BEKG: A built environment knowledge graph

Xiaojun Yang, Haoyu Zhong, Zhengdong Wang, Penglin Du, Keyi Zhou, Heping Zhou, Xingjin Lai, Yik Lun Lau, Yangqiu Song and Liyaning Tang

School of Information Engineering, Guangdong University of Technology, Guangzhou, People’s Republic of China; Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, People’s Republic of China; School of Architecture and Built Environment, The University of Newcastle, Callaghan, Australia

ABSTRACT
In recent years, the digitalization of the built environment has progressed rapidly due to the development of modern design and construction technologies. However, the need for extensive professional knowledge in this field has not been met by practitioners and scholars. To address this problem, a study was conducted to build a knowledge graph in the built environment domain, which stores entities and their connections in a graph data model. To achieve it, this research collected more than 80,000 paper abstracts from the built environment domain. To ensure the accuracy of entities and relationships in the knowledge graph, two well-annotated datasets were created with 29 types of relationships, each containing 2000 and 1450 instances, respectively, for Named Entity Recognition (NER) and relationship extraction (RE) tasks. Two BERT-based models were trained on these datasets and achieved over 85% accuracy in both tasks. Using these models, over 200,000 high-quality relationships and entities were extracted from abstract data. This comprehensive knowledge graph will help practitioners and scholars better understand the built environment domain.

Introduction
Data plays a vital role in driving economic growth and social progress. Studies demonstrate numerous applications of artificial intelligence (AI) in analyzing data in the built environment. For instance, Wei et al. (2018) used machine learning models to predict city-level electricity consumption, improving demand-side management. Yu et al. (2010) proposed decision tree models for forecasting building energy demand. Newton (2018) utilized optimization algorithms to classify 3D building features, while Ghadai et al. (2018) developed a deep-learning framework for anticipating challenges in borehole fabrication. The trend is towards using advanced AI tools for analysis and forecasting in this field.

However, improving data analysis and forecasting tools alone is insufficient to support decision-making for future built environment projects. Data collection and knowledge management are essential due to the interdisciplinary nature of such projects. Effective knowledge management assists project managers in making informed decisions and can be reused in other construction projects. For example, past project knowledge aids in future risk assessment (Nieto-Morote & Ruz-Vila, 2011; Tah, 2001) and analyzing cooperation networks (Sun et al., 2019). As project knowledge expands, effective management becomes crucial. Udeaja et al. (2008) developed a prototype to capture and reuse project knowledge, while Ozorhon et al. (2014) proposed a web-based tool for creating organizational memory.

In addition to knowledge management, data collection is crucial for successful project management. However, processing the massive, multi-sourced, and heterogeneous data becomes challenging as its volume increases. In the built environment field, most raw data is unstructured and not directly usable. Extracting valuable information from this data using traditional methods like rule-based systems and expert intervention is time-consuming and cannot meet business needs.

In view of the exponential data growth, knowledge mapping tools have been developed to aid knowledge management and decision-making throughout project management to boost efficiency and effectiveness in communications. Developing big data and knowledge management tools will be beneficial to facilitate scholars, practitioners, and decision-makers to quickly acquire knowledge from existing projects and make...
reasonable decisions in the field of the built environment.

One way to manage knowledge intuitively is through Knowledge Graphs (KGs). KGs organize and present structured knowledge by linking entities and defining relationships, facilitating understanding and management. In the built environment field, there is specialized terminology. Manual extraction of these terms and relations is accurate but time-consuming. To address scalability, an automated KG construction approach is preferred. This ensures accurate entity recognition and relation classification while benefiting from expert annotations. Initially proposed by Google in 2012 to enhance its search engine, KGs have been widely adopted for efficient knowledge management in various information systems. KGs can be categorized as general or domain-specific, with significant advancements in recent years.

A general KG typically contains knowledge from various sectors and scenarios, much of which is common knowledge. Thus, a general KG is suitable for recommendation systems and question-answering (QA) systems targeting general users. Existing general KGs like Freebase (Bollacker et al., 2008), DBpedia (Auer et al., 2007), YAGO (Suchanek et al., 2007), ConceptNet (Liu & Singh, 2004), and Microsoft Concept Graph (Ji et al., 2019) have demonstrated strong potential in enhancing knowledge acquisition efficiency across numerous applications.

A domain-specific knowledge graph (KG) differs from a general KG as it offers specialized knowledge tailored to specific industries, targeting experts rather than the general public. Its purpose is to support complex analysis applications and decision-making processes, requiring precise and professional knowledge. Domain-specific KGs have been developed in various fields, including medical (Ernst et al., 2014; Goodwin et al., 2013; Rotmensch et al., 2017; Shi et al., 2017), cyber security (Jia et al., 2018), financial (Elnagdy et al., 2016; Liu et al., 2019), common sense (Zhang et al., 2020), and news (Ciampaglia et al., 2015; Rospocher et al., 2016; Rudnik et al., 2019), providing specialized knowledge to users.

KGs have practical applications in various domains. For instance, cybersecurity experts use KGs for defence (Piplai et al., 2020), while a geographic knowledge representation model extracts information on features (Wang et al., 2019). KGs support knowledge mapping in higher education (Yu et al., 2021) and aid in investigating machinery failures and suggesting repairs in industrial maintenance (Qin & Jin, 2020). In the oil and gas industry, KGs integrate domain-specific knowledge to enhance exploration and development management (Zhou et al., 2020). Public service benefits from KGs, such as transforming legal information systems into legal KGs (Filtz et al., 2021). Moreover, KGs offer enhanced interoperability in the built environment, supporting decision-making throughout the building life cycle (Pauwels, 2014). An information management framework similar to a KG effectively manages construction project information (Olawumi & Chan, 2019), helping achieve management goals, including improving information sharing, enhancing decision support, optimizing resource management, promoting innovation, and fostering standardization in the construction industry.

In this paper, various methods have been employed to construct a KG in the domain of the built environment. The objective is to create a highly accurate KG that facilitates knowledge acquisition for scholars, practitioners, and decision-makers. The major achievements include leveraging large-scale text data to build a domain-specific KG, which reduces labour costs and time involved in the creation process, while also yielding promising results. Compared to traditional data management approaches in the built environment field, the KG approach offers greater intuitiveness and scalability for knowledge acquisition and management. The source code is found at: https://github.com/HKUST-KnowComp/BEKG.

Related works

The KG construction process could generally be divided into two core steps, data collection and information extraction. The data collection step helped filter out undesirable data and retain valuable data, while the information extraction step revealed meaningful data and transformed it into a structured data format.

Dataset collection

The dataset collection stage poses challenges in defining unknown relation types. Unsupervised machine learning algorithms, like clustering, can detect patterns and reveal relations in a large-scale corpus. Hasegawa et al. (2004) proposed an unsupervised machine-learning approach for relation extraction based on identical semantic entities. Rozenfeld and Feldman (2006) improved upon this by clustering entity pairs in the same corpus and eliminating candidate pairs with multiple relations. Plank and Moschitti (2013) achieved domain adaptation by using artificially constructed features like word clustering and dependency trees.

Information extraction

Various machine learning methods, including Integer Linear Programming (ILP) and semi-supervised
learning, can be used for information extraction. Roth and Yih (2004; 2007) proposed an ILP approach for solving information extraction tasks. Semi-supervised learning can train on a small set of seed-labelled training samples to predict a large number of unlabelled samples, reducing the manual labelling efforts required to obtain a classification model. The Dual Iterative Pattern Relation Extraction (DIPRE) system (Brin, 1999) was an early proposed semi-supervised entity relation extraction method based on bootstrapping with a few labelled samples. Yang et al. (2021) applied DIPRE’s techniques for acquiring new information.

Meanwhile, Mintz et al. (2009) proposed leveraging distant supervision signals from a remote knowledge base for entity relation extraction tasks. By aligning the data with the information in the remote knowledge base, labelled samples are obtained for the massive data in the open domain. Etzioni et al. (2008) proposed open information extraction and the first domain-independent Open Information Extraction (OIE) system, TextRunner, which could be extended to a large-scale Web corpus. Fader et al. (2011) defined lexical and grammatical constraints on binary relation verbs, improving the problem of incoherence and insufficient information in the OIE system.

