Ontology-based distant supervision for extracting entity-property relations in construction documents

Junjie Jiang [a], Chengke Wu [a,*], Wenjie Sun [a], Yong He [c], Yuanjun Guo [a,b], Yang Su [d], Zhile Yang [a,b]

[a] Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
[b] Guangdong Institute of Carbon Neutrality (Shaoguan), Shaoguan, China
[c] China Construction Third Bureau First Engineering & MEP Co., Ltd., Shenzhen, China
[d] School of Design and the Built Environment, Curtin University, Perth, Australia

Abstract

Engineering documents of construction management contain rich information, including various project entities and their properties, forming entity-relation-entity triples. Such information is valuable for management activities, such as quality inspection and compliance checking, which is however challenging to process due to unstructured form of original texts. Thus, triple extraction enabled by deep learning (DL) has become an urgent need in the industry. However, fully supervised methods are impractical as they require enormous labelled data. Distant supervision (DS) alleviates the burden of manual labelling but introduces much noise and requires establishing priori large-scale knowledge bases (KBs). Thus, we propose a novel framework for entity-property triple extraction, i.e., Ont4RE. The Ont4RE leverages an ontology to integrate domain knowledge and automatically annotate a construction corpus without manual intervention, before a DL model is employed to extract triples. In experiments, we verified that Ont4RE can realize precise auto-labelling while is adapt to different extractors with varying DL structures. Specifically, the AUPRC, mean P@N, and F1-score among the tested DL models are improved by 6.1%, 3.3%, and 11.7% in average, respectively. The best performance is reached when naïve BERT serving as the relation extractor（i.e., 92.3% AUPRC and 93.5% F1-score). A controlled experiment was conducted. The results demonstrate that when searching for project information, the Ont4RE was superior to manual approaches and artificial intelligence tools (i.e., ChatGPT) in terms of precision, recall, and robustness, which also reduced information extraction time from 98 seconds to 2.5 seconds. As such, Ont4RE can substantially improve information extraction efficiency in construction management, thereby facilitating downstream management tasks and decision-making.

Keywords: Ontology; Distant Supervision; Relation Extraction; Semantic Similarity; Deep Learning; Construction Industry

1. Introduction

In the digital era, the concept of 'smart construction' is attracting increasing attention, which leverages Artificial Intelligence (AI) and cutting-edge information technologies (ITs) to boost construction productivity, quality, and safety, [2, 19, 12]. According to McKinsey, the industry has globally received over $50 billion invested in smart construction during 2020-2022 [20], an 85% increase compared to the last three-year period. Construction Engineering and Management (CEM) permeates the entire project lifecycle; thus, it is the core of applying smart construction technologies [1]. CEM depends on collecting, analyzing, and disseminating various information of a project, such as task progress, material properties, and costs. Over 80% of such information is buried in textual project documents, such as plans, manuals,

* Corresponding author.
Email address: ck.wu@siat.ac.cn (Chengke Wu)

standards, and contracts [14]. Most documents take digital forms but are still human-written texts, which are uncertain, non-standard, and unstructured [17]. Thus, it is challenging to directly extract valuable information from the documents. The delay and low quality of information can lead to project delays, resource waste, quality issues, and even safety risks [58]. As such, information extraction (IE) plays an essential role in smart CEM.

Fortunately, natural language processing (NLP) methods underpinned by Deep Learning (DL) models, such as Text Convolutional Neural Network (TextCNN), Recurrent Neural Network (RNN), and Transformer, have realized efficient, robust, and accurate IE in general dataset (e.g., DBpedia) and have been applied in the construction industry [23]. Recent IE in the industry focuses on triple extraction (ERE) [3]. Each triple contains a head entity, a tail entity, and a relation between them. For instance, the 'concrete' and its type (e.g., C30) are extracted as the head and tail, respectively, and the relation is 'has-type'. As such, engineers can conveniently extract entities and relevant properties, regardless they are scattered in different sources [6, 7, 18]. Thus, ERE can facilitate many downstream applications, such as compliance checking [5], knowledge sharing [4, 11], and cost and schedule monitoring [22, 13]. However, existing ERE methods heavily rely on supervised learning, which entails laborious and time-intensive manual annotation of triples in the training dataset. To address the issue, the distantly supervised (DS) annotation strategy [8] is proposed, which eliminates manual annotation with an external large-scale knowledge base (KB). A KB is formed by numerous triples serving as prior knowledge, which are mapped and compared to the training data (e.g., text sentences). If both the head and tail entities in a valid KB triple are found in one sentence, the sentence is assumed to hold the same relation with the corresponding triple in the KB, and the sentence is labelled by the triple automatically [9, 32]. The naïve DS strategy can introduce a lot of noise, because the corpus can contain ambiguous entities with similar semantics but different names. For instance, in a working plan, the two entities with different texts, 'crew' and 'worker', share the same semantics of 'labor'. Moreover, the co-occurrence of entities does not imply the presence of a pre-defined relation [37]. Thus, without a sufficiently large and carefully crafted KB, conventional DS-based models, such as Piece-wise CNN (PCNN) and PCNN with self-attention (PCNN-ATT) [36, 37] easily fail in practical ERE tasks. On the other hand, ontologies are knowledge models that employ standard logics to define unambiguous concepts and relations in specific domains [15]. The knowledge in an ontology is not limited to individual texts, but to groups of texts with similar semantics [15]. Moreover, as an ontology only defines abstract concepts rather than specific entities, its size is significantly smaller than a KB. However, the combination of ontology, DS-based IE, and cutting-edge relation extraction models has not been adequately discovered in CEM.

Therefore, in this study, we propose a novel ERE framework, namely, Ontology for Relation Extraction (Ont4RE). The Ont4RE consists of three components: 1) a domain ontology, 2) a DS strategy improved by the ontology, and 3) a downstream relation extractor. The extractor can recognize 12 different entity-property relations (also called triples), i.e., 'head entity has-property tail entity' (see Table 1). Such entity-property relations are the backbone of CEM, because detailed numerical or semantic descriptions of project entities are required in most management activities, which however are the most prone to errors or omissions [24, 25]. Additionally, we use Chinese documents to validate the framework. However, given the widely recognized challenge of Chinese NLP, such as the prevalent ambiguity and non-existence of word segmentation [27], we believe that the Ont4RE can be transferred to other languages following the same methodology in Section 3. The study consists of four aspects:

1) Establishing an ontology for CEM (i.e., CEMO), which provides unambiguous representations of domain classes (i.e., concepts) and relations pertinent to CEM.

2) Proposing an ontology-based DS strategy, which automatically annotates sentences by taking the CEMO as the external knowledge source.

3) Developing a relation extractor which consists of a Transformer-based encoder and a relation classifier.

4) Validating the Ont4RE. Through comprehensive experiments (i.e., evaluation of sentence annotation and model implementation), we found that the proposed DS strategy significantly outperformed the conventional one. We also tested the performance using different DL models as the triple encoder. The models with the Transformer structure were superior

to the competitors. The best performance was achieved by BERT, which was 92.3% AUPRC and 93.5% F1-score. Additionally, a controlled experiment was conducted to compare Ont4RE with manual searching and the cutting-edge ChatGPT in practical ERE tasks. The results proved that Ont4RE reduced IE time by 97% while maintaining higher extraction accuracy.

2. Literature Review

This section reviews relevant studies from two aspects: 1) current progress of applying NLP and DL models in IE tasks in construction projects, and 2) studies on distant supervision for relation extraction.

2.1 Information extraction in the construction industry

IE in CEM can be divided into three groups based on the methods they rely on: 1) rule or template-matching, 2) traditional ML, and 3) DL-based. 1) Rule-based methods utilize syntactic and semantic features of texts to extract entities and relations. For instance, Zhang et al. proposed a pattern-matching method to extract rules and conditions in construction regulation [5]. However, due to the diversity and complexity of patterns, it is impossible to create abundant rules covering all situations. Thus, rule-based methods become less popular. 2) ML-based IE follows the pipeline of feature design, data annotation, and model training. The methods often work with simple ML models, such as Latent Dirichlet Allocation, Support Vector Machine, and Decision Tree. However, manual feature design and annotation are time-consuming, and the models often only perform well in shorter texts with simple semantics [28, 30]. 3) DL-based IE automatically recognizes features hidden in texts, thanks to the powerful deep neural structure. As such, neural network-based DL models like CNN, Long-Short-Term Memory (LSTM), and RNN avoid the tedious feature design and are increasingly adopted in different IE tasks in the industry. Additionally, ontologies are structural and unambiguous representations of domain knowledge, which have been applied to improve DL-based IE methods in terms of accuracy and efficiency [48, 50]. Generally, the standard and general information in ontologies is used to classify specific texts (i.e., assigning a domain class to each candidate entity or relation) before feeding texts to learning models. This strategy facilitates model training by reducing noise caused by varying texts [49]. However, DL models require excessive manual data annotation, which is the main barrier to practical application. To alleviate the burden, the DS strategy proposed by Mintz et al. [8] became a feasible solution. DS could realize semi-automated labeling, which however needs an external large-scale KB as the supervisor (more details are introduced in Section 2.2).

The IE methods are extensively used in downstream applications of CEM. For instance, Zhong et al. combined Bi-LSTM and Conditional Random Fields (CRF) to extract project constraint knowledge [3]. Li et al. proposed a framework that combined IE and spatial reasoning to identify non-compliance phenomena in underground utility projects to prevent accidents [29]. Ajayi et al. developed a two-stage classification method to extract information from factory bidding documents and identify risky clauses that could increase cost and working capital loss [33]. Wang et al. developed six DL models to assist with health and safety risk management to prevent occupational accidents and equipment damage [34]. Zhou et al. proposed a new IE model to mine valuable safety knowledge from engineering project reports, thereby enhancing system safety and awareness [34]. Roth et al. proposed an IE method that combined text classification and ontology-based pattern matching. They tested it in extracting energy constraints from the Energy Conservation Code, to support fully automatic energy compliance checking in buildings [35]. Xu et al. combined ontology-based IE and rules to extract requirements from specifications [45]. Similarly, Wu et al. proposed an integrated method with ontology-based rules and Bi-LSTM-CRF to realize information searching [46]. These studies indicate that IE has broad prospects and potential in the construction industry. However, current applications still require excessively manual intervention and are restricted to specific data and scenarios, resulting in the limited generalization ability and application.

2.2 Distant supervision strategy

Conventional supervised ERE requires enormous manually labeled data to train the learning model, which are not cost-effective in practices [31]. Thus, DS paradigm was proposed [8]. The naïve DS used an external large-scale KB to label a vast corpus by assuming that if two entities participate in a relation in the KB, any sentence that contains those entities express that relation. Unfortunately, this strong assumption leads to incorrect annotations and issues with long-tail data. Studies aiming to solve the issues can be grouped into feature-based and model-based methods. Feature-based methods tend to introduce specific features, such as entity types and context information, to reduce noise. For instance, Zeng et al. improved the DS hypothesis with at-least-one strategy to exactly describe the relations between entities [36], which however introduced noisy relations due to the rarity of correctly annotated sentences. Riedel et al. developed a famous corpus New York Times (NYT). They combined DS and the Multi-Instance Multi-Label Learning (MIML) strategy to allow an entity pair to have multiple relation labels [9], thereby alleviating the impact from false annotation. Recently, DL models have been increasingly applied to improve DS since they can utilize deep neural structures to automatically recognize to distinguish useful triple patterns from noise. For example, Lin et al. proposed a PCNN relation extractor, which followed the MIML strategy and used a convolutional architecture with piecewise max pooling to learn sentence features [37]. Ru et al. used word embedding and CNN to measure the semantic similarity between relation phrases in KB and two entities in sentences, thereby filtering incorrect labels [48]. Lin et al. and Ji et al. introduced a sentence-level attention model based on PCNN, making full use of context information in sentences and effectively reducing the impact of incorrect labels [37, 38]. In addition, pre-trained language models, such as BERT and GPT, have gained much popularity as they can learn abundant language knowledge without feature design, and some studies attempted to combine these models with DS [43, 44]. For example, a new framework called BERE was proposed by Hong et al. [39] to extract biomedical relations from a large-scale literature library automatically. Zeng et al. developed a BERT-based system that predicted molecular properties and extract biomedical relations from multiple sources without supervision [40]. Additionally, Alt et al. adapted GPT for DS settings and fine-tuned the NYT dataset to predict a broader range of relation types [41]. A recent study used ChatGPT's natural reasoning ability to generate triples to improve the long tail issue of the dataset, thereby enhancing the relation annotation with DS strategy [65]. Overall, the studies lack the usage of the domain knowledge and often require manually developed mapping rules for annotation or rely on existing large-scale KBs.

