatory knowledge extraction and domain knowledge representation in the AEC domain.

2.1. Regulatory knowledge extraction in the AEC domain

In the past two decades, great progress has been made in automatic knowledge extraction for regulation documents. Several semi-automated rule-based mapping approaches that capture required information through predefined rules or patterns have been developed to extract target knowledge from regulatory documents. For example, Feijo and Krause [7] combined hypertext with graphs and sentences in mathematical logic to navigate regulatory documents. Hjelseth and Nisbet [8] proposed a semantic mark-up (RASE) methodology to capture normative constraints in target construction documents. Beach et al. [9] developed a rule-based semantic approach to extract regulations from textual documents by annotating regulatory documents. Li et al. [11] proposed an information extraction method based on chunk-based rules for utility regulations. Lau and Law [10] developed a shallow parser that can consolidate different formats of regulations into extensible mark-up language (XML) to semi-automate the rule translational process. These conventional hard-code rule-based methods can only be used to extract some regulatory knowledge with simple logic and fixed format. To improve extraction performance, some research proposed ontology-driven approaches, which utilise the domain or application ontology to aid in extracting related semantic information from target documents [29]. Yurchyshyna and Zarli [18] conducted research in which norms were extracted from electronic regulations and structured as SPARQL queries using the industry foundation classes (IFC) ontology. Anantharangachar et al. [19] proposed an information extraction approach by mapping the entities and relations pre-defined in the domain ontology. Following a similar workflow, Zhou and EI-Gohary [17] proposed a rule-based ontology-enhanced information extraction method for extracting building energy requirements from energy conservation codes and then formatted the extracted information into a B-Prolog representation.

To improve generalisation within the construction domain, some machine learning-based methods are introduced for automatic regulatory knowledge extraction. Some ML algorithms (e.g., Decision Trees, Support Vector Machines, Hidden Markov Models, Conditional Random Fields) and artificial neural networks (ANN) are adopted to extract information (clauses, entities, and relations) from engineering documents such as project reports, design code, etc [30]. For clause classification, Salama and EI-Gohary [31] proposed a machine learning-based text classification algorithm, which utilized a deontic model to classify clauses and subclauses into different topics. To further improve the performance, Zhou and EI-Gohary [32] developed an ontology-based clause classification algorithm based on the deontic model, which can leverage the semantic features of the text. In addition, Zhou and EI-Gohary [33] also introduced a machine learning (ML)-based algorithm for classifying clauses based on the topic hierarchy in environmental regulatory documents. Zhu and Li [4] proposed an LSTM-based neural network model that can automatically recognise the qualitative rule sentences from engineering standards with an accuracy of 98%. In terms of entities and relations, Wang and El-Gohary [10] proposed a hybrid bi-directional long and short-term memory (BiLSTM) and convolutional neural network (CNN) model, which can automatically identify entities in building safety regulations. Moreover, an attention-based convolutional neural network model that can identify and classify relations mentioned in regulation documents was proposed by them in 2022 [14]. Liu and El-Gohary proposed an automated information extraction method for bridge inspection reports based on CRFs [12]. Zhang and El-Gohary [16] proposed an LSTM-based method to generate semantically enriched building-code sentences, which achieves an accuracy of 87% on their domain dataset. Current ML-based approaches applied in the AEC domain demonstrate excellent generalisation in automatic information extraction. After being trained on several representative regulation documents, they can achieve ideal results on other different regulation documents [15]. However, due to the inability of parsing logic, these approaches can only handle simple knowledge extraction tasks (e.g., named entity identification, relationship extraction and clause classification) for regulation documents, and it is still not capable to interpret the complex relations in regulatory knowledge.

Given the advantages of natural language processing in parsing complex logic and relationships, several recent studies have attempted to combine NLP techniques (e.g., Part of Speech, Parsing Tree, etc.) with conventional approaches to interpret regulatory knowledge. Zhang and EI-Gohary [34] proposed a semantic NLP-based approach named Regex-E, which annotates text in the building codes with the help of POS tags and domain ontologies and uses semantic mapping to transform single-requirement to logical clauses. Based on this research, Zhang and EI-Gohary [13] introduced a phrase structure grammar (PSG)-based information extraction method, which can reduce the number of needed patterns for semantic mapping. For regulatory knowledge with multiple entities and relations, Zhou et al. [20] developed an automated rule extraction method based on syntax trees, where clauses are firstly distinguished from descriptions by ML algorithm, then predefined semantic elements (e.g., prop, cmp, Rprop, ARprop, etc.) and a set of context-free grammars (CFGs) are utilized to transfer textual regulatory rules into pseudocode formats. Using the same semantic parsing method, Zheng et al. [35] proposed a knowledge-informed framework, which identified clauses with the help of predefined classes and properties in domain ontology, then these clauses were parsed and transformed into SPARQL queries by pattern-matching rules. Xu and Cai [36] proposed an ontology and rule-based NLP framework to automate the interpretation of utility regulations into deontic logic (DL) clauses, where pattern-matching rules are used for information extraction; pre-learned model and domain-specific hand-crafted mapping methods were also adopted for semantic alignment between rules and ontology.

2.2. Knowledge representations in the AEC domain

As an effective cross-disciplinary approach to organise and utilise dispersed knowledge, knowledge modelling-related research has attracted significant attention since the 1980s and has been intensively studied by many researchers [28]. So far, there are various widely used forms of knowledge representation, such as logical representation, semantic network representation, frames representation [37], and production rules. Among all these knowledge representation methods, semantic network representation is one of the most studied and effective methods [28]. A semantic network is a graphic notation for representing knowledge in patterns of interconnected nodes and arcs, which can represent knowledge and support automated systems for reasoning about knowledge. For a specific domain or subject, there is a commonly used graph-based knowledge representation form called ontology, which represents a set of concepts/instances within a domain and the relationships between them [38]. So far, there are many published ontology-based knowledge representations in the AEC domain. For example, there is the long-standing IFC ontology that has been available in the Web Ontology Language since 2016 [39]. Furthermore, there exists the combination of Linked Building Data (LBD) ontologies [6], which are diverse combinations of building topology ontology (BOT) [40], building element ontology (BEO) [41], Building Product Ontology (BPO) [42], and MEP ontology, etc. When considering asset management, ontologies, such as SAREF4BLDG, the Damage Topology Ontology (DOT) [43], and Real Estate Core (REC), can be of additional use. For ecosystems, Brick [44], the most predominant ontology, is an open-source effort to standardize semantic descriptions of the physical, logical, and virtual assets in buildings and the relationships between them. The Flow System Ontology (FSO) [45] allows users to define systems, components, and connections in the HVAC system. It can be used to define and compute flows through a system and design its dimensions accordingly to acquire an optimised HVAC system. In addition to the typical ontologies mentioned above, many other ontologies can be easily aligned and combined with the above ontologies, such as QUDT, SSN/SOSA, O&M, Time, etc [46]. Apart from domain ontologies, there is another semantic network-based knowledge representation, knowledge graph, which has become increasingly prevalent in recent years. Zhou et al. [47] introduced a method to collect and formalize building codes and transform them into a knowledge graph representation by connecting building clauses via their indexing numbers. Jiang et al. [48] proposed a graph-based knowledge model to support the representation of building codes in a more logical manner. Zhang and Ashuri [49] designed a systematic methodology to generate a knowledge graph of the design social network to examine the relationship between its characteristics and the production performance of designers. Compared with other forms of knowledge representation, semantic network-based knowledge representation (e.g., ontology and knowledge graph) provides a more powerful and flexible semantic representation, which makes it more desirable to represent knowledge with complex logic and relations. Semantic inference and retrieval can be easily implemented to facilitate querying the relevant knowledge. With the above advantages, graph-based knowledge models are currently the most widely used knowledge representation in the AEC domain [24]. Many downstream applications driven by this knowledge have been developed and implemented, such as automatic compliance checking, regulatory knowledge inference, regulatory knowledge fusion, multi-objective decision-making, etc. Although many graph-based knowledge models (ontologies and knowledge graphs) have been developed in the AEC domain, the work of knowledge modelling mainly relies on domain experts. Some semi-automated knowledge modelling methods [47,48] still require the assistance of conceptual ontologies that are manually predefined.

2.3. Research gaps

Despite great efforts in the interpretation of regulatory knowledge in the AEC domain, there are still some limitations in automatic knowledge extraction and knowledge representation.

First, before knowledge extraction, text classification is normally required to filter out irrelevant text. Existing approaches mainly rely on classes and properties defined in domain ontologies to separate clauses and descriptions [17,19,32,34–36]. Although Zhou and EI-Gohary [33] and Salama and EI-Gohary [31] have introduced some machine learning-based algorithms, a predefined domain deontic model that contains similar components (concepts, relations, and axioms) to ontology is still required. These customized semantic models make the above approaches only capable of interpreting the knowledge in a particular regulation. Intensive labour is still required in constructing ontologies if different regulation documents are to be processed.

Second, the existing studies mainly focus on interpreting quantitive requirements in technical regulations (e.g., International Building Code (IBC), GB 50016–2014), while the qualitative requirements are usually ignored. Compared with qualitative or management requirements, technical requirements are simpler in the representation of logic and relations. For example, "the protection layer shall use noncombustible material and the thickness of protection layer shall not be less than 10mm." and "the appointing party shall establish the requirements that tendering organizations shall meet within their tender response" (ISO 19650). As covering most of the simple logical relations (e.g., "shall use", "not less than"), the 7 semantic element labels defined by Zhou [20] and Zheng [35] performed quite well on the extraction of complex knowledge in technical regulations. However, this simplified semantic label set fails to fulfil the flexible and complex logical scenarios in non-technical regulations. Furthermore, all existing studies rely on the syntax tree and POS tags to parse the structure of clauses, which is limited in the interpretation of complex logic and can be improved by dependency parsing approaches.

Third, although the RDF-based graph shows great preponderance in semantic inference and retrieval and has widely been used in the AEC domain, existing interpretation approaches mainly convert textual clauses into pseudocode [20], plain description logic [17,36] or SPARQL queries [18,35], which are flawed in compatibility and knowledge expressed in these formats are is difficult to accumulate, integrate, and be reused by other applications.

3. The overarching framework for autonomous knowledge mining and modelling (AKMM)

To fill the abovementioned gaps, an autonomous knowledge mining framework for the textual regulation documents is proposed, which integrates multiple advanced techniques in NLP, linguistics, and transfer learning. The proposed framework innovatively adopts fine-tuned BERT model to identify clauses in regulation documents, which eliminates the extensive labour involved in constructing ontologies and enables automated extraction capacity for knowledge in different kinds of regulations. The injection of linguistic knowledge and the adoption of dependency parsing further improve the interpretation performance for clauses with complex logic and multiple relations. The extracted knowledge is automatically modelled as an RDF graph, which can be easily inferred, queried and applied to different kinds of downstream applications. Fig. 1 illustrates the proposed autonomous regulatory knowledge mining framework, which consists of the following four main parts:

1. **Domain dataset establishment.** A domain regulation dataset with samples manually selected from several engineering standards (i.e., ISO 14001, IBC 2015) are established in this part. The data augmentation technique and Delphi method [50] are applied to further enhance the quality and reliability of this domain dataset.
2. **Transfer learning-based clause extraction.** In this part, a clause extractor based on a pre-trained BERT base model is developed. This extractor is then fine-tuned with the domain dataset (generated in Part 1) to acquire the ability to precisely identify and extract clauses from different regulation documents. The fine-tuning process follows the general neural network training workflow, including pre-processing, tokenisation, data packaging, training, and testing.
3. **NLP-based extraction of constituents.** In this part, an NLP-based constituent extraction engine is developed based on the Seven Clause theory [51] and syntactic parsing. This constituent extractor consists of a clause classifier, two complex clause processors, and a constituent extractor. The clause classifier categorises raw clauses (extracted in Part 2) into coordinate, compound, and simple clauses via joint mapping of part-of-speech (POS) tags and dependency parsing (DP) labels. The coordinate and compound clauses are then further simplified as simple clauses by two clause processors, respectively. All the simple clauses are finally processed by the constituent extractor, which is formed by a tuple extraction algorithm and an attribute extraction algorithm. The tuple extraction algorithm is developed based on the rule-based mapping approach and the Seven Clause theory, which transfers clauses into quintuples. The attribute extraction algorithm extracts modifiers of noun chunks based on the phrase structure parsing method and stores extracted information as attribute matrixes.
4. **Automated knowledge modelling.** The extracted tuples and attributes are classified into four categories of constituents (entities, relations, attributes, and constraints). The knowledge integration engine developed based on an external Python library (RDFLib) reorganises the regulatory constituents as RDF triples/reifications and assembles them as a graph-based knowledge model.