In recent years, deep learning has gained popularity in improving information extraction. Socher et al. (2012) and Zeng et al. (2014) proposed Recursive Neural Network (RNN) and Convolutional Neural Network (CNN) methods, respectively. Zeng et al. (2015) used multi-instance learning to address noise in distant supervision, while Jiang et al. (2016) proposed a multi-instance, multi-label CNN model for solving relation overlapping. Miwa and Bansal (2016) and Kattiyar and Cardie (2017) represented entities and relations using neural networks, with the latter incorporating attention mechanisms. Although deep learning shows promise, it requires large amounts of labelled data. Few-shot learning techniques were proposed by Gao et al. and Han et al. (2018), and Kolluru et al. (2020) developed a neural OpenIE system with iterative labelling. Huguet Cabot and Navigli (2021) introduced REBEL, a seq2seq model for end-to-end relation classification.

**Knowledge graph used in the built environment**

KGs are suitable for domain knowledge management and visualization as they can effectively organize entities and relations. Pauwels and Costin (2022) analyzed the application of KG representation in the built environment, including product data, sensor data, and 3D geometric data. They found that KGs are beneficial for managing and visualizing product data. Additionally, sensor data, when contextualized with the built asset, can also be effectively represented in KGs. However, representing Constructive Solid Geometry (CSG) in KGs for 3D geometric data may not be meaningful, but a condensed mesh representation can serve as a suitable alternative.

In the field of the built environment, various ontologies have been developed to cater to different data categories. For the Architecture, Engineering, and Construction (AEC) domain, the IFC (Industry Foundation Class) Ontology (Pauwels et al., 2017) and the Building Topology Ontology (Rasmussen et al., 2021) have been established. In the smart building domain, ontologies such as Brick, Haystack Tagging Ontology (Charpenay et al., 2015), SAREF4Building Ontology (Poveda-Villallon & Garcia-Castro, 2020), and Real Estate Core Ontology have been created. However, there is still a lack of a comprehensive concept ontology and knowledge graph in the built environment, which hinders systematic knowledge management for scholars, practitioners, and decision-makers. This gap may become a new research interest.

**Research methodology**

KG is a graph-based knowledge base that represents entities and their relations. It uses an ‘entity-relation-entity’ triple format as the primary unit, forming a graph that describes complex concepts in the real world. In Figure 1, a typical subgraph illustrates this idea. Nodes represent entities, representing concepts and real-world objects, while edges represent relations between entities. Arrows indicate the direction of the relations. For example, the concept of ‘thermal comfort’ is influenced by ‘acoustic conditions’ and can be analyzed through ‘physical measurements’. This knowledge is represented by triples like ‘acoustic conditions, affect, thermal comfort’ and ‘physical measurements, analyze, thermal comfort’.

The research methodology flow chart is shown in Figure 2. The first step was data acquisition and processing, involving 85,000 abstracts from 54 journals in the built environment field. A subset of 10,000 abstracts was sampled and pre-processed for relation analysis, determining the schema for the Built Environment Knowledge Graph (BEKG) and enhancing its accuracy and comprehensiveness. Next, 50 instances of each relation in the schema were annotated, and a dataset was created for NER and Relation Classification model training. The trained model was then applied to the unlabelled data to extract information from the remaining journal abstracts. Finally, the extracted entities and relations were used to develop the BEKG.
Data acquisition and processing

Journal abstracts in the built environment field were the base of the BEKG construction presented in this paper. These raw data however required data preprocessing before putting data into knowledge graph construction, the overall steps include data collection, cleaning, and pre-extraction.

Data collection and cleaning

Abstracts in the built environment field provide valuable insights into research work in this domain. They serve as concise summaries of research elements, including problems, methodologies, and results. Abstract data is easily accessible due to the availability of electronic publications, making it a suitable choice for developing the BEKG. The abstracts of journal papers in the built environment field were collected from the Microsoft Academic Graph (MAG) database (Wang et al., 2020), which contains 166,192,182 papers with various properties, including abstracts, authors, titles, and journals.

Around 85,000 abstracts were obtained from 54 highly regarded journals in the built environment field using the Microsoft Academic Graph’s Azure API. The selection of these journals was based on the ERA 2018 Journal list, which ranks journals according to their quality and relevance in their respective fields. By using this list, the abstracts collected through the Azure API can be trusted as they come from reputable sources. The selected journals all belong to the Field of Research (FoR) categories of 1202 (Building) or 1205 (Urban and Regional Planning). This ensures the reliability of the data and enables researchers to derive meaningful insights and conclusions from the analysis.

Figure 1. An example of a typical KG subgraph.

Figure 2. Research methodology flowchart.
Table 1 displays the Top-5 journals, their quantities, and the corresponding Field of Research (FoR). The abstract data spans approximately 40 years, starting from the earliest publication. Alongside the aforementioned data, the titles and authors were extracted from the MAG. The abstract data obtained from MAG were compressed, necessitating a decompression process to access the original abstracts. Data cleaning techniques were applied to eliminate escape characters and redundant punctuation marks or symbols, ensuring the integrity of the abstract data.

### Knowledge Pre-extraction

After obtaining the clean abstract data, pre-extraction was conducted to bridge the gap with the dataset for model training. This involved creating a candidate set of pre-extraction triples to enhance human annotation efficiency and serve as a corpus for constructing the KG schema. To minimize manual annotation and labeling, various information extraction tools were tested during the pre-extraction process. Among them, Ollie (Schmitz et al., 2012) demonstrated effective and efficient performance in automatically identifying and extracting entities and relations from English sentences. After multiple experiments, Ollie showed the best results. Therefore, the Ollie tool was chosen for entity and relation extraction.

In the first step, 10,000 cleaned abstracts were randomly sampled from the entire dataset mentioned in the previous section to ensure the generalization of the Knowledge Graph (KG) schema. Each sentence from the abstracts was processed using Ollie to extract numerous triples by identifying relations mediated by nouns, adjectives, and more based on learned pattern templates. Subsequently, a context-analysis module was employed to incorporate contextual information into these triples. Finally, a confidence score was assigned to each triple using a confidence function.

In Figure 3, the blue and red entities represent the head and tail entities respectively. The yellow box represents the predicted relation, and the black box indicates the confidence score of the extraction result. To ensure triple quality, the sentence output with the highest confidence was selected. Ultimately, 60,318 pre-extracted triples (head, predicate, and tail) were obtained from the 10,000 abstracts, which were then used for further processing to establish the Built Environment Knowledge Graph (BEKG) schema.

### Knowledge graph schema

During pre-extraction, numerous triples were extracted from input sentences. However, these extracted entities and relations were excessively redundant for direct utilization in model training and implementation. To address this issue, the KG schema played a crucial role. Each KG possesses its own schema, which defines the data and relation types within it. Generally, a KG schema, functioning as an ontology layer, encompasses four key definitions: (1) Universal recognition of knowledge within the schema; (2) Conceptual description of entities with precise specifications; and (3) Formalizability of knowledge within the schema.

The BEKG schema aimed to achieve the following goals. The defined relations should be relevant and comprehensive in the built environment domain, covering a wide range of knowledge. Additionally, the schema should reduce redundancy by grouping similar semantic relations extracted by Ollie into the same type. Lastly, ensuring the precision of the schema was crucial for optimal model performance and final extraction.

The initial approach involved the manual construction of the schema based on statistical analysis. The aim was to select the top 30 frequently occurring relation types from the Ollie pre-extraction results. However, these relations proved insufficient in terms of comprehensiveness and precision to meet the stated objectives. To address this, clustering was employed as an unsupervised learning technique. By measuring similarity and grouping similar samples together, clustering

---

**Table 1.** Top-5 Journals with the quantity of accessed abstract data.

| Journal name                      | No. of abstracts collected | Earliest year published in | Latest year published in |
|-----------------------------------|----------------------------|-----------------------------|---------------------------|
| Energy and Buildings              | 1202                       | 9759                        | 1977                      | 2021                      |
| Building and Environment          | 1202                       | 7612                        | 1976                      | 2021                      |
| Cement and Concrete Research      | 1202                       | 7553                        | 1966                      | 2021                      |
| Environment and Planning A        | 1205                       | 5336                        | 1969                      | 2020                      |
| Land Use Policy                   | 1202                       | 5042                        | 1984                      | 2021                      |

---

**Figure 3.** One Ollie pre-extraction result of a sentence.
effectively facilitated the aggregation of entities with shared attributes into meaningful categories. This approach enabled a deeper understanding of the dataset’s relationships, identification of data anomalies, and potential trends. In the context of relationship classification, clustering played a crucial role in revealing patterns within the dataset, enhancing the understanding of entity relationships, and uncovering valuable patterns and structures.

