Automation in Construction 161 (2024) 105345

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Natural language instructions for intuitive human interaction with robotic assistants in field construction work

Somin Park [a], Xi Wang [b], Carol C. Menassa [a,*], Vineet R. Kamat [a], Joyce Y. Chai [c]

[a] Dept. of Civil and Env. Engineering, University of Michigan, USA
[b] Dept. of Construction Science, Texas A&M University, USA
[c] Dept. of Elec. Engineering and Computer Science, University of Michigan, USA

ARTICLE INFO

Keywords:
Human-Robot Collaboration (HRC)
Natural interaction
Natural language instruction
Natural Language Processing (NLP)
Natural language understanding
Drywall installation

ABSTRACT

Human-Robot Collaboration (HRC) has shown promise of combining human workers' flexibility and robot assistants' physical abilities to jointly address the uncertainties inherent in construction work. In HRC, natural language-based interaction can enable human workers who are non-experts in robot programming to intuitively communicate with robot assistants. However, limited research has been conducted on this topic in construction. This paper proposes a framework to allow human workers to interact with construction robots based on natural language instructions for pick-and-place construction operations. The proposed method consists of three modules: Natural Language Understanding (NLU), Information Mapping (IM), and Robot Control (RC). A case study for drywall installation evaluates the proposed approach. Results indicate over 99% accuracy in NLU and IM, allowing a robot to perform tasks accurately for a given set of natural language instructions. It highlights the potential of using natural language-based interaction to replicate human-like communication in human-robot teams.

1. Introduction

Robotics is considered an effective means to address issues of labor shortages and stagnant growth of productivity in construction [1–3]. However, it is challenging for robots to work on construction sites due to evolving and unstructured work environments [4,5], differing conditions from project to project [6], and the prevalence of quasi-repetitive work tasks [7]. This is in contrast to automated manufacturing facilities that have structured environments [4].

Collaboration between humans and robots has the potential to address several such challenges inherent in the performance of construction tasks in the field. The advantage of collaborative robots lies in the opportunity to combine human intelligence and flexibility with robot strength, precision, and repeatability [8,9]. Collaboration can increase productivity, improve quality and enhance human safety [10,11]. It can also reduce physical exertion for humans since repetitive tasks will be carried out by robots. Therefore, in Human-Robot Collaboration (HRC), skills of human operators and robots can complement each other to complete designated tasks.

On today's construction sites, communication between workers is essential allowing work crews to have many degrees of freedom in organizing and coordinating the work, and dealing with the dynamic and unpredictable environments [12]. Similarly, when collaborative robots assist human workers, interaction between humans and robots is critical [1]. In human-robot construction teams, most of the robots are currently in the lower level of robot autonomy where human workers determine task plans and robots execute them [13]. To deliver plans generated by human workers to robots, human operators need proper interfaces [14]. However, designing intuitive user interfaces is one of the key challenges of HRC since interaction with robots usually requires specialized knowledge in humans [15]. Therefore, intuitive and natural interaction enables human operators to easily interact with robots while taking full advantage of human skills [15,16].

1.1. Enhancing HRC in construction through natural interaction

Perceived ease of use and usefulness were emphasized as having critical roles in encouraging construction personnel to engage with HRC [17]. The complexity involved in learning and using new technologies was identified as a substantial barrier that negatively impacts workers'

* Corresponding author.
E-mail addresses: somin@umich.edu (S. Park), xiwang@tamu.edu (X. Wang), menassa@umich.edu (C.C. Menassa), vkamat@umich.edu (V.R. Kamat), chaijy@umich.edu (J.Y. Chai).

https://doi.org/10.1016/j.autcon.2024.105345
Received 29 September 2023; Received in revised form 6 February 2024; Accepted 24 February 2024
Available online 1 March 2024

willingness to work in the HRC in construction [17]. Thus, it is imperative for human workers to experience an effortless learning process, ensuring that interactions with the robotic system are straightforward and uncomplicated. In this context, we define "natural and intuitive interaction" within the context of HRC in the construction industry as a mode of communication between humans and robots that is inherently understandable and easy to use, requiring minimal training and cognitive effort from human operators. This definition emphasizes that the interaction should not simply mirror colloquial human-to-human communication, but must be easily adapted to the specific context of construction tasks while being inherently understandable and straightforward to implement for human operators.

Several recent studies have investigated natural HRC in the construction industry using various communication channels such as gesture [18], Virtual Reality (VR) [19], brainwaves [20], and speech [21]. Among them, speech interaction has been considered as the most natural and intuitive way of communication in the human-robot interaction field [22–25]. Natural language instructions, delivered through a speech channel, allow human operators to deliver their requests accurately and efficiently [26]. Users' intents about action, tools, workpieces, and location for HRC can be accurately expressed through natural language without information loss in ways distinct from other simplified requests [27,28]. In addition, users do not need to design informative expressions when communicating through existing languages, making the interaction efficient.