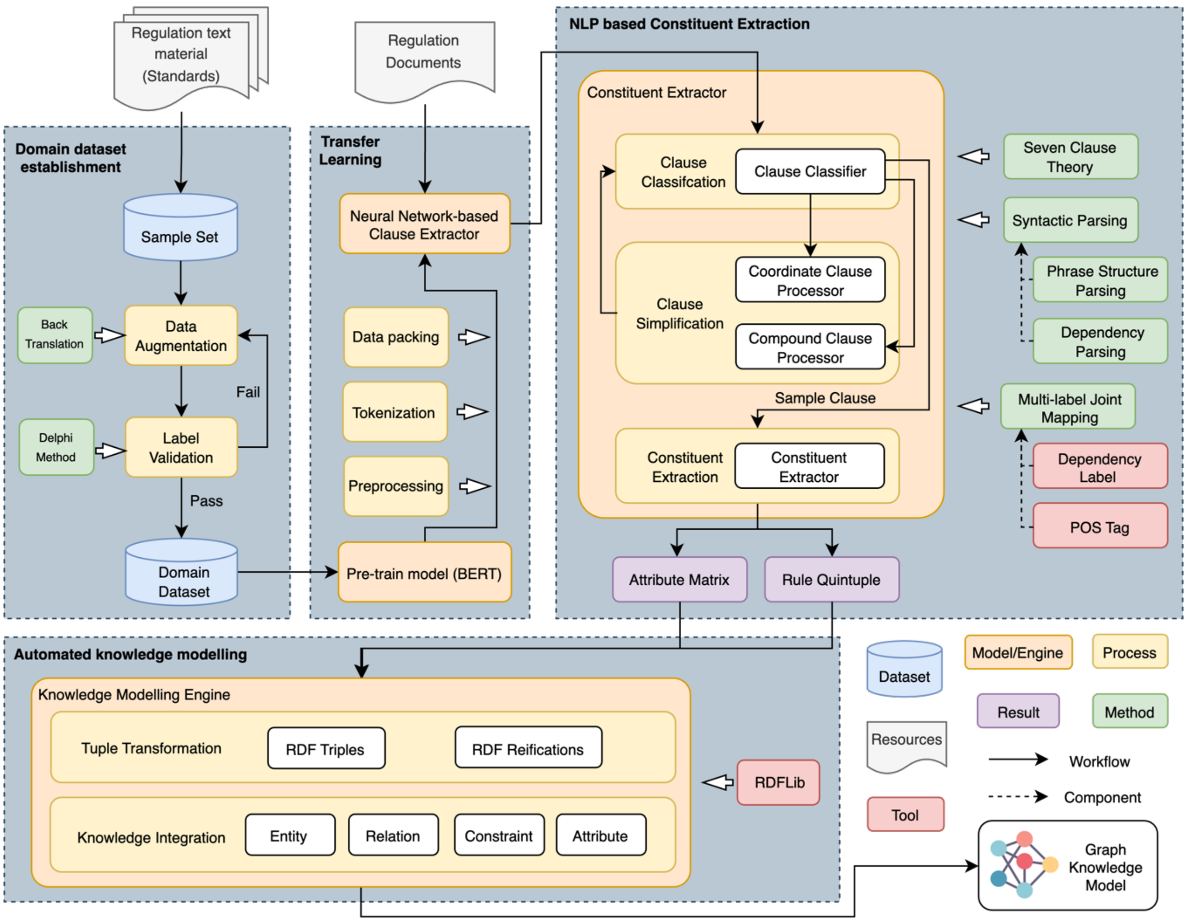


Fig. 1. The proposed autonomous knowledge mining and graph modelling framework

4. AKMM framework development and implementation

This section elaborates on the specific development process of the proposed autonomous knowledge mining and modelling framework. To better illustrate the proposed framework, some regulation texts from practical engineering standards, such as ISO 19650 and IBC 2015, are selected as examples to explain the fundamental principles and demonstrate the process specifically.

4.1. Dataset establishment

Currently, no public datasets are available used for the training of clause extraction. Hence, the authors manually created a domain dataset, which is formed by clause samples and description samples extracted from regulation documents. To ensure the representativeness of clauses and descriptions, various engineering standard files released by different institutions were selected as data sources. Furthermore, some samples from engineering-related standards (e.g., ISO 9001) were added to enhance the generalisation of the neural network model. As shown in Table 1, 826 samples were manually extracted from the selected standards. Generally, there are more clauses than descriptions in a regulation document, so these 826 samples comprise 573 clause samples and 253 description samples.

Table 1. The composition of the original domain dataset

Standard Code	Description	Published by	No. of Clauses extracted
ISO 9001	Quality management systems Requirements	ISO[a]	216
ISO 14001	Environmental management systems -Requirements with guidance for use	ISO	169
ISO 50001	Energy management systems - Requirements with guidance for use	ISO	153
ISO 19650-1	Organization and digitization of information about buildings and civil engineering works, including building information modeling (BIM)	ISO	199
2015 IBC	International Building Code	ICC[b]	48
GB/T 51212	Unified standard for building information modeling	MOHURD[c]	41

[a]International Organization for Standardization, [b]ICC - International Code Council, [c]MOHURD – Ministry of Housing and Urban-Rural Development of China

It is well-known that the performance of the neural network model significantly relies on the quality and the size of the data. Normally, a number of 1000 samples of each category are considered sufficient to acquire good performance when training neural network models for classification tasks. Hence, the original domain dataset needs to address the problem of data shifting and deficient samples. Furthermore, the clause and description samples in the original dataset are manually labelled, which is subjective and error-prone. To address the above problems, the authors first expanded and balanced the samples in the original dataset through data augmentation, then applied the Delphi method to remove the subjectivity and uncertainty in sample labelling.

Data augmentation is a technique for artificially extending a training dataset by making a limited amount of data produce more equivalent data [52]. There are currently several text-specific data augmentation methods, such as lexical substitution, back translation, and regular expressions-based transformation [53]. Back translation is a sentence-level data augmentation method, which generates more variants by running reverse translation in a different language to augment the unlabelled clause samples. Considering that the samples are all sentences and used for the training of sentence classification, back translation is the most preferable. Thus, the authors used the back translation approach to augment the existing samples and obtain a larger dataset with 1000 clause samples and 1000 description samples. French and Mandarin are chosen as intermediate languages. The specific procedure for back translation is shown in Fig. 2 and illustrated as follows:

1) Count the number of positive and negative samples to be augmented.
2) Randomly select a corresponding number of positive and negative samples in the dataset according to the statistical results.
3) The selected samples are translated into French and Mandarin one by one with the help of a third-party translator (Google translation). Then, another translator (DeepL) translates them back into English to form the translated samples.

4) The newly translated samples are compared with the original samples, and if their expressions are different, the translated samples are added to the dataset as augmented samples.

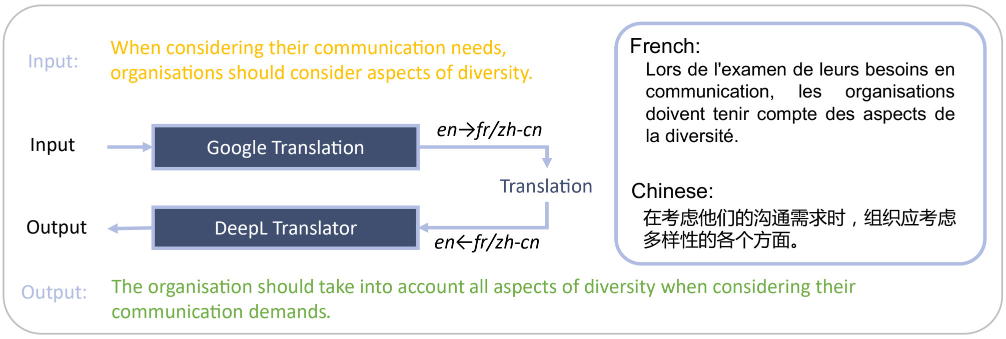


Fig. 2. The procedure of Back Translation with a specific example

The Delphi method [50] is a feedback anonymous correspondence method that can obtain relatively objective information, opinions, and insights through several experts' independent and repeated subjective judgments. The general process is that after obtaining experts' opinions on the issue to be predicted, they are collated, summarised, counted, and then anonymously fed back to each expert, consulted, pooled, and fed back again until a consensus is obtained. In the proposed framework, the authors follow the Delphi method and bring together five experts with an excellent understanding of engineering standards to validate the labels in the dataset. The specific validation process is shown in Fig. 3 and illustrated as follows:

1) All the sample and label data undergo the first round of expert group validation.
2) The research group collates and tallies the validation results from the expert group. For samples where all experts agree, they can pass the validation directly. If more than half of the experts agree on the sample, the research group modifies the sample appropriately based on the experts' opinions and then validates it in the next round. Suppose the sample passes the validation by only a few experts. A new sample of the same label type is generated by data augmentation to replace the original sample and then is validated in the next round.
3) All adjusted samples are subjected to the above two steps of validation again until all experts come to an agreement.

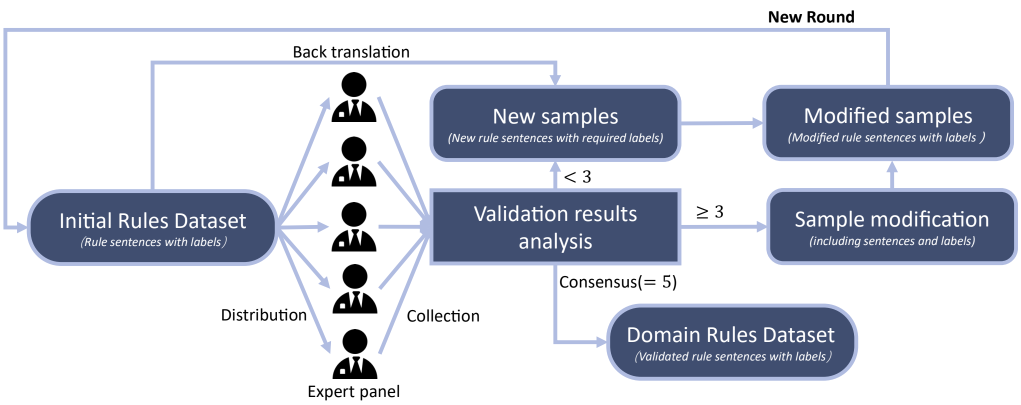


Fig. 3. The procedure of the Delphi method for sample validation

After four rounds of validation and modification, the five experts eventually agreed on all the dataset sample labels. With the help of data augmentation and the Delphi method, the domain dataset with balanced samples and validated labels was established.

4.2. Transfer learning-based clause extraction

Deep learning is a method for learning representations of information based on an artificial neural network architecture, which has achieved astonishing performance compared to conventional machine learning algorithms and has been widely applied in computer vision, speech recognition, machine translation, etc. As a data-driven approach, training a deep learning model requires large amounts of training data, which is time-consuming and labour-intensive [54]. To reduce the training cost, the concept of transfer learning is introduced, which reuses the common knowledge that has already been learned instead of starting from scratch to save the resources and time for training. Generally, the transfer learning process can be divided into two steps: pre-training and fine-tuning [55]. The pre-training phase aims to generate models that contain reusable knowledge, i.e., pre-trained models. The fine-tuning phase involves designing and adding fine-tuned layers to the pre-trained model based on specific task requirements. In the proposed framework, a pre-trained model that has already learnt universal language representations is fine-tuned to capture the semantic features of clauses and then utilised as an extractor to mine clauses from regulation documents.