Therefore, the NLP tool (Reimers & Gurevych, 2019) and a clustering algorithm were utilized for the schema development process, as demonstrated in Figure 4. The text embeddings were calculated by the NLP tool, which transformed high-dimensional texts into low-dimensional vectors for clustering purposes. The entities from the Ollie triple results were then embedded, and the clustering algorithm grouped entities with similar semantics into clusters. Subsequently, the original Ollie triples were updated based on the clustering results. Finally, the updated Ollie triples were transformed into embeddings and clustered again to generate the knowledge graph schema.

The selected NLP tool for calculating text embeddings was the ‘BERT-STSb-large’ model, which achieved the best performance in Sentence-BERT (Reimers & Gurevych, 2019). This model adds a mean pooling operation to the BERT-large model output, resulting in a fixed-size sentence embedding. It was trained on the STS benchmark (Cer et al., 2017) with a max sequence length of 128, which is a popular dataset for evaluating supervised Semantic Textual Similarity (STS) systems. The model is user-friendly, allowing direct input of entities or triples to obtain a 768-dimensional embedding, significantly improving efficiency without compromising effectiveness. The choice of embedding dimension is crucial, as it determines the size of the text’s vector representation. A dimension of 768 strikes a balance between efficiency and effectiveness, capturing sufficient information while minimizing computational costs.

As for the clustering algorithm, the k-means (Lloyd, 1982) is adopted. The optimization criterion of the k-means algorithm is expressed as Equation (1):

$$J = \min \sum_{i=1}^{k} \sum_{x \in C_i} ||x - u_i||^2, \quad u_i = \frac{1}{|C_i|} \sum_{x \in C_i} x, \quad (1)$$

where $x$ denotes the embeddings of the entity or triple, $C = \{C_1, C_2, \ldots, C_k\}$ denotes the set of clusters. $u_i$ denotes the centroid of each cluster $C_i$. $J$ was optimized iteratively to reach a minimum until $u_i$ was fixed so that the similarity of samples in the same cluster becomes as high as possible. The flow chart of the clustering algorithm is shown in Figure 5.

The number of entity and triple clusters follows the approach of Wang et al. (2018), where the total number of clusters was divided by 5. For triples clustering, the entity closest to the centroid represented each cluster, and all entities in the corpus were updated accordingly. The final knowledge graph schema was obtained in a similar manner for obtaining relations.

Throughout the implementation, duplicates were removed from approximately 58,000 Ollie pre-extracted triples during implementation, resulting in 9800 unique entities. These entities were then clustered into 56 groups. The Ollie pre-extracted triples were updated based on the entity clustering, resulting in around

**Figure 4.** The process of building a KG schema.

**Figure 5.** The flow chart of the clustering algorithm.
3000 triples. The KG schema was constructed with 29 relation types obtained from the triples clustering. Table 2 displays some relation types in the schema along with their respective representative entities. The ‘Top-3 Similar Entities’ row lists the top three similar entities for each cluster representative. Each example in the ‘Triple’ line represents a triple in the triple cluster, updated according to the entity clustering.

**Human annotation**

The next step was annotating the data following the obtained KG schema to build a dataset. To obtain higher quality triples to create a dataset for model training, several annotators were organized to annotate the triples on the Brat platform (Stenetorp et al., 2012), a web-based annotation tool for employing each type of annotation on text easily and effectively. The Brat platform was used to structure the original unstructured text to obtain the annotation corpus required for the named entity recognition model and relation classification model training.

The human annotation workflow is shown in Figure 6. First, depending on the BEKG schema, Ollie triples from pre-extraction were classified to each relation. Then, several well-educated annotators were organized to filter the Ollie extraction results of high-quality belonging to each relation using the Brat platform. The Brat platform illustrated each Ollie triple by displaying the original sentence with two highlighted entities linked by a line with their relation above.

During the first filtering, annotators aimed at looking for the good Ollie results whose entities and results corresponded to the built environment without mistakes. However, the amount of these well-annotated Ollie results was small. For other Ollie extraction results, the annotator had to correct some entity annotation mistakes that appeared in them.

In general, annotators were asked to ensure the high-quality instance with precisely annotated entities and the correctness of relations labelled. As the whole process relied on subjective recognition, some criteria were provided for them during annotating.

Specifically, the clause and adverb in the annotated entities were removed to lower the redundancy of the annotation. In addition, the coordinate entities in one annotated entity were split, which would not confuse the NER model during training. A few examples of the high-quality Ollie extraction results obtained after the first filtering are shown in Figure 7.

After the first filtering, an annotator who was not involved in the first round of filtering would be asked to validate annotations and correct them if necessary to ensure mutual agreements among annotators. The annotation consumed considerable time and human resources. However, it was difficult to find annotators familiar with the BE domain. Therefore, considering the balance of efficiency and effectiveness, the number of well-annotated instances for each relation was set as 50.

Table 2. Examples of some types of relations and their instances, blue for the head entity and red for the tail entity.

| Relation type | Top-3 Similar Head Entities | Top-3 Similar Tail Entities |
|---------------|----------------------------|-----------------------------|
| Relation_1: Be Made | Entity_198: aluminum metal | Entity_134: market expansion process |
| | Entity_218: cement Entity_450: Portland limestone | Entity_138: clinker |
| | | Entity_264: Climate condition moisture source |
| | Triple: Cements < Entity_210 > made from < Relation_1 > clinker < Entity_308 >. | |
| Relation_2: Lead to | Entity_126: pore water pressure transducer | Entity_283: work experience performance. |
| | Entity_104: steam curing | Entity_9: lower porosity and fewer micropores. |
| | Entity_308: paper analysis cost | Entity_61: activity engagement. |
| | Triple: Steam curing < Entity_104 > led to < Relation_2 > lower porosity and fewer micropores < Entity_9 >. | |
| Relation_3: Related to | Entity_297: transformative adaptation | Entity_266: wall friction effect |
| | Entity_402: discomfort probability | Entity_311: manufacturing industry |
| | Entity_46: transitions theories | Entity_81: regional scholarship and practice |
| | Triple: transitions theories < Entity_46 > related to < Relation_2 > regional scholarship and practice < Entity_81 >. | |
| Relation_4: Improve | Entity_308: paper analysis cost | Entity_130: survey inquiry research |
| | Entity_230: study contribution | Entity_386: model predictive controller |
| | Entity_371: The addition of fly ash | Entity_146: the microstructure of MPC |
| | Triple: The addition of fly ash < Entity_371 > improves < Relation_4 > the microstructure of MPC < Entity_146 >. | |
| Relation_5: Include | Entity_254: accuracy measure exposure | Entity_166: heat strain prediction |
| | Entity_349: field data information | Entity_419: rate estimation method |
| | Entity_29: Modern concrete construction | Entity_173: the use of a variety of special concretes |
| | Triple: Modern concrete construction < Entity_29 > includes < Relation_5 > the use of a variety of special concretes < Entity_173 >. | |
Finally, 1450 high-quality instances in 29 relations were obtained for developing relation classification datasets. Regarding the dataset split method adopted by Han et al. (2018), 29 relations were shuffled first, then 18, 5, and 6 of them were allocated to relation classification training, validation, and testing set correspondingly.

As the performance of the NER model plays a crucial role in the final extraction results, 550 more annotated instances were added to the NER dataset. A total of 1,600 and 400 instances were randomly selected and allocated to the training and validation set respectively. There was no test set in the NER dataset as the model performance was evaluated by humans at the extraction phase, which could increase the amount of data in the training set. The statistics of these two datasets are shown in Table 3.

**Model training**

Training models using the annotated dataset were necessary to achieve better performance. According to the BEKG schema, two datasets containing more than 1400 instances were built. There was a pair of entities and their relation in each instance. Besides, a suitable named entity recognition model and relation classification model were in need to extract entities and relations in the data extraction stage.