Pick-and-place operations, which are commonly performed by industrial robots, have increasingly been guided by natural language instructions [28–31]. In the construction domain, such operations are critical for tasks on structures (e.g., bricklaying and concrete block installation), surface (e.g., tile and drywall installation), and fixtures (e.g., glass panel installation). However, while there is significant potential in applying natural language instructions for these construction tasks, collaborating with robots remains a challenge. The primary challenge lies in the need for a comprehensive system that can integrate the analysis of language instructions with the subsequent robot controls. Moreover, there is a need for a language model to extract task-specific information for construction as well as a method to map the extracted information onto the dynamic construction sites.

1.2. Objective and structure of this study

To address this research gap, this study proposes a framework aimed at enhancing natural interactions with construction robots consisting of three modules: 1) Natural Language Understanding (NLU): to extract task-specific information through a language model, 2) Information Mapping (IM): to employ conditional statements to deal with discrepancies between NLU outputs and building component information, and 3) Robot Control (RC): to execute action plans using a virtual construction robot. The framework supports pick-and-place construction operations through natural language instructions.

Table 1 shows the main characteristics of this study. Diverse interaction channels have been considered for interaction with construction robots, but no prior research has directly investigated how to collaborate with the robots using natural language instructions in pick-and-place construction operations. While other language instructions used in the previous studies describe target objects and destination, pick-and-place operations for construction activities require one more piece of information about placement orientation. To address this issue, a deep learning-based language model is trained and tested on language instructions data for construction tasks. To describe target objects and destination in natural language instructions, building component information and working records available from the construction project information are used. The target objects and destination are described using their IDs, dimension, position or working records. To demonstrate and evaluate the proposed approach, a set of experiments on drywall installation is conducted as a case study.

Table 1
Characteristics of this study.

#	Characteristics
1	Communication using natural language instructions
2	Pick-and-place construction operations: target, destination, and placement orientation
3	Use of the building component information (e.g., designs, materials) and working records
4	Natural language instruction data for drywall installation
5	Object description: ID, dimension, position, and previous working records
6	Demonstration of drywall installation

2. Literature review

Through the review of existing works, the need for this study and research gaps are identified. The first section establishes the need for analyzing natural language instructions for HRC in the construction domain. The second section examines the characteristics of data and approach used in other domains in relation to natural language understanding. The third section investigates studies that performed information extraction in the construction industry.

2.1. Interaction between human workers and robots in the construction industry

Advanced interaction methods for HRC enable human workers to collaborate with robots easily and naturally. In construction, research using gestures, VR, brain signals, and speech has been proposed for interaction with robots. Gesture-based interaction using operators' body movements can enhance the intuitiveness of communication [32] and can be used in noisy environments encountered on construction sites [33]. In 2021, Wang and Zhu [33] proposed a vision-based framework for interpreting nine hand gestures to control construction machines. Sensor-based wearable glove systems were proposed to recognize hand gestures for driving hydraulic machines [18] and loaders [34]. However, when using hand gestures, the operators' hands are not free, and they have to keep pointing to the endpoint, which may lead to fatigue [35].

VR interfaces have been used in the construction industry for visual simulation, building reconnaissance, worker training, safety management system, labor management and other applications (e.g., [36–39]). It can also provide an opportunity for users to control robots without safety risks [40]. Regarding interaction with robots, Zhou et al. [41] and Wang et al. [14] tested VR as an intuitive user interface exploring the virtual scene for pipe operation and drywall installation, respectively. Both studies sent commands to robots by handheld controllers, which determined desired poses and actions of robots.

In addition to the purpose of operating robots, in 2022, Adami et al. [19] investigated the impacts of VR-based training for remotely operating construction robots. In the interaction with a demolition robot, operators used the robot's controller consisting of buttons and joysticks based on digital codes. However, head mounted devices as visual displays may be uncomfortable for operators due to onset of eye strain and hand-held devices may limit the operators in their actions [42,43]. In addition, the connection between the headset and the controllers can be interrupted, and the working space is limited due to cables attached to the computer [44].

Recently, brain-control methods have been proposed for HRC in construction, translating the signals into a set of commands for robots. To control robots, users can attempt to convey their intention in a direct and natural way by manipulating their brain activities [45]. In construction, in 2021, Liu et al. [20] and Liu et al. [46] proposed systems for brain-computer interfaces to allow human workers to implement hands-free control of robots. Users' brainwaves were captured from an electroencephalogram (EEG) and interpreted into three directional

commands (left, right, and stop) [20]. In the other study [46], brainwaves were classified into three levels of cognitive load (low, medium, and high), and the results were used for robotic adjustment. This communication using brain signals enables physiologically-based HRC by evaluating workers' mental states [45]. However, systems using brain signals have to overcome challenges of time consumption for user training, non-stationarity of signals affected by the mental status of users, and user discomfort from the wearable equipment [47]. It is also challenging for users to deliver high-dimensional commands to collaborative robots because of the limited number of classifiable mental states [45].