Extracting clauses from regulation documents is a binary classification task for sentences. Therefore, the authors select the *BertForSequenceClassification* model as the pre-training model, which has a sequence classification head on top of a BERT model. BERT [56], namely Bidirectional Encoder Representations from Transformer, extracts feature information from the input sequence based on the bidirectional encoder provided by Transformer. With the help of its attention mechanism, the BERT model can capture long-distance dependencies and generate a feature vector for each sequence element (word) based on the contextual features of the input sequence. Therefore, it performs better than other deep neural network models (e.g., RNN, LSTM) in terms of efficiency and stability, with a wider range of applications [22]. Fig. 4 shows the structure of the selected pre-trained BERT base model.

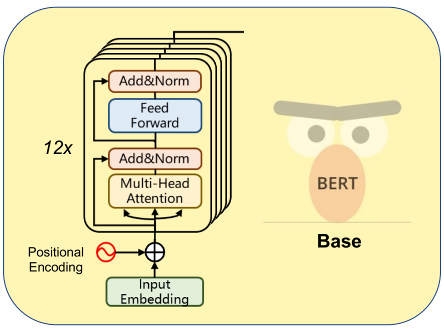


Fig. 4. The architecture of the pre-trained BERT base model

To fine-tune the pre-trained BERT model for clauses extraction, the following steps are implemented: pre-processing, tokenisation, data packing, training, and testing. During the fine-tuning process, only the parameters in the linear layer are updated and all the parameters in other layers are locked. Fig. 5 demonstrates the process of fine-tuning with a clause example.

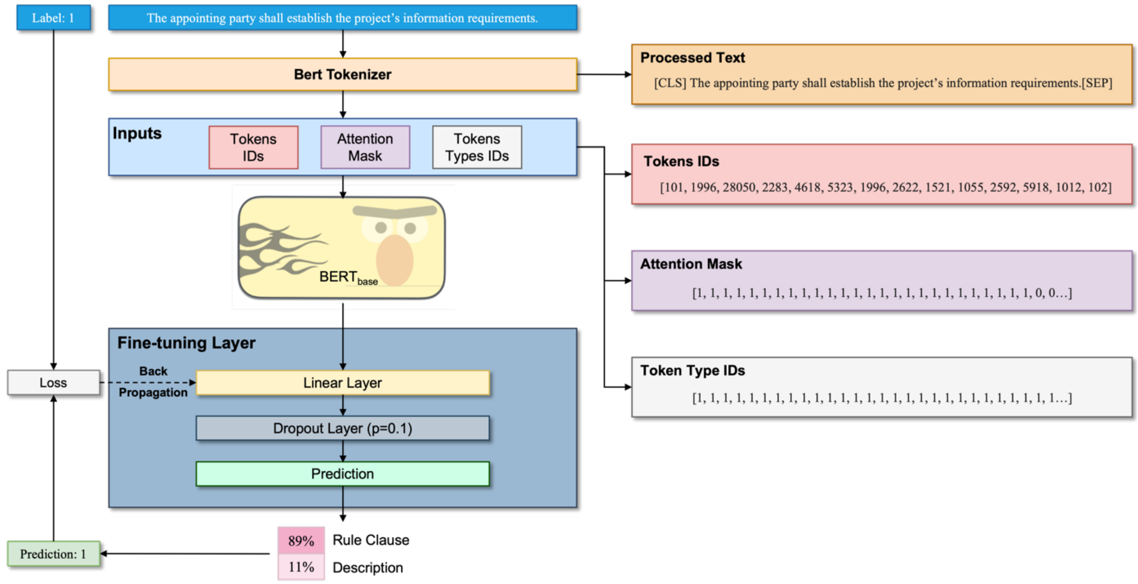


Fig. 5. The fine-tuning process in the proposed framework with a specific clause sample

4.2.1. Pre-processing and tokenisation

The samples in the domain dataset are all derived from regulation documents, which have a normative expression. Thus, conventional pre-processing methods, such as removing web links, stop words, and special characters, are unnecessary. However, some additional pre-processing steps are essential to make the outcoming model more effective, including:

1) Replace semicolons with dots. Many regular texts are written with semicolons, making it difficult to split and tokenise.
2) Remove all repeating whitespace and multiple subsequent spaces. This is a standard step in pre-processing texts.
3) Remove sentences that contain more than 200 words. Sentences that are this long usually have loads of words/numbers that do not make an actual sentence.
4) Count the dataset's characteristics, such as the number of samples, maximum sequence length, lexicon size, etc. These parameters will be used in the later steps.

Before loading data into the pre-trained model, two more steps are required in the pre-processing stage. One is inserting two special tags (CLS and SEP) to transfer the original sample to *[CLS]+sentence+[SEP]* format. The other is replacing the words in the original sentence with *[MASK]* and random words (rnd). After completing the pre-processing of the data, a pre-trained tokenizer (BertTokenizer) is applied to transform the textual samples into a sequence of IDs in the corpus. A tensor of token type ids and a tensor of attention mask are generated at the same time.

4.2.2. Data packing

The created dataset has a small size of 2000 samples. The training, validation, and test sets are divided with a ratio of 80%, 10% and 10% respectively. For the fine-tuning tasks, the batch size of 16 or 32 is recognised as appropriate. The authors set the batch size for fine-tuning as 32 through trial and error.

4.2.3. Training and testing

Several hyperparameters must be determined before training, including batch size (mentioned in Section 4.2.2), training epoch and learning rate. According to the minimal hyperparameter tuning strategy suggested by Devlin et al. [56], learning rates should stay between 2e-5 to 5e-5, and the number of training epochs should be 3 or 4. To avoid underfitting, the authors set the initial value of the training epoch as 5, which will be further adjusted according to the state of the model. Regarding the learning rate, the authors selected the minimum value (2e-5) and adopted an optimisation strategy for the learning rate named linear warmup to avoid overfitting and maintain the stability of the model. AdamW optimiser was adopted to calculate the gradient and update parameters in the network model. The testing process is the same as the training process, except that the gradient is set to zero during backpropagation. More details of testing results can be found in Section 5.

4.3. NLP-based extraction of constituents

Constituent extraction is essentially an information extraction for regulatory clauses. As one of the key techniques in NLP, syntactic parsing has been applied by many researchers to extract information from regulation documents. There are two imperative attributes of text syntactic: Part of Speech (POS) tags and Dependency Grammar (DG). Part of Speech tagging specifies the property or attribute of the word. Each word in a sentence is associated with a part of speech tags, such as nouns, verbs, adjectives, and adverbs. Dependency grammar is a segment of syntactic text analysis that determines the relationship among the words. This relationship is illustrated as a labelled arrow between the governor and the dependent. Fig. 6 shows a clause example labelled by POS tagging and DG. The meaning of some commonly used POS tags and dependency parsing labels are listed in Table 2. and Table 3. According to the representation of the syntactic structure, syntactic parsing can be divided into phrase structure parsing and dependency syntactic parsing. Phrase structure parsing identifies the phrase structures and their hierarchical syntactic relationships in the sentence based on POS tags. Dependency syntactic parsing (or dependency parsing) recognizes the interdependencies between words in the sentence based on dependency grammar. Phrase analysis is fast and accurate but can only identify fixed patterns of POS tag combinations. In other words, this method is quite effective in analysing phrases but has difficulties dealing with complex logic in long sequences. Dependency parsing represents the grammatical structure of sentences through various dependencies between words, allowing it to capture the complex logic in long sequences accurately. These two approaches are combined in the proposed framework to leverage the above-mentioned advantages.

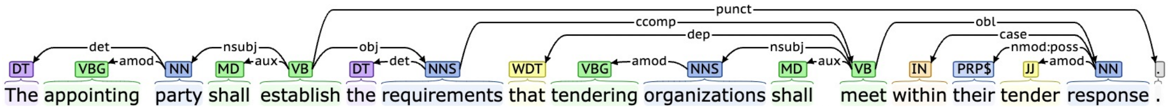


Fig. 6. An example of dependency parsing-based clause derivation

Table 2. Label and description of some common dependency parsing labels

Label	**Description**
acl	Clausal modifier of noun
advcl	Adverbial clause modifier
amod	Adjectival modifier
aux	Auxiliary
cc	Coordinating conjunction
ccomp	Clausal complement
compound	Compound modifier
det	Determiner
mark	Marker of an adverbial clause modifier or a clausal complement
nsubj	Nominal subject
dobj	Direct Object
pobj	Object of preposition

Table 3. Label and description of some common POS tags

Label	**Description**
CC	Coordinating conjunction
DT	Determiner
IN	Preposition/subordinating conjunction
JJ	Adjective
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
RB	Adverb
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund/present participle
VBN	Verb, past participle
VBP	Verb, present tense not 3rd person singular
VBZ	Verb, present tense with 3rd person singular
WDT	Wh-determiner (e.g., that, what, which)
WP$	Possessive wh-pronoun (e.g., whose)
WRB	Wh-adverb (e.g., how, where, why)

According to the Seven Clause theory, all sentences are composed of five components (subject (S), verb (V), object (O), complement (C), and adverbials (A) [57]) and can be classified into seven different types according to the grammatical function of their constituents [51], as illustrated in Table 4. Sentences containing only one S-V structure and where all the sentence components consist of only words or phrases are defined as simple clauses. Except for simple clauses, there are two other common types of clauses in the regulation documents: 1) coordinate clause, which contains conjunctions or multiple subject-predicate constructions, and 2) compound clause, some of whose components are expressed as an individual sentence rather than words or phrases. As mentioned in Section 2.2, the existing information extraction approaches for regulation documents can only deal with simple clauses but are not applicable to complex clauses (e.g., coordinate clauses and compound clauses). To remedy this deficiency, the authors apply a multi-label-based joint mapping approach to process complex clauses.

Table 4. Patterns and clause types (based on [51]*)*

Pattern	Clause type	Example
	Basic patterns	
SV_i	SV	A. Einstein died.
SV_cA	SVA	A. Einstein remained in Princeton.
SV_cC	SVC	A. Einstein is smart.
$SV_{mt}O$	SVO	A. Einstein has won the Nobel Prize.
$SV_{dt}O_iO$	SVOO	RSAS gave A. Einstein the Nobel Prize.
$SV_{ct}OA$	SVOA	The doorman showed A. Einstein to his office.
$SV_{ct}OC$	SVOC	A. Einstein declared the meeting open.
	Some extended patterns	
SVAA	SV	A. Einstein died in Princeton in 1955.
SV_iAA	SVA	A. Einstein remained in Princeton until his death.
SV_eCA	SVC	A. Einstein is a scientist of the 20^{th} century.
$SV_{mt}OA$	SVO	A. Einstein has won the Nobel Prize in 1921.
$ASV_{mt}O$	SVO	In 1921, A. Einstein has won the Nobel Prize.

S: Subject, V: Verb, C: Complement, O: Direct object, O_i: Indirect object, A: Adverbial, V_i: Intransitive verb, V_c: Copular verb, V_e: Extended-copular verb, V_{mt}: Monotransitive verb, V_{dt}: Ditransitive verb, V_{ct}: Complex-transitive verb

Based on the techniques and theory mentioned above, an NLP-based constituent extraction engine is developed to automate the information extraction of clauses. The architecture comprises a clause classifier, two clause processors, and a constituent extractor. Two external pipelines (DependencyParser and Tagger) provided by SpaCy are embedded in the parsing engine to generate dependency parsing (DP) labels and POS tags, respectively. The process of constituent extraction consists of the following three stages:

4.3.1. Clause classification

In this stage, raw clauses extracted by the clause extractor are first analysed by the parser, and then the parsing label of each word in the clause is generated. After this, the raw clauses are classified into coordinate clauses, compound clauses, and simple clauses by the clause classifier via joint mapping of specific dependency labels and POS tags. For example, the clause will be classified as an adverbial clause if it includes a word whose dependency label is *mark* and whose POS tag is IN. Table 5 presents some complex clause examples with the corresponding markers, correspondence between clause types, and the marker's parsing labels.