**Named entity recognition**

Named entity recognition refers to the task of recognizing entities in an input sentence. The named entity recognition model could annotate each token in the sentence using one of the ‘BIO’ sets. ‘B’ means the beginning token of the entity. ‘I’ represents the token inside the entity, and ‘O’ means the last token in the entity or token that does not belong to any entity.

The named entity recognition was performed by BERT-CRF, a combination of BERT (Devlin et al., 2018), which is a large masked language pre-trained model for text representation, and CRF (Lafferty et al., 2001), an objective of structured output prediction. The BERT-CRF model combines the strengths of both BERT and CRF. BERT is used to encode the input text into a high-dimensional vector representation, which is then fed into a CRF layer for sequence labelling. The output of the CRF layer is a sequence of labels corresponding to the input tokens, which represent the named entities in the text. By using BERT-CRF for NER, state-of-the-art performance on the task of named entity recognition was achieved. The forward of BERT-CRF could be expressed as Equation (2):

$$\text{Pred} = C_{\text{decode}}(l(d(B(X))))$$

where $B$ denotes the BERT model, $d$ and $l$ denote the dropout and linear layer respectively. The linear layer outputs the emission score of each token in the input sequence for the ‘BIO’ label. $C_{\text{decode}}$ denotes the Viterbi algorithm for finding the best tag sequence given an emission score tensor.

The BERT-CRF was trained to maximize the log-probability of the correct tag sequence. Therefore, the loss function of the model was designed as Equation (3):

$$\ell = -C(\log(S(\text{emission})))$$

where $\text{emission}$ denotes the output of the linear layer, $S$ denotes the softmax layer, and $C$ denotes the CRF layer.

A demonstration of how the named entity recognition model worked is shown in Figure 8. BERT-CRF took the tokenized sentence as input and output each token’s ‘BIO’ annotation. A model was trained on the dataset since the pre-trained model’s performance was not satisfied. The dataset contained 2000 instances,
each consisting of a sentence and two annotated pairwise entities.

**Few-shot learning-based relation classification**

Compared to named entity recognition, relation extraction requires a model with better generalization. Few-shot learning addressed this challenge by enabling effective generalization with limited annotated samples. BERT-Pair (Gao et al., 2019) was a model designed for few-shot learning in relation extraction. It fine-tuned a pre-trained BERT model on limited annotated data, enhancing few-shot learning and enabling the model to learn new tasks effectively. This approach was valuable in domains with limited annotated data. Additionally, BERT-Pair reduced the time and effort required to build accurate relation extraction models, achieving comparable results with fewer annotations. In this paper, BERT-Pair was employed for relation extraction using few-shot learning. The process involved a support set and query sets, typically constructed using the ‘N-way K-shot Q-query’ setting. These parameters determined the selection of N classes and obtaining K instances for the support set (Equation 4).

\[
S = \left\{ \left( x_{1}^{1}, r_{1} \right), \left( x_{1}^{2}, r_{1} \right), \ldots, \left( x_{K}, r_{1} \right), \ldots, \left( x_{1}^{N}, r_{N} \right), \left( x_{2}^{N}, r_{N} \right), \ldots, \left( x_{K}^{N}, r_{N} \right) \right\}
\]

where \(r_{N}\) denotes the Nth class and \(x_{K}^{N}\) denotes the K-th instance of the N-th class in the support set. Then, Q instances were picked from each selected class’s training instances, excluding instances that have been selected for the support set to construct N query sets. It can be expressed as Equation (5).

\[
Q = \left\{ \left( y_{1}^{1}, r_{1} \right), \left( y_{1}^{2}, r_{1} \right), \ldots, \left( y_{Q}, r_{1} \right), \ldots, \left( y_{1}^{N}, r_{N} \right), \left( y_{2}^{N}, r_{N} \right), \ldots, \left( y_{Q}^{N}, r_{N} \right) \right\}
\]

where \(y\) denotes the instance different from instances in the support set. \(y_{Q}^{N}\) denotes the Q-th instance of the N-th class in the support set. To illustrate these two sets, a demonstration of constructing a ‘5 way 1 shot 1 query’ setting is shown in Figure 9.

To create the support set and query set, each query and support instance was tokenized and their respective
embeddings were calculated for each token. These tokenized query embeddings were then concatenated with the tokenized embeddings of each support instance, as illustrated in Figure 10. The BERT-Pair model, which is built upon the BERT sequence classification model, was employed to compute the relation similarity between the two instances in each concatenated embedding. The maximum length of the concatenated embedding was set to 128. The entire similarity calculation process for each query can be expressed as Equation (6).

\[ O_{pred} = \max_{r \in \{1, ..., K\}} \frac{1}{K} \sum_{j=1}^{K} (BP(y_r, x'_j)) \]  

where \( O_{pred} \) denotes the score of the predicted label and \( BP \) denotes the BERT-Pair model. For better understanding, Figure 10 demonstrates the calculation process of a 5-way 1-shot setting. Each row in Figure 10 table indicates the similarity score of a query instance with each class in the support set. The highest score for the first row was 8.6. Therefore, the first query is classified into the ‘lead_to’ class.

At the stage of model training, both the BERT-CRF and BERT-Pair models were trained on Titan Xp. During the BERT-CRF training, the batch size was set to 16. The learning rate started from 5e-8 and the Adam optimizer (Kingma & Ba, 2014) was adopted for model optimization.

The BERT-Pair training process utilized consistent N-way K-shot settings throughout both training and evaluation. Considering the limited number of relations in the dataset, the recommended 5-way 1-shot configuration from Han et al. (2018) was adopted. This configuration demonstrated optimal performance during the evaluation of the validation set, which contained only five relations. Consequently, the 5-way 1-shot learning strategy was employed for training the BERT-Pair model. An initial learning rate of 2e-5 was selected, and no warm-up strategy was applied during model training due to the utilization of pre-trained BERT models for both BERT-CRF and BERT-Pair. To address GPU memory limitations, a batch size of 2 was set.

**BEKG integration and visualization**

The methods above were proved suitable to accomplish the goal of developing a BEKG. With the trained model, it can be used to extract the entities and relations in the abstract data. The process of constructing the BEKG can be separated into four parts: entity extraction, relation extraction, post-processing, and visualization. Appendix A provides detailed information on the process and system visualization used to construct BEKG with the approach employed in this research.

**Result evaluation and analysis**

The quality of the extracted knowledge was sensitive to the accuracy of each model for extraction. Therefore, having a proper way to evaluate the performance at each stage is vital in the experiments. In the following sections, this paper will analyze the experiment results and elucidate how it improved the model performance.

**Accuracy of each stage**

During NER model training, a validation set consisting of 400 annotated instances was used to evaluate model performance. The results are presented in Table 2. In addition to BERT-CRF, other common NER methods were trained and evaluated to compare their performance with BERT-CRF on the dataset. Four metrics were employed to assess the validation set’s performance from various perspectives. BERT-Span’s accuracy metric was not applicable due to its output format not being suitable for the NER problem. The results indicate that the inclusion of a CRF layer as the final layer in
BERT yielded superior performance compared to other methods in almost all four metrics on the validation set. Among these metrics, Accuracy measures the ratio of correctly predicted tokens to all tokens, evaluating performance at the token level. Precision measures the ratio of correctly predicted entities to all predicted entities, providing an entity-level metric. Recall measures the ratio of correctly predicted entities to all labelled entities. F1-score calculates the harmonic mean of precision and recall, providing an overall evaluation metric.

The results in Table 4 did not reveal the model’s robustness as the constructed dataset is of a small size. To assess the model performance objectively, a k-fold cross-validation was performed to split the original dataset with 2000 instances into k folds of the same size to conduct evaluations.

For keeping the same size of the dataset as the original one, the 5-fold setting was chosen which split the dataset into training and validation in the ratio 4:1. Other settings during training and validation were also kept in accordance with previous ones.

The cross-validation results on the validation set are shown in Table 5. It is worth noting that the model performance on fold-2 was worse than on other folds. The standard deviation (STD) of four metrics on the 5-fold cross-validation also showed that the model robustness was affected by the limited size of the dataset slightly. Despite discrepancies between the averaged results and individual folds in Table 4, the performance was approximately within one standard deviation.