On the other hand, speech is the most natural way of communication in humans, even if the objects of their communication are not other humans but machines or computers [22,24]. Natural language can be a flexible and familiar medium for construction workers to communicate with robots, and can be leveraged for hands-free and eyes-free interaction with low-level training [48]. Enabling robots to understand natural language commands also facilitates flexible communication in human-robot teams [49]. Despite the advantages of the speech channel and natural language in interaction, there are few studies examining natural language instructions for human-robot collaboration in construction. In 2018, Follini et al. [21] proposed a robotic gripper system integrated with voice identification/authentication for automated scaffolding assembly, but it was based on a very limited number of simple voice commands like *stop, grip*, and *release*. In the construction industry, speech and natural language-based HRC could be further investigated due to the potential benefits discussed above.

2.2. Natural language instructions for non-construction HRC

Many studies in which humans give instructions to robots using natural language commands have been conducted for manipulation tasks, focusing on the identification of target and destination. Regarding the placing task, Paul et al. [28] and Bisk et al. [29] leveraged spatial relations in natural language instructions to allow robots to move blocks on the table. Paul et al. [28] proposed a probabilistic model that incorporates notions of cardinality and ordinality as well as abstract spatial concepts. A neural architecture, consisting of encoder, representation stages, and grounding to predict three task elements, was suggested for interpreting unrestricted natural language commands in moving blocks identified by a number or symbol [29]. In 2020, Mees et al. [50] developed a network to estimate pixelwise placing probability distributions used to find the best placement locations for household objects. However, in order to make a robot perform various construction tasks, it is necessary to use different kinds of attributes (e.g., dimension, material, and ID) describing objects as well as spatial information (e.g., vertical and horizontal arrangement) of the objects.

Several multimodal studies have mapped visual attributes and language information by using two types of input (an image and an instruction). In 2018, Hatori et al. [30] integrated deep learning-based object detection with LSTM-based language model to deal with attributes of household items, such as color, texture, and size. In 2019, Magassouba et al. [31] proposed a deep neural sequence model including Bi-LSTM-based model to process language instructions. The model aimed to predict a target-source pair in the scene from an instruction sentence for domestic robots. In 2021, Ishikawa and Sugiura [51] proposed a transformer-based model [52] including text embedder and multi-layer transformer to model the relationship between everyday objects for object-fetching instructions. In 2023, Guo et al. [53] developed an audio-visual fusion framework for robot placing tasks, employing a bi-GRU encoder with a hierarchical attention module [54] to extract text features. A combination of linguistic knowledge with visual information can describe targets in many ways. To utilize these methods for assembly tasks at unstructured and complex construction sites, there is a need for vast collections of image-text pairs as previous studies [30,51]. However, limited datasets of image-text pairs in the context of construction sites present challenges in applying previous multimodal studies to HRC in construction.

Some methods interpreted natural language instructions given to robots without relying on visual information. Language understanding using background knowledge [55] and commonsense reasoning [56] have been explored to infer missing information from incomplete instructions for kitchen tasks. In 2018, Nyga et al. [55] generated plans for a high-level task in partially-complete workspaces through a probabilistic model to fill the planning gaps with semantic features. In 2020, Chen et al. [56] utilized an RNN-based model to formalize commonsense reasoning as outputting the most proper complete verb-frame by computing scores of candidate verb frames. However, unlike kitchen tasks, it can be challenging to infer targets in construction activities using general knowledge or pre-defined verb frames. In 2018, Brawer et al. [57] proposed a logistic regression model that estimates the action probability to select one target among 20 candidates by contextual information such as the presence of objects and the action history. The context information can also be leveraged in HRC for construction activities, but the proposed model is limited to analyzing language instructions for the pick-up action.

2.3. Natural language processing in the construction industry

Natural language processing (NLP) is a research domain exploring computer-assisted analytical technique to automatically interpret and manipulate natural language [58]. With the advance of machine learning and deep learning, NLP has been increasingly adopted in the construction industry. NLP applications in construction have been explored in many areas, such as knowledge extraction, question-answering system, factor analysis, and checking [59]. Various documents, such as accident cases [60,61], injury reports [62], compliance checking-related documents [63], legal texts [64], and construction contracts [65] have been analyzed in construction. Analysis on natural language instructions for HRC has not been explored in the construction industry.

Collaboration with a construction robot using natural language instructions requires extracting useful information from the instructions so the robot can start working. Previous studies extracted keywords based on frequency features [66] and handcrafted rules [67]. These approaches are not robust to unfamiliar input which includes misspelled or unseen words rather than the keywords. To address these challenges, machine learning and deep learning models have been used to extract information about infrastructure disruptions [68] and project constraints [69,70]. However, entities used in these studies, such as task/procedures [70], interval times [69], and organization [68] are not suitable for identifying important information from natural language instructions for construction activities. A new group of entities should be defined to give essential information to construction robots. For example, entities for pick-and-place tasks are relevant to characteristics of the tasks such as target objects, placement location, and placement orientation.

