Latent Dirichlet Allocation-based Approach for Automatically Mapping Components to Tasks in Modular Construction

Xiao Li[1], Chengke Wu[2]*, Weisheng Lu[3], Fan Xue[4]

Abstract: A large portion of cross-knowledge domain tasks have interdependent relationships with varied components in modular construction (MC). The MC components serve as the critical resources to support the task planning and execution for generating excellent MC products and services. Meanwhile, dynamic changes of tasks may adversely affect the design, procurement, and assembly of components. Furthermore, manually mapping components to tasks will be time-consuming and prevent forming effective work packages to achieve collaborative working. Thus, this study aims to develop an approach for automatically connecting components with tasks, which helps workers efficiently know the relationships between tasks and components. To this end, the latent Dirichlet allocation (LDA) approach is customized to this task-component mapping scenario. Moreover, compared with other leading unsupervised clustering techniques, e.g., K-means, the customized LDA demonstrated better performance on accuracy and efficiency for task-component mapping, and it can pave the way

[1] Xiao Li
Department of Real Estate and Construction, The University of Hong Kong, Hong Kong, China
E-mail: xl1991@hku.hk

[2]* Chengke Wu
Corresponding author, Department of Construction Management, Curtin University, Perth, Australia

[3] Weisheng Lu
Department of Real Estate and Construction, The University of Hong Kong, Hong Kong, China

[4] Fan Xue
Department of Real Estate and Construction, The University of Hong Kong, Hong Kong, China

for effective work package formation in MC.

Keywords: Modular Construction; Latent Dirichlet Allocation; Components; Tasks; Work Package.

1 Introduction

Modular construction (MC) is an innovative construction method to manufacture the facility (e.g., infrastructure, building) products in the factory and deliver them to the site for assembly, which has been widely used in military barracks, prison cells, school dormitories, hotels, healthcare, residences and even infrastructure [1-3]. MC has the potentials to reach Construction 4.0 and has been recognized with compelling advantages over traditional cast-in-situ construction by using advanced manufacturing production systems, such as shortened construction times [4], ensured quality [5], reduced site labor [6], and better working environment [7]. However, as MC involves multi-specialty and cross-domain knowledge in producing a facility, it in prefabrication factories still requires assigning tasks to various subcontractors, specialized work teams, or even robotics and automatic machinery [8]. For example, a single housing module requires a doze of trades to work on various systems, such as structure, door/window, wall, wet, print, mechanical, electrical and plumbing equipment [9]. Moreover, these physically connected systems in a prefabricated module are normally manufactured by coordinating interdependent tasks, which need seamless interface planning [10].

WBS is an efficient tasks planning tool that was jointly developed by the U.S. Department of Defense (DoD), NASA and the U.S. aerospace industry in 1962 [11]. It has now been widely extended for construction project management and planning [12-15]. A WBS is a hierarchical decomposition of the total work scope in a project, and a work package is the smallest element in WBS for planning one or more executable tasks [16]. The responsibility for executing a work package is normally assigned to a single person or organizational unit. The benefits of using work packages for MC project planning have also been recognized [17-21]: (i) it offers the fragmented MC project team members with clear instructions of their roles and responsibilities; (ii) it allows concurrent tasks to be simultaneously executed on an MC project; (iii) it helps measure the performance of schedule and cost by using techniques, such as earned value methods; (iv) it also supports risks, constraints, and disturbances (RCD) management at a task level. However, the tasks in work packages are mainly decomposed by project managers manually according to their experience and knowledge. This kind of task generation method can not ensure the accuracy and integrity of work packages in WBS.

Massive tasks in MC are to process and assemble the components for forming a final prefabricated module. Thus, mapping components with tasks can help generate more accurate work packages. For example, Ibrahim et al. [22] automatically generated the work packages using predefined attributes of BIM components. Isaac et al. [17] further considered the topological relations, sequences, and interfaces between specific BIM components to form work packages. However, despite previous studies bridging object-oriented BIM components with the tasks-oriented work packages, forming work packages in product-oriented MC, challenges still exist. For example, mapping and modeling relationships between MC components and tasks are difficult. Furthermore, dynamic changes in tasks will adversely affect the needs of MC components, vice versa. It happens because the tasks and components are closely interconnected.

This study aims to develop a latent Dirichlet allocation (LDA)-based modeling approach for automatically mapping the components with tasks in MC. To this end, three concrete objectives are designed. (1) to analyze the corresponding relations between components and tasks; (2) to establish the LDA approach for component-task mapping; (3) to validate the proposed approach and compare it with the state-of-the-art method via an MC case study.

2 Literature Review

The first type of relevant study is to generate WBS in project management. WBS has been widely used to hierarchically decompose a project into manageable pieces (e.g., work package) for reducing project complexity. Golpayegani and Emamizadeh[24] used neural networks to recognize the components and relationships in the project WBS. Siami-irdemoosa et al.[12] then applied a similar method to generate WBS for the complex underground construction project. Torkanfar and Azar[25] further developed a similarity measurement to conduct a semantic comparison of WBSs for achieving knowledge reuse. To improve the dynamic WBS generation, Lee et al. [26] proposed a system to support bi-directional transformation between processes and WBS by using the design structure matrix. Park and Cai[27] also established an automated linking mechanism between tasks and BIM objects to help generate a dynamic BIM database. In modular construction, the incompatibility between product-oriented off-site manufacturing and activity-oriented on-site construction can reduce the seamless interface and integration, Sutrisna et al.[15] proposed a hybrid WBS-matrix to bridge the off-site PBS and on-site WBS, and each prefabricated module is defined as a work package. Many efforts have been made to generate efficient WBS in project management. However, previous works mainly focusing on forming the static structure of WBS rather than defining dynamic models for mapping components to tasks, which are useful connections to develop work packages.