Table 5. Correspondence between clause types and marker's parsing labels

Examples with Markers	Clause Type	DP Label	POS Tag
The appointing party should understand **what** information is required concerning their asset(s) or project(s). (Clause 5.1 of ISO19650-1)	Compound (Object)	ccomp, mark	IN, WDT
If the review is successful, the lead appointed party shall authorize the information model and instruct each task team to submit their information. (Clause 5.7.2 of ISO 19650-2)	Compound (Adverbial)	advcl, mark	IN, WRB

The requirements should be expressed in such a way **that** they can be incorporated into project-related appointments. (Clause 5.5 of ISO19650-1)	Compound (Relative)	relcl	WDT, WRB, WP$
Exterior load-bearing walls **and** nonload-bearing walls shall be mass timber construction. (Clause 602.4 of IBC 2015)	Coordinate	cc, conj	CC

4.3.2. Clause simplification

After being classified by clause classifier, simple clauses are directly passed to the extraction phase. Coordinate clauses and compound clauses are simplified into several simple clauses by the proposed coordinate and compound clauses processors, respectively.

For coordinate clauses, the processor locates the juxtaposed elements based on dependency labels and POS tags. Since repeated contents are commonly omitted in coordinate clauses, the juxtaposed element's sentence parts (S, P, O, A, C) are determined according to its dependency label and POS tag. Then, the processor decomposes the coordinate clause into two individual clauses with the same clause pattern (Table 4). Taking a clause from ISO 19650-1 as an example, the original text is "The complexity of project information management functions should reflect the extent and complexity of project information". The juxtaposed elements are "the extent" and "the complexity", the objects in the clause. Therefore, the missing components (subject and predicate) need to be added when decomposing the sentence. The output sentences would be "The complexity of project information management functions should reflect the extent of project information" and "The complexity of project information management functions should reflect the complexity of project information".

For compound clauses, considering that subject clauses rarely appear in regulation documents, the compound clause processor mainly focuses on predicative clauses, object clauses, attributive clauses, and adverbial clauses. Table 5 lists some examples of the compound clause in the regulation documents. Similar to the coordinate clause processor, the compound clause processor also adopts joint mapping of dependency labels and POS tags to identify the sentence part of the subordinate clause. But the identified subordinate clauses are kept separately and labelled with corresponding sentence parts. The specific process of classification and simplification is revealed in Fig. 7 with a complex clause from ISO 19650.

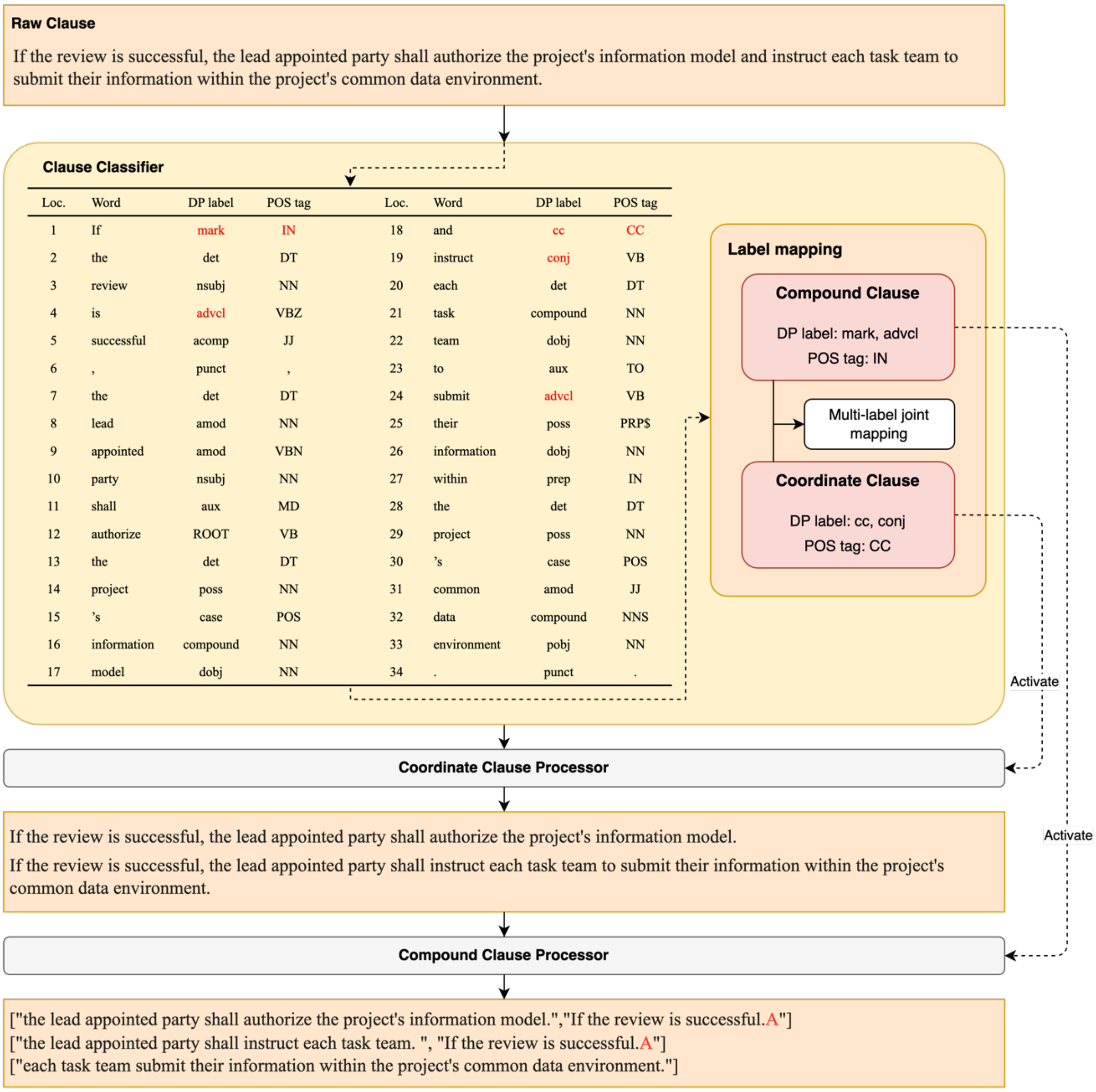


Fig. 7. The specific procedure of clause classification and simplification with a clause example from ISO 19650

The process of clause classification and simplification is cyclical. All simplified clauses are sent back to the original clauses set and reclassified until all the clauses are passed to the next stage as simple clauses.

4.3.3.Constituent extraction

The constituent extractor is developed based on syntactic parsing and the Seven Clause theory, which comprises a tuple extraction algorithm and an attribute extraction algorithm. The tuple extraction algorithm is developed based on dependency parsing, which aims to recognise constituents of the clauses by mapping specific tags or sequences of tags. According to the seven clauses theory, all simple clauses are formed by the following five components: subject (S), predicate (P), object (O), complement (C), and adverbial (A), or part of them. Therefore, a quintuple (S, P, O, V, C) is generated to store the corresponding constituents. The attribute extraction algorithm is developed based on phrase structure parsing to extract the attributes of entities mentioned in clauses. The extracted attributes and related entities are stored in an attribute matrix. Fig. 8 illustrates the constituent extraction process and results of the first simplified clause in Fig. 7.

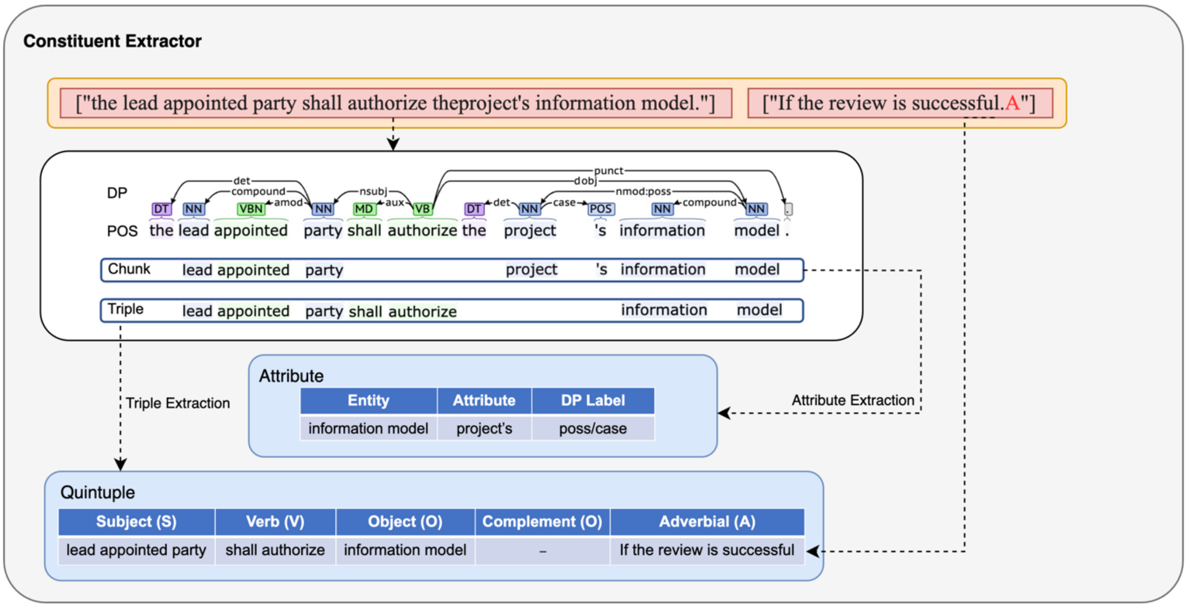


Fig. 8. The constituent extraction process on simplified clause

The specific extraction process of the constituent can be divided into the following five main steps:

1. Predicate extraction. Words parsed with ROOT (DP label) and verb-related POS tags (VB/VBD/VBG/VBN/VBP/VBZ) are extracted as relations. If this relation is preceded by words with labels like *neg* or *aux*, these words will be spliced together as the relation.
2. Subject extraction. Words that meet the following two conditions are extracted as subject: the DP label of its head word is ROOT, and the DP label of itself is nsubj or nsubjpass.
3. Object extraction. Words parsed with DP labels like dobj or dative, and whose head word's DP label is ROOT, are extracted as the object.
4. Adverbial extraction. There are several fixed patterns of label combination for the adverbials in clauses, such as prep + (pcomp+) dobj, advmod, agent+pobj, etc. The triple extraction algorithm extracts the adverbials by mapping these label sequences.
5. Attributes extraction. The attributes are extracted by the attribute extraction algorithm, which runs in parallel with the triplet extraction algorithm. This algorithm first recognises noun chunks in the clauses. Then extract the words with labels such as nummod, quantmod, poss, case, etc., as attributes of the central noun (entity). Furthermore, the complements extracted by the tuple extraction algorithm are also stored as attributes.

4.4. Automated graph modelling

The regulatory constituents extracted by the constituent extractor are all in tuple format, which cannot be directly used. To convert these tuples to a serviceable knowledge representation, a knowledge modelling engine is developed in the proposed framework to automatically assemble the separate tuples into a graph-based regulation knowledge model. The proposed knowledge modelling engine comprises two algorithms: the tuple transformation algorithm and the knowledge integration algorithm. The tuple transformation algorithm is developed based on the Seven Clause Theory, which aims to transfer the quintuples and attribute matrixes extracted by the constituent extractor into RDF triples (node, edge, node) or RDF reifications (statement, subject, predicate, object). The knowledge integration algorithm adopts an external Python library named RDFLib to assemble the generated RDF triples and RDF reifications into a graph-based regulatory knowledge model based on OWL and RDF schema.