In the above review, two research gaps were found.

1) In CEM, existing ERE methods rely on rules or annotated data, which substantially restrict their performances in the CEM sector where the data are sparse and dynamic, and large-scale manual annotation is not practical.

2) The DS strategy requires a KB as a supplementary supervisor. However, current KBs are developed for general world knowledge instead of domain-specific knowledge. Moreover, developing a KB containing sufficient prior triples is time-consuming. On the other hand, leveraging ontology as a KB for DS is a potential solution to the challenge, yet current studies have not exploited the option.

To address the gaps, the Ont4RE framework is proposed, which combines an ontology of the CEM industry, an improved DS strategy that leverages the knowledge in the ontology to automatically label entity-property triples in texts, and the transformer-based downstream DL model to extract valid relations.

3. Methodology

The Ont4RE includes three steps: building an ontology in the CEM domain (named CEMO), developing the ontology-based DS strategy, and training a relation extraction model.

3.1 Ontology development and representation

The CEMO is developed by drawing upon our works [46, 48, 49] and other popular ontologies in the industry, such

as ifcOWL and DiCon. The ontologies have been used in different management functions, including information sharing and extraction. We establish the CEMO by modifying existing ontologies by following the Stanford 101 ontology development guide [26], while using the ontology editor Protégé and web ontology language (OWL).

First, we identify main subject classes (also called concepts) of CEM from existing ontologies, namely, Man, Machinery, Material, and Environment. The classes are organized into the tree-like taxonomy that contains four levels. For example, the general class 'Material' includes sub-classes with finer granularity, such as "Steel Material", "Concrete Material", and "Adhesive Material"; the class "Steel Material" can be further divided into sub-classes such as "Rebar", and "Steel Plate ". Second, we identify 12 property classes of the subject classes. They can be categorized as geometric and non-geometric, thereby forming another branch in the taxonomy. Third, we define semantic relations among the classes to transform the taxonomy to an ontology [59]. The relation between a fine-grained class and coarse-grained class is defined as "has-parent". Additionally, as the Ont4RE concerns extracting entity-property relations, we enrich the "has-property" by defining 12 sub-relations. Table 1 lists the details. As such, the CEMO can automatically recognize and label more types of entity-property relations in the DS process.

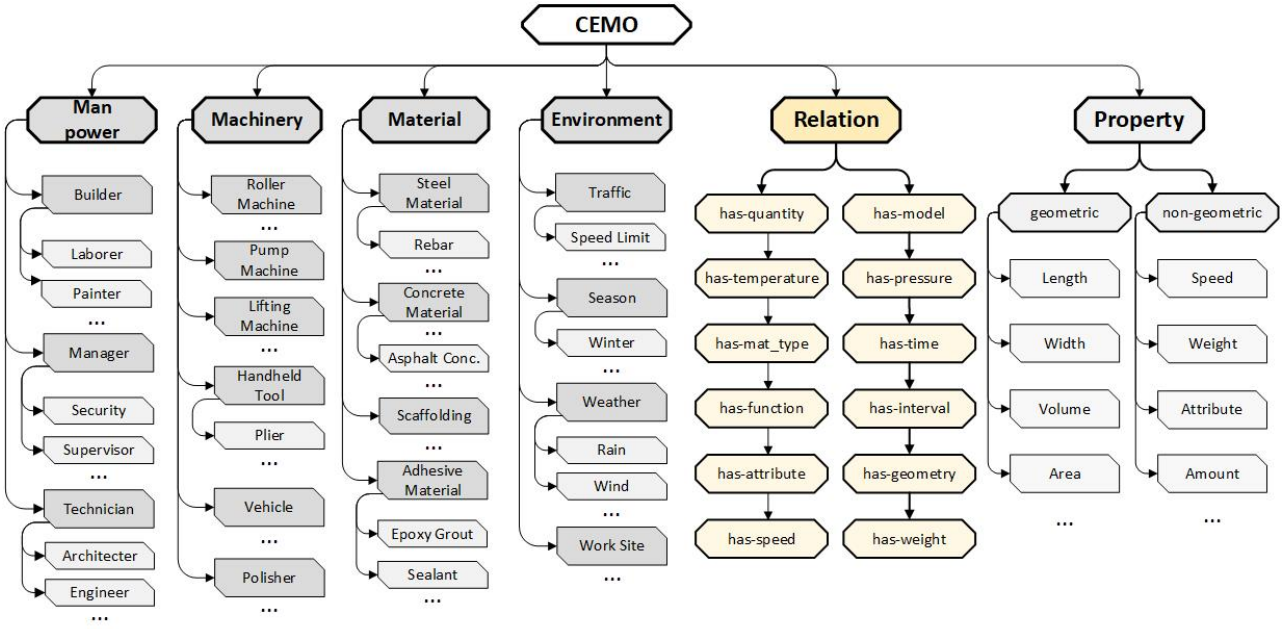


Figure 1. Hierarchy of the ontology CEMO (not fully expanded).

To enable the ontology-based DS, the CEMO is represented with a set of triples $\boldsymbol{T} = t^1, t^2 \ldots t^T$. A triple includes three elements, the head, relation, and tail. For example, the "Rebar" class has the "Weight" class as one property, which in triple representation is expressed as "Rebar has-weight Weight". As such, each triple t in $\boldsymbol{T}$ consists of a subject class ***SC*** as the head, an entity-property relation r, and a property class ***PC*** as the tail.

Table 1. The overview of entity-property relations in CEMO. The entities that exhibit has-property relations have been highlighted using the italic form, and their corresponding ontology classes are included in the parenthesis.

Entity-Property Relations	Subject Class (SC)	Property Class (PC)	Examples
has-time	Material /Environment	Time	The basic curing time of structural ***adhesive (SC)*** is about ***24h (PC)***
has-temp	Material /Environment	Temperature	The shrinkage of ***concrete (SC)*** is regulated by proper ***temperature (PC)***
has-model	Machine	Item Type	Emulsified asphalt ***spraying vehicle (SC)*** is the main equipment in asphalt ***auxiliary machinery (PC)***

has-speed	Machine	Velocity	The rolling speed of the ***roller (SC)*** is maintained at ***2-3km/h (PC)***
has-weight	Material/Machine	Weight	The weight of the ***material (SC)*** should be limited to ***500kg/m3 (PC)***
has-interval	Material	Spacing	Drill holes to ensure the spacing between each ***anchor bar (SC)*** is ***30cm (PC)***
has-pressure	Material	Pressure	Before the strength of ***concrete*** reaches ***1.2Mpa (PC)***, various load pressures should be avoided
has-quantity	Material/Machine /Man	Amount	There are ***four (PC)*** rubber ***rollers (SC)*** on this site
has-function	Material /Man	Characteristic	The ***contractility (PC)*** of ***sealant (SC)*** is an important index to measure its quality
has-attribute	Material /Man	Physical ability	The ***viscosity (PC)*** of ***asphalt concrete (SC)*** will affect its cracking resistance and water resistance
has-geometry	Material/Machine	Size	When the ***thickness (PC)*** of the ***steel plate (SC)*** is greater than 5mm, it should be bonded by pressure
has-mat_type	Material	Material type	Use ***C25 (PC) concrete (SC)*** to restore the upper side wall of the arch

3.2 Ontology-based DS strategy

The ontology-based DS strategy relies on the CEMO and takes three steps: 1) pre-processing raw documents, 2) identifying entity candidates from each text sentence, which could form valid entity-property relations, and 3) annotating sentences by considering the candidates and domain knowledge in CEMO. The DS process is shown in Figure 2.

3.2.1 Raw document pre-processing

The pre-processing is responsible for generating sentences, tokens, and semantic embeddings based on raw CEM documents. Frist, we define regular expressions by drawing upon common punctuations marks in Chinese, with which we recognize and extract sentences from the documents and form the sentence pool. Second, for each sentence s in the pool, we employed the Sentence-Transformer model for tokenization and semantic embedding generation [57]. The model is pre-trained on Chinese corpus, which not only has learned general semantics of the language but also adapts to the challenging context when tokenizing Chinese sentences. The pre-processing produces a list of tokens $e^1, e^2 \dots e^M$ and their corresponding embeddings $v(e^1), v(e^2) \dots v(e^M)$. The classes in CEMO are also converted by the Sentence-Transformer, resulting in $v(c^1), v(c^2) \dots v(c^N)$.

3.2.2 Entity candidate identification

For each sentence, we apply a double-layered loop and cosine similarity to measure the semantic similarity between each e^m and c^i based on their embeddings, which can be defined as Eq. (1).

$$\text{Cosine}(v(c), v(e)) = \frac{v(c) \cdot v(e)}{\| v(c) \| \times \| v(e) \|} \tag{1}$$

As such, each token e^i corresponds to N similarity values of all CEMO classes. Then, two consecutive max-pooling operations are applied. The first pooling operates at the token level, which identifies the most similar class for each token and produces a class list $c_{e^1}^{max}, c_{e^2}^{max} \dots c_{e^M}^{max}$. The second pooling operates on the list. It orders the list and takes two tokens with the highest class similarities as entity candidates, namely e^{c1} and e^{c2}.

3.2.3 Automatic sentence annotation

Based on the entity candidates, we follow a heuristic rule to realize automatic sentence annotation. Specifically, if a relation exists in CEMO connecting the classes of the two candidate entities, then the current sentence is automatically

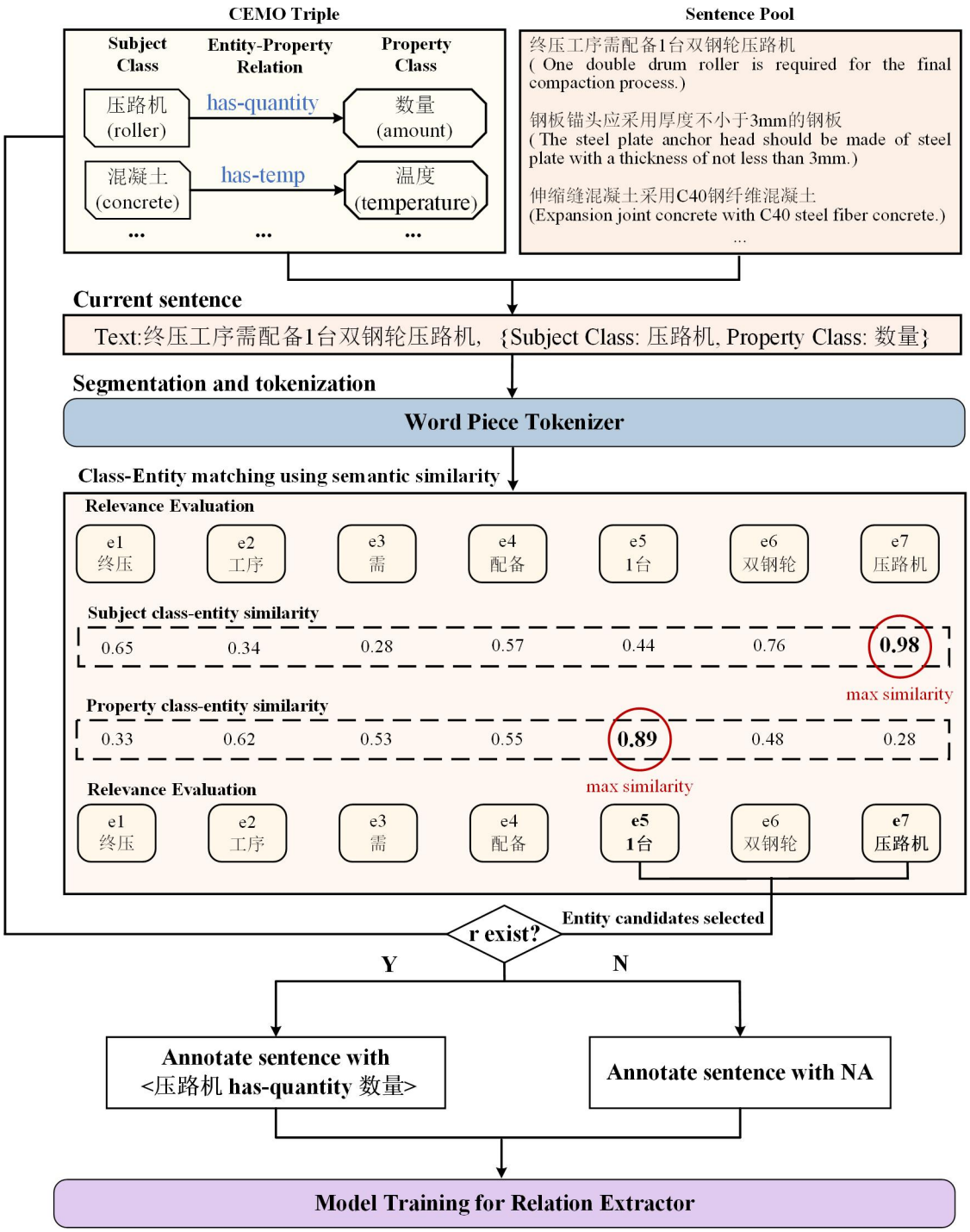


Figure 2. The pipeline for ontology-based distantly supervised annotation process.