Several studies have used natural language queries to change or retrieve Building Information Modeling (BIM) data [71–73]. In 2016, Liu et al. [71] retrieved wanted BIM information by mapping extracted keywords from queries and IFC entities. However, the proposed method supported only simple queries such as “quantity of beams on the second story” or “quantity of steel columns in the check-in-zone.” In 2021, Shin and Issa [72] developed a BIM automatic speech recognition (BIMASR) framework to search and manipulate BIM data using a human voice. They conducted two case studies for a building element, a wall, but a quantitative evaluation of the framework was excluded. A question-answering system for BIM consisting of natural language understanding and natural language generation was developed [73]. The system achieved an 81.9 accuracy score with 127 queries. For example, users can obtain answers to questions like “What is the height of the second floor?”, “What is the object of door 302?”, or “What is the model

creation date?". These studies have analyzed text inputs to retrieve useful project information from language queries. However, the text inputs do not address construction-specific information that is requisite for HRC commands. Additionally, it is important to note that their proposed methods do not primarily aim to interact with robots for construction tasks.

In recent research developments (2023), two studies employed ChatGPT, a large language model, to develop an interactive virtual AI assistant for construction tasks. Xu et al. [74] introduced a system combining AR, Optical Character Recognition (OCR), and the GPT language model to optimize user performance in operations and maintenance tasks. Notably, their system relies on language instructions that are set at the beginning of tasks to fine-tune the GPT model, which limits the scope for ongoing interaction during the tasks. Moreover, their framework does not specifically cater to interactions with construction robots. Ye et al. [75] investigated the influence of ChatGPT in fostering trust in HRC assembly tasks. In the study, the robot is programmed to assist human operators by fetching tools or objects, following simple language commands such as "get closer to me" and "give me the screw." However, this approach is limited as it only involves identification of objects or tools by name, lacking the integration of more complex descriptors such as size, location, object IDs, or historical data of past interactions. These observations highlight that while existing research has made strides in integrating natural language processing with robotics, there remains a significant opportunity for advancement in applying this technology to the specific needs and complexities of construction environments.

There has been no research to plan robot tasks based on natural language commands which require interpretation of information from both language commands, BIM, and working history.

2.4. Robot control commands

The interpretation of natural language instructions is conducted entirely independently, and prior to, aspects of robot control [76]. To facilitate this, semantic information from human instructions must be decoded into structured commands that a robot can comprehend and execute. For example, the directive to "take the cable from the floor" or to "start painting wall A in room 123" requires a translation into a semantically structured input for the robot. This translation is important for bridging the communication gap between human language and robotic actions, ensuring that the robot performs tasks as intended by the operator.

The translation of natural commands into robot actions can take various forms. One direct method, as demonstrated by Ralph et al. [77], involves mapping natural language instructions to individual robot motions—such as pairing the command "Move Up" with the action "translate along +Z world axis," or "Tilt Down" with "pitch down tool frame." This approach creates a direct link between human commands and robot movements. On the other hand, a more structured approach incorporates an action verb and relevant contextual information into the command. This method, used by Matuszek et al. [78] for robot navigation, involves commands like (*move-to forward-loc*) which combine a directive with a spatial reference. Similarly, She and Chai [79] explored grounded verb semantics in HRI, employing expressions that vary in complexity based on the action's requirements, like (*Grasp(Kettle1)*) or (*Keep(Kettle1, on Stovefire4)*). Moreover, Chen et al. [56] addressed the challenge of interpreting incomplete instructions by using a complete verb frame, such as (*pour, water, bowl*), which details the action, object, and destination. This diversity in approaches showcases the adaptability of robotic systems to various levels of command detail, depending on the robot's capabilities and the complexity of the task at hand.

3. System architecture

The proposed system aims to make a robot assistant perform construction activities after receiving verbal (natural language) instructions from a human partner. Specifically, the construction activities targeted in this study are pick-and-place construction operations. Essentially, the system is designed exclusively to manage the actions involving the picking up and placing of materials. Developing this system necessitates the integration of three modules. Fig. 1 shows critical components and data workflows of the system, which comprises three modules: Natural language understanding (NLU), Information Mapping (IM), and Robot Control (RC). In this system, the three modules work together to enable a human operator to interact with a construction robot.

The NLU module takes a natural language instruction as input and employs a trained language model to perform sequence labeling tasks, generating word-tag pairs. In certain contexts, the word-tag pairs can directly provide the final message to the robot, ensuring unambiguous communication. However, language instructions can often demand contextual understanding and the consideration of historical data. To address this, the IM module integrates the interpretation of building component information and action history with the output of a language model to generate executable robot control commands. Finally, the RC module utilizes three types of task information (target, final location, and placement method) to control the robot's movement for pick-and-place tasks. Within this system, detailed instructions for minor adjustments, such as 'tilt', 'fit', or 'avoid', are not necessary for the collaborative robot to complete construction tasks. This assumption is grounded in the robot's own cognitive capabilities to address minor geometric deviation and workspace uncertainties, demonstrating its adeptness in detecting geometric discrepancies between as-designed and as-built work, as shown in Lundeen et al. [80]. Although the application of the robot's capabilities is out of the scope of the current study, they hold potential for future integration into the RC module of the proposed system to address discrepancies between the robot's expected information and actual conditions, thus improving a practical implementation of on-site construction robots.