The second type of relevant study is to use the topic modeling approach for mapping two entities, such as business processes and software components. Aversano et al.[28] proposed an approach including modeling and measuring activities for evaluating the alignment level between a business process and the supporting software systems. Marcus and Maletic[29] used latent semantic indexing to automatically identify traceability links from system documentation to program source code. Pessiot et al.[30] then extended the Probabilistic Latent Semantic Analysis (PLSA) model for document clustering. Al-Anazi et al.[31] compared three clustering techniques: k-means, k-means fast, and k-medoids in document clustering using measures of cosine similarity, Jaccard similarity, and correlation coefficient. Baskara et al.[31] used LDA to discover a traceability link between business processes and software components. However, the mapping rules between MC components and tasks are only manually described rather than automatically modeled. Particularly for product-oriented MC with massive components and related tasks, there is a lack of a mapping model to match the MC components with tasks.

LDA is an unsupervised probabilistic model extensively applied to analyze discrete and unstructured data, such as texts. The LDA model first learns to identify main topics from a large archive of text documents (i.e., the training process). In this stage, the LDA model, in essence, clusters documents based on the topics. The number of topics can be pre-defined according to certain criteria, e.g., perplexity and similarity. The documents used for training the LDA model are also called a text corpus. After training, the LDA model can assign topics to a new document (i.e., the validation or testing process)[33]. LDA has been widely used in many areas[43], such as social networks, software engineering, crime science, geography, political science, medical, and linguistic science. In the construction sector, LDA has been adopted by many studies for topic modeling in various aspects, including identifying main onsite issues and their changes over time[34], understanding the perceptions of Chinese governments towards mitigating environmental impacts of highway construction projects[35], investigating main types of lawsuit cases[36], and categorizing main hazards from injury reports[37]. Although Dirichlet topic distribution cannot capture correlations among words, the components in MC do not require sequential correlations for component-task mapping. Meanwhile, the LDA model is highly modular and can therefore be easily extended and embedded in more complicated models to improve the accuracy of component-task mapping. For example, LDA has the potential to be enriched with topology ontology models to analyze spatial relations among semantically related components and then cluster them based on certain categories.

3 Analysis of the Relations between Components and Tasks

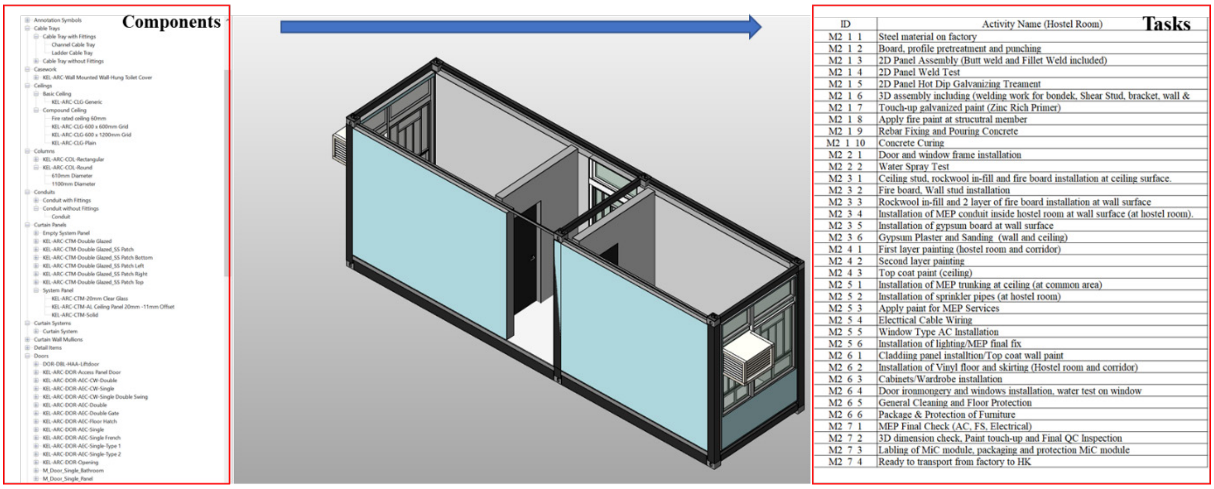
The bill of material (BoM) generated from BIM can serve as the product breakdown structure (PBS), and the tasks decomposed from the work breakdown structure (WBS) can be used for project planning. A PBS and WBS in modular construction can be represented by two eight-tuples, respectively. Here we name all elements and materials to form the final prefabricated products as the components.

$$\text{Definition 1: } PBS = (CN, CID, CS, CM, CW, CO, CHR, CC) \quad (1)$$

$$\text{Definition 2: } WBS = (TN, TID, TP, TSE, TR, TP, THR, TD) \quad (2)$$

Where PN is component name, CID is component ID, CS is component size, CM is component material, CW is component weight, CO is the component origin, CHR is hierarchical relation between components, CC is constraint relation between components; TN is task name, TID is task ID, TP is task period, TSE is the start and end time, TR is the resource, THR is hierarchical relation between tasks, TD is predecessor or successor dependency between tasks.

The relations between PBS and WBS can be deduced based on the above definition. (1) components in PBS and tasks in WBS can be well mapped through name, ID, or other similar properties; (2) As PBS and WBS follow similar decomposition philosophy by using top-down, from coarse to fine, stepwise refinement, both components and tasks can form hierarchical relations; (3) geometric and non-geometric constraints between components may lead to dependencies between tasks.