annotated by this relation. In other words, we only label the triple in a sentence only if a valid ontological triple with sufficient semantic similarity can be found in CEMO. For instance, as demonstrated in Figure 2, in the sentence “One double drum roller is required for the final compaction process (终压工序需配备1台双钢轮压路机),” the words “roller” and “one” are identified as the entity candidates, and the corresponding classes CEMO are "Roller Machine" and "Amount", respectively. As the relation "has-quantity" exists between the two classes, the ontology-based DS strategy will annotate the sentence using the triple "Roller Machine has-quantity Amount ". Formally, each annotation includes the head entity, relation, tail entity, and positions of entities. The whole process is shown in Algorithm 1.

Algorithm 1 The ontology-based distantly supervised annotation using semantic similarity

Input: sentence pool, triple sets **T** ($t^1, t^2, ..., t^T$), triple t (sc, r, pc), JIEBA, Sentence-Transformer
Output: $corpus$ ($c = c_1, c_2, ..., c_n$)

$corpus$ = {} // initialize an empty corpus
first_entry = True // variable for finding correct triplet
for sentence in sentence pool **do**
  found = False // add a flag to track if matching success
  $words$ = JIEBA(sentence) // word segmentation
  $embeddings$ = Sentence-Transformer.encode($words$) // tokenize each words
  $entities$ = list($embeddings$)
    for t in **T do**
      compute cosine similarity between sc (pc) and $entities$ using Equation 1
      $SimList$ = $entities$.MaxPooling(**T**) // find classes with the highest similarity with the entities
      $entity\ candidates$ = MaxPooling($SimList$) // find entities with the top 2 highest similarity as the candidates
      found = True
      head entity ← sc, tail entity ← pc, relation ← r
      h_pos ← [sc, sc + len(sc)] // get the pos of head entity
      t_pos ← [pc, pc + len(pc)] // get the pos of tail entity
      if first_entry **then**
        first_entry == False
      end if
      write the head entity, tail entity, pos, and relation to $corpus$ // Distant supervision
      if found = False **then**
        write NA to $corpus$ // write NA if no relation is found
      end if
    end for
end for
return $corpus$

3.3 Entity-property relation extractor

Entity-property extraction can be regarded as a relation classification task when the two entity candidates are known. As such, the downstream relation extractor consists of a Transformer encoder for sentence representation and a decoder for relation classification.

3.3.1 Input sentence representation

As illustrated in Figure 3, the Transformer model takes tokens $e^1, e^2 \ldots e^M$ as the initial input sequence, which is padded or cut to a fixed length $\boldsymbol{L}$ for easy handling in the subsequent model layers. A specific token [CLS] is added to the sequence's end as the aggregated representation of the entire sequence. The embedding of each token to be fed into the Transformer contains three components, namely, the semantic embedding $v(e^i)$, the sentence boundary embedding $v(e_s^i)$, and position embeddings $v(e_p^i)$. $v(e_s^i)$ is embedded for detecting if the sequence exists special token, e.g., [CLS]. $v(e_p^i)$ is embedded for capturing the token's relevant position in the sequence. Thus, the final representation of a sentence s is concatenating above three components, i.e., $E^s \in \mathbb{R}^{3d \times L}$, supposing the vectors share the same dimension $d = 768$.

3.3.2 Transformer-based encoder

Theoretically, any deep-learning model can serve the purpose of the encoder. However, as demonstrated in Section 4, different model structures exhibit varying degrees of compatibility with the Ont4RE framework, while the Transformer structure outperforms the others.

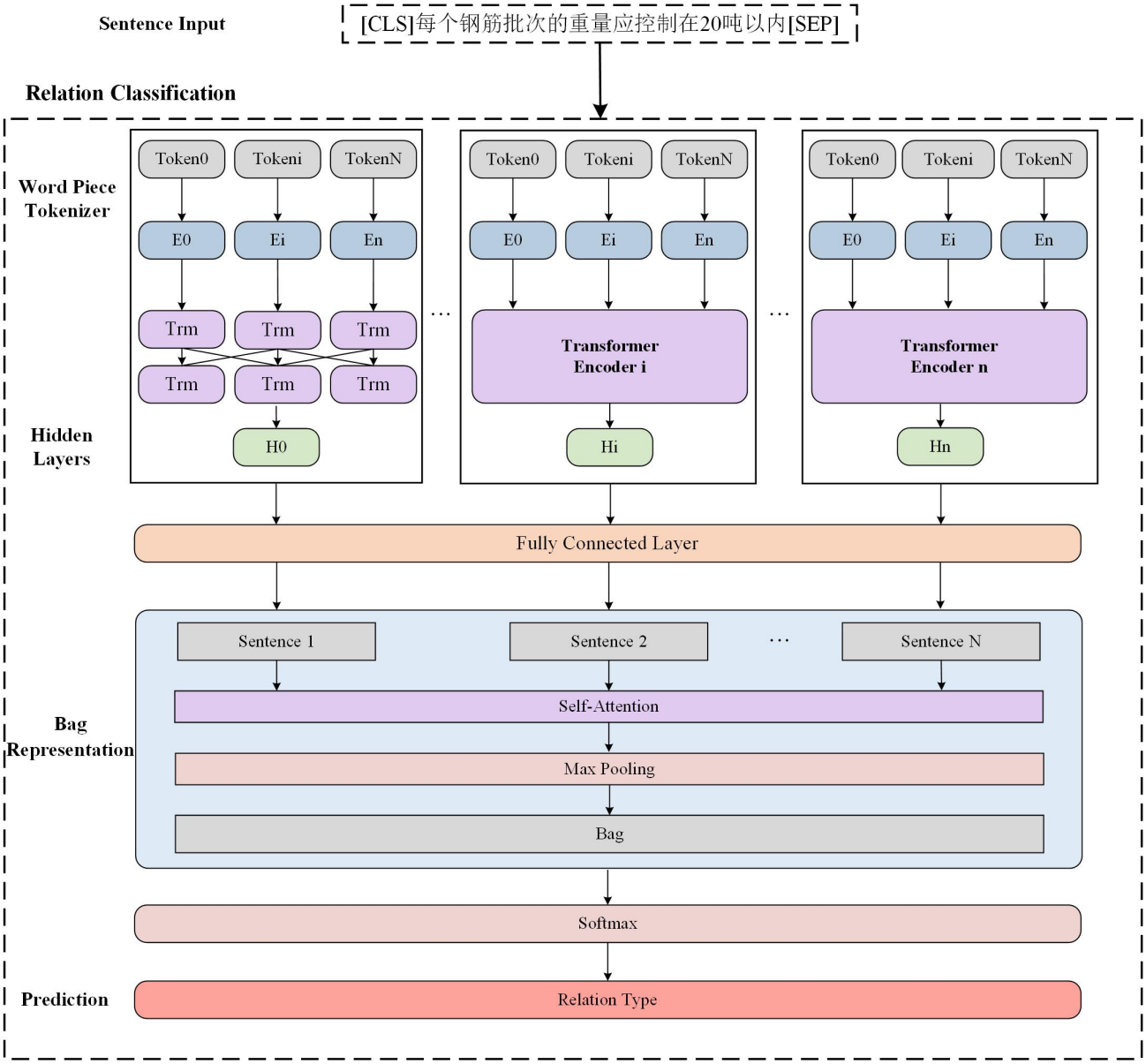


Figure 1. Workflow of the relation extractor.

The Transformer-based encoder consists of $\boldsymbol{U}$ multi-head self-attention layers, which is the critical component enabling the encoder to fully capture the context and understand the in-depth semantics. For each attention head in each layer, the input $v(e^i)$ is transformed into three matrices, i.e., $\boldsymbol{Q}$, $\boldsymbol{K}$, $\boldsymbol{V}$, by multiplying another three trainable matrices respectively, i.e., W^Q, W^K, and W^V, which can be shown in Eq. (2).

$$Q = v(e^i)W^Q, K = v(e^i)W^K, V = v(e^i)W^V \tag{2}$$

Then, the self-attention operation is used to produce the attention value Att_s of the entire sequence, which is represented in Eq. (3).

$$Att_s(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d}})V \tag{3}$$

If more than one attention heads are used, the aggregated value is obtained by applying a linear transformation to the concatenation of the results from different heads, as represented in Eq. (4).

$$Att_s^i = (Att_s^1 \oplus Att_s^2 \ldots \oplus Att_s^H)W^o + b^o \tag{4}$$

Where $i = 1,2..H$, $\boldsymbol{H}$ is the number of heads, and the W^0 is a trainable matrix. The output in a layer O^u is produced by a fully connected feed-forward layer as Eq. (5).

$$O^u = Act\left(Att_s^i\right)W^1 + b^1 \tag{5}$$

Where the W^1 is a trainable matrix, and the ***Act*** indicates activation functions, such as Relu, Tanh, and sigmoid. Finally, we take the output at the final layer, i.e., O^U, then extract the vector at the position of [CLS] as the sequence representation after encoding, i.e., $v(rep)$, which should follow the shape $\in \mathbb{R}^{3d}$.

3.3.2 Relation classifier

In CEM documents, a specific pair of entities can appear in multiple sentences, which however can contain different and even noisy information about the relations between entities. Hence, we adopt the bag-level method for relation classification. The method groups all sentences containing the same pair of entities into a bag and employs the self-attention mechanism to weight the bag members. As such, the extractor can fully utilize the context information to prioritize sentences and minimize the impact of noise [60]. Formally, for a bag $B = \{s_1, s_2, \ldots, s_N\}$, it includes $\boldsymbol{N}$ sentences all containing the same pair of entities. Accordingly, a set of sentence representation vectors $\{v_{rep}^{s^1}, v_{rep}^{s^2}, \ldots, v_{rep}^{s^N}\}$ is created by encoding the bag. Taking use of the vectors, the self-attention mechanism assigns a weight to each sentence, indicating its importance (i.e., attention) in the bag. The weights are evaluated based on the association between one sentence and others. Specifically, in a sentence s, the attention α_s is computed using Eq. 6.

$$\alpha_i = \frac{\exp(sim^i)}{\sum_{i=1}^{N}\exp(sim^i)} \tag{6}$$

Where sim^i indicates the sum of cosine similarities between of s_i and other sentences, namely, $\mathcal{C}\text{osine}(v_{rep}^{s^i}, v_{rep}^{s^j})$. The bag representation v^{bag} is the sum of sentence representations weighted by the attention values using Eq. 7.

$$v^{bag} = \sum_{i=1}^{n}\alpha_i v_{rep}^{s^i} \tag{7}$$

After the bag representation v^{bag} is obtained, the similarity between v^{bag} and relation embeddings r is calculated to get the confidence that the bag is predicted as a certain relation type, the process can be shown in Eq. 8.

$$vc = br + \epsilon \tag{8}$$

Here, ϵ is a bias vector, the outcome $vc \in \mathbb{R}^R$, and $\boldsymbol{R}$ is the number of relation types (in our case $\boldsymbol{R}$=12). Finally, the softmax function is utilized to compute the probability of predicting bag B as the k^{th} relation type using Eq. 9.

$$p(\,k \mid B, \theta\,) = \frac{\exp(c_k)}{\sum_{i=1}^{R}\exp(c_i)} \tag{9}$$

Where θ defines all trainable parameters in the relation extractor. The cross-entropy is employed to compare model predictions and ground-true labels while evaluating the loss, which is defined as Eq. 10.

$$L(\theta) = \sum_{i=1}^{NB}\log\big(p(\,k \mid B_i, \theta\,)\big) \tag{10}$$

Where NB is the number of bags. Finally, the Adam optimizer is applied to update model parameters in training. The entire process is presented in Algorithm 2.