3.1. Dataset generation and labeling

In the proposed system, two pieces of information source are needed for a robot to execute tasks: one from BIM and the other from natural language instructions. First, it is assumed that BIM encompasses details about construction materials at construction sites. Specifically, BIM contains the ID, dimension, and position of a workpiece, which are essential data for pick-and-place construction operations. In this regard, it is assumed that users have access to mobile devices (e.g., tablet) to obtain building component information such as a name, a unique ID, a dimension, and an initial position of each workpiece on a future construction site.

Given the potential use of mobile or wearable technologies in the construction industry [14,81,82], such technologies could be used to provide project information to construction workers making it easier to unambiguously specify which workpieces are to be installed and corresponding location to the robot assistants. As a result, natural language instructions will specify targets and destinations based on their ID, dimension, or position. Second, natural language instruction serves as the medium through which human operators convey task-specific information for pick-and-place construction tasks to robots.

In data generation, a single natural language instruction for pick-and-place construction operations consists of one or multiple sentences. There are three rules to generate natural language instruction dataset in this study. First, each instruction should contain attributes of three key pieces of information, which are a target, a final location, and how to place the target, exactly once. For example, it is unacceptable to mention two targets in one instruction or solely reference two out of the three key information components. Second, expressions clearly indicating features related to these three types of information should appear only once in each instruction. Human teammates are expected to

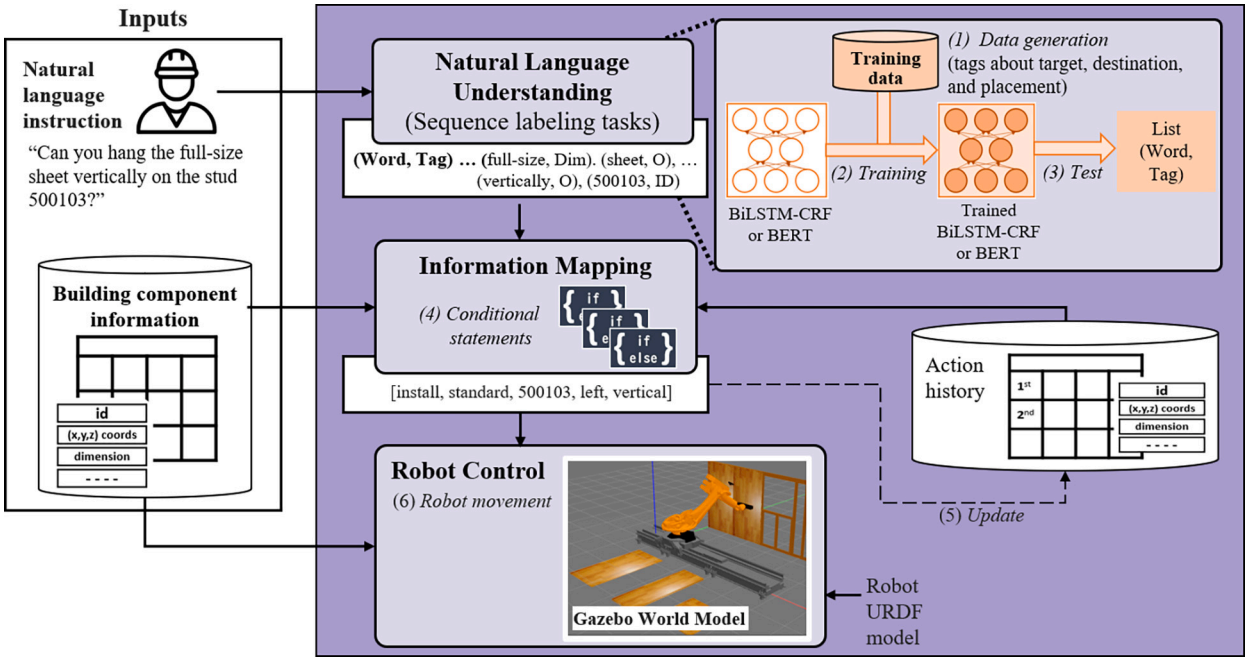


Fig. 1. The proposed system using natural language instructions for HRC in construction.

articulate each piece of task-specific information singularly. For instance, it is not acceptable to describe a target using both its ID and dimensions within a single natural language instruction. Lastly, a fine-grained annotation is employed to assign corresponding labels to attributes of the three types of information. In the annotation process, terms indicating targets are not labeled as 'target'; instead, individual attributes are annotated with precise information as '*ID*' or '*length*'.