- ⊞ Annotation Symbols
- ⊟ Cable Trays
  - ⊟ Cable Tray with Fittings
    - Channel Cable Tray
    - Ladder Cable Tray
  - ⊞ Cable Tray without Fittings
- ⊟ Casework
  - ⊞ KEL-ARC-Wall Mounted Wall-Hung Toilet Cover
- ⊟ Ceilings
  - ⊟ Basic Ceiling
    - KEL-ARC-CLG-Generic
  - ⊟ Compound Ceiling
    - Fire rated ceiling 60mm
    - KEL-ARC-CLG-600 x 600mm Grid
    - KEL-ARC-CLG-600 x 1200mm Grid
    - KEL-ARC-CLG-Plain
- ⊟ Columns
  - ⊞ KEL-ARC-COL-Rectangular
  - ⊟ KEL-ARC-COL-Round
    - 610mm Diameter
    - 1100mm Diameter
- ⊟ Conduits
  - ⊞ Conduit with Fittings
  - ⊟ Conduit without Fittings
    - Conduit
- ⊟ Curtain Panels
  - ⊞ Empty System Panel
  - ⊞ KEL-ARC-CTM-Double Glazed
  - ⊞ KEL-ARC-CTM-Double Glazed_SS Patch
  - ⊞ KEL-ARC-CTM-Double Glazed_SS Patch Bottom
  - ⊞ KEL-ARC-CTM-Double Glazed_SS Patch Left
  - ⊞ KEL-ARC-CTM-Double Glazed_SS Patch Right
  - ⊞ KEL-ARC-CTM-Double Glazed_SS Patch Top
  - ⊟ System Panel
    - KEL-ARC-CTM-20mm Clear Glass
    - KEL-ARC-CTM-AL Ceiling Panel 20mm -11mm Offset
    - KEL-ARC-CTM-Solid
- ⊟ Curtain Systems
  - ⊞ Curtain System
- ⊞ Curtain Wall Mullions
- ⊞ Detail Items
- ⊟ Doors
  - ⊞ DOR-DBL-HAA-Liftdoor
  - ⊞ KEL-ARC-DOR-Access Panel Door
  - ⊞ KEL-ARC-DOR-AEC-CW-Double
  - ⊞ KEL-ARC-DOR-AEC-CW-Single
  - ⊞ KEL-ARC-DOR-AEC-CW-Single Double Swing
  - ⊞ KEL-ARC-DOR-AEC-Double
  - ⊞ KEL-ARC-DOR-AEC-Double Gate
  - ⊞ KEL-ARC-DOR-AEC-Floor Hatch
  - ⊞ KEL-ARC-DOR-AEC-Single
  - ⊞ KEL-ARC-DOR-AEC-Single French
  - ⊞ KEL-ARC-DOR-AEC-Single-Type 1
  - ⊞ KEL-ARC-DOR-AEC-Single-Type 2
  - ⊞ KEL-ARC-DOR-Opening
  - ⊞ M_Door_Single_Bathroom
  - ⊞ M_Door_Single_Panel

ID	Activity Name (Hostel Room)
M2 1 1	Steel material on factory
M2 1 2	Board, profile pretreatment and punching
M2 1 3	2D Panel Assembly (Butt weld and Fillet Weld included)
M2 1 4	2D Panel Weld Test
M2 1 5	2D Panel Hot Dip Galvanizing Treament
M2 1 6	3D assembly including (welding work for bondek, Shear Stud, bracket, wall &
M2 1 7	Touch-up galvanized paint (Zinc Rich Primer)
M2 1 8	Apply fire paint at strucutral member
M2 1 9	Rebar Fixing and Pouring Concrete
M2 1 10	Concrete Curing
M2 2 1	Door and window frame installation
M2 2 2	Water Spray Test
M2 3 1	Ceiling stud, rockwool in-fill and fire board installation at ceiling surface.
M2 3 2	Fire board, Wall stud installation
M2 3 3	Rockwool in-fill and 2 layer of fire board installation at wall surface
M2 3 4	Installation of MEP conduit inside hostel room at wall surface (at hostel room).
M2 3 5	Installation of gypsum board at wall surface
M2 3 6	Gypsum Plaster and Sanding (wall and ceiling)
M2 4 1	First layer painting (hostel room and corridor)
M2 4 2	Second layer painting
M2 4 3	Top coat paint (ceiling)
M2 5 1	Installation of MEP trunking at ceiling (at common area)
M2 5 2	Installation of sprinkler pipes (at hostel room)
M2 5 3	Apply paint for MEP Services
M2 5 4	Electtical Cable Wiring
M2 5 5	Window Type AC Installation
M2 5 6	Installation of lighting/MEP final fix
M2 6 1	Claddiing panel installtion/Top coat wall paint
M2 6 2	Installation of Vinyl floor and skirting (Hostel room and corridor)
M2 6 3	Cabinets/Wardrobe installation
M2 6 4	Door ironmongery and windows installation, water test on window
M2 6 5	General Cleaning and Floor Protection
M2 6 6	Package & Protection of Furniture
M2 7 1	MEP Final Check (AC, FS, Electrical)
M2 7 2	3D dimension check, Paint touch-up and Final QC Inspection
M2 7 3	Labling of MiC module, packaging and protection MiC module
M2 7 4	Ready to transport from factory to HK

Fig.1 Components from BoM of BIM and Tasks from WBS

4 Research Method

The proposed method in this study comprises three processes, including data preprocessing, LDA generation and similarity matching. The output of these processes could be a traceability matrix between components and tasks. Fig.2 presents the flow of the proposed method. Firstly, MC tasks name and components names are extracted by text pre-processing and combined into a set of documents. Then, the document-topic and topic-word distributions are generated by training the LDA model. Finally, new components are mapped to the relevant MC tasks considering the similarity between tasks' names and the components' names in the training dataset.