In the implementation, all the extracted quintuples (S, P, O, C, A) were firstly classified into rule triples and rule quaternions by tuple transformation algorithm based on the Seven Clause theory. Quintuples composed of (S, P, O)/(S, P, A)/(S, P, C) were classified as rule triples, which can be directly used as RDF triples. Quintuples with (S, P, O, A)/(S, P, O, C) were classified as rule quaternions, and these quaternions required to be converted into RDF reifications (A/C, S, P, O) before integration. The attributes of the entity in the attribute matrixes were expressed as RDF triples with a fixed pattern (entity, should_be, attribute). After the tuple transformation, the generated RDF triples and RDF reifications were further assembled as a regulation graph by the knowledge integration algorithm, which automatically generated IRIs for each element in the tuples based on a predefined namespace and associates other triples and reifications according to this unique IRI. To integrate extracted knowledge, the authors defined four knowledge representation rules based on RDF syntax in the integration algorithm, which are: 1) subject (S)/object (O) → rdf:type → OWL.NamedIndividual; 2) predicate (P) → subPropertyOf → topObjectProperty; 3) subject (S)→ should_be → complement (C); 4) adverbials (A) → rdf:type → rdf:Statement. After being processed by the above two knowledge modelling algorithm, a graph-based knowledge model was established, which contains all the regulatory knowledge extracted from regulation documents and can be queried and visualised by external services. Fig. 9 illustrates the tuple transformation and knowledge integration process based on the previous extraction results.

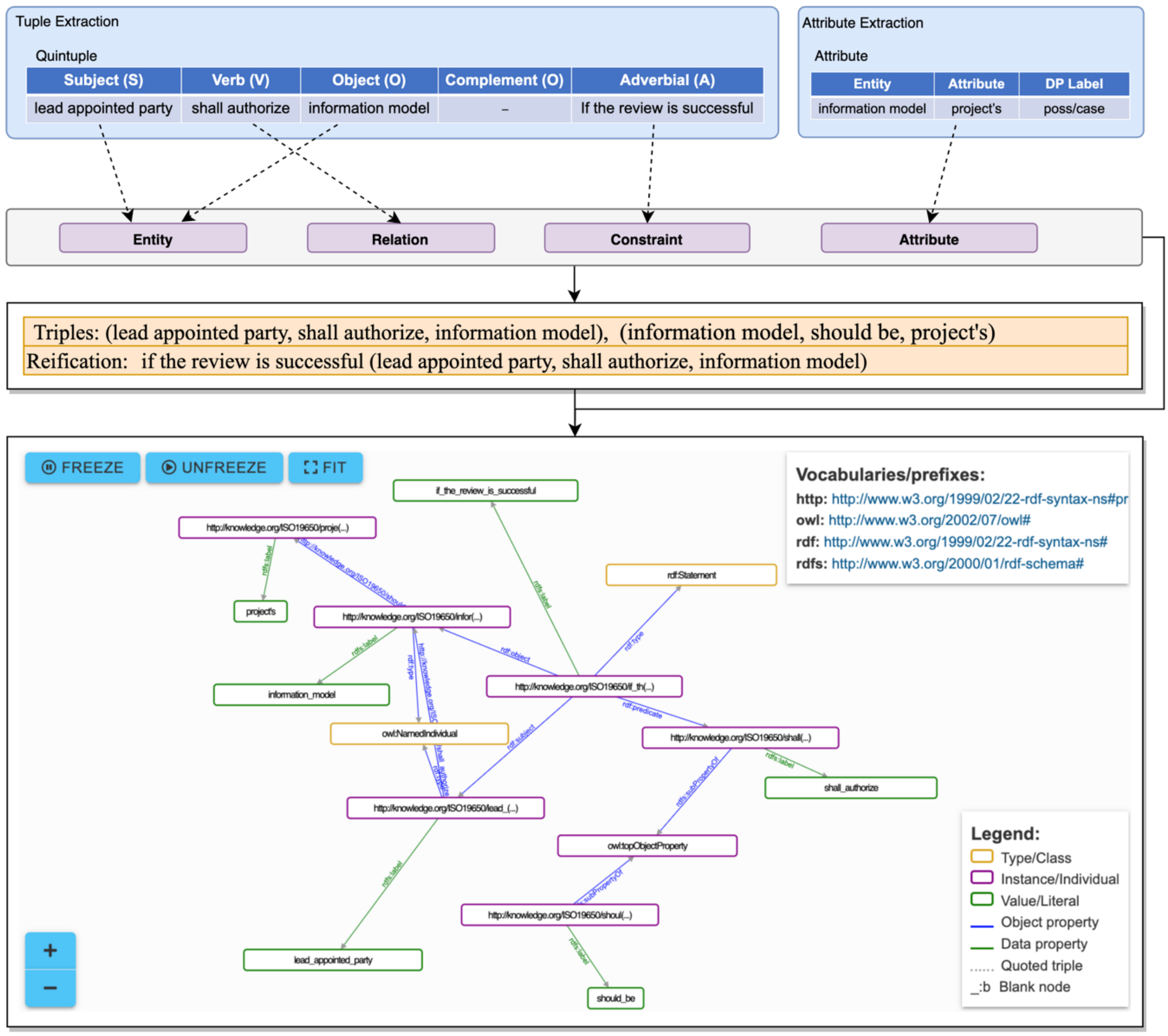


Fig. 9. The process of knowledge modelling based on the previous extraction results

5. Validation

The validation work of the proposed autonomous knowledge mining framework consists of two parts, a preliminary validation during the development phase and a practical validation based on a qualitative engineering standard. The preliminary validation evaluates the performance of specific functional modules in the proposed framework, and the practical validation assesses the effectiveness of the whole framework in practical scenarios.

5.1. Preliminary validation

The preliminary validation in the development phase mainly focuses on the results of transfer learning and the performance of the constituent extraction.

As a deep learning-based binary classification, the performance of transfer learning-based clause extraction is evaluated on the test set and measured by some commonly used indicators (accuracy, precision, recall, and F-measure), which are calculated using the following equations:

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN} \tag{1}$$

$$Precision\ (P) = \frac{TP}{FP + TP} \tag{2}$$

$$Recall\ (R) = \frac{TP}{FN + TP} \tag{3}$$

$$F1 = 2 * \frac{P * R}{P + R} \tag{4}$$

where TP, TN, FP and FN stand for the numbers of true positive, true negative, false positive and false negative, respectively.

Fig. 10 presents the variation of training loss with the training epoch. The pre-trained BERT model reaches its optimum after four epochs of fine-tuning, consistent with the parameter suggestions in Devlin's research [56]. The result of clause extraction is shown in the confusion matrix (Fig. 10). According to the split ratio, 400 samples are randomly selected from the domain dataset to form the test set, which includes 208 clauses and 192 description clauses. 94.4% precision and 98.1% recall are achieved in the clause extraction from the test set, indicating an F1 score of 0.96.

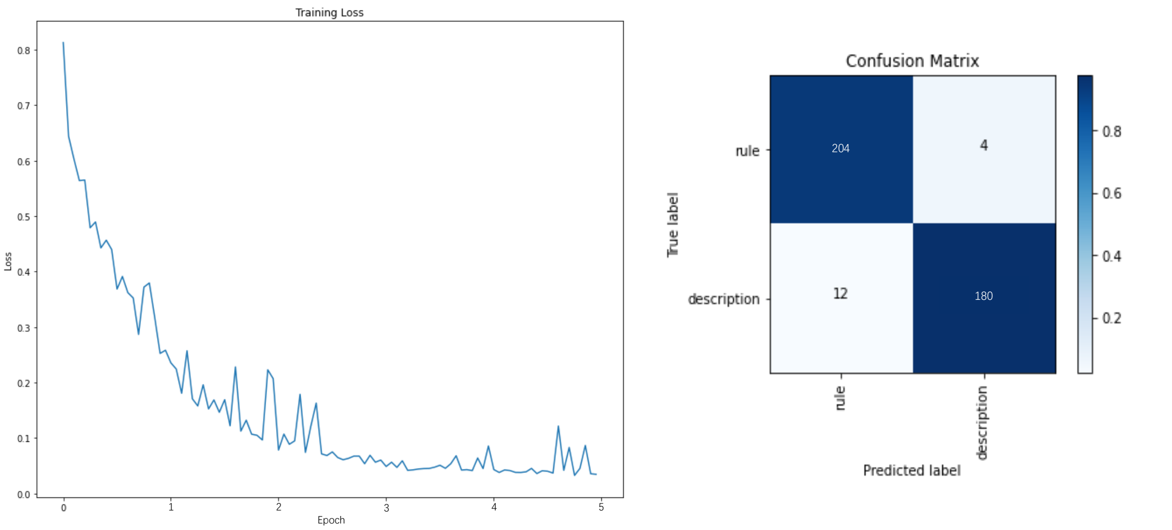


Fig. 10. The variation of training loss and confusion matrix of extraction result on the test set

To better evaluate the performance of clause extraction, the proposed clause extractor is compared with several best-performing text classification models, including RNN, LSTM and pre-trained BERT. Table 6 illustrates the details of the extraction results. The distribution of classification results and Receiver Operating Characteristic (ROC) curves of each model are presented in Figure 11.

Table 6. Comparison of the clause extraction results between different deep learning models

Model	Accuracy	Precision	Recall	F1-value
RNN	0.798	0.828	0.752	0.789
LSTM	0.893	0.911	0.887	0.899
Bi-LSTM	0.932	0.947	0.919	0.933

BERT-pre	0.814	0.810	0.820	0.815
BERT-ft	0.960	0.944	0.981	0.962

RNN: recurrent neural network, LSTM: Long short-term memory, Bi-LSTM: bidirectional LSTM, BERT-pre: pre-trained BERT model, BERT-ft: fine-tuned BERT model

The result in Table 6 demonstrates the proposed fine-tuned BERT extractor achieves the highest accuracy in clause extraction and its AUC value (Fig. 11) also proves this extractor outperforms the state-of-the-art deep learning models. Moreover, fine-tuning the pre-trained BERT model also saves significant training resources. In this experiment, the pre-trained BERT model achieved 96% accuracy on the test set after 4 epochs of training, while the conventional model (RNN and LSTM) reached its optimum after about 30 epochs of training.

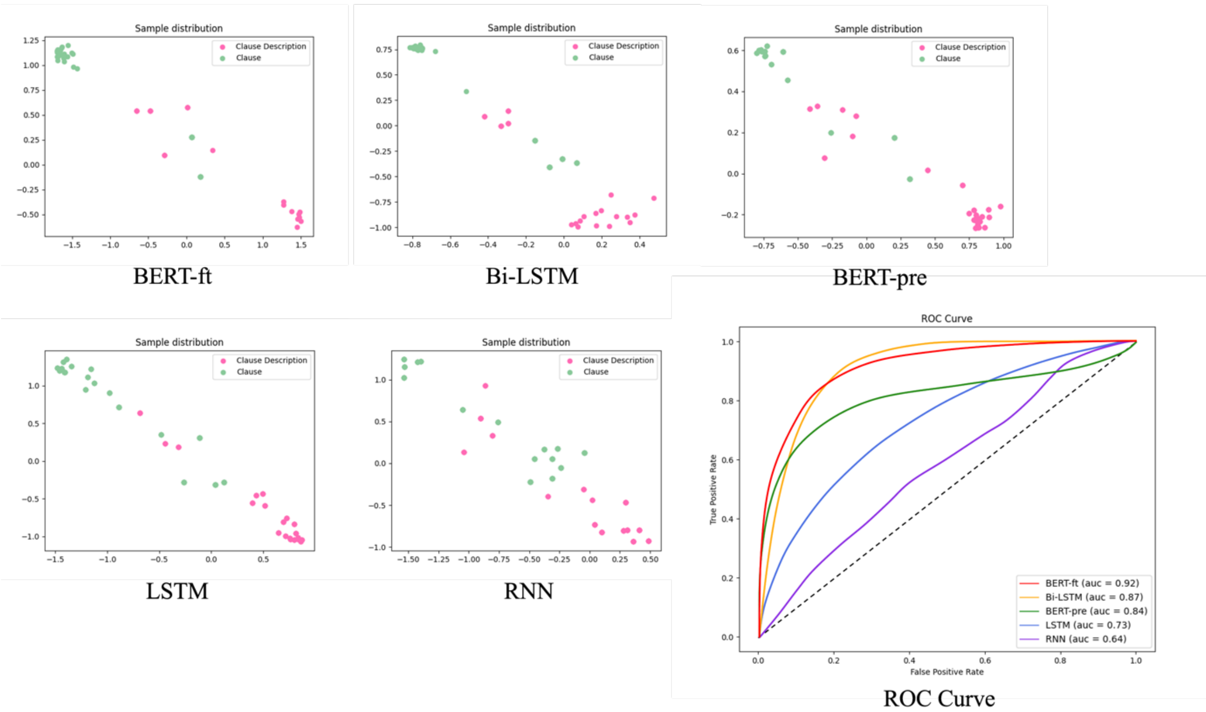


Fig. 11. Comparison of different models on ROC curve and classification result distribution.