Within the dataset, co-reference issues might arise. This is when words referring to a target object, a final location, and a placement method can be included multiple times within a single instruction. For example, in an instruction "Please pick up the object A. Move it on to the object B", words 'the object A' and 'it' denote the same object. Relying on the second and third rules, only 'A', which indicates a feature of the target object, will be annotated as 'ID' during the labeling. The second and third rules facilitate the identification of unique workpiece characteristics to resolve the co-reference issues.

In data labeling, IDs in language instructions can be tagged with a label such as '*ID*'. BIM models used in previous studies have allocated a five to seven-digit number to every building element [83–85]. A list of digits can be read out in the working environments such as warehouses or factories to increase work performance [86–88]. While it may not be common to utter long digits in today's construction workers' practice, this study suggests that using IDs could be one of the effective ways for workers to unambiguously indicate a target object or a final location when interacting with robots to ensure accurate selection and installation of workpieces, particularly in BIM-driven construction workflows.

Workpiece dimensions in language instructions can be labeled with labels like '*length*', '*width*', or '*dimension*'. For example, when a target object is described in numbers such as "4 by 8 feet", "12 by 12 feet", or "its length is 12", the numeric values are annotated as '*length*' or '*width*'. Within the construction industry, there are workpieces conforming to established standard sizes widely prevalent in the industry. When describing the dimensions of workpieces using terms like "full-size" or "standard", the words representing the size of the workpieces are annotated as the label '*dimension.*' Both the target object and the final location can also be labeled based on their locations. Instead of specifying precise coordinates to describe the placement of workpieces, expressions such as 'left' 'right' or 'second to the left' are employed with labels '*Loc*' in the process of data labeling.

Finally, regarding how to place target objects in tasks, we consider both vertical and horizontal placement. When a target object is positioned either vertically or horizontally, the corresponding terms can be annotated as the labels '*Vr*' or '*Hr.*' Diverse situations can be explored, including situations where a target is placed to the upper, to the bottom, to the left, or to the right side of the final location.

The selection of tags for the system was designed to accurately represent the key attributes of construction materials such as ID, size, and location, as well as the placement method for tasks. This deliberate selection of tags is critical in enhancing the effectiveness of task execution, particularly when operators rely solely on voice commands. This approach is supplemented by detailed information about construction materials. The inspiration for tag selection stems from previous studies in the fields of BIM integration with construction robotics [89,90]. In these studies, unique identifiers, dimensions, positions or main axis of building elements have been used as inputs for robot control systems.

3.2. Natural Language Understanding (NLU)

A NLU module aims to predict semantic information from the user's input which is in natural language. Two main tasks of the NLU are intent classification (IC) predicting the user intent and slot filling extracting relevant slots [91]. The NLU module of this study focuses on the slot filling which can be framed as a sequence labeling task to extract semantic constituents. It extracts semantic information for target, destination, and placement orientation based on characteristics of construction materials that were previously unexplored in prior research.

Fig. 2 shows an example of the slot filling for the user command "Install the object A on the object B" on a word-level. The word 'tag' is

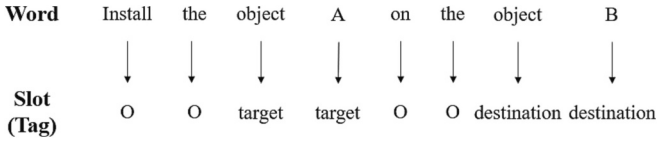


Fig. 2. An example of an instruction labeling for slot filling.

used to refer to the semantic label. The objective of the slot filling task is to produce word-tag pairs as its output. In this study, two language models, which are the typical deep learning architectures for this task, are tested to evaluate their capability in assigning the correct tags to each word in a user command. This evaluation seeks to determine which model offers the most effective and accurate results in the proposed system. The first architecture is the Bidirectional Long Short-Term Memory (BiLSTM) layer [92] with a Conditional Random Fields (CRF) layer [93]. The second architecture is based on the Bidirectional Encoder Representations from Transformers (BERT) architecture [94].

BiLSTM-CRF is a neural network model that has been used for sequence labeling [95–97]. BiLSTM incorporates a forward LSTM layer and a backward LSTM layer in order to leverage the information from both past and future observations of the sequence. A hidden forward layer is computed based on the previous hidden state ($\overrightarrow{h}_{t-1}$) and the input at the current position while a hidden backward layer is computed based on the future hidden state ($\overleftarrow{h}_{t+1}$) and the input at the current position as shown in Fig. 3. At each position t, the hidden states of the forward LSTM ($\overrightarrow{h}_t$) and backward LSTM ($\overleftarrow{h}_t$) are concatenated as input to the CRF layer. The CRF layer generates the sequence labeling results by adding some effective constraints between tags. Each tag score output by the BiLSTM is passed into the CRF layer, and the most reasonable sequence path is determined according to the probability distribution matrix. The BiLSTM-CRF model consists of the BiLSTM layer and the CRF layer, which can process contextual information and consider the dependency relationship between adjacent tags, resulting in higher recognition performance in comparison to a single CRF model with an identical set of features [95].