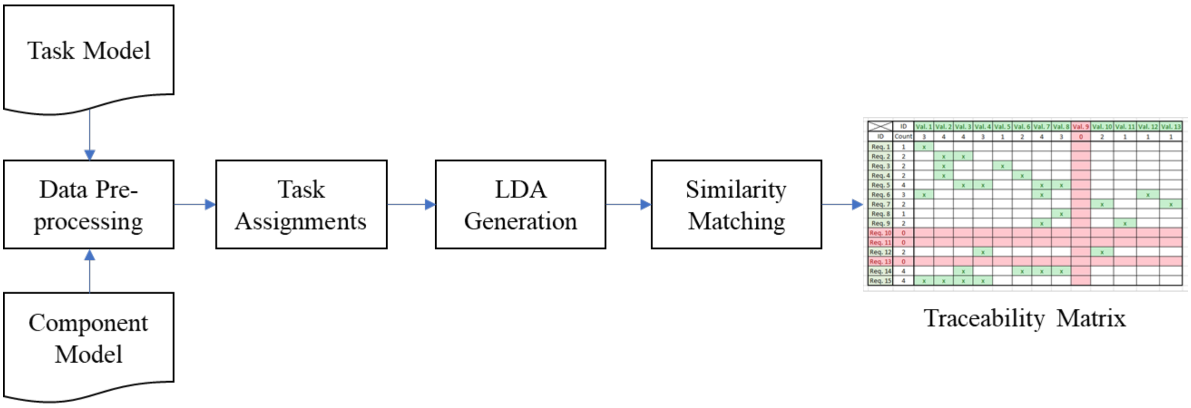


Fig.2 Component-task mapping process

3.1 Data Preprocessing

The conventional LDA model relies on text data, but the noise of texts, such as stopwords that do not carry important meanings (e.g., ‘the’ and ‘a’) and different forms of words (e.g., plural and singular forms), can affect model performance [38]. In this study, the LDA model aims to map MC components to tasks. The data inputs consist of two models, i.e., task model and MC component model. For the task model, the MC production workflow is used as an input. Each task name in the workflow is extracted as a unique task type. As for the MC component model, the BoM is used as an input. Each material inside the BoM represents a component. For training the LDA model, components in the component model are manually assigned to task types in the task model, forming task assignments (TAs) as model inputs. Each TA includes one task type and a set of components. However, the names of tasks and components are still texts. Hence, data preprocessing is required to reduce noise, including four steps: tokenization (i.e., separating a name into words), lowercasing, lemmatization (i.e., converting different forms of a word to its basic form), and stopwords removal. To realize the latter two steps, the dictionaries that record basic forms of words and common stopwords in the MC domain are developed and applied. Then, the LDA model can be trained to get topics (i.e., task types) over documents (i.e., TAs), probability distributions and words (i.e., components) over topics (i.e., Task types) probability distributions.

3.2 Latent Dirichlet Allocation Development and Training

The LDA model is a hierarchical Bayesian model, which assumes that documents comprise a random collection of words over potential topics, and each topic has a distribution over words[39]. The LDA model relies on two Dirichlet distributions, i.e., **DD1** $P(\theta;\alpha)$ and **DD2** $(\phi;\beta)$, and two types of multinominal distributions, i.e., **MD1** $P(Z|\theta)$ and **MD2** $P(W|Z)$. The α and β are hyper-parameters, θ and ϕ are latent variables, and Z and W represent topics (i.e., task types) and words (i.e., components), respectively. The **DD1** maps TAs to task types, determining the probabilities that a TA belongs to each type. The **DD2** maps task type to components, determining the probabilities of assigning different components in a particular task. The LDA model is trained by simulating the processing of generating documents based on the topics of documents and words. Thus, **MD1** and **MD2** are derived from the Dirichlet distributions to generate TAs. Specifically, suppose there are *N* components in a TA to be generated, an **MD1** is created to determine the task type of each component (i.e., $\{c_1, c_2 \ldots c_N\}$). Then, for each component (in this stage, a component only has an associated task type but does not have specific names), an **MD2** is created to determine its name based on the task type. In other words, for generating *N*-specific components, *K* **MD2** distributions covering all task types should be developed. The process of generating TAs using the LDA model is illustrated in Eq.(3) and Figure 3, where *M*, *K*, and *N* present the number of TA, task types, and components in a TA, respectively.

$$\prod_{j=1}^{M} P(\theta_j; \alpha) \prod_{i=1}^{K} (\phi_i; \beta) \prod_{n=1}^{N} P\left(Z_{j,n} | \theta_j\right) P(W_{j,n} | Z_{j,n}) \quad (3)$$

Training the LDA model requires estimating the parameters α and β so that the model can maximize the probability of generating similar TAs in the training corpus. This can be achieved by Gibbs sampling. The sampling works as follows: 1) randomly assigns a task type to each component in the training corpus, 2) summarizes the number of components belonging to each task type in each TA as well as the number of components belonging to each task type in the entire corpus, 3) for each component, recalculate the **MD2** based on the statistics obtained in the previous step, 4) reassigns a new task type to the current component using the newly obtained MD2, 5) repeats above process until parameters in **MD1** and **MD2** converge. In practice, the maximum number of iterations is commonly defined to save computation power and time[40]. After training, each task type can be represented by a set of top key components ranked by the task-component (**MD2**) distributions.