Algorithm 2 The training process

Input: sentence s, relation set R, max_epoch $T_{\max}$, *Adam* optimizer
Output: The trained relation extractor

group bag of the same entity pairs into a set G: $G_1, G_2, ..., G_r$
$T \leftarrow 0$
while $T < T_{\max}$ **do**
  for G_i in G **do**
    for s_i in G_i **do**
      calculate input representation $v(e^i)$, $v(e^i_s)$, $v(e^i_p)$
      Transform $v(e^i)$ into matrices Q, K, V using Equation 2
      Compute attention values Att_s from different attention heads using Equation 3 and 4
      Compute sentence representation $v^{(s_i)}_{\text{rep}}$ using Equation 5
    end for
    Compute sentence association weights α_i using Equation 6
    calculate the bag representation v^{bag} using Equation 7
    calculate the confidence c of the bag G_i using Equation 8
    calculate the probability of the relation of the bag G_i using Equation 9
  end for
  update θ by *Adam* using Equation 10
  $T \leftarrow T + 1$
end while

4. Experimental results and evaluation

A series of experiments were conducted, including the similarity-based DS annotation evaluation, cross-comparisons using mainstream DL models as the relation extractor, and a controlled experiment mimicking real-world ERE in CEM. The models and experiment settings were realized in Ubuntu 20.04 and Python 3.7 environment with Nvidia GeForce RTX 3090 and Core i9-10920X. The developed dataset can be found in: https://github.com/Construction-Material/Construction-Dataset-CONSD.

4.1 Dataset construction

4.1.1 Raw data collection and pre-processing

To collect raw CEM documents, both online public databases and file systems of the cooperative company (China Construction Technology Group) were accessed, and 212 documents were collected. Then, the documents were filtered following the criteria: 1) Formatting check. We removed documents containing garbled characters or formatting errors 2) Content filtering and pre-processing. We checked document contents by manual screening to filter out those obviously irrelevant to the study, which resulted in 38 work plans, 13 manuals, and 21 standards. We followed the pre-processing steps in Section 3.2.1, which produced 4714 valid sentences. The mean length and variance of the sentences were 45.7 and 132.3, respectively. The Sentence-Transformer were implemented to divide sentences to tokens and then convert tokens into semantic embeddings $\in \mathbb{R}^{768}$. 3) Data augmentation. ChatGPT was used to enrich the data, increasing the number of sentences to 10K. More details of enriching are introduced in Section 4.1.2.

4.1.2 Data augmentation with ChatGPT and corpus development

As mentioned in the above section, over 4.7K sentences were kept for model training. However, we found that the annotation results with DS exhibited long-tail effect, a common issue in real-world data (see Figure 4). Specifically, many entities are associated with only a few entity-property relations, such as relation ‘has-geometry’ (the left of deep red area), whereas many other relations are associated with a few entities (the right of deep red area). This could lead to poor

performance of ERE when extracting low-frequency entity-property relations, more details can be found in evaluation metrics in Section 4.2.1.

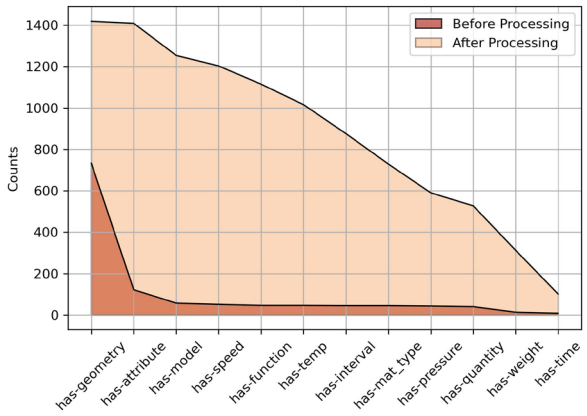


Figure 4. The long-tail effect of data distribution and its solution.

Fortunately, the text generation capability of ChatGPT, is suitable to generate semantically coherent and contextually appropriate sentences [52]. For the sake of alleviating long-tail effect, ChatGPT was used to enrich the sentence pool. Firstly, we filled existing sentences and entity pairs in the prompt and asked ChatGPT to generate synonymous sentences with consistent quality, similar semantics, but different expressions. An example is demonstrated in Figure 5.

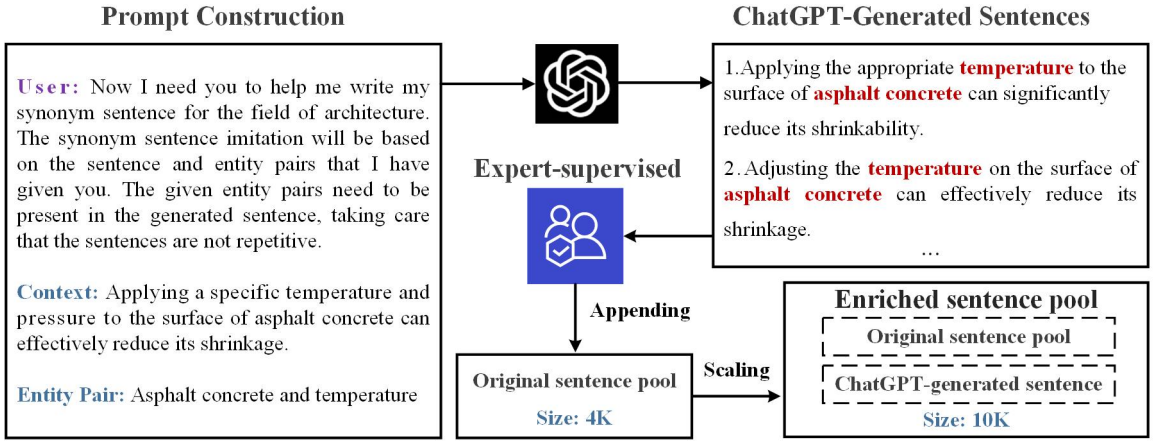


Figure 5. An example prompt representation of data augmentation using ChatGPT.

The whole process was expert-supervised, indicating that all generated sentences are CEM-related, unduplicated, and true positive samples (i.e., sentences with entity-property relations). Through this augmentation, the number of sentences was effectively enriched to 10K. Then, both the traditional DS strategy and the proposed Ont4RE were applied to annotate the 10K sentence pool, which generated two datasets, CONSD [1] and CONSD [2], respectively. The annotation performances using the two DS strategies can be found in Section 4.2.1. Table 2 demonstrates a few examples in the annotated datasets.

Table 2. Annotated data examples.

1. The prestressed cold-drawn rebars of other bridges need to be tested in batches, and the weight of each batch must not exceed 20 tons. （需对其他桥梁的预应力冷拉钢筋进行分批检测，每批重量不得超过20吨）
2. Inject compressed gas into the crack, and adjust the air pressure between 0.2-0.4MPa through the grouting nozzle. （将压缩气体注入裂缝中，通过压浆嘴调节气压在0.2-0.4MPa之间）

instance No.	relation	h_id	h_name	h_pos	t_id	t_name	t_pos
1	has-weight	397	Rebars (钢筋)	[12,14]	3726	Weight (重量)	[23,25]
2	has-pressure	2536	compressd gas (压缩气体)	[1,5]	2037	Pressure (压力)	[28,31]

Each sentence was treated as a single data sample with eight features, namely, instance number, relations, head entity ID

(h_id), head entity name (h_name), head entity position (h_pos), tail entity ID (t_id), tail entity name (t_name), and tail entity position (t_pos).

In addition, the distribution of the two datasets annotated by two DS strategies is shown in Table 3. The total size of CONSD [1] is more than CONSD [2] because the traditional DS strategy without considering semantic features could cause more negative samples (i.e., sentences without any relations) and duplicated entity matching. Stratified sampling was carried out when splitting both datasets to training, validation, and testing sub-datasets with a ratio of 8:2:1. This ensures the proportions of triples of each relation type in the three subsets remained consistent. Figure 6 presents an overview of the dataset after applying the augmentation and sampling.

Table 3. Distribution of the dataset generated through two different DS strategies.

Dataset	Training Set	Val Set	Test Set	Total
CONSD [1]	10923	3117	1572	15612
CONSD [2]	7440	2123	1077	10640

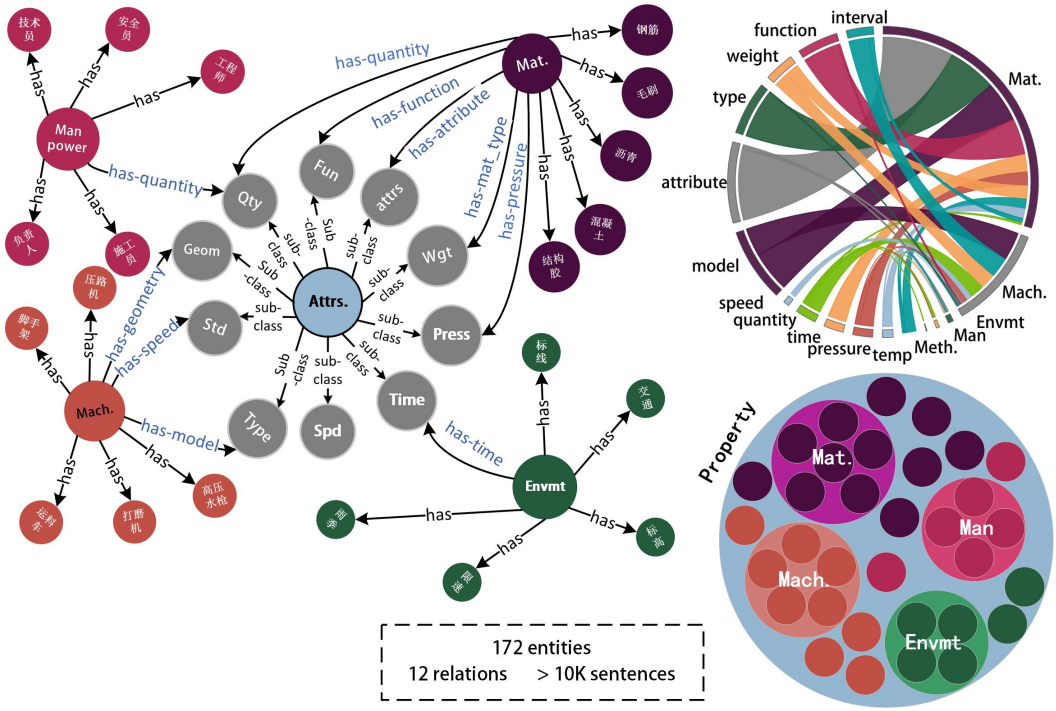


Figure 6. The data example composition of CONSD[2].