BERT, Bidirectional Encoder Representations from Transformers, is a bidirectional language model that achieves outstanding performance on various NLP tasks including sequence labeling [94]. The architecture of BERT is a multilayer transformer structure which is based on the attention mechanism developed by Vaswani et al. [52] in 2017. BERT is trained to predict words from its left and right contexts using Masked Language Modeling (MLM) [94] to mask the words to be predicted. The general idea of BERT is to pre-train the model with large-scale dataset, and parameters of the model can be updated for the given tasks during fine-tuning.

In this study, pre-trained BERT-base model [94] is fine-tuned for sentence tagging tasks. As shown in Fig. 4, the input text is tokenized and special token like [CLS], which stands for classification, is added at the beginning. It is needed to create an attention mask. The input for BERT is the masked sequence and the sum of the token and position embeddings (E_i). Then, the final hidden vector is denoted as T, which is the contextual representation for each token. The token-level classifier is a linear layer using the last state of the sequence as input. In this study, when a word is composed of several tokens and the prediction results of the tokens are different, the class of the word is determined by the token corresponding to more than half of the tokens.

3.3. Information Mapping (IM)

The information mapping module aims to generate a final command for the robotic system using output of the NLU module, building component information, and action history. This module is necessary in the proposed system since the results of the NLU module (word-tag pairs) cannot be directly used as inputs for the robot control. This module is designed to extract three necessary types of information crucial for a successful pick-and-place construction operation, including the identification of a target object, its destination, and placement orientation.

In the IM module, NLU outputs, building component information, and action history are mapped by using conditional statements, and the mapping result is recorded in the action history (Fig. 5). Conditional statements play a role to find out essential information for tasks by dealing with vocabulary discrepancies between words of NLU outputs and building component information. The action history record includes information about the previously installed object, including its IDs, dimension, where it is placed, and how it is placed. The previous action record can be used as one of the inputs for the conditional statements to find out a target object and its final location for the current action. The final command to be delivered to the RC module is determined based on the mapping result.

To address inconsistencies in the vocabularies between the NLU output, building component information, and action history, the module incorporates a procedure that uses conditional statements to extract information about the target object, destination, and placement method. These conditional statements are designed to utilize the ID, position, and dimension information of each component, which can be obtained from the building component information.

The appropriate conditional statement to use is determined based on the tag of each word in the NLU output. For instance, if the NLU output contains a tag '*ID_target*' that refers to the target object's ID, the corresponding word is mapped to the ID in the building component information. The component information associated with that ID is then added to the action history as the target object's information. Similarly, if the NLU output contains a tag '*Position_target*' that refers to the

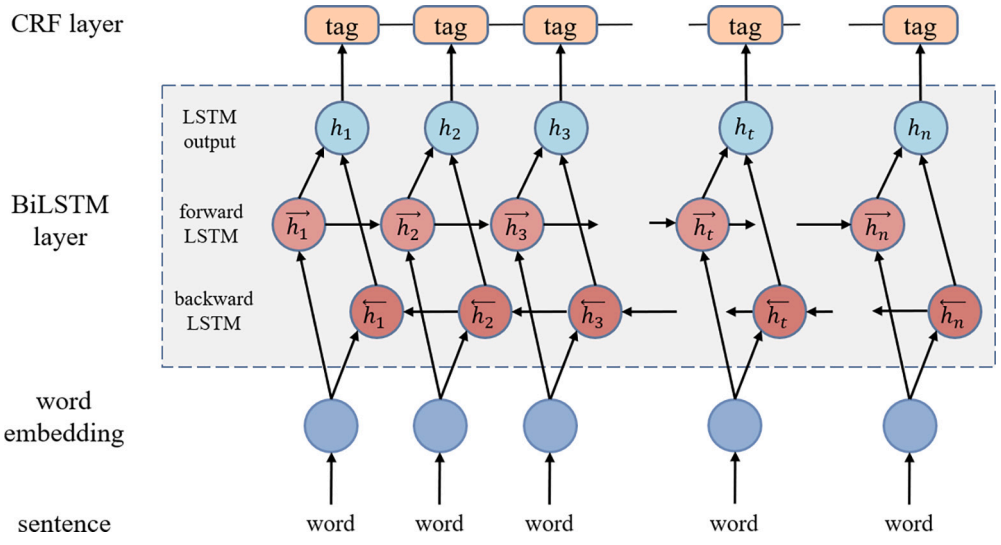


Fig. 3. A BiLSTM-CRF structure.