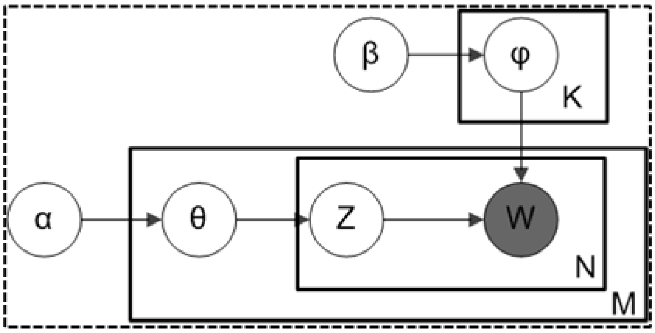


Figure 3 Mechanism of mapping components to MC tasks

3.3 Similarity Matching

During model testing, the trained LDA model takes in new MC components and tasks for component-task mapping. After the pre-processing process introduced before, the names of most MC components can be converted into standard forms. However, as the same task can be expressed by different names (e.g., rebar fixing and rebar binding), it is very difficult to convert the names of MC tasks in this way. Thus, the LDA model can encounter tasks with names that do not exist in the training corpus. To minimize the impact of such ambiguity, a similarity matching method is applied to task names. Specifically, all the names of task types in the training corpus serve as references, when a new task comes in, it 1) enumerates each existing task type, 2) computes similarity values $\boldsymbol{s}$ between their names following Eq. (4), 3) finds the task type with the maximum similarity as the standard form to which components can be assigned. To facilitate similarity computation (i.e., step 2), word embeddings are employed. This is because task names can be represented by words consisting of different characters but expressing similar meanings. Word embeddings are low-dimension (e.g., 50-300) real-valued vectors that can effectively capture the meanings of words. As such, the $\boldsymbol{e^c}$ and $\boldsymbol{e^e}$ in Eq. (4) are obtained by 1) tokenizing the new and existing task names into words and 2) averaging embeddings of these words. Word embeddings should be learned through ML models, e.g., the famous word2vec model [41]. However, training such models is very data demanding and time-consuming. Hence, word embeddings trained in previous work can be utilized, which is also a common practice [42].

$$\boldsymbol{s} = \frac{\sum_{i=1}^{n} e_i^c \times e_i^e}{\sqrt{\sum_{i=1}^{n}\left(e_i^c\right)^2} \times \sqrt{\sum_{i=1}^{n}\left(e_i^e\right)^2}} \quad (\boldsymbol{n}\text{=the dimension of the word embedding vector}) \qquad (4)$$

3.4 Improved Latent Dirichlet Allocation for Component-task mapping

Following the conventional approach to develop LDA models, a TA only consists of associated products. However, such simple data cannot suit situations in practical MC projects. One common problem is that when producing a room module, a product is required in multiple tasks thus should be included in multiple TAs. For instance, a gypsum board simultaneously belongs to the tasks ‘board pre-treatment and punching’ and ‘installation of the board at wall surface’. Therefore, the conventional LDA model cannot distinguish such differences and can only make a random guess when mapping a new ‘gypsum board’ product.

To address this issue, spatial relations of module products are identified by referring to the BIM model and MTO to enrich TA data. The process has three steps: 1) as the BoM is derived from the BIM model, associated spatial instances (i.e., instances of MTO classes) of each product can be extracted from the BIM model (e.g., a gypsum board is related to a ‘wall’ instance in BIM); 2) a ‘contains’ spatial relation is set up between the spatial instance and product; and 3) the triple taking the form (i.e., ‘spatial instance contains product’) is added to the TA data. If no spatial relation is found, for instance, the product is used in pre-installation tasks (e.g., board punching), a triple ‘non contains product’ is added. As such, the same products can be distinguished by different spatial relations with a spatial instance.

Moreover, after investigating the workflow charts and schedules of MC projects in Hong Kong, it is found that 1) the unit for managing MC projects is individual rooms (e.g., house modules); 2) distinct spatial instances are considered as one system, and tasks performed on these instances are managed as one package (e.g., installing studs in all walls of a room is treated as a single package). Therefore, it is unlikely that the same triple (e.g., ‘wall contains studs’) appears multiple times in the TA data and confuses the model.

4 Experiment and Result

The proposed approach is demonstrated in a case study, which is a student residence modular project located in Hong Kong Island. This modular student residence includes two 17-storey student residence tower buildings on top of a three-storey podium structure. A total of 1224 student places will be offered, and supporting facilities such as canteen, common room, laundry rooms and car park will be fitted into the podium. The typical floor layout comprises 28 prefabricated modules. This project has five types of modules with different dimensions. The production of modules is considered as the scenario to demonstrate the component-task mapping. The MC tasks are extracted from prefabricated module production workflow and MC components from BoMs. As such, 220 TAs were manually labeled to train the LDA model, and several examples are shown in Table 1.

Table 1 Example List of MC tasks and components

Task type	Components
Door and window ironmongery installation	door ironmonger, aluminum window, FRP timber door, glass panel, non-FRP timber door
Cabinets installation	wall cabinet, toilet-sink cabinet, TV cabinet
Door and window frame installation	aluminum cladding, rockwool acoustic insulation door threshold
Installation of sprinkler pipes	galvanized steel pipes, galvanized steel pipe fittings, grooved pipe fittings and couplings, sprinkler heads, smoke detectors, fire alarm bells
Packaging and protection of furniture	chair, mattress, hand sliding window curtain, hand sliding shower curtain, table, bed, mirror
Rebar fixing	spacer and bar chair, tie wire, steel reinforcement bar

First, text processing is performed to standardize data and remove noise. Then, topics over documents probabilities are generated using LDA. The hyperparameters used in this experiment are $\alpha = 2.38$, $\beta = 0.01$ and K = 23. We used this setting because it allows the model to converge quickly while can provide the best result from our observation. Fig. 4 shows the top words probability distributions over the topics that have been generated by the model. From the words probability distributions, we can interpret that Topic 0 (Figure 4(a)) is about installing cabinets, Topic 1 (Figure 4(b)) is about installing sprinkler pipes, Topic 2 (Figure 4(c)) is about installing door and window frames, and Topic 3 (Figure 4(d)) is about installing door and window ironmongery. The LDA model was created using Python libraries, i.e., Natural Language Toolkit (NLTK), Gensim, and pyLDAvis for preprocessing, model training, and result visualization.

In addition, word embeddings trained in the works [42] were adopted. The embeddings (n=300) were trained on large databases (e.g., Wikipedia) and have covered more than 400000 English words, which should be comprehensive for the study. From this similarity matching process, a set of components related to the task are obtained.

For evaluating the capacity of the LDA model, 50 MC components were randomly selected, then 1) MC tasks of the components were labeled manually as ground truth, 2) the components were fed into the trained LDA model, which returned their corresponding predicted MC tasks, 3) the manually labeled and predicted tasks were compared, and the mapping accuracy was computed.