4.2 Model implementation and evaluation

The Ont4RE employed different mainstream DL models as the triple extractor, and the performances were cross-compared. Specifically, we tested Text-CNN [64], PCNN-based variants, i.e., PCNN+ONE [9] and PCNN+ATT [37], BERT [43], and BERT-based variants, i.e., SpERT [61], CasRel [62], and RoBERTa [63]. Text-CNN applies the conventional convolutional neural network (CNN) to text data. PCNN stands for piece-wise CNN with at least one (ONE) strategy, which is a classical algorithm and is a benchmark for relation extraction. PCNN+ATT further adds an attention layer (ATT) to dynamically evaluate weights of text tokens. BERT is a pre-trained DL model with an encoder-decoder Transformer structure while also relies on the attention mechanism. SpERT, CasRel, and RoBERTa are the increasingly popular relation extraction SOTA with also a BERT-based architecture.

4.2.1 Evaluation metrics

ERE is essentially a multi-classification task. Thus, the precision (P), recall (R), and F1-Score were included in the evaluation. The area under curve-receiver operating characteristic (AUC-ROC), the area under precision-recall curve

(AUPRC), and P@N values were also reported. More details of the metrics are listed in Table 4. The AUC-ROC (ROC) curve describes the association between true positive rate (TPR) and false positive rate (FPR), assessing the classification ability of a model on the entire dataset. The AUPRC (PR) curve focuses on the capability of predicting positive samples, which is important in handling unbalanced classification [52]. Although the long tail effect was mitigated with data augmentation, the sample size for each relation category was still far less than that of negative samples ('NA'). Thus, if the model can accurately predict most negative samples (true negative), it can cause a decrease in false positive rate and an increase true positive rate, creating the illusion of excellent classification performance in the ROC curve. This is because that model learns more about 'NA' sample features instead of positive samples that we are more interested in. Thus, we adopted the PR curve to prevent excessive influence from negative instances. In this regard, we argue that the model possesses excellent classification ability when dealing with unbalanced samples, if the ROC curve approaches the upper left corner (i.e., high true positive rate, low false positive rate), and the PR curve approaches the upper right corner.

Table 4. The definition of evaluation metrics.

Metrics	Description
$P = \frac{TP}{TP + FP}$	Precision reflects how many entity pairs predicted by the model to have a specific relation actually possess that relation.
$R = \frac{TP}{TP + FN}$	Recall reflects how many of all the entity pairs that truly have a certain relation are predicted by the model.
$F1 = \frac{2PR}{P + R}$	F1-score is the mean of precision and recall, used to comprehensively evaluate the performance of the model.
AUPRC	The area under the curve where the precision and recall metrics form the vertical and horizontal axis, respectively
AUC-ROC	The area under the curve where the TPR and FPR metrics form the vertical and horizontal axis, respectively
P@N	P@N indicates the proportion of correct results in the first N results of retrieval to the total number N.
Extraction Time	Extraction Time serves as an essential indicator in subsequent case study. It refers to the time required for the ERE task to extract relations from the texts.

4.2.2 Automatic annotation results

Before training the relation extractor, we first verified the reliability of the auto-labelling process. We extracted 200 sentence samples from the sentence pool and manually annotated the entity-property relations. These annotations were regarded as the gold standard (i.e., reference benchmark). Subsequently, we applied the Ont4RE and traditional DS to annotate the same 200 samples. Considering the long tail effect, the distribution of different relation types in the selected samples was balanced to ensure the fairness of the evaluation. Finally, the results of the two DS methods with the gold standard are compared using the evaluation metrics defined in Section 4.2.1. As presented in Table 5, the Ont4RE can achieve 83.33% accuracy compared to the gold standard, significantly better than the 74.02% using original DS.

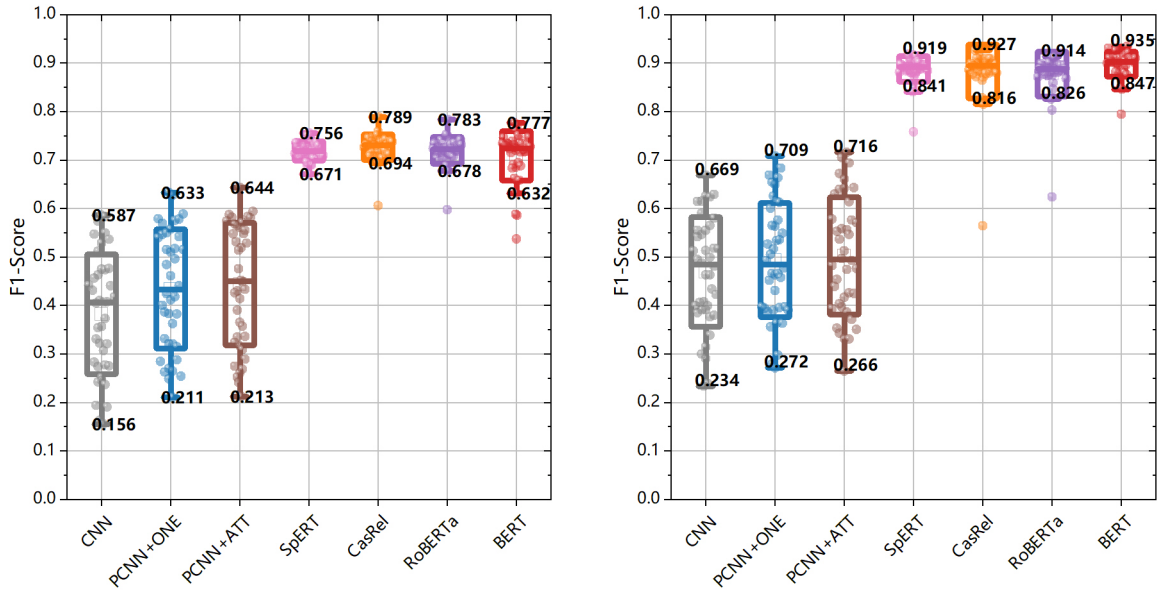
Table 5. The evaluation of the proposed DS annotation

DS annotation	Semantics	Rules needed	Accuracy	Precision	Recall	F1-score
Original DS	-	✓	74.02%	79.89%	89.38%	84.37%
Ont4RE	✓	-	**83.33% (+9.31)**	**95.09% (+15.2)**	**85.64% (-3.74)**	**90.12% (+5.75)**

The results indicate that the ontology-based DS performs better in overall correctness (i.e., all the true and false labels). In terms of precision, Ont4RE reaches 95.09%, compared to 79.89% for the original DS, demonstrating higher annotation accuracy in positive samples. For the recall, although the Ont4RE can miss some positive samples, the comprehensive metric F1-score is still nearly 6% higher than the original DS. It is suggested that the DS performance with Ont4RE is reliable and can achieve semantic-based annotations without predefined rules (i.e., mapping entities that have special characters). Therefore, these annotated data can be effectively used for subsequent experimental research.

4.2.3 Relation extraction results

Table 6 and Figure 7 compare ERE performances between using the On4RE and traditional DS and when different DL models were applied as the triple extractor. For Text-CNN, PCNN+ONE, and PCNN+ATT, the F1 scores are all lower than that of BERT variants in both the original DS and Ont4RE contexts, demonstrating superiority of the transformer structure. Apparently, the Ont4RE outperforms the conventional DS, where the performance of all baseline models is improved on CONSD [2] compared to CONSD [1]. Specifically, the F1-scores of RoBERTa, CasRel, and PCNN-ONE are increased by 13.1%, 13.8%, and 7.6%, respectively. Particularly, the highest F1 (i.e., 0.935) could be reached when the naïve BERT serving as the relation extractor. The results highlight the robust generalization capabilities of Ont4RE, which can effectively integrate with various DL models.



(a) Performance of traditional DS (b) Performance of Ont4RE

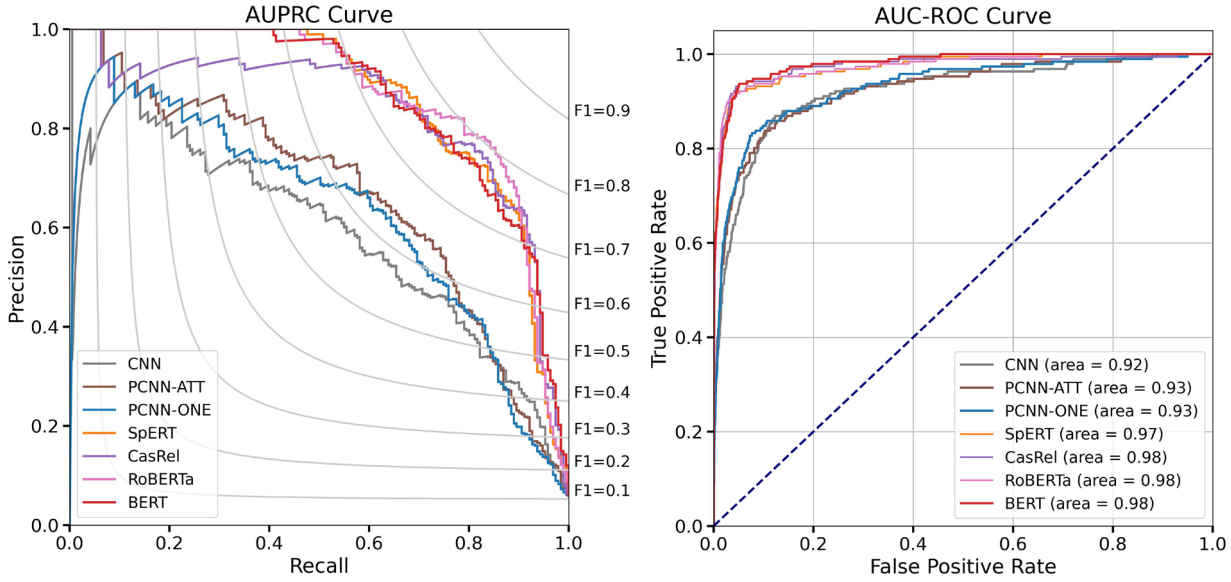
Figure 7. Comparison of F1 score across different models on CONSD [1] (a) and CONSD [2] (b).

Table 6. P@N values and AUPRC of different models on CONSD [1] and CONSD [2].

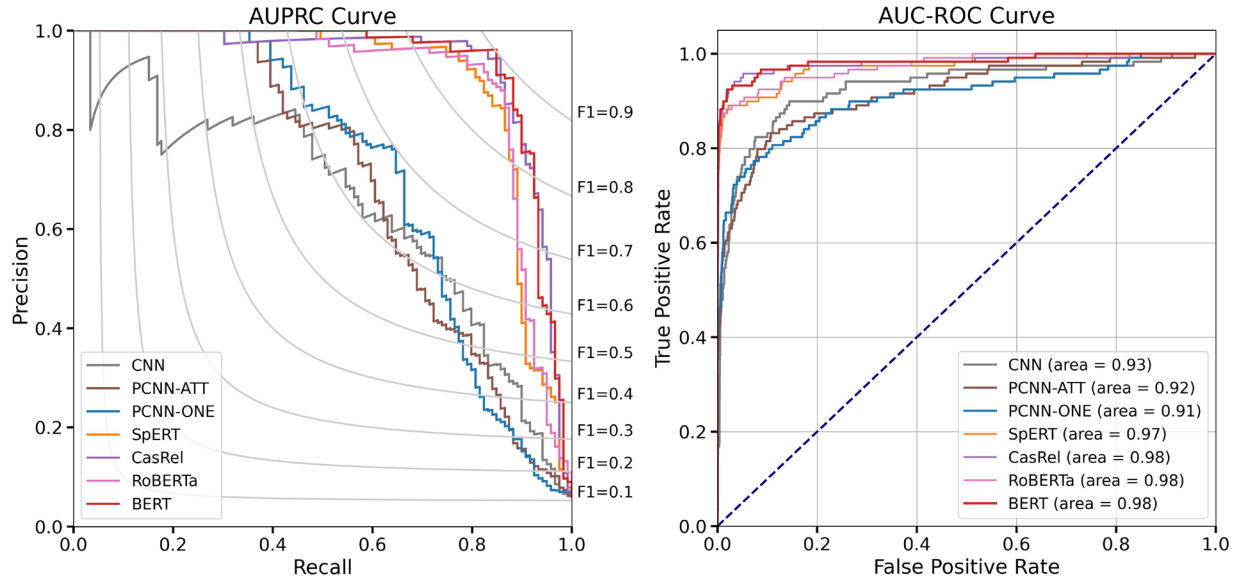
Learning Strategy	Method	P@N				AUPRC (%)	F1-score (%)
	P@N (%)	100	200	300	Mean		
Original DS	CNN	70	55	44	56.3	57.6	58.7
	PCNN+ONE	73	61.5	48.3	60.9	62.2	63.3
	PCNN+ATT	78	62.5	49	63.2	65.8	64.4
	SpERT	94	75	58	77	85.6	75.6
	CasRel	93	77	58.7	76.2	82.7	78.9
	RoBERTa	97	78	58.3	77.8	86.4	78.3
	BERT	**98**	**74.5**	**57.7**	**76.7**	**85.7**	**77.7**
Ont4RE	CNN	78	61.1	49	62.7**(+6.4)**	63.9(+6.3)	66.9+(8.2)
	PCNN+ONE	76	69.9	54.6	66.7(+5.8)	70.7(+8.5)	70.9+(7.6)
	PCNN+ATT	80.5	64.3	50.5	65.1(+1.9)	68.5(+2.7)	71.6+(7.2)
	SpERT	94	78	60	77.3(+0.3)	88.9(+3.3)	91.9+**(16.3)**
	CasRel	96	76.5	63.2	78.6(+2.4)	91.9**(+9.2)**	92.7+(13.8)
	RoBERTa	96	80.5	66.7	81.1(+3.3)	92.4(+6.0)	91.4+(13.1)
	BERT	**96**	**80.2**	**62.1**	**79(+2.7)**	**92.3**(+6.6)	**93.5+(15.8)**