In terms of constituent extraction, the proposed extraction engine is compared to two existing information extraction tools (OpenIE [58] and ClauseIE [57]) with random samples of 50 clauses selected from the domain dataset. These samples comprise both complex and simple clauses. Table 7 presents the total number of extractions for each method and Table 8 shows the detailed extraction comparison of different IE tools on the same clause.

Table 7. Comparison of extraction numbers on selected samples.

Clause samples	OpenIE	ClauseIE	Our extraction engine
50	72	123	176

Compared with the OpenIE and ClauseIE, the proposed extraction engine extracts similar information but provides higher granularity. The existing IE approaches stay at the noun chunk level, such as the lead appointed party, the information model, etc. The proposed engine can dig deeper and extract the attributes of the central noun (e.g., project's). This characteristic is essential for quantitative clauses because quantitative requirements are generally embedded in noun chunks. For example, "there shall be an approved alarm-initiating device at not more than 150-foot intervals." (Clause 415.5.2 of IBC 2015), where 150-foot is the quantitative requirement for intervals of the alarm-initiating device. The construction of reifications (R1~R3) is another advantage over the other two approaches. Reifications can represent the conditions and constraints of the required actions better than an individual triplet (C1). In terms of extraction accuracy, 166 out of 176 triples are correctly extracted. Manual extraction has also been implemented to obtain the ground truth. The result of manual extractions is 191, which indicates the developed engine achieves 86.9% accuracy on constituent extraction. Based on the above-observed results, the proposed framework outperforms other existing approaches in clause extraction, constituent extraction and is superior in representing constituents.

Table 8. A comparison of the proposed framework and existing tools on information extraction for regulation clause

Clause: If the review is successful, the lead appointed party shall authorize the project's information model and instruct each task team to submit their information within the project's common data environment.
Triplets extracted by OpenIE:
O1: (the lead appointed party, shall authorize, information model) O2: (the lead appointed party, shall authorize, information model and instruct each task team to submit their information within project's common data environment)
Triplets extracted by ClausIE:
C1: (the review, is, successful) C2: (the lead appointed party, shall authorize, information model) C3: (the lead appointed party, instruct, each task team) C4: (the lead appointed party, submit, their information)
Triplets extracted by proposed framework:
T1: (lead appointed party, shall authorize, information model) T2: (information model, should be, project's) T3: (lead appointed party, shall instruct, task team) T4: (task team, submit, information) T5: (information, should be, their) T6: (common data environment, should be, project's) R1: if the review is successful (lead appointed party, shall authorize, information model) R2: if the review is successful (lead appointed party, shall instruct, task team) R3: within the project's common data environment (task team, submit, information)

5.2. Practical validation

Considering that the representativeness of the samples in the self-developed dataset also affects the performance of the proposed method, a validation with practical qualitative engineering standards is required to validate the whole knowledge mining and modelling process in practical application scenarios. ISO 19650 series standard is a typical regulation document with complex knowledge, which specifies requirements for information management within the context of the delivery phase of assets. Therefore, the authors selected Section 5.2 of ISO 19650-2 as a case study to validate the performance of regulatory knowledge extraction and presentation.

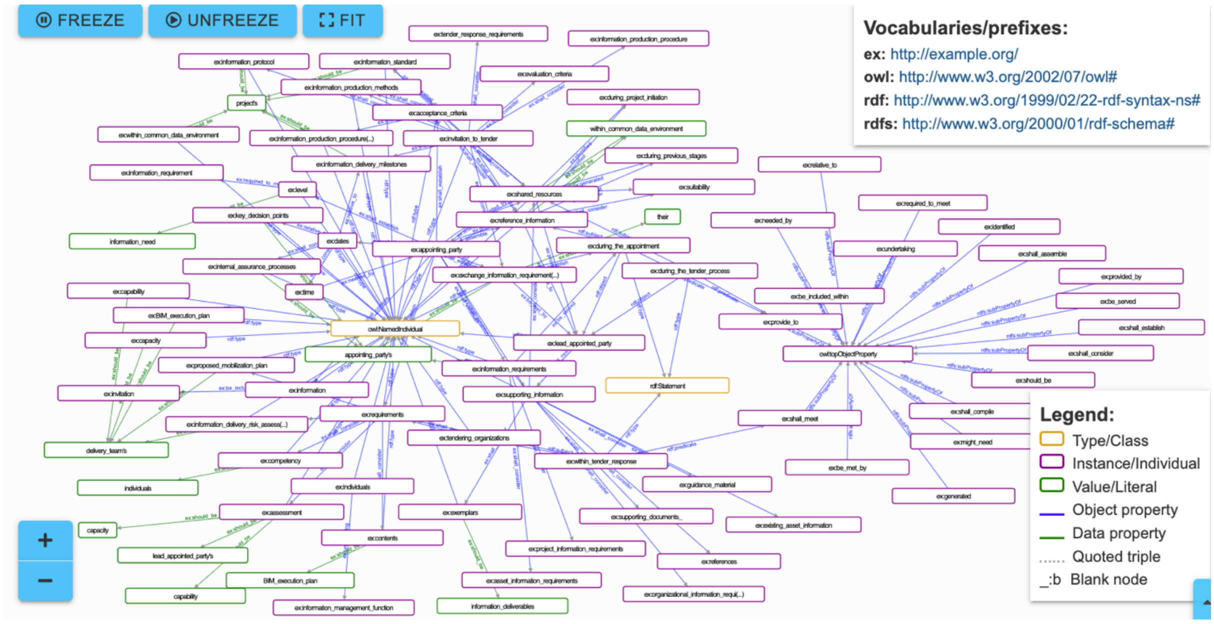


Fig. 12. An overall visualisation of the RDF graph generated by the proposed framework

Since the graph-based knowledge models represent knowledge in patterns of interconnected nodes and arcs [59], the validation of the generated RDF graph (Fig. 12) mainly concentrates on the instances and their relations extracted from regulation documents. To validate the generated RDF graph, the authors have invited three domain experts to manually construct an ontology based on the same content. The regulatory ontology development follows the processes in Ontology Development 101 [60]. The domain experts are all familiar with ISO 19650 series standards and have several years of ontology-related research experience. Hence, the domain ontology generated by domain experts can be treated as the golden standard, which contains all the essential regulatory knowledge. The results of alignment between the RDF graph and expert ontology can be considered as the performance of the proposed framework in the practical scenario. The metrics of the domain ontology constructed by experts and the RDF graph generated by the proposed framework are shown in Table 9.

Table 9. Ontology metrics of the generated ontology and expert ontology

	RDF graph	**Expert ontology**
Axiom	113	201
Logical axiom	80	119
Declaration axiom	0	82
Class	1	22
Object property	12	16
Individual	51	45
Annotation assertion	0	0

The validation process is divided into two parts: 1) element checking (whether the required instances and relations are defined), 2) connectivity checking (whether the instances are connected by correct relations). Intersection-over-Union (IoU) is used to measure the accuracy of the checking, which is calculated using the following equation:

$$IoU = \frac{Number\ of\ Overlap}{Number\ of\ Union}$$

5.2.1. Element checking

As indicated in Table 7, the proposed framework automatically extracted 51 instances and 18 object properties, while the experts defined 45 instances and 16 object properties. Fig. 13 presents the mapping result of these two sets of object properties. The RDF graph shares 14 of the same properties with the expert ontology. If the built-in object property, "should_be" (which is utilized to connect instances and their attributes), is ignored, the accuracy of object property mining achieves 73.7%. For example, the RDF graph and expert ontology have 36 common instances, which means the accuracy is 60%. However, some instances in different wording have similar meanings and refer to the same thing in the regulation documents (e.g., existing asset information and asset information). Taking these instances into consideration, the number of common instances increases to 41, indicating the accuracy of instances mining reaches 74.5%.

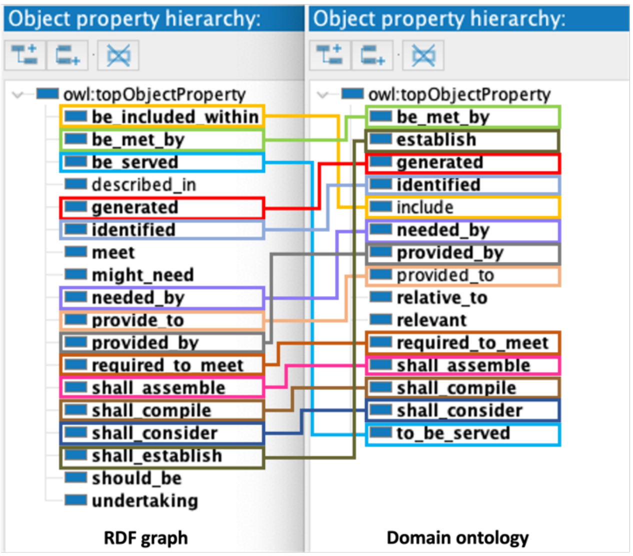
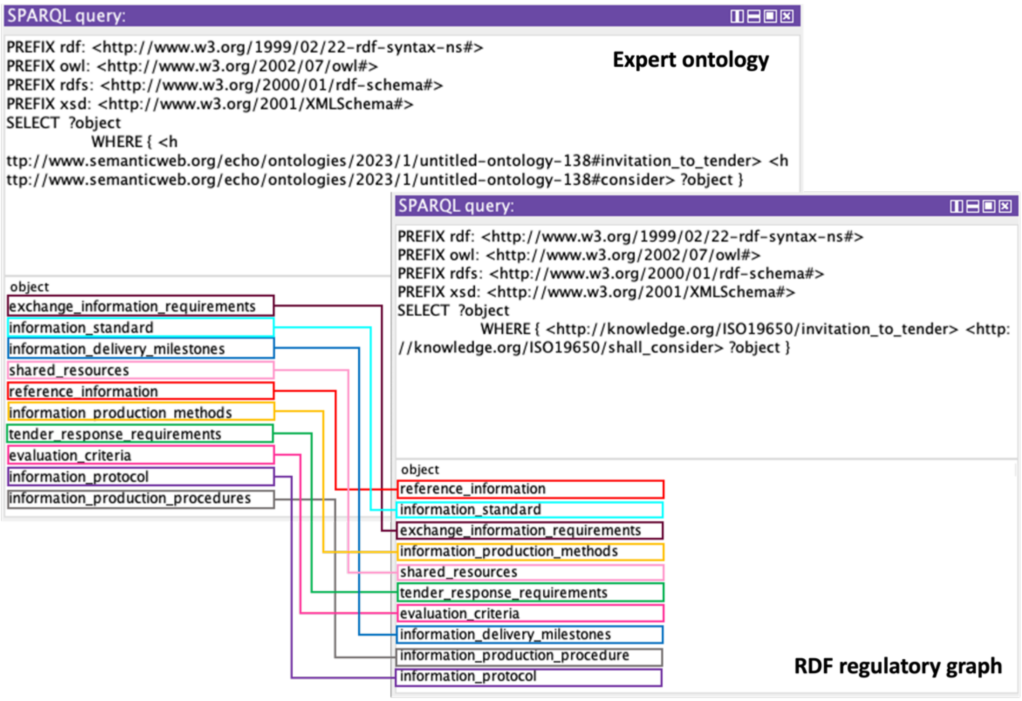