As for the relation classification, results were evaluated separately at the training and large-scale extraction stage in two ways. The validation and test set were used to assess the model performance throughout the model training to prevent overfitting. The evaluation results on each set are shown in Table 6.

The 5-fold cross-validation was also conducted on the relation classification task. The division among the training, validation, and test set in each fold reuses the previous fold’s allocation with a four-relations right rotation, ensuring each relation was deployed for validation or test set. The allocation of the original dataset was the initial setting for Fold-1.

The results of each fold’s validation and test set were obtained after 1000 iterations of few-shot relation classification. As shown in Table 7, the average accuracy on validation and test set were both close to the accuracy in Table 6.

The standard deviation of accuracy on the 5-fold validation set and test set cross-validation were higher than in the NER task. Some relations containing more types of relation words would be a challenge for the BERT-Pair model to classify which pulled down the accuracy of individual folds. At the same time, as adopting a different way to allocate dataset for each fold following the few-shot learning, a domain gap between the training and validation or test set also affected the accuracy.

In the final stage of data extraction, many instances were obtained. To reduce the manual validation workload, a sample of 100 instances per relation out of a total of 29 relations was selected for validation. Each instance was independently validated by two examiners who determined whether the relation between the two entities in the instance was 'True' or 'False'. The validation result was considered valid when both examiners reached a consensus. In cases where consensus was not achieved, a third annotator made the final judgment to ensure accuracy and objectivity. The results of this phase are summarized in Table 8, indicating an overall acceptable quality of the knowledge in BEKG, with an accuracy exceeding 80%.

To analyze the model’s generalization across different relations, human evaluations were conducted using the accuracy metric. Based on the statistics presented in Table 9, BERT-Pair demonstrated good performance on most relations but encountered challenges with

### Table 4. Comparison of the accuracy of different NER methods on the validation set.

| Methods       | Accuracy | Precision | Recall | F1-Score |
|---------------|----------|-----------|--------|----------|
| BERT-CRF      | 92.31%   | 71.48%    | 77.24% | 74.25%   |
| BERT-Softmax  | 87.49%   | 69.15%    | 64.36% | 66.67%   |
| BERT-Span     |          | 72.53%    | 62.85% | 67.34%   |
| ALBERT-Softmax| 88.23%   | 71.22%    | 67.63% | 69.38%   |
| BERT-BiLSTM-CRF| 91.82%  | 68.37%    | 73.53% | 70.86%   |
| SpanBERT-CRF  | 90.79%   | 66.31%    | 71.48% | 68.80%   |

### Table 5. NER tagging results of BERT-CRF on the validation set using 5-fold cross-validation.

|          | Fold-1 | Fold-2 | Fold-3 | Fold-4 | Fold-5 | Average | STD  |
|----------|--------|--------|--------|--------|--------|---------|------|
| Acc      | 91.62% | 91.26% | 91.82% | 92.14% | 92.24% | 91.82%  | 0.40%|
| Precision| 69.03% | 64.55% | 68.97% | 69.33% | 70.78% | 68.53%  | 2.34%|
| Recall   | 75.70% | 71.12% | 76.37% | 76.40% | 77.55% | 75.43%  | 2.50%|
| F1-Score | 72.21% | 67.68% | 72.48% | 72.69% | 74.01% | 71.81%  | 2.41%|

### Table 6. Accuracy of relation classification at the training phase.

|          | Validation Set | Test Set |
|----------|----------------|----------|
| Avg Acc  | 85.51%         | 86.3%    |

The cross-validation results on the validation set are shown in Table 5. It is worth noting that the model performance on fold-2 was worse than on other folds. The standard deviation (STD) of four metrics on the 5-fold cross-validation also showed that the model robustness was affected by the limited size of the dataset slightly. Despite discrepancies between the averaged results and individual folds in Table 4, the performance was approximately within one standard deviation.

As for the relation classification, results were evaluated separately at the training and large-scale extraction stage in two ways. The validation and test set were used to assess the model performance throughout the model training to prevent overfitting. The evaluation results on each set are shown in Table 6.

The 5-fold cross-validation was also conducted on the relation classification task. The division among the training, validation, and test set in each fold reuses the previous fold’s allocation with a four-relations right rotation, ensuring each relation was deployed for validation or test set. The allocation of the original dataset was the initial setting for Fold-1.

The results of each fold’s validation and test set were obtained after 1000 iterations of few-shot relation classification. As shown in Table 7, the average accuracy on validation and test set were both close to the accuracy in Table 6.

The standard deviation of accuracy on the 5-fold validation set and test set cross-validation were higher than in the NER task. Some relations containing more types of relation words would be a challenge for the BERT-Pair model to classify which pulled down the accuracy of individual folds. At the same time, as adopting a different way to allocate dataset for each fold following the few-shot learning, a domain gap between the training and validation or test set also affected the accuracy.

In the final stage of data extraction, many instances were obtained. To reduce the manual validation workload, a sample of 100 instances per relation out of a total of 29 relations was selected for validation. Each instance was independently validated by two examiners who determined whether the relation between the two entities in the instance was 'True' or 'False'. The validation result was considered valid when both examiners reached a consensus. In cases where consensus was not achieved, a third annotator made the final judgment to ensure accuracy and objectivity. The results of this phase are summarized in Table 8, indicating an overall acceptable quality of the knowledge in BEKG, with an accuracy exceeding 80%.

To analyze the model’s generalization across different relations, human evaluations were conducted using the accuracy metric. Based on the statistics presented in Table 9, BERT-Pair demonstrated good performance on most relations but encountered challenges with
part_of' and 'identify'. The 'part_of' relation encompasses various relation words with similar meanings, such as 'composed of', 'formed of', 'served as', and others. On the other hand, 'identify' is a polysemous relation, leading to ambiguities in different contexts and posing difficulty for the relation classification model. Despite these challenges, BERT-Pair benefited from its large-scale pre-trained corpus settings and was able to correctly interpret the meaning, covering most relations in the dataset. However, it showed a slight decrease in performance when dealing with relations that exhibit significant diversity.

### Overall Statistics of the BEKG

From a total number of 84,588 abstracts, 400,200 sentences were obtained, resulting in 771,648 entities identified using the trained BERT-CRF model. These entities were combined to form 458,726 pairwise entities, each paired with its corresponding sentence. Each of these instances was then processed by the BERT-Pair model, utilizing the annotated dataset as a supporting set to extract the final relation. The statistics are found in Table 10. Each triple was attached with a score indicating the possibility of the output. The number of triples harvested over different thresholds was presented as a line graph in Figure 11.

### Table 7. Relation classification results on both validation set and test set using 5-fold cross-validation.

|       | Fold-1   | Fold-2   | Fold-3   | Fold-4   | Fold-5   | Average | STD  |
|-------|----------|----------|----------|----------|----------|---------|------|
| Val   | 85.03%   | 76.24%   | 95.32%   | 92.16%   | 87.68%   | 87.29%  | 7.34%|
| Test  | 89.98%   | 90.70%   | 89.56%   | 81.42%   | 70.46%   | 84.42%  | 8.67%|

### Table 8. Accuracy of relation extraction at the data integration phase.

|                           | True | False | Total | Accuracy   |
|---------------------------|------|-------|-------|------------|
| Mutually agreed           | 1965 | 236   | 2201  | 89.28%     |
| Disagreed and annotated with a third annotator | 456  | 243   | 699   | 65.24%     |
| Total                     | 2421 | 479   | 2900  | 83.48%     |

### Table 9. Accuracy of relation extraction at the data integration phase.

| Relation Type | Accuracy | Relation Type | Accuracy |
|---------------|----------|---------------|----------|
| represent     | 84.00%   | calculate     | 71.00%   |
| discuss       | 73.00%   | find          | 80.00%   |
| draw_on       | 91.00%   | improve       | 91.00%   |
| part_of       | 66.00%   | identify      | 68.00%   |
| require       | 83.00%   | be            | 90.00%   |
| include       | 92.00%   | explore       | 88.00%   |
| report        | 89.00%   | lead_to       | 86.00%   |
| used_as       | 73.00%   | reveal        | 87.00%   |
| affect        | 84.00%   | related_to    | 87.00%   |
| consider      | 70.00%   | produce       | 92.00%   |
| examine       | 84.00%   | offer         | 88.00%   |
| analyse       | 74.00%   | develop       | 86.00%   |
| collect       | 86.00%   | concern       | 87.00%   |
| be_made       | 86.00%   | focus_on      | 86.00%   |
| describe      | 92.00%   |               |          |

### Table 10. Results statistics at data integration phase.

| Final Data Extraction | Abstract | Sentences | Entities | High-quality Triples |
|-----------------------|----------|-----------|----------|----------------------|
| Total                 | 84,588   | 400,200   | 425,650  | 223,443              |

---

**Figure 11.** Distribution of triples by relation classification score.
A graph with 223,443 high-quality triples filtered with an empirical threshold was obtained. The threshold was determined to balance the quantity and quality of the triples extracted. With this threshold, triples with obviously incorrect relations were filtered, which could ensure the preciseness of the BEKG to retain high-quality triples. A breakdown in the number of triples and entities belonging to each relation with the selected threshold was reported in Table 11.