The outputs of training the LDA model are **MD1** (i.e., mapping components to task types) and **MD2** (i.e., assigning names to components based on their task types). As the research aims to automatically map MC components to tasks, **MD2** is more important in the experiments. Specifically, **MD2** consists of lists, where each list records the most representative components of the current task. Figure 4(a) illustrates the **MD2** list of the MC task-cabinet installation as an example. The blue and red bars represent the frequency of the component appearing in the entire dataset (i.e., all TAs) and TAs of the current task type. In the example, it is obvious that different types of cabinets can largely represent the MC task.

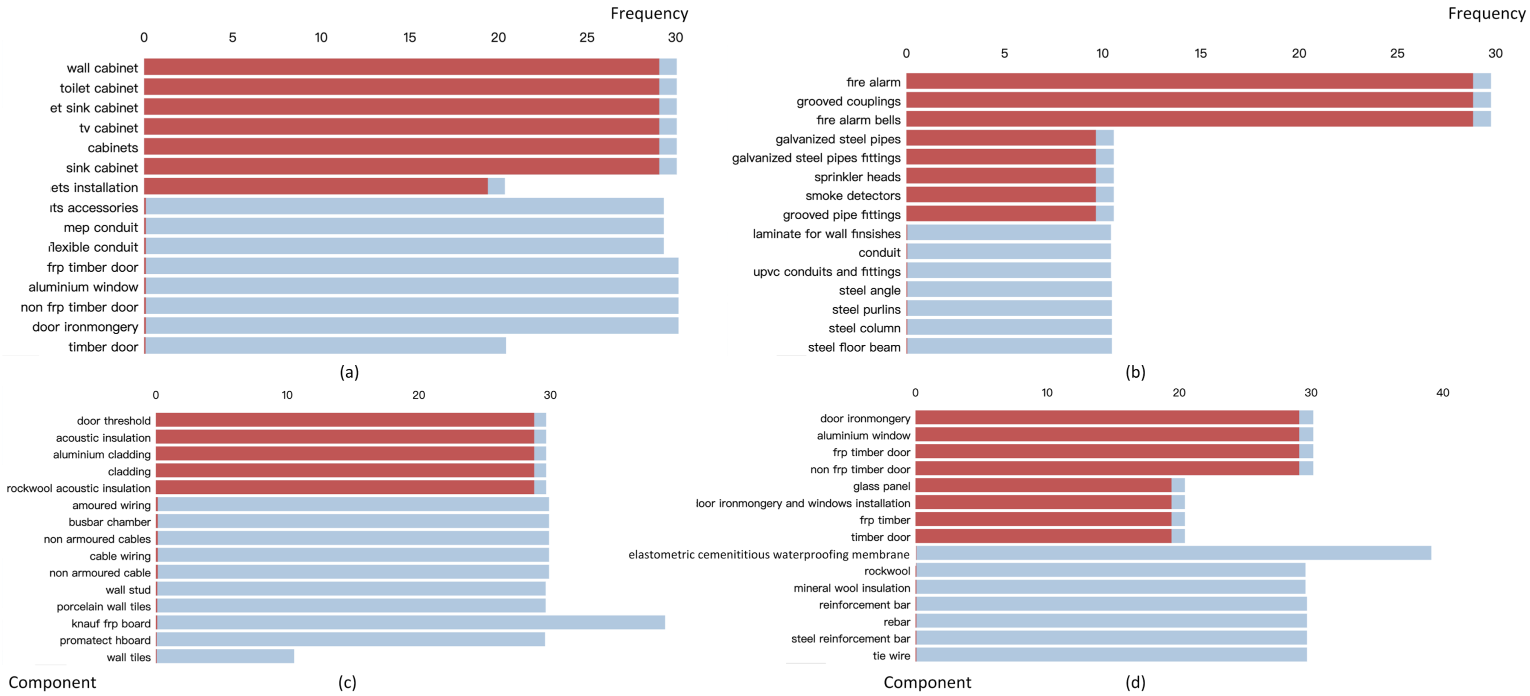
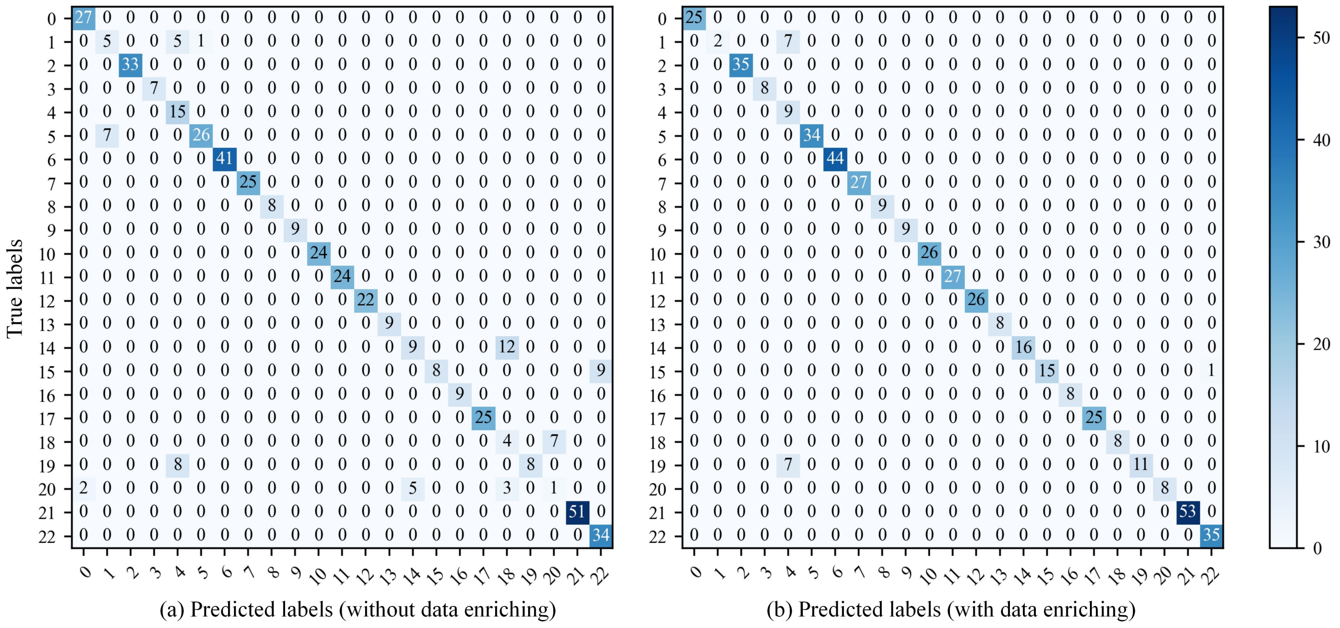