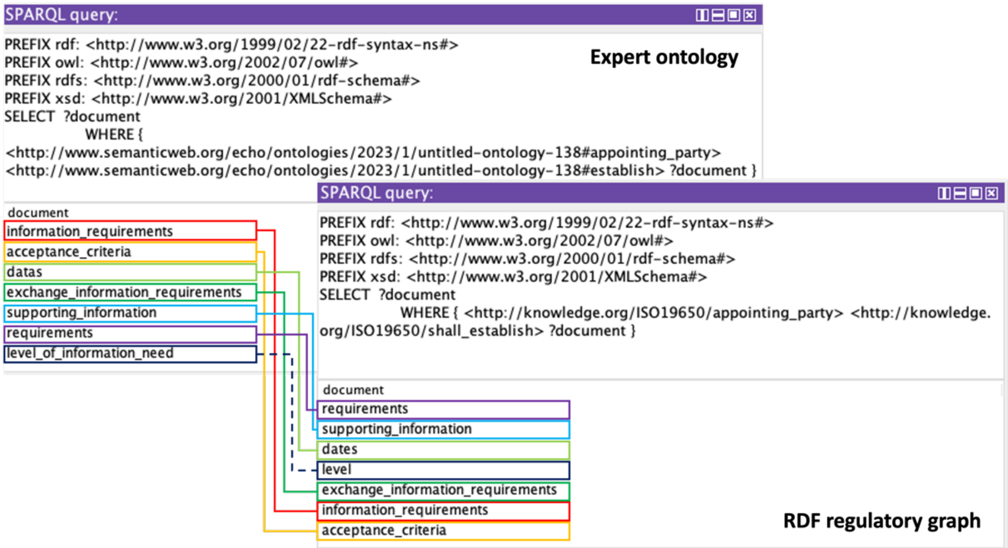
Fig. 13. The mapping of the object properties in the generated RDF graph and expert ontology

5.2.2. Connectivity checking

Since the connectivity between instances and properties is difficult to be directly validated, the authors apply SPARQL queries to verify the connectivity between graph elements. These queries are defined by domain experts based on the content of Section 5.2 of ISO 19650-2 and covers all the regulatory knowledge mention in this section. The connectivity of graph elements is checked via comparing the query results of the generated graph and expert ontology. Fig. 14 shows the mapping results of three examples from all queries.



(a) Mapping results for querying what document shall be considered in the invitation to tender stage.



(b) Mapping results for querying what document appointing party shall establish in the invitation to tender stage.

(c) Mapping results for querying all regulatory knowledge related to information requirements.

Fig. 14. The mapping results of three query examples between generated RDF graph and expert ontology

As shown in Fig. 14, the results of queries that consist of common relations and instances (e.g., exchange information requirements, consider, establish, etc.), are essentially the same. Some instances in the result list may differ in wording but refer to the same instance (e.g., level and level of information need). However, any queries that include unique instances or properties (e.g., might need, undertaking, etc.) of the RDF graph show no result in the expert ontology and vice versa. This is reasonable because an RDF triple can only exist if the instances and the property that comprises this triple are defined. Based on the above analysis, the connections between common graph elements (instances and properties) in the generated RDF graph are complete and correct. The discrepancies between query results are entirely caused by differences in instances or properties Considering the proposed method achieves 73.7% and 74.5% accuracy in the automatic extraction of instances and properties, the authors believe the proposed framework achieved about 74% accuracy in autonomous implicit knowledge mining and modelling compared to experts' manual approach.

6. Limitations

After the in-depth analysis of the mapping results, the authors identified the limitations of the proposed method. Firstly, the proposed framework only can identify verb-centric relations in clauses and does not work effectively with adjectival relations (e.g., relative_to). Secondly, the recognition and analysis of clauses remain at the sentence level. The referential relationships between clauses have not been recognised, which leads to the generation of some redundant instances. For example, the "information requirements" established by the appointing party (Clause 5.2.1) are referenced as "requirements" in the following paragraph. Due to it cannot perceive their correlation, this approach defines them as two separate instances. Furthermore, the fine-tuned language model in the proposed framework discriminates clauses from descriptions through the expression pattern of sentences. This feature allows the model to process different standard documents after training but loses the hierarchical relationship (e.g., classes in ontology) between different entities. In addition, as a sentence-level classifier, the language model cannot filter out the unnecessary relations and instances in clauses, which results in the generation of some redundant triplets. For example, Clause 5.2.1 of ISO 19650-2, "when establishing the exchange information requirements, the appointing party shall establish the supporting information that the prospective lead appointed party might need.", where the triplet (appointing party, shall establish, the supporting information) is necessary, while the triplet (lead appointed party, might need, the supporting information) is superfluous. Finally, the proposed NLP-based approach can only extract the entities and relations that are defined or mentioned in the regulation document. It is not able to generate summarised entities or relations as domain experts.

7. Contributions to the Body of Knowledge

This research is important for regulatory knowledge interpretation from the perspective of automation, scope of application, and knowledge representation. For automatic knowledge interpretation, this research introduces a novel method that combines transfer learning with advanced NLP techniques to extract and parse clauses. Compared with existing ontology-driven approaches, this method is more automated and can be directly applied to different regulations without constructing ontologies for each regulation. In terms of the application scope, this research is the first in the AEC domain that addresses complex knowledge interpretation for qualitative regulatory requirements. To deal with flexible qualitative requirements, this research innovatively integrates linguistic knowledge (The Seven Clause Theory) into syntax parsing to identify and simplify complex clauses (compound clause and coordinate clause) into simple clauses. Additionally, the combination of Phrase structure grammar (PSG) with Dependency grammar in this research allows the proposed method to be capable of parsing complex logic, multiple relations and attributes. The multi-label joint mapping approach is proposed to cope with complicated parsing results and precisely assemble the words in clauses as knowledge constituents. The above innovations allowed the proposed method to outperform existing algorithms (OpenIE and ClauseIE) in qualitative requirements parsing and achieve 74% accuracy on a comparison against the manual work of domain experts. For knowledge representation, this research develops a knowledge modelling engine to automatically reorganise extracted regulatory knowledge as an RDF graph, which allows the extracted knowledge to be stored, accumulated, and reused by different types of downstream applications.

The impact of our research on regulatory knowledge interpretation in the AEC domain could be profound. First, this research brings automatic compliance checking for qualitative standards (e.g., ISO 19650) one step closer to reality. Qualitative requirements in project-related regulatory documents (e.g., project contracts, information requirements) can be extracted and checked against actual project documents and records. The time and cost spent on process management and quality control would be significantly reduced and the enterprises can improve and optimize their workflow accordingly. Second, through the automatic and cumulative approach proposed in this research, various standards from different sectors in the AEC domain can be quickly processed and the extracted regulatory knowledge can be easily merged to form a comprehensive large-scale knowledge base, which enables some cross-domain engineering applications, such as multi-objective optimisation and holistic decision-making.

8. Conclusion and future work

This paper introduced a novel autonomous complex knowledge mining framework that enables fully automated regulatory knowledge transformation from textual regulation documents to a graph-based knowledge model. The proposed framework first establishes a reliable domain dataset via data augmentation and the Delphi validation. Then a BERT-based clause extractor that can extract clauses from different regulation documents is developed by fine-tuning with the domain dataset. After that, the extracted clauses are processed by a linguistics- and NLP-supported constituent extractor, where the regulatory constituents in the clauses are automatically extracted. Finally, these constituents are automatically integrated as RDF triples/reifications and assembled as a regulation knowledge graph by a modelling engine. The proposed framework achieved 74% accuracy on ISO 19650-2 (Section 5.2), which indicates that the proposed autonomous knowledge mining and modelling framework is promising for downstream applications.

The contributions of this research are highlighted as follows: 1) the proposed framework introduces an efficient method for establishing reliable domain datasets, which can be adapted by other researchers for similar research. 2) the proposed framework unprecedentedly uses fine-tuned a large language model as the source of domain knowledge to identify regulatory statements in documents, which eliminates extensive manual work involved in constructing ontologies and realises full automation in clause extraction for different regulation documents. 3) the proposed framework innovatively incorporates linguistic knowledge and dependency parsing into the extraction of constituents, which significantly improves the performance in parsing regulatory knowledge with complex logic and multiple relations. 4) a more compatible representation (RDF graph) is utilised to store the extracted regulatory knowledge, which enables the knowledge to be managed, queried and reused by various downstream applications.

In conclusion, the proposed autonomous complex knowledge mining framework may have a profound impact on regulatory knowledge transformation in the AEC domain. It fills the gap in fully automated knowledge mining for regulation documents with qualitative requirements and significantly improves the efficiency of regulatory knowledge interpretation. The advent of the autonomous complex knowledge mining method makes it possible to perform large-scale knowledge extraction from qualitative regulation documents, as well as facilitates the digitization of regulatory knowledge. It brings some downstream applications, such as multi-regulation knowledge fusion, automated compliance checking, multi-objective optimisation, and holistic decision-making, one step closer to reality. In future work, research efforts are focusing on enhancing adjectival relation recognition and coreference resolution. The knowledge modelling algorithm can be improved in the aspect of constructing the T-box. In addition, other types of regulation documents in the AEC domain, such as existential requirements, building codes, and contractual documents, will be tested to further improve the performance of the proposed constituent extractor.

Reference

[1] N.O. Nawari, Building Information Modeling : Automated code checking and compliance processes, (2018), pp.164. https://doi.org/10.1201/9781351200998, (accessed March 4,2023).

[2] R. Sacks, U. Gurevich, P. Shrestha, A review of building information modeling protocols, guides and standards for large construction clients, Journal of Information Technology in Construction (ITcon), 21(29) (2016), pp. 479-503. http://www.itcon.org/paper/2016/29.

[3] A.T. Kovacs, A. Micsik, BIM quality control based on requirement linked data, International Journal of Architectural Computing. 19 (2021), pp.431–448. https://doi.org/10.1177/14780771211012175.

[4] X. Zhu, H. Li, G. Xiong, H. Song, Automated qualitative rule extraction based on bidirectional long short-term memory model, 29th EG-ICE International Workshop on Intelligent Computing in Engineering 2022, pp. 227–237. https://doi.org/10.7146/AUL.455.C213.

[5] M. Wagner Alibali, K.R. Koedinger, The developmental progression from implicit to explicit knowledge: a computational approach, Behavioral and Brain Sciences, 22(5) (1999), pp. 755-756. https://doi.org/10.1017/S0140525X99522188.

[6] V. Prathap Reddy M, P. P.V.R.D, M. Chikkamath, K. Ponnalagu, Extracting conjunction patterns in relation triplets from complex requirement sentence, International Journal of Computer Trends and Technology. 60 (2018), pp. 133–143. https://doi.org/10.14445/22312803/IJCTT-V60P121.

[7] B. Feijó, W.G. Krause, D.L. Smith, P.J. Dowling, A hypertext model for steel design codes, Journal of Constructional Steel Research, 28 (1994), pp.167–186. https://doi.org/10.1016/0143-974X(94)90041-8.

[8] E. Hjelseth, N. Nisbet, Capturing normative constraints by use of the semantic mark-up RASE methodology, Proceedings of CIB W78-W102 Conference, (2011), pp.1–10. http://itc.scix.net/data/works/att/w78-2011-Paper-45.pdf.

[9] T.H. Beach, Y. Rezgui, H. Li, T. Kasim, A rule-based semantic approach for automated regulatory compliance in the construction sector, Expert Systems with Applications. 42 (2015), pp.5219–5231. https://doi.org/10.1016/J.ESWA.2015.02.029.

[10] G.T. Lau, K. Law, An information infrastructure for comparing accessibility regulations and related information from multiple sources, International Conference on Computing in Civil and Building Engineering, ICCCBE, 10, (2004), pp.1-11. https://doi.org/10.25643/bauhaus-universitaet.192.