### Conclusion

This paper presents a framework for constructing a KG in the built environment domain, called the BEKG. The construction of the BEKG is accomplished through the extraction of information from scientific paper abstracts. By utilizing this approach, this work bridges the gap between the need for a high-quality and comprehensive KG in the field and the absence of an existing scalable KG.

BEKG has made significant improvements to existing knowledge graphs in terms of accuracy, coverage, and usability. In terms of accuracy, BEKG addresses the complexity and variability of the built environment domain by incorporating diverse and comprehensive data. It provides accurate and reliable information while maintaining data currency. In terms of coverage, BEKG connects information from various domains in the built environment, including building materials, construction techniques, energy-efficient technologies, and sustainable practices, offering a comprehensive perspective. BEKG’s expanded coverage extends the boundaries of knowledge and provides a comprehensive reference tool. In terms of usability, BEKG prioritizes user experience and feedback, continuously improving interface design and functionality. It offers an intuitive and user-friendly interface to meet user needs. By revealing the connections between building materials and energy efficiency, as well as the relationship between the built environment and sustainable development goals, BEKG provides valuable insights to benefit stakeholders. These insights can influence decision-makers in areas such as architectural design, material selection, and policymaking, promoting the development of the built environment towards greater sustainability and user-friendliness.

The significance of the methods used to construct BEKG can be reflected in terms of scalability, efficiency, and adaptability to different domains. In terms of scalability, through large-scale data processing techniques, BEKG can gather and integrate built environment-related data from scientific abstracts, handling large volumes of data, thus constructing a comprehensive and rich knowledge graph. In terms of efficiency, by employing automated data extraction and data cleansing techniques, BEKG effectively processes large-scale data and transforms it into structured knowledge representations with minimum human annotation efforts. This efficiency helps reduce manual operations and time costs while improving the speed and accuracy of KG construction. In terms of adaptability to different domains, the methodology employed in this study can also be applied or extended to other domains or challenges. For instance, this methodology can be applied to domains such as urban planning, traffic management, and environmental conservation, offering decision support for urban development and sustainable goals.

Despite the significant progress achieved in our research, there are still some limitations and areas for improvement. First, ’BERT-STSb-large,’ as an NLP tool, was used for text similarity calculation in our experiments. However, over time, this tool can become outdated, and there may be more advanced tools and techniques available for more accurate text processing and semantic analysis. Second, although the generalizability of BEKG can be limited. Due to the wide-ranging and ever-evolving knowledge and concepts in the field of built environment, BEKG may not be able to cover all new developments and changes in the domain. Third, the relationship classification models in BEKG may have some limitations in predicting complex relationships. Particularly, there is room for improvement in the accuracy of certain specific relationships. Finally, the application and promotion of built environment ontology and related tools still face challenges in real-world settings. Implementing BEKG may require addressing issues related to data collection and integration, as well as ensuring timely updates and maintenance of the knowledge graph. To improve BEKG, in the future, it is recommended to explore new technologies,
including incorporating more diverse and comprehensive data into BEKG, designing intelligent models to enhance accuracy and generalizability, and focusing on interface design and user-friendliness.

**Acknowledgment**

This paper was supported by the National Natural Science Foundation of China (U20B2053), the Key-Area Research and Development Program of Guangdong Province (2018B010115001), and the Key Laboratory Open Foundation (WDZC20215250118), the GRF (16211520), and the RIF (R6020-19 and R6021-20) from RGC of Hong Kong. Authors also thank the UGC Research Matching Grants (RMGS20EG01-D, RMGS20CR11, RMGS20CR12, RMGS20EG19, RMGS20EG21).

**Disclosure statement**

No potential conflict of interest was reported by the author(s).

**Funding**

This work was supported by Key-Area Research and Development Program of Guangdong Province: [Grant Number 2018B010115001]; Key Laboratory Open Foundation: [Grant Number WDZC20215250118]; RIF: [Grant Number R6020-19]; GRF: [Grant Number 16211520]; National Natural Science Foundation of China: [Grant Number U20B2053].

**References**

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. (2007). DBpedia: A nucleus for a web of open data. In 6th international semantic web conference/2nd Asian semantic web conference (ISWC 2007/ASWC 2007), Busan, South Korea, pp. 722–735. https://doi.org/10.1007/978-3-540-76298-0_52

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008). Freebase. Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 1247–1250. https://doi.org/10.1145/1376616.1376746

Brin, S. (1999). Extracting patterns and relations from the world wide Web. Springer. Berlin Heidelberg, pp. 172–183. https://doi.org/10.1007/978-3-540-47629-8

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., & Specia, L. (2017). SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. Association for Computational Linguistics, Vancouver, Canada, pp. 1–14. https://doi.org/10.18653/v1/S17-2001

Charpenay, V., Käbisch, S., Anicic, D., & Kosch, H. (2015). An ontology design pattern for iot device tagging systems. In 2015 5th international conference on the internet of things (IOT), IEEE, pp. 138–145. https://doi.org/10.1109/IOT.2015.7356588

Ciampaglia, G. L., Shiralkar, P., Rocha, L. M., Bollen, J., Menczer, F., & Flammini, A. (2015). Correction: Computational fact checking from knowledge networks. *PloS One*, 10(10), e0141938. https://doi.org/10.1371/journal.pone.0141938

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805, 2018. https://arxiv.org/abs/1810.04805.

Elmagy, S. A., Qiu, M. K., & Gai, K. K. (2016). IEEE understanding taxonomy of cyber risks for cybersecurity insurance of financial industry in cloud computing. In 3rd IEEE International Conference on Cyber Security and Cloud Computing (IEEE CSCloud) / 2nd IEEE International Conference of Scalable and Smart Cloud (IEEE SSC), Beijing, Peoples R China, pp. 295–300. https://doi.org/10.1109/CSCloud.2016.46

Ernst, P., Meng, C., Siu, A., & Weikum, G. (2014). Knowlife: A knowledge graph for health and life sciences. In IEEE 30th International Conference on Data Engineering (ICDE), Chicago, IL, pp. 1254–1257. https://doi.org/10.1109/ICDE.2014.6816754.

Etzioni, O., Banko, M., Soderland, S., & Weld, D. S. (2008). Open information extraction from the Web. *Communications of the ACM*, 51(12), 68–74. https://doi.org/10.1145/1409360.1409378

Fader, A., Soderland, S., & Etzioni, O. (2011). Identifying relations for open information extraction. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pp. 1535–1545. https://www.aclweb.org/anthology/D11-1142.pdf.

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

Filtz, E., Krrane, S., & Polleres, A. (2021). The linked legal data landscape: linking legal data across different countries. *Artificial Intelligence and Law*, 29(4), 485–539. https://doi.org/10.1007/s10506-021-09282-8

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

Ghadai, S., Balu, A., Sarkar, S., & Krishnamurthy, A. (2018). Learning localized features in 3D CAD models for manufacturability analysis of drilled holes. *Computer Aided Geometric Design*, 62, 263–275. https://doi.org/10.1016/j.cagd.2018.03.024

Goodwin, T., Harabagiu, S. M., & Soc, I. C. (2013). Automatic generation of a qualified medical knowledge graph and its usage for retrieving patient cohorts from electronic medical records. In 7th IEEE International Conference on Semantic Computing (ICSC), Irvine, CA, pp. 363–370. https://doi.org/10.1109/ICSC.2013.68

Han, X., Zhu, H., Yu, P., Wang, Z., Yao, Y., Liu, Z., & Sun, M. (2018). FewRel: A large-scale supervised Few-shot relation classification dataset with state-of-the-Art evaluation. Association for Computational Linguistics, Brussels, Belgium, pp. 4803–4809. https://doi.org/10.18653/v1/D18-1514

Hasegawa, T., Sekine, S., & Grishman, R. (2004). Discovering relations among named entities from large corpora. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04),
pp. 415–422. https://www.aclweb.org/anthology/P04-1053.pdf.