Figure 4 The frequencies of representative components of four tasks

Figure 5 (a) shows the confusion matrix of mapping results. There are several wrongly assigned components out of the 50 components used for testing, reaching an overall accuracy of 88.0%. In addition, the time to manually assign these components was 638 seconds, nearly 2000 times that of using the LDA model, which only consumed 0.32 seconds. The experiment results have proved that the LDA model is a practical tool in MC projects to effectively map components to tasks and facilitate work package-based management approaches.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	5	0	0	5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	7	0	0	0	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	22	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	12	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	9
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	7	0	0
19	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0
20	2	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	3	0	1	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	51	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	34

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	2	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	26	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	1
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0
19	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	53	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35

Figure 5 Confusion matrix of product-task mapping (both x and y-axis are task ID, the tasks

are not arranged in sequence in the product-task mapping stage)

Additionally, it is found that the errors are caused by the fact that some MC components can belong to multiple tasks, which misleads the LDA model. For instance, the component Knauf FRP board is required in both board pre-treatment and wall stud installation. Thus, the LDA model can only make a random guess when assigning such a component. To address this issue, the relations (especially spatial and topological relations) among the components can be considered when preparing the training data (The results are presented in Fig.5(b)). For instance, each BoM includes not only individual components but also triples which are formed by two components and one spatial relation (e.g., under and above). The relations can be directly extracted from BIM models of the MC modules. In this way, when the model encounters ambiguous components, it can take the triple data to make the final assignment. However, errors still occur for a few products which cannot be distinguished even spatial relations are utilized. For instance, the product ‘ceiling panel’ should be used in task 9, ‘installing studs at the ceiling’, as stated in the workflow. However, the LDA model wrongly assigns it to task 15‘layer and coat painting ceiling’ because the product has the same spatial relations in both tasks. Nevertheless, the accuracy is still increased by more than 7%.

5 Conclusion

This study has proposed an approach using Latent Dirichlet Allocation (LDA) to discover traceability links between components and tasks in modular construction (MC). LDA is applied to get the topic probability distributions of components and tasks. When the topic probability distributions are determined, the similarity of topic probability can be computed. A threshold is set for the similarity matching to gather the most relevant components for a given task. However, the LDA model sometimes may only make a random guess when assigning the component, which leads to mapping errors. To address this issue, the relations (especially spatial and topological relations) among the components are considered when preparing the training data to improve the conventional LDA model.

The experiment results show that LDA can mine topics on components and tasks and can be used to discover traceability links between them. And it is a practical tool in MC projects to effectively map components to tasks and facilitate work package-based management approaches. Future studies will focus on transforming the semantic-enrich tasks into work packages for further collaborative project planning.

References

[1] Gann, D. M. (1996). Construction as a manufacturing process? Similarities and differences

between industrialized housing and car production in Japan. *Construction Management and Economics*, *14*(5), 437-450.

[2] Teng, Y., Mao, C., Liu, G., & Wang, X. (2017). Analysis of stakeholder relationships in the industry chain of industrialized buildings in China. *Journal of Cleaner Production*, *152*, 387-398.

[3] Li, X., Wu, C., Wu, P., Xiang, L., Shen, G. Q., Vick, S., & Li, C. Z. (2019). SWP-enabled constraints modeling for the on-site assembly process of prefabrication housing production. *Journal of Cleaner Production*, *239*, 117991.

[4] Eriksson, P. E., Olander, S., Szentes, H., & Widén, K. (2014). Managing short-term efficiency and long-term development through industrialized construction. *Construction Management and Economics*, *32*(1-2), 97-108.

[5] Goh, M., & Goh, Y. M. (2019). Lean production theory-based simulation of modular construction processes. *Automation in Construction*, *101*, 227-244.

[6] Gong, P., Teng, Y., Li, X., & Luo, L. (2019). Modeling constraints for the on-site assembly process of prefabrication housing production: a social network analysis. *Sustainability*, *11*(5), 1387.

[7] Hammad, A. W., Akbarnezhad, A., Wu, P., Wang, X., & Haddad, A. (2019). Building information modeling-based framework to contrast conventional and modular construction methods through selected sustainability factors. *Journal of Cleaner Production*, *228*, 1264-1281.

[8] Arashpour, M., Kamat, V., Bai, Y., Wakefield, R., & Abbasi, B. (2018). Optimization modeling of multi-skilled resources in prefabrication: Theorizing cost analysis of process integration in off-site construction. *Automation in Construction*, *95*, 1-9.

[9] Lawson, M., Ogden, R., & Goodier, C. (2014). *Design in modular construction*. CRC Press.

[10] Salama, T., Salah, A., Moselhi, O., & Al-Hussein, M. (2017). Near optimum selection of module configuration for efficient modular construction. *Automation in Construction*, *83*, 316-329.

[11] DOD and NASA. (1962). DOD and NASA Guide: PERT COST Systems Design.