In contrast, P@N is an indicator calculated only on the testing dataset, which pays more attention to the part of the

results with higher confidence. It is a more efficient and acceptable metric for engineers to obtain a few most relevant searching results rather than only one most possible option. From Table 6, the improvements in both P@N (Mean) and AUPRC for the different models persist (compared to the original DS), indicating the effect and generalization capability of Ont4RE. For example, the naïve BERT reaches 92.3% AUPRC compared to 85.7% in the original DS. For P@100, although naïve BERT and RoBERTa are less 1%-2% compared to Ont4RE, there is no doubt that Ont4RE can achieve holistically optimal performance (i.e., F1-score, P@N, AUPRC). Additionally, models with original DS are more likely to learn false negative samples as more falsely annotated relations, so the actual performance is less reliable than the models with Ont4RE. Thus, with Ont4RE, models can more effectively utilize deep semantic knowledge to discern the sentences with true positive entity-property relations. Figure 8 reveals the model performances in terms of AUPRC (PR) and AUC-ROC (ROC) curves using two DS strategies.



(a) Performance of traditional DS



(b) Performance of Ont4RE

Figure 8. PR and ROC curves on two datasets of CONSD [1] (a) and CONSD [2] (b).

Based on PR and ROC results, we again verify that Ont4RE performs better than the conventional DS regardless of the

structure of the downstream relation extractor. Particularly, models with the Transformer structure fits the Ont4RE well and performs best, reaching 92.3% AUPRC and 93.5% F1-score. More details are presented below.

1) For PR curves, the performance improvements when utilizing Ont4RE are evident compared to original DS across different models. Specifically, BERT and RoBERTa performed equally well (with high precision in high recall areas). It indicates that both models can accurately recognize the specific relation among 'non-NA' samples while maintaining a low error rate.
2) For DL models with lower performances, such as CasRel, SpERT, PCNN+ATT, the AUPRC metrics can be improved by up to 9.2% when adopting Ont4RE framework, demonstrating its robust ability when improving DS.
3) The performances of transformer-based variants (i.e., BERT, RoBERTa, CasRel, and SpERT) are closer to the upper rights (PR curves) and the upper lefts (ROC curves) using either the original DS and Ont4RE. It indicates the advantage of bidirectional attention encoders for capturing the deep semantics in text when carrying out ERE.
4) For ROC curves, the CNN-based models reach excellent metrics (i.e., over 90%). However, when we observe both PR curves, models that perform well on the ROC do not stand out. As mentioned before, it is because models are more likely to be affected by features from 'NA' samples, while the transformer-based variants can accurately predict positive samples and their relations amidst most 'NA' samples.
5) The ROC performances of both BERT and CasRel are closer to the upper left corner. It is suggested that with a small number of incorrect predictions ('NA' predicted as 'non-NA'), both can predict more true entity-property relations.

4.2.4 Significant test

To minimize the effects of random variation on performance metrics, we conducted two independent sample t-tests on DL models' F1-score and P@N using the traditional DS strategy and Ont4RE. The t-test evaluates whether there is a significant difference between two sets of experimental results. Thus, we formulated the null hypothesis: "There is no significant difference in the performance (F1 or P@N) of models when using Ont4RE compared to the traditional DS". The alternative hypothesis stated, "The performance (F1 or P@N) of models when using Ont4RE is significantly better than the traditional DS." We set the significance level α at 0.05, which means that the test results can be trusted with 95% confidence. In the test, the F1-score and P@N of each model were obtained by randomly training five times. Since seven models were evolved on both datasets, 70 sets of data were generated, each containing two groups of metrics, the F1-score and P@N. We employed the third-party library $scipy\ stats$ toolkit to perform t-test on the metrics of BERT and the others. The statistical results are shown in Table 7. The t-statistic is a statistic obtained from a paired sample t-test of metrics and is used to measure the significance of the mean difference in both metrics between the traditional DS and Ont4RE. A large t-value indicates a significant difference. The corresponding p-value is the probability of obtaining that t-value, with a low p-value implying that the Ont4RE significantly outperforms the traditional DS strategy.

Table 7. The result of significance test.

Models	F1 t-statistic	F1 p-value	P@N t-statistic	P@N p-value
CNN	4.8333	0.0084	7.6358	0.0016
PCNN+ONE	6.3560	0.0031	17.8639	5.77e-05
PCNN+ATT	5.7603	0.0045	8.2622	0.0012
SpERT	26.0894	1.28e-05	20.9867	3.05e-05
CasRel	20.1878	3.55e-05	15.6427	9.75e-05
RoBERTa	26.3857	1.27e-05	19.0448	4.48e-05
BERT	26.8908	1.14e-05	18.444	5.09e-05

From the Table 7, several insights were drawn: 1) For CNN-based models like CNN, PCNN+ONE, and PCNN+ATT, the t-statistics of F1-score and P@N are ranged from 4.83 to 17.86. It means the performance difference exists between CNN-based models with the traditional DS and those with Ont4RE. Additionally, the corresponding p-values are all lower than

α, which indicates that the performance improvement of CNN-based models adopting Ont4RE is statistically significant. 2) For transformer-based models like SpERT, CasRel, RoBERTa, and BERT, the t-statistic and p-value are superior to those of the CNN-based models, which not only highlight the performance differences between the models adopting Ont4RE and the traditional DS, but also verify that Ont4RE coupled with the transformer-based models can substantially improve the performance of the ERE task.

Given the above results, we could reject the null hypothesis and accept the alternative hypothesis, i.e., the F1-score and P@N of models using Ont4RE were significantly best.

4.3 Controlled experiment in practical project management

Finally, to verify the usefulness and practicality of Ont4RE in CEM, we conducted a practical experiment to compare the performance of Ont4RE, traditional manual searching, and the large language model (LLM) ChatGPT in a routine information searching task. We included ChatGPT as exploring its effects and potential application in construction projects is essential, considering its excellent performance in daily life and several industries such as medicine, education, and finance [53-55].

4.3.1 Experiment description

We selected an ongoing construction project as the subject of the experiment, which must comply with multiple construction regulations and safety standards. First, we collected data from various documents and records of the project to build a small-scale entity-property relation dataset. When creating the dataset, we grouped and stratified each relation category to ensure that all the 12 predefined relation types were covered. Then, we crafted 200 samples, including both positive samples and 'NA' samples. Manual ERE was conducted by 10 CEM professionals, including master's students, doctoral students, and management personnel. As for the ChatGPT, we set up prompts to ask it to work as a construction manager to perform project information searching task using few-shot learning (i.e., giving a few examples to ChatGPT). As shown in Figure 9, the prompt included a description of the task, explanations of the relation types, several demonstrative examples, so that ChatGPT could understand the task requirements and produce prediction results in the

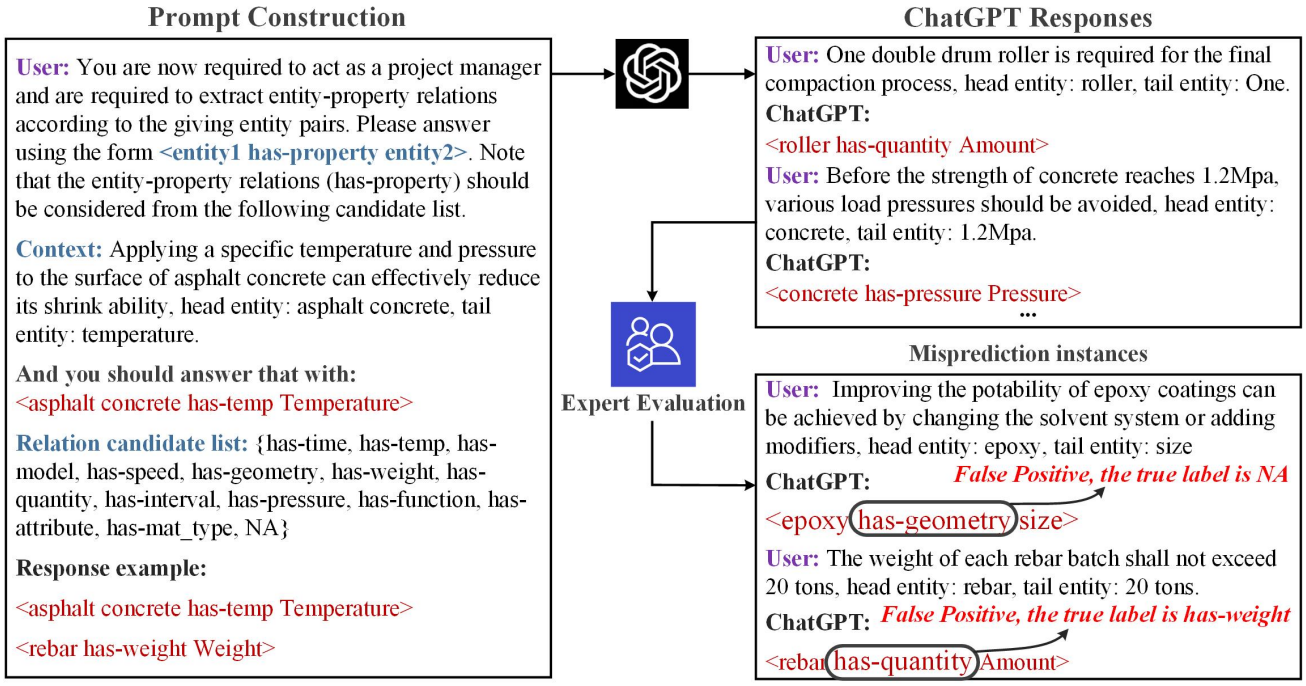


Figure 9. The ChatGPT prompt example and misprediction instances in few-shot relation extraction.

form of relation labels. We meticulously recorded the time to process the same sample, using a stopwatch for humans and

built-in time recorder for ChatGPT and Ont4RE, respectively. When the experiment ended, we summarized and compared the predicted labels and ground truth to evaluate the accuracies.

4.3.2 Method comparison and analysis

In Figure 10, we present confusion matrices of using the three methods to perform the information searching tasks. The vertical and horizontal axis represents true labels and predictions, respectively. The diagonal line represents the true positive rate, while the other cells indicate misclassification.