Huguet Cabot, P. L., & Navigli, R. (2021). Rebel: Relation extraction by End-to-end language generation. Association For Computational Linguistics, Punta Cana, Dominican Republic, pp. 2370–2381. https://doi.org/10.18653/v1/2021.findings-emnlp.204

Ji, L., Wang, Y. J., Shi, B. T., Zhang, D. W., Wang, Z. Y., & Yan, J. (2019). Microsoft concept graph: Mining semantic concepts for short text understanding. Data Intelligence, 1 (3), 238–270. https://doi.org/10.1162/int_a_00013

Jia, Y., Qi, Y. L., Shang, H. J., Jiang, R., & Li, A. P. (2018). A practical approach to constructing a knowledge graph for cybersecurity. Engineering, 4(1), 53–60. https://doi.org/10.1016/j.engan.2018.01.004

Jiang, X., Wang, Q., Li, P., & Wang, B. (2016). Relation extraction with multi-instance multi-label convolutional neural networks. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 1471–1480. https://www.aclweb.org/anthology/C16-1139.pdf.

Katiyar, A., & Cardie, C. (2017). Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In 55th Annual Meeting of the Association for Computational Linguistics (ACL). Vancouver, Canada, pp. 917–928. https://www.aclweb.org/anthology/P17-1085.pdf.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980, 2014. https://arxiv.org/abs/1412.6980.

Kolluru, K., Adlakha, V., Aggarwal, S., Chakrabarti, S., & Mausam. (2020). Openie6: Iterative grid labeling and coordination analysis for open information extraction. Association for Computational Linguistics, pp. 3748–3761. https://doi.org/10.18653/v1/2020.emnlp-main.306

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proc. 18th International Conf. on Machine Learning, 2001. https://repository.upenn.edu/cis_papers/159/.

Liu, H., & Singh, P. (2004). Conceptnet — A practical commonsense reasoning tool-Kit. Bt Technology Journal, 22 (4), 211–226. https://doi.org/10.1023/B:BTIT.0000047600.45421.6d

Liu, J., Lu, Z. C., & Du, W. (2019). Proceedings of the annual Hawaii international conference on system sciences. In 52nd Hawaii International Conference on System Sciences (HICSS), Hi, pp. 1247–1255. https://doi.org/10.24251/HICSS.2019.153

Lloyd, S. (1982). Least squares quantization in PCM. IEEE Transactions on Information Theory, 28(2), 129–137. https://doi.org/10.1109/TIT.1982.1056489

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. https://www.aclweb.org/anthology/P09-1113.pdf.

Miwa, M., & Bansal, M. (2016). End-to-End relation extraction using LSTMs on sequences and tree structures. In 54th Annual Meeting of the Association-for-Computational-Linguistics (ACL), Assoc Computational Linguistics-ACL, Berlin, Germany, pp. 1105–1116. <Go to ISI>:WOS:000493808001015.

Newton, D. (2018). Multi-Objective qualitative optimization (MOQO) in architectural design, 36th International Conference on Education and Research in Computer Aided Architectural Design in Europe (eCAADe), Lodz univ technol, Fac civil engn architecture & environm engn, Lodz, Poland, pp. 187–196. http://papers.cunimad.org/cgi-bin/works?Show_id=ecaaade2018_323.

Nieto-Morote, A., & Ruiz-Vila, F. (2011). A fuzzy approach to construction project risk assessment. International Journal of Project Management, 29(2), 220–231. https://doi.org/10.1016/j.ijproman.2010.02.002

Olavumi, T. O., & Chan, D. W. M. (2019). Building information modelling and project information management framework for construction projects. Journal of Civil Engineering and Management, 25(1), 53–75. https://doi.org/10.3846/jcem.2019.7841

Ozorhon, B., Karatas, C. G., & Demirkesen, S. (2014). A Web-based database system for managing construction project knowledge. In 27th World Congress of the International-Project-Management-Association (IPMA), Dubrovnik, Croatia, pp. 377–386. https://doi.org/10.1016/j.jsbspro.2014.03.043

Pan, Z., Su, C., Deng, Y., & Cheng, J. (2021). Video2Entities: A computer vision-based entity extraction framework for updating the architecture, engineering and construction industry knowledge graphs. Automation in Construction, 125, 103617. https://doi.org/10.1016/j.autcon.2021.103617

Pauwels, P. (2014). Supporting decision-making in the building life-cycle using linked building data. Buildings, 4(3), 549–579. https://doi.org/10.3390/buildings4030549

Pauwels, P., & Costin, A. (2022). M.H. Rasmussen, Knowledge Graphs and Linked Data for the Built Environment, Industry 4.0 for the Built Environment, Springer 2022, pp. 157–183. https://doi.org/10.1007/978-3-030-82430-3

Pauwels, P., Krijnen, T., Terkaj, W., & Beetz, J. (2017). Enhancing the IFCOWL ontology with an alternative representation for geometric data. Automation in Construction, 80, 77–94. https://doi.org/10.1016/j.autcon.2017.03.001

Piplai, A., Mittal, S., Joshi, A., Finin, T., Holt, J., & Zak, R. (2020). Creating cybersecurity knowledge graphs from malware after action reports. IEEE Access, 8, 211691–211703. https://doi.org/10.1109/ACCESS.2020.3039234

Plank, B., & Moschitti, A. (2013). Embedding semantic similarity in tree kernels for domain adaptation of relation extraction. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1498–1507. https://www.aclweb.org/anthology/P13-1147.pdf.

Poveda-Villallon, M., & Garcia-Castro, R. (2020). ETSI, SAREF Extension for Building. https://saref.etsi.org/saref4bldg/v1.1.1.2/.

Qin, H., & Jin, J. (2020). Intelligent maintenance of shield tunnelling machine based on knowledge graph. In 2020 IEEE 18th International Conference on Industrial Informatics (INDIN), IEEE, pp. 793–797. https://doi.org/10.1109/INDIN45582.2020.9442126

Rasmussen, M. H., Lefrançois, M., Schneider, G. F., & Pauwels, P. (2021). BOT: the building topology ontology
of the W3C linked building data group. *Semantic Web*, 12 (1), 143–161. https://doi.org/10.3233/SW-200385

Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks, arXiv preprint arXiv:1908.10084, 2019. https://arxiv.org/abs/1908.10084

Rosocher, M., van Erp, M., Vossen, P., Pokkens, A., Aldabe, I., Rigau, G., Soroa, A., Ploeger, T., & Bogaard, T. (2016). Building event-centric knowledge graphs from news. *Journal of Web Semantics*, 37–38, 132–151. https://doi.org/10.1016/j.websem.2015.12.004

Roth, D., & Yih, W.-T. (2007). Global inference for entity and relation identification via a linear programming formulation, Introduction to statistical relational learning 553–580.

Roth, D., & Yih, W.-T. (2012). A linear programming formulation for global inference in natural language tasks. In Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004), HLT-NAACL, pp. 1–1. https://doi.org/10.1109/NMDC.2012.6527569

Rotmensch, M., Halpern, Y., Tlimat, A., Horng, S., & Sontag, D. (2017). Learning a health knowledge graph from electronic medical records. *Scientific Reports*, 7(1), 5994. https://doi.org/10.1038/s41598-017-05778-z

Rozenfeld, B., & Feldman, R. (2006). High-performance unsupervised relation extraction from large corpora. In 6th IEEE International Conference on Data Mining, Hong Kong, Peoples R China, pp. 1032–1037. https://doi.org/10.1109/ICDM.2006.82.