[12] Siami-Irdemoosa, E., Dindarloo, S. R., & Sharifzadeh, M. (2015). Work breakdown structure (WBS) development for underground construction. *Automation in Construction*, *58*, 85-94.

[13] Li, D., & Lu, M. (2017). Automated generation of work breakdown structure and project network model for earthworks project planning: a flow network-based optimization approach. *Journal of Construction Engineering and Management*, *143*(1), 04016086.

[14] Jung, Y., & Woo, S. (2004). Flexible work breakdown structure for integrated cost and schedule control. *Journal of Construction Engineering and Management*, *130*(5), 616-625.

[15] Sutrisna, M., Ramanayaka, C. D., & Goulding, J. S. (2018). Developing work breakdown structure matrix for managing offsite construction projects. *Architectural Engineering and Design Management*, *14*(5), 381-397.

[16] Project Management Institute. (2013). Guide to the Project Management Body of Knowledge (PMBOK Guide). *Project Management Institute*, Newtown Square, PA, 5th ed.

[17] Isaac, S., Curreli, M., & Stoliar, Y. (2017). Work packaging with BIM. *Automation in Construction*, *83*, 121-133.

[18] Li, X., Shen, G. Q., Wu, P., Xue, F., Chi, H. L., & Li, C. Z. (2019). Developing a conceptual framework of smart work packaging for constraints management in prefabrication housing production. *Advanced Engineering Informatics*, *42*, 100938.

[19] Liu, H., Al-Hussein, M., & Lu, M. (2015). BIM-based integrated approach for detailed construction scheduling under resource constraints. *Automation in Construction*, *53*, 29-43.

[20] Liu, H., Lu, M., & Al-Hussein, M. (2016). Ontology-based semantic approach for construction-oriented quantity take-off from BIM models in the light-frame building industry. *Advanced Engineering Informatics*, *30*(2), 190-207.

[21] 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.

[22] Ramasesh, R. V., & Browning, T. R. (2014). A conceptual framework for tackling knowable unknown unknowns in project management. *Journal of Operations Management*, *32*(4), 190-204.

[23] Ibrahim, Y. M., Lukins, T. C., Zhang, X., Trucco, E., & Kaka, A. P. (2009). Towards automated progress assessment of work package components in construction projects using computer vision. *Advanced Engineering Informatics*, 23(1), 93-103.

[24] Golpayegani, S. A. H., & Emamizadeh, B. (2007). Designing work breakdown structures using modular neural networks. *Decision Support Systems*, *44*(1), 202-222.

[25] Torkanfar, N., & Azar, E. R. (2020). Quantitative similarity assessment of construction projects using WBS-based metrics. *Advanced Engineering Informatics*, *46*, 101179.

[26] Lee, J., Deng, W. Y., Lee, W. T., Lee, S. J., Hsu, K. H., & Ma, S. P. (2010). Integrating process and work breakdown structure with design structure matrix. *Simulation*, *7*, 8.

[27] Park, J., & Cai, H. (2017). WBS-based dynamic multi-dimensional BIM database for total construction as-built documentation. *Automation in Construction*, *77*, 15-23.

[28] Aversano, L., Grasso, C., & Tortorella, M. (2016). Managing the alignment between

business processes and software systems. *Information and Software Technology*, *72*, 171-188.

[29] Marcus, A., & Maletic, J. I. (2003, May). Recovering documentation-to-source-code traceability links using latent semantic indexing. In *25th International Conference on Software Engineering, 2003. Proceedings.* (pp. 125-135). IEEE.

[30] Pessiot, J. F., Kim, Y. M., Amini, M. R., & Gallinari, P. (2010). Improving document clustering in a learned concept space. *Information processing & management*, *46*(2), 180-192.

[31] Al-Anazi, S., AlMahmoud, H., & Al-Turaiki, I. (2016). Finding similar documents using different clustering techniques. *Procedia Computer Science*, *82*, 28-34.

[32] Baskara, A. R., Sarno, R., & Solichah, A. (2016, October). Discovering traceability between business process and software component using Latent Dirichlet Allocation. In *2016 International Conference on Informatics and Computing (ICIC)* (pp. 251-256). IEEE.

[33] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *The Journal of Machine Learning Research*, *3*, 993-1022.

[34] Lin, J. R., Hu, Z. Z., Li, J. L., & Chen, L. M. (2020). Understanding On-Site Inspection of Construction Projects Based on Keyword Extraction and Topic Modeling. *IEEE Access*, *8*, 198503-198517.

[35] Wu, L., Ye, K., Gong, P., & Xing, J. (2019). Perceptions of governments towards mitigating the environmental impacts of expressway construction projects: A case of China. *Journal of Cleaner Production*, *236*, 117704.

[36] Jallan, Y., Brogan, E., Ashuri, B., & Clevenger, C. M. (2019). Application of natural language processing and text mining to identify patterns in construction-defect litigation cases. *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, *11*(4), 04519024.

[37] Zhong, B., Pan, X., Love, P. E., Sun, J., & Tao, C. (2020). Hazard analysis: a deep learning and text mining framework for accident prevention. *Advanced Engineering Informatics*, *46*, 101152.

[38] Hoffman, M., Bach, F., & Blei, D. (2010). Online learning for latent Dirichlet allocation. *advances in neural information processing systems*, *23*, 856-864. [38] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *The Journal of Machine Learning Research*, *3*, 993-1022.

[39] Gelfand, A. E. (2000). Gibbs sampling. *Journal of the American Statistical Association*, *95*(452), 1300-1304.

[40] K. W. Church, *Word2Vec,* Natural Language Engineering. 23(1) (2017) pp. 155-162.

[41] Baker, H., Hallowell, M. R., & Tixier, A. J. P. (2020). Automatically learning construction injury precursors from the text. *Automation in Construction*, *118*, 103145.

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

[43] Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y., & Zhao, L. (2019). Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications*, *78*(11), 15169-15211.