[11] S. Li, H. Cai, M. Asce, V.R. Kamat, Integrating natural language processing and spatial reasoning for utility compliance checking, Journal of Construction Engineering and Management. 142 (2016), pp.04016074. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001199.

[12] K. Liu, N. El-Gohary, Ontology-based semi-supervised conditional random fields for automated information extraction from bridge inspection reports, Automation in Construction. 81 (2017), pp.313–327. https://doi.org/10.1016/J.AUTCON.2017.02.003.

[13] J. Zhang, N.M. El-Gohary, A.M. Asce, Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking, Journal of Computing in Civil Engineering. 30 (2013), pp. 04015014. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000346.

[14] X. Wang, N. El-Gohary, Deep learning-based relation extraction from construction safety regulations for automated field compliance checking, Construction Research Congress 2022 (2022), pp.290–297. https://doi.org/10.1061/9780784483961.031.

[15] X. Wang, N. El-Gohary, Deep learning-based named entity recognition from construction safety regulations for automated field compliance checking, Computing in Civil Engineering. (2021), pp.164–171. https://doi.org/10.1061/9780784483893.021.

[16] R. Zhang, N. El-Gohary, A machine-learning approach for semantically-enriched building-code sentence generation for automatic semantic analysis, Construction Research Congress 2020: Computer Applications. (2020), pp.1261–1270. https://doi.org/10.1061/9780784482865.133.

[17] P. Zhou, N. El-Gohary, Ontology-based automated information extraction from building energy conservation codes, Automation in Construction. 74 (2017), pp.103–117. https://doi.org/10.1016/J.AUTCON.2016.09.004.

[18] A. Yurchyshyna, A. Zarli, An ontology-based approach for formalisation and semantic organisation of conformance requirements in construction, Automation in Construction. 18 (2009), pp. 1084–1098. https://doi.org/10.1016/J.AUTCON.2009.07.008.

[19] R. Anantharangachar, S. Ramani, S. Rajagopalan, Ontology guided information extraction from unstructured text, International Journal of Web & Semantic Technology (IJWesT). 4 (2013), pp.19-36 https://doi.org/10.5121/ijwest.2013.4102.
[20] Y.C. Zhou, Z. Zheng, J.R. Lin, X.Z. Lu, Integrating NLP and context-free grammar for complex rule interpretation towards automated compliance checking, Computers in Industry. 142 (2022), pp.103746. https://doi.org/10.1016/J.COMPIND.2022.103746.
[21] E. Soysal, I. Cicekli, N. Baykal, Design and evaluation of an ontology based information extraction system for radiological reports, Computers in Biology and Medicine. 40 (2010), pp. 900–911. https://doi.org/10.1016/J.COMPBIOMED.2010.10.002.
[22] X. Xue, J. Zhang, Part-of-speech tagging of building codes empowered by deep learning and transformational rules, Advanced Engineering Informatics. 47 (2021), pp.101235 https://doi.org/10.1016/J.AEI.2020.101235.
[23] K. Adnan, R. Akbar, Limitations of information extraction methods and techniques for heterogeneous unstructured big data, International Journal of Engineering Business Management. 11 (2019), pp.1-23. https://doi.org/10.1177/1847979019890771.
[24] Y. Rui, V. Ivan, S. Carmona, M. Pourvali, Y. Xing, W.-W. Yi, H.-B. Ruan, Y. Zhang, Y. Rui, V.I.S. Carmona, M. Pourvali, Y. Xing, W.W. Yi, H.B. Ruan, Y. Zhang, Knowledge mining: a cross-disciplinary survey, Machine Intelligence Research. 19 (2022), pp. 89–114. https://doi.org/10.1007/s11633-022-1323-6.
[25] A. Sara Ismail, K. Nita Ali, N.A. Iahad, A review on BIM-based automated code compliance checking system, 2017 International Conference on Research and Innovation in Information Systems. IEEE,(2017), pp. 1-6. https://doi.org/10.1109/ICRIIS.2017.8002486.
[26] C. Eastman, J. min Lee, Y. suk Jeong, J. kook Lee, Automatic rule-based checking of building designs, Automation in Construction. 18 (2009), pp. 1011–1033. https://doi.org/10.1016/J.AUTCON.2009.07.002.
[27] B. Zhong, X. Xing, H. Luo, Q. Zhou, H. Li, T. Rose, W. Fang, Deep learning-based extraction of construction procedural constraints from construction regulations, Advanced Engineering Informatics. 43 (2020), pp.101003. https://doi.org/10.1016/J.AEI.2019.101003.
[28] W. Yun, X. Zhang, Z. Li, H. Liu, M. Han, Knowledge modeling: a survey of processes and techniques, International Journal of Intelligent Systems. 36 (2021), pp.1686–1720. https://doi.org/10.1002/INT.22357.
[29] A. Moreno, D. Isern, A.C. López Fuentes, Ontology-based information extraction of regulatory networks from scientific articles with case studies for Escherichia coli, Expert Systems with Applications. 40 (2013), pp. 3266–3281. https://doi.org/10.1016/J.ESWA.2012.12.090.
[30] P.J. Tierney, A qualitative analysis framework using natural language processing and graph theory, The International Review of Research in Open and Distributed Learning. 13 (2012), pp.173–189. https://doi.org/10.19173/IRRODL.V13I5.1240.

[31] D.M. Salama, N.M. El-Gohary, Semantic text classification for supporting automated compliance checking in construction, Journal of Computing in Civil Engineering. 30 (2013), pp.04014106. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000301.

[32] P. Zhou, S.M. Asce, N. El-Gohary, A.M. Asce, Ontology-based multilabel text classification of construction regulatory documents, Journal of Computing in Civil Engineering. 30 (2015), pp.04015058. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000530.

[33] P. Zhou, S.M. Asce, N. El-Gohary, A.M. Asce, Domain-specific hierarchical text classification for supporting automated environmental compliance checking, Journal of Computing in Civil Engineering. 30 (2016), pp.04015057. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000513.

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

[35] Z. Zheng, Y.C. Zhou, X.Z. Lu, J.R. Lin, Knowledge-informed semantic alignment and rule interpretation for automated compliance checking, Automation in Construction. 142 (2022), pp.104524. https://doi.org/10.1016/J.AUTCON.2022.104524.

[36] X. Xu, H. Cai, Ontology and rule-based natural language processing approach for interpreting textual regulations on underground utility infrastructure, Advanced Engineering Informatics. 48 (2021), pp.101288. https://doi.org/10.1016/J.AEI.2021.101288.

[37]M. Minsky, A framework for representing knowledge, Frame Conceptions and Text Understanding. (1979), pp.1–25. https://doi.org/10.1515/9783110858778-003.(accessed March 4,2023).

[38] M.M. Taye, Understanding semantic web and ontologies: theory and applications, Journal of Computing. 2 (2010), pp.182-192.https://doi.org/10.48550/arXiv.1006.4567.

[39] P. Pauwels, W. Terkaj, Express to OWL for construction industry: towards a recommendable and usable ifcOWL ontology, Automation in Construction. 63 (2016), pp.100–133. https://doi.org/10.1016/J.AUTCON.2015.12.003.

[40] K. Janowicz, M.H. Rasmussen, M. Lefrançois, G.F. Schneider, P. Pauwels, BOT: the building topology ontology of the W3C linked building data group, Semantic Web. 12 (2020), pp.143–161. https://doi.org/10.3233/SW-200385.

[41] P. Pauwels, Building element ontology, https://pi.pauwel.be/voc/buildingelement/index-en.html,(2018), (accessed March 4,2023).

[42] A. Wagner, W. Sprenger, C. Maurer, T.E. Kuhn, U. Rüppel, Building product ontology: core ontology for linked building product data, Automation in Construction. 133 (2022), pp.103927. https://doi.org/10.1016/J.AUTCON.2021.103927.

[43] A. Hamdan, M. Bonduel, R. Scherer, An ontological model for the representation of damage to constructions, CEUR Workshop Proceedings. 2389(6) (2019), pp. 64-77. https://ceur-ws.org/Vol-2389/05paper.pdf.

[44] B. Balaji, A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, D. Hong, A. Johansen, J. Koh, J. Ploennigs, Y. Agarwal, M. Bergés, D. Culler, R.K. Gupta, M.B. Kjærgaard, M. Srivastava, K. Whitehouse, Brick: Metadata schema for portable smart building applications, Applied Energy. 226 (2018), pp.1273–1292. https://doi.org/10.1016/j.apenergy.2018.02.091.

[45] V. Kukkonen, A. Kücükavci, M. Seidenschnur, M.H. Rasmussen, K.M. Smith, C.A. Hviid, An ontology to support flow system descriptions from design to operation of buildings, Automation in Construction. 134 (2022), pp.104067. https://doi.org/10.1016/J.AUTCON.2021.104067.

[46] P. Pauwels, G. Fierro, A reference architecture for data-driven smart buildings using brick and LBD ontologies, CLIMA 2022 Conference. (2022), pp.1-8. https://doi.org/10.34641/CLIMA.2022.425.

[47] Y. Zhou, J. Lin, Z. She, Automatic construction of building code graph for regulation intelligence, International Conference on Construction and Real Estate Management 2021. (2021), pp.248–254. https://doi.org/10.1061/9780784483848.028.

[48] L. Jiang, J. Shi, Z. Pan, C. Wang, N. Mulatibieke, A multiscale modelling approach to support knowledge representation of building codes, Buildings. 12(2022), pp.1638. https://doi.org/10.3390/BUILDINGS12101638.

[49] L. Zhang, B. Ashuri, BIM log mining: discovering social networks, Automation in Construction. 91 (2018), pp.31–43. https://doi.org/10.1016/J.AUTCON.2018.03.009.

[50] C.-C. Hsu, B. Sandford, The Delphi technique: making sense of consensus, Practical Assessment, Research, and Evaluation. 12 (2007), pp.1-8. https://doi.org/10.7275/PDZ9-TH90.

[51] H. Weiss, Randolph Quirk/Sidney Greenbaum/Geoffrey Leech/Jan Svartvik, A comprehensive grammar of the English language, English World-Wide. 8 (1987), pp.123–128. https://doi.org/10.1075/EWW.8.1.10WEI.

[52] BayerMarkus, KaufholdMarc-André, ReuterChristian, A survey on data augmentation for text classification, ACM Computing Surveys. 55(7) (2021), pp.1-39. https://doi.org/10.1145/3544558.

[53] C. Coulombe, Text data augmentation made simple by leveraging NLP cloud APIs, arXiv preprint arXiv，1812.04718 (2018), pp.1-32. https://doi.org/10.48550/arxiv.1812.04718.

[54] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on Knowledge and Data Engineering. 22 (2010)， pp.1345–1359. https://doi.org/10.1109/TKDE.2009.191.

[55] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks? Advances in Neural Information Processing Systems. 4 (2014), pp.3320–3328. https://doi.org/10.48550/arxiv.1411.1792.

[56] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 1 (2019), pp.4171–4186. https://doi.org/10.48550/arxiv.1810.04805.

[57] L. del Corro, R. Gemulla, ClausIE: clause-based open information extraction, Proceedings of the 22nd International Conference on World Wide Web. (2013), pp.355-366. https://doi.org/10.1145/2488388.

[58] S. Zhou, B. Yu, A. Sun, C. Long, J. Li, J. Sun, A survey on neural open information extraction: current status and future directions, IJCAI International Joint Conference on Artificial Intelligence. (2022) pp.5694–5701. https://doi.org/10.24963/ijcai.2022/793.

[59] J. Sowa, B. Woods, G. Hirst, N. Sondheimer, R. Thomason, Principles of semantic networks, 1991. https://doi.org/10.1016/C2013-0-08297-7 (accessed March 1, 2023).

[60] N. Noy, Deborah L. McGuinness, Ontology development 101: a guide to creating your first ontology,(2001).http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html.(accessed March 4,2023).