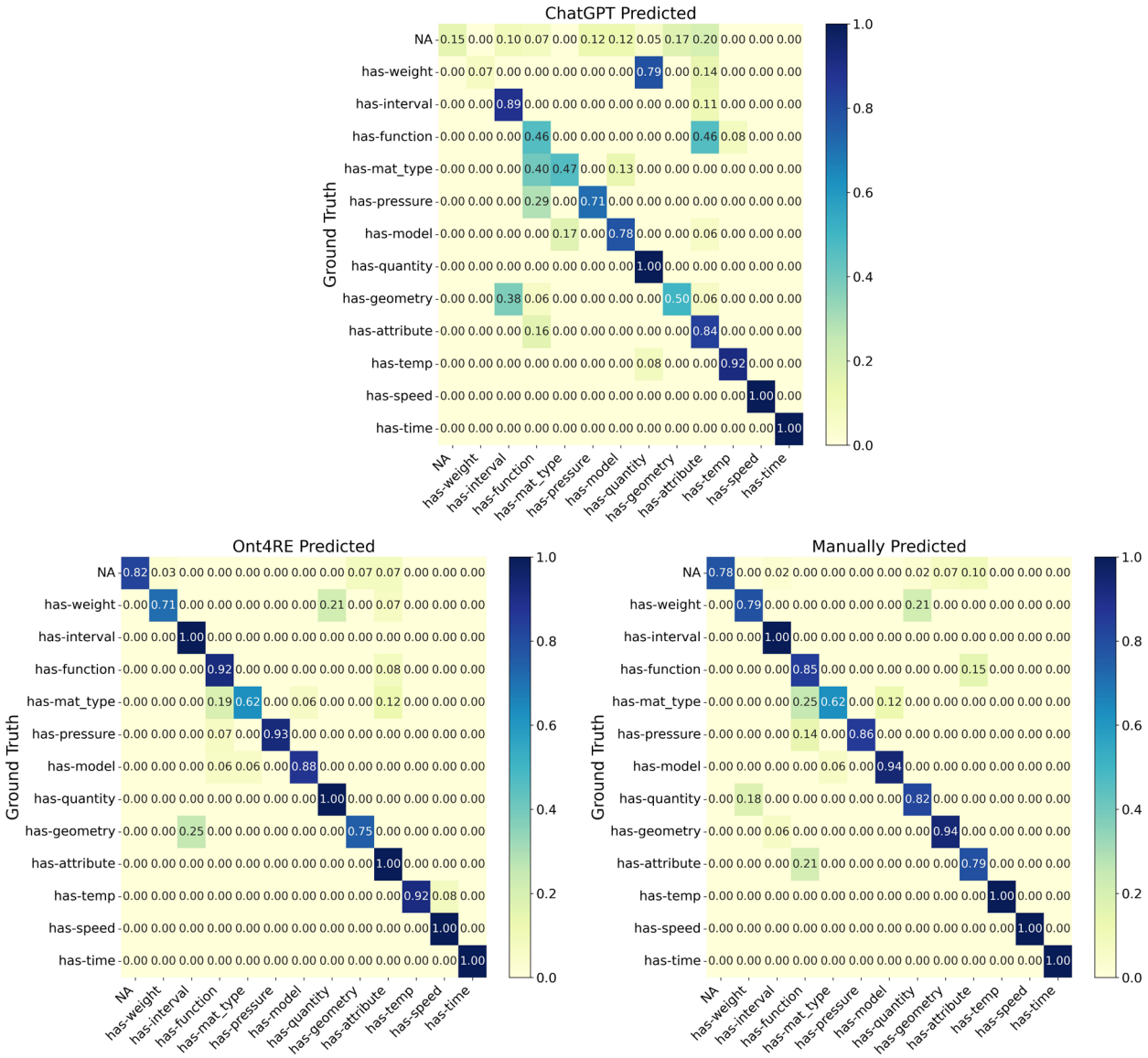


Figure 10. Confusion Matrices of ChatGPT, Ont4RE and Manual Searching.

Compared to ChatGPT, the values in the diagonal area are closer to 1 when using Ont4RE, indicating superior triple extraction performance. ChatGPT performed poorly when dealing with negative samples, where detection rate for ‘NA’ is only 15%, and it erroneously predicted many ‘NA’ negative samples as ‘has-weight,’ ‘has-function,’ and ‘has- geometry’. In addition, as shown in Figure 10, when dealing with ‘has-weight’ relation samples, ChatGPT could reach a high 79% probability of misclassifying them as ‘has-quantity’, such misprediction instances can be seen in the from Figure 9). One reason behind is that the training corpus of ChatGPT is more general rather than domain-specific, akin to having a non-professional carry out this task. For instance, ChatGPT could easily identify simple triples, such as ‘has-speed' and 'has-time'. However, when dealing with more complex semantic relations such as 'has-function,' 'has-interval,' and 'has-model,'

its performance is inferior to Ont4RE trained on specific construction corpora. Besides, Ont4RE features a misclassification rate of only 18% when dealing with 'NA' samples, proving its robustness in the ERE task. Nevertheless, it is still possible to misclassify 'has-weight' and 'has-geometry' samples as 'has-quantity' and 'has-interval,' respectively. It is challenging for the model to distinguish descriptions of quantification well when capturing sentence features, especially those with mathematical expressions.

On the other hand, the overall performance of manual searching is comparable to Ont4RE. However, its ability to distinguish negative samples (78%) is slightly inferior to the latter. In addition, manual searching might misjudge some easily confused relations, such as 'has-weight' and 'has-quantity,' 'has-function' and 'has-attribute,' 'has-mat_type' and 'has-model'. Figure 10 reflects the phenomenon with a symmetrical distribution characteristic along the diagonal line, which Ont4RE rarely exhibits. This is because Ont4RE can utilize context to learn semantics, which can avoid misjudgment caused by the inherent ambiguity of semantics. Specific ERE performance metrics of the three methods are listed in Table 8. For most types of relations, Ont4RE and manual searching could outperform ChatGPT, in terms of precision and recall, achieving higher F1 scores of 87.3% and 86.1%, respectively. However, the time for information retrieval is much longer than Ont4RE and ChatGPT, reaching 98 seconds. It is because ERE task requires recognizing keywords from numerous project documents, then searching and judging constraints between entities, which are time-consuming activities when being undertaken by human. Furthermore, due to the long processing and Internet lag when using the API of ChatGPT, it generally requires more time than the Ont4RE. As such, the precision, recall, and F1 scores of Ont4RE are superior to the competitors, and it also requires shorter execution time, demonstrating its significant advantages in practical ERE.

Table 8. The performance result in the controlled experiment of ERE.

Method	Precision	Recall	F1-score	Extraction Time
ChatGPT	69.9	67.6	61.0	3.2s
Manual Searching	86.2	87.5	86.1	98s
Ont4RE	**87.6**	**88.9**	**87.3**	**2.5s**

5. Discussion

We develop a computational framework named Ont4RE that can realize accurate and automatic entity-property triple extraction in CEM. The contributions of our study are twofold.

Firstly, in terms of computational novelty, Ont4RE improves the DS strategy by leveraging a domain ontology, i.e., CEMO. Traditionally, the fully supervised ERE involves large human annotation, and the current DS strategies produce excessive noise due to its strong assumption, which also needs an external large-scale KB as a supervision source, thereby further increasing the application requirements. Recent studies adopt generative AI (e.g., ChatGPT) to generate or enrich the external KB [65], which contains only general-world information and is not capable of supervising the annotation of domain-specific texts. In contrast, the Ont4RE integrates domain knowledge encoded in the ontology into the DS process to realize automatic annotation without developing external KBs. The DL-based relation extractor with Ont4RE as the upper stream annotator can improve by an average of 6.1% AUPRC and 11.7% F1-score than those using traditional DS. Specifically, the classes and relations in CEMO can be regarded as a highly compressed version of a KB, and the entity-class mapping is realized by measuring semantic similarities rather than text-based matching in conventional DS. As such, the annotation accuracy is increased by 9.3% compared to conventional DS, because the general and unambiguous semantics in the ontology can effectively handle the noise caused by entities with similar semantics but different names. Moreover, knowledge in the CEMO is domain-specific while cross-project applicable, the strategy can be flexibly transferred to different projects in the CEM sector without re-developing or continuous updating the ontology like the conventional DS usually does.

Secondly, from an application perspective, the study is the pioneer that explores the framework combining DS and

ontology for enabling DL models in ERE tasks in the construction industry. We proved that the Ont4RE is adaptable to different DL structures (e.g., CNN-based and Transformer-based) serving as the relation extractor. More importantly, we found that BERT (including its variants) is the most compatible downstream structure, which achieves the best ERE performance (i.e., 92.3% AUPRC and 93.5% F1-score). We verify the usefulness of Ont4RE in a controlled experiment mimicking practical information searching in CEM. The Ont4RE outperformed both the traditional manual searching and ChatGPT in terms of accuracy and efficiency. For one thing, we attribute the superior performance of Ont4RE over manual searching to various subjective and objective factors. For instance, fatigue commonly occurs when many items need to be checked by human engineers, and the variability in expertise also lead to inconsistencies of searching results. For another, many argued that LLMs are capable of handling NLP tasks in various domain, such as medicine, education, and finance [53-55]. However, LLMs are developed to understand human semantics and generate data instead of searching and extracting information. As such, even the SOTA ChatGPT significantly underperforms in the experiments compared to Ont4RE (the F1-score is 26.3% less than that of Ont4RE). Although LLMs can be fine-tuned using domain corpus, it requires excessively more high-quality labelled data and computation power. Thus, the Ont4RE is the more feasible and cost-effective option in automating ERE and creating values for CEM. Last but not least, we develop two datasets (i.e., CONSD [1] and CONSD [2]) especially for training and testing ERE methods coupled with DS strategies. As there is few public corpus in CEM, the datasets are also valuable for developing or applying relevant methods in the industry.

Nonetheless, there are some limitations to our study. Firstly, the study is primarily based on Chinese data. Despite that the Ont4RE is not language-bound and can be applied to any language corpus, we will construct datasets in more languages (e.g., English) to further verify the approach and shed lights on cross-language applications. Secondly, Ont4RE faces challenges when dealing with sentences containing complex quantitative mathematical expressions. For example, Ont4RE can misjudge 'has-weight' and 'has-geometry' as 'has-quantity' and 'has-interval.' Future work will consider introducing more complex model structures, such as developing domain LLMs and sophisticated knowledge engineering to improve such complex semantic understanding. Besides, due to the LLMs' increasing window size of the input token and powerful context awareness, it is worth exploring its potential as a DS annotator in the context of domain LLMs to improve triple labelling accuracy. Despite the limitations, our research still provides a new effective method for ERE in CEM. We look forward to further improving our method in future work and applying it to broader scenarios and language environments.

Conclusion

Automatically and efficiently identifying and extracting entity-property triples from text documents is an urgently needed function in the CEM sector. This study introduces a novel computational framework, Ont4RE, to extract entity-property relations from CEM documents. Firstly, we construct an ontology called CEMO, which contains domain-specific classes and relations in an unambiguous structured topological form. Secondly, we leverage the CEMO to automatically recognize and annotate potential triples in sentences by mapping the semantics between text tokens and ontology classes. Third, we develop a DL-based downstream relation extractor, which is trained on bags of sentences labelled by the ontology-based DS and predicts valid entity-property triples. Through comprehensive experiments and significant testing, we found that the proposed framework can improve DL models by an average of 6.1% AUPRC, 3.3% mean P@N, and 11.7% F1-score. Additionally, Ont4RE is adaptable to various DL structures as the relation extractors and consistently outperforms the conventional DS strategy. The models with the Transformer structure, such as BERT, reaches the best performance, i.e., 92.3% AUPRC and 93.5% F1-score. We also conducted a controlled experiment mimicking practical information searching. The results show that compared to traditional manual searching and the SOTA AI tool ChatGPT, Ont4RE possesses accuracy and efficiency advantages while substantially reducing the searching time. In summary, the Ont4RE framework offers an innovative and cost-effective solution for ERE tasks in current construction projects, exhibits practical value in real-world CEM routines by saving information searching time for engineers and facilitating relevant

management activities, and sheds insights on improving information extraction in relevant fields.