Rudnik, C., Ehrhart, T., Ferret, O., Teyssou, D., Troncy, R., & Tannier, X. (2019). Acm, searching news articles using an event knowledge graph leveraged by Wikidata. In World wide Web conference (WWW), San Francisco, CA, pp. 1232–1239. https://doi.org/10.1145/3308560.3316761

Schmitz, M., Soderland, S., Bart, R., & Etzioni, O. (2012). Open language learning for information extraction. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 523–534. https://www.aclweb.org/anthology/D12-1048.pdf

Shi, L., Li, S., Yang, X., Qi, J., Pan, G., & Zhou, B. (2017). Semantic health knowledge graph: Semantic integration of heterogeneous medical knowledge and services. *BioMed Research International*, 2017, 1–12. https://doi.org/10.1155/2017/2858423

Socher, R., Huval, B., Manning, C. D., & Ng, A. Y. (2012). Semantic compositionality through recursive matrix-vector spaces. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 1201–1211. https://www.aclweb.org/anthology/D12-1110.pdf

Song, D., Schilder, F., Hertz, S., Saltini, G., Smiley, C., Nivarthi, P., Hazai, O., Landau, D., Zaharkin, M., Zielund, T., Molina-Salgado, H., Brew, C., & Bennett, D. (2019). Building and querying an enterprise knowledge graph. *IEEE Transactions on Services Computing*, 12(3), 356–369. https://doi.org/10.1109/TSC.2017.2711600

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., & Tsujii, J. I. (2012). BRAT: a web-based tool for NLP-assisted text annotation. In Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics, pp. 102–107. https://www.aclweb.org/anthology/E12-2021.pdf.

Suchanek, F. M., Kasneci, G., & Weikum, G. (2007). Yago. In Proceedings of the 16th International Conference on World Wide Web, pp. 697–706. https://doi.org/10.1145/1242572.1242667

Sun, J., Ren, X., & Anumba, C. J. (2019). Analysis of knowledge-transfer mechanisms in construction project cooperation networks. *Journal of Management in Engineering*, 35(2), 0418061. https://doi.org/10.1061/(ASCE)ME.1943-5479.0000663

Tah, J. H. M. (2001). Knowledge-Based approach to construction project risk management. *Journal of Computing in Civil Engineering*, 15(3), 170–177. https://doi.org/10.1061/(ASCE)0887-3801(2001)15:3(170)

Udeaja, C. E., Kamara, J. M., Carrillo, P. M., Anumba, C. J., Bouchlaghem, N., & Tan, H. C. (2008). A web-based prototype for live capture and reuse of construction project knowledge. *Automation in Construction*, 17(7), 839–851. https://doi.org/10.1016/j.autcon.2008.02.009

Wang, K. S., Shen, Z. H., Huang, C. Y., Wu, C. H., Dong, Y. X., & Kanakia, A. (2020). Microsoft Academic Graph: When experts are not enough. *Quantitative Science Studies*, 1(1), 396–413. https://doi.org/10.1162/qss_a_00021

Wang, S., Zhang, X. Y., Ye, P., Du, M., Lu, Y. X., & Xue, H. N. (2019). Geographic knowledge graph (GeoKG): A formalized geographic knowledge representation. *ISPRS International Journal of Geo-Information*, 8(4), 184. https://doi.org/10.3390/ijgi8040184

Wang, X., Zhang, Y., Li, Q., Chen, Y., & Han, J. (2018). Open information extraction with meta-pattern discovery in biomedical literature. In Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, Association for Computing Machinery, Washington, DC, USA, pp. 291–300. https://doi.org/10.1145/3233547.3233594

Wei, Y. X., Zhang, X. X., Shi, Y., Xia, L., Pan, S., Wu, J. S., Han, M. J., & Zhao, X. Y. (2018). A review of data-driven approaches for prediction and classification of building energy consumption. *Renewable and Sustainable Energy Reviews*, 82, 1027–1047. https://doi.org/10.1016/j.rser.2017.09.108

Yang, X., Liao, L., Yang, Q., Sun, B., & Xi, J. (2021). Limited-energy output formation for multiagent systems with intermittent interactions. *Journal of the Franklin Institute*, 358 (13), 6462–6489. https://doi.org/10.1016/j.jfranklin.2021.06.009

Yu, X. B., Stahr, M., Chen, H., & Yan, R. M. (2021). IEEE, design and implementation of curriculum system based on knowledge graph. In IEEE International Conference on Consumer Electronics And Computer Engineering (ICCECE), Guangzhou, Peoples R China, pp. 767–770. https://doi.org/10.1109/ICCECE51280.2021.9342370

Yu, Z., Haghighat, F., Fung, B. C. M., & Yoshino, H. (2010). A decision tree method for building energy demand modeling. *Energy and Buildings*, 42(10), 1637–1646. https://doi.org/10.1016/j.enbuild.2010.04.006

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. https://www.aclweb.org/anthology/D15-1203.pdf. https://doi.org/10.18653/v1/D15-1203
Zeng, D., Liu, K., Lai, S., Zhou, G., & Zhao, J. (2014). Relation classification via convolutional deep neural network. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 2335–2344. https://www.aclweb.org/anthology/C14-1220.pdf.

Zhang, H. M., Liu, X., Pan, H. J., Song, Y. Q., Leung, C. W. K., & Assoc Comp, M. (2020). ASER: A large-scale eventuality knowledge graph. In 29th Web Conference (WWW), Taipei, Taiwan, pp. 201–211. https://doi.org/10.1145/3366423.3380107

Zhou, X. G., Gong, R. B., Shi, F. G., & Wang, Z. F. (2020). Petrokg: Construction and application of knowledge graph in upstream area of PetroChina. Journal of Computer Science and Technology, 35(2), 368–378. https://doi.org/10.1007/s11390-020-9966-7
Appendix

Appendix A: BEKG Integration and Visualization

This appendix provides a complete BEKG build process and system visualization details.

As the relation classification model requires a pairwise entity as the input, it is necessary to conduct entity extraction prior to the relation classification. The whole process of extraction is shown in Figure A.1. During the entity extraction process, all sentences in the journal paper abstracts were first tokenized into the independent token. Then each token in each sentence was passed to the BERT-CRF model, outputting each token’s ‘BIO’ annotation. Since more than two entities may be extracted from a sentence, all the possible entity pairs were extracted and passed into the relation classification model to confirm whether any relation existed before further categorizing into more fine-grained relations. Therefore, these entities should be paired into the pairwise entity as the input of the relation classification model.

After combining the sentence along with the pairwise entity, these inputs were passed into the model to extract the relation between the pairwise entity. During the relation extraction stage, the whole process was regarded as a ‘29-way 1-shot 1-query’ setting few-shot classification. The input sentence and pairwise entities were taken as the query instance. Then, the BERT-Pair model calculated the similarity between the query instance and every 29 support instances.

Figure A1. An example of the whole extraction process from a sentence during final data extraction: blue for the head entity and red for the tail entity.

Figure A2. The user interface of the KG visualization system.

Figure A3. Demonstration of the search results gained from the KG visualization system.
and output the most similar relation from the 29 types of relations.

Despite having obtained a large number of extracted entities and relations, it was difficult for users to analyze such a diverse and massive amount of data effectively. A practical method is to use the current widely used KG web visualization technique, which could effectively represent information. Figure A.2 shows the initial user interface of the visualization system, which combines Navigation Bar, Search Box, and Graphical User Interface. The interface is designed to be user-friendly such that users can search and browse the target graph within a few simple steps.

Specifically, users can switch pages between the Navigation Bar and enter words with interests related to the built environment in the search box. The user interface of the search results is shown in Figure A.3. The nodes selected by users from the generated candidates related to the search words will appear in the Graphical User Interface.

Figure A.4 shows what the visualization system did when the users interacted with it. After the user clicked the nodes drawn by the front end, it immediately sent the request to the back end. The back end returned the linked nodes and relations back to the front end for visualization. The design of the front end was based on the characteristics of KGs. At the front end, the Vue framework was adopted for the basic graph interface and the D3 framework for the graph layout algorithm. At the back end, all triples were stored inside the Neo4j graph database, assessed by a functional framework called Mybatis. The spring-boot framework provided the basic logistic functions for web applications.

A sub-graph illustration related to the entity ‘cements’ of the complete BEKG is shown in Figure A.5. The nodes represent entities, and the lines between the nodes represent their relations. At the same time, the source paper title related to each entity can be found in its specific details. In this way, the visualization system offers an efficient way for users to retrieve and trace the research of interest.