Acknowledgements

The work described in this paper is supported by grants from the National Natural Science Foundation of China (52208324, 52077213, 62003332), China Postdoctoral Science Foundation (2022M713275), the Guangdong Basic and Applied Basic Research Foundation (2023A1515011162), Youth Innovation Promotion Association CAS (2021358), Shenzhen Excellent Innovative Talents (RCYX20221008093036022), and Shenzhen Science and Technology Program (RCBS20221008093307015 and JSGG20220831105800002).

References

[1] Akhavian, R., & Behzadan, A. H. (2015). Construction equipment activity recognition for simulation input modeling using mobile sensors and machine learning classifiers. Advanced Engineering Informatics, 29(4), 867-877..

[2] Wu, C., Li, X., Guo, Y., Wang, J., Ren, Z., Wang, M., & Yang, Z. (2022). Natural language processing for smart construction: Current status and future directions. Automation in Construction, 134, 104059.

[3] Zhong, B., Xing, X., Luo, H., Zhou, Q., Li, H., Rose, T., & Fang, W. (2020). Deep learning-based extraction of construction procedural constraints from construction regulations. Advanced Engineering Informatics, 43, 101003.

[4] Al Qady, M., & Kandil, A. (2010). Concept relation extraction from construction documents using natural language processing. Journal of construction engineering and management, 136(3), 294-302.

[5] Zhang, J., & El-Gohary, N. M. (2016). Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking. Journal of Computing in Civil Engineering, 30(2), 04015014.

[6] Nayak, T., Majumder, N., Goyal, P., & Poria, S. (2021). Deep neural approaches to relation triplets extraction: A comprehensive survey. Cognitive Computation, 13, 1215-1232.

[7] Tixier, A. J. P., Vazirgiannis, M., & Hallowell, M. R. (2016). Word embeddings for the construction domain. arXiv preprint arXiv:1610.09333.

[8] Mintz, M., Bills, S., Snow, R., & Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (pp. 1003-1011).

[9] Riedel, S., Yao, L., & McCallum, A. (2010). Modeling relations and their mentions without labeled text. In Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III 21 (pp. 148-163). Springer Berlin Heidelberg.

[10] El-Gohary, N. M., & El-Diraby, T. E. (2010). Domain ontology for processes in infrastructure and construction. Journal of Construction Engineering and Management, 136(7), 730-744.

[11] Wang, H., & Meng, X. (2019). Transformation from IT-based knowledge management into BIM-supported knowledge management: A literature review. Expert Systems with Applications, 121, 170-187.

[12] Jiang, J., Yang, Z., Wu, C., Guo, Y., Yang, M., & Feng, W. (2023). A compatible detector based on improved YOLOv5 for hydropower device detection in AR inspection system. Expert Systems with Applications, 225, 120065.

[13] Yamaguchi, K., Asahi, R., & Sasaki, Y. (2022). Superconductivity information extraction from the literature: A new corpus and its evaluations. Advanced Engineering Informatics, 54, 101768.

[14] Sun, J., Lei, K., Cao, L., Zhong, B., Wei, Y., Li, J., & Yang, Z. (2020). Text visualization for construction document information management. Automation in construction, 111, 103048.

[15] Yang, C., Zheng, Y., Tu, X., Ala-Laurinaho, R., Autiosalo, J., Seppänen, O., & Tammi, K. (2023). Ontology-based knowledge representation of industrial production workflow. Advanced Engineering Informatics, 58, 102185.

[16] Mohemad, R., Hamdan, A. R., Othman, Z. A., & Mohamad Noor, N. M. (2011). Ontological-based information

extraction of construction tender documents. In Advances in Intelligent Web Mastering–3: Proceedings of the 7th Atlantic Web Intelligence Conference, AWIC 2011, Fribourg, Switzerland, January, 2011 (pp. 153-162). Springer Berlin Heidelberg.

[17] Fan, H., Xue, F., & Li, H. (2015). Project-based as-needed information retrieval from unstructured AEC documents. Journal of Management in Engineering, 31(1), A4014012.

[18] Leng, J., & Jiang, P. (2016). Mining and matching relationships from interaction contexts in a social manufacturing paradigm. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47(2), 276-288.

[19] Zhou, Z., Wei, L., Yuan, J., Cui, J., Zhang, Z., Zhuo, W., & Lin, D. (2023). Construction safety management in the data-rich era: A hybrid review based upon three perspectives of nature of dataset, machine learning approach, and research topic. Advanced Engineering Informatics, 58, 102144.

[20] Blanco, J. L., Dohrmann, T., Julien, J. P., Law, J., & Palter, R. (2019). Governments can lead construction into the digital era. New York: Capital Projects and Infrastructure.

[21] Blanco, J. L., Rockhill, D., Sanghvi, A., Law, J., and Torres, A. (2023). From start-up to scale-up: Accelerating growth in construction technology. Private Equity and Principal Investors Practice.

[22] Na, X. U., Ling, M. A., Liu, Q., Li, W. A. N. G., & Deng, Y. (2021). An improved text mining approach to extract safety risk factors from construction accident reports. Safety science, 138, 105216.

[23] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. In The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007+ ASWC 2007, Busan, Korea, November 11-15, 2007. Proceedings (pp. 722-735). Springer Berlin Heidelberg.

[24] Chen, P. P. S. (1976). The entity-relationship model—toward a unified view of data. ACM transactions on database systems (TODS), 1(1), 9-36.

[25] Lin, Y., Liu, Z., & Sun, M. (2016). Knowledge representation learning with entities, attributes and relations. ethnicity, 1, 41-52.

[26] Noy, N. F., & McGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology.

[27] Cui, Y., Che, W., Liu, T., Qin, B., & Yang, Z. (2021). Pre-training with whole word masking for chinese bert. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29, 3504-3514.

[28] Jallan, Y., & Ashuri, B. (2020). Text mining of the securities and exchange commission financial filings of publicly traded construction firms using deep learning to identify and assess risk. Journal of Construction Engineering and Management, 146(12), 04020137.

[29] Li, S., Cai, H., & Kamat, V. R. (2016). Integrating natural language processing and spatial reasoning for utility compliance checking. Journal of Construction Engineering and Management, 142(12), 04016074.

[30] Ur-Rahman, N., & Harding, J. A. (2012). Textual data mining for industrial knowledge management and text classification: A business oriented approach. Expert Systems with Applications, 39(5), 4729-4739.

[31] Zheng, Z., Lu, X. Z., Chen, K. Y., Zhou, Y. C., & Lin, J. R. (2022). Pretrained domain-specific language model for natural language processing tasks in the AEC domain. Computers in Industry, 142, 103733.

[32] Chen, C., Wang, T., Zheng, Y., Liu, Y., Xie, H., Deng, J., & Cheng, L. (2023). Reinforcement learning-based distant supervision relation extraction for fault diagnosis knowledge graph construction under industry 4.0. Advanced Engineering Informatics, 55, 101900.

[33] Ajayi, A., Oyedele, L., Owolabi, H., Akinade, O., Bilal, M., Davila Delgado, J. M., & Akanbi, L. (2020). Deep learning models for health and safety risk prediction in power infrastructure projects. Risk Analysis, 40(10), 2019-2039.

[34] Wang, Z., Zhang, B., & Gao, D. (2022). A novel knowledge graph development for industry design: A case study on indirect coal liquefaction process. Computers in Industry, 139, 103647.

[35] Roth, B., Barth, T., Wiegand, M., & Klakow, D. (2013). A survey of noise reduction methods for distant supervision. In Proceedings of the 2013 workshop on Automated knowledge base construction (pp. 73-78).

[36] Zeng, D., Liu, K., Chen, Y., & Zhao, J. (2015). Distant supervision for relation extraction via piecewise convolutional neural networks. In Proceedings of the 2015 conference on empirical methods in natural language processing (pp. 1753-1762).

[37] Lin, Y., Shen, S., Liu, Z., Luan, H., & Sun, M. (2016). Neural relation extraction with selective attention over instances. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 2124-2133).

[38] Ji, G., Liu, K., He, S., & Zhao, J. (2017). Distant supervision for relation extraction with sentence-level attention and entity descriptions. In Proceedings of the AAAI conference on artificial intelligence (Vol. 31, No. 1).

[39] Hong, L., Lin, J., Li, S., Wan, F., Yang, H., Jiang, T., ... & Zeng, J. (2020). A novel machine learning framework for automated biomedical relation extraction from large-scale literature repositories. Nature Machine Intelligence, 2(6), 347-355.

[40] Zeng, Z., Yao, Y., Liu, Z., & Sun, M. (2022). A deep-learning system bridging molecule structure and biomedical text with comprehension comparable to human professionals. Nature communications, 13(1), 862.

[41] Alt, C., Hübner, M., & Hennig, L. (2019). Fine-tuning Pre-Trained Transformer Language Models to Distantly Supervised Relation Extraction. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 1388-1398).

[42] Ru, C., Tang, J., Li, S., Xie, S., & Wang, T. (2018). Using semantic similarity to reduce wrong labels in distant supervision for relation extraction. Information Processing & Management, 54(4), 593-608.

[43] Kenton, J. D. M. W. C., & Toutanova, L. K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of naacL-HLT (Vol. 1, p. 2).

[44] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

[45] Xu, X., & Cai, H. (2021). Ontology and rule-based natural language processing approach for interpreting textual regulations on underground utility infrastructure. Advanced Engineering Informatics, 48, 101288.

[46] 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.

[47] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[48] Wu, C., Wu, P., Wang, J., Jiang, R., Chen, M., & Wang, X. (2021). Ontological knowledge base for concrete bridge rehabilitation project management. Automation in construction, 121, 103428.

[49] Li, X., Wu, C., Xue, F., Yang, Z., Lou, J., & Lu, W. (2022). Ontology-based mapping approach for automatic work packaging in modular construction. Automation in Construction, 134, 104083.

[50] Rajpathak, D., Xu, Y., & Gibbs, I. (2020). An integrated framework for automatic ontology learning from unstructured repair text data for effective fault detection and isolation in automotive domain. Computers in Industry, 123, 103338.

[51] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.

[52] Davis, J., & Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. In Proceedings of the 23rd international conference on Machine learning (pp. 233-240).

[53] Vaishya, R., Misra, A., & Vaish, A. (2023). ChatGPT: Is this version good for healthcare and research?. Diabetes & Metabolic Syndrome: Clinical Research & Reviews, 17(4), 102744.

[54] Dalalah, D., & Dalalah, O. M. (2023). The false positives and false negatives of generative AI detection tools in education and academic research: The case of ChatGPT. The International Journal of Management Education, 21(2),

100822.

[55] Leippold, M. (2023). Sentiment spin: Attacking financial sentiment with GPT-3. Finance Research Letters, 103957.

[56] Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

[57] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (pp. 3982-3992).

[58] Liu, M., Luo, X., Wang, G., & Lu, W. Z. (2023). Intelligent information extraction from government on-site inspection reports of construction projects: A graph-based text mining approach. Advanced Engineering Informatics, 58, 102163.

[59] Wu, C., Li, X., Jiang, R., Guo, Y., Wang, J., & Yang, Z. (2023). Graph-based deep learning model for knowledge base completion in constraint management of construction projects. Computer-Aided Civil and Infrastructure Engineering, 38(6), 702-719.

[60] Ye, Z. X., & Ling, Z. H. (2019). Distant Supervision Relation Extraction with Intra-Bag and Inter-Bag Attentions. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 2810-2819).

[61] Eberts, M., & Ulges, A. (2019). Span-based joint entity and relation extraction with transformer pre-training. In Proceedings of the 24th European Conference on Artificial Intelligence (ECAI) (pp. 2006-2013).

[62] Wei, Z., Su, J., Wang, Y., Tian, Y., & Chang, Y. (2020). A Novel Cascade Binary Tagging Framework for Relational Triple Extraction. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 1476-1488).

[63] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

[64] Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics.

[65] Li, J., Jia, Z., & Zheng, Z. (2023). Semi-automatic Data Enhancement for Document-Level Relation Extraction with Distant Supervision from Large Language Models. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (pp. 5495-5505).