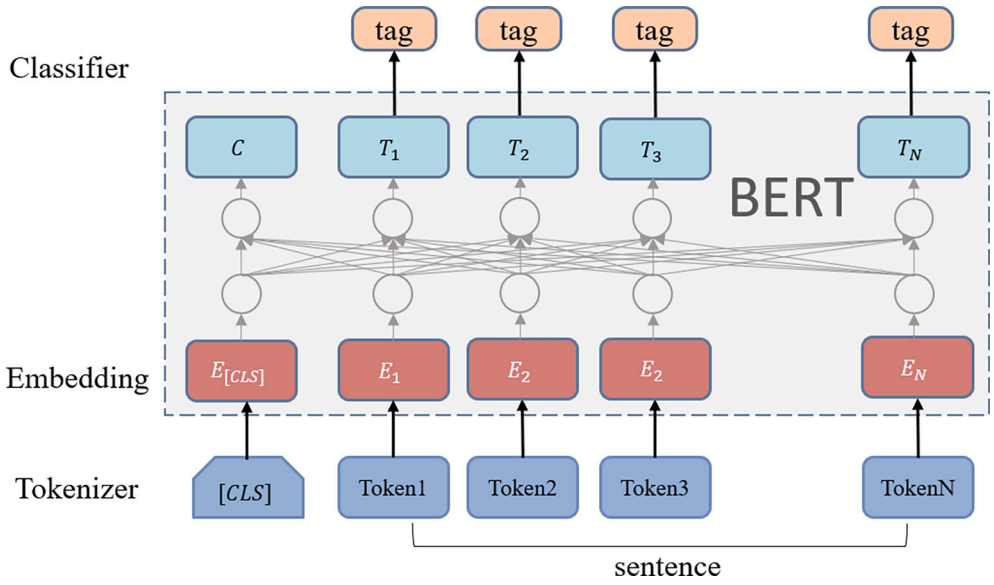


Fig. 4. BERT for sentence tagging tasks.

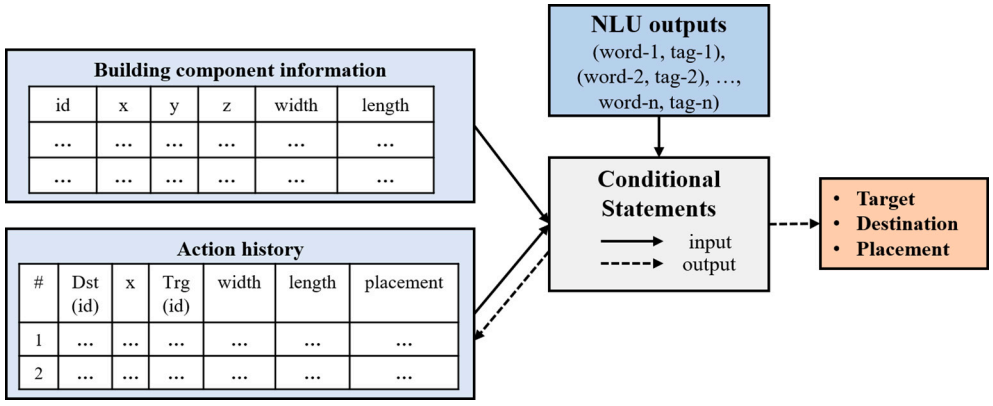


Fig. 5. Inputs and outputs of the IM module.

position of the target object, the corresponding word in a language instruction is mapped to a component in the building component information within the conditional statement processing the position information. Then, all the information associated with that component is then added to the action history as the latest record.

When the vocabularies in the NLU output, representing the target object, destination, and placement method, are accurately mapped to their respective items in the building component information, the IM module's execution is regarded as successful. The performance is closely linked to the output generated by the Natural Language Understanding (NLU) module, as the latter's output serves as the input for the former. This interdependence implies that the accuracy of the IM module depends on the performance of the NLU module. If there are inaccuracies or misinterpretations in the results predicted by the NLU module, it can lead to errors in the conditional statements of the IM module, hence influencing its operational integrity. This relationship underscores the importance of precision of the first component in the system, highlighting the interplay of accuracy across modules.

Once the action history is updated, the final command for robot control is determined as the target object type, destination ID, and placement methods from the action and transferred to the Robotics Control (RC) module.

3.4. Robot Control (RC)

This study uses a virtual robot digital twin to plan and execute actions following natural language instructions and building component information processed by the previous modules. Fig. 6 shows the process flow for pick-and-place operations implemented in the RC module. The initial step is to calculate the precise coordinates for the target and destination, as depicted in the figure. This calculation is critical in bridging the gap between abstract instructions and actionable data for task execution. This process utilizes the geometric points and dimension information of the objects, which is derived from the building component information. A robot in this study is simulated using Robot Operating System (ROS) and Gazebo that is the virtual environment offered by the Open-Source Robotics Foundation. The robot is a 6 degrees-of-freedom KUKA robotic arm, whose movements are informed by a previous study described in Wang et al. [14].

The robot's movements are executed through a sequence of phases. The robot establishes a pose target and devises a motion plan. The robot arm finds a motion from its original base location at first. Should the initial plan prove unfeasible, the robot's base position is adjusted accordingly (Pre-Pick). Once a valid path of the robot's base is determined, a motion plan for the movement of the robotic arm is generated. This plan ensures that a robot's end-effector aligns precisely with the center of the object. In this phase, the orientation of the end-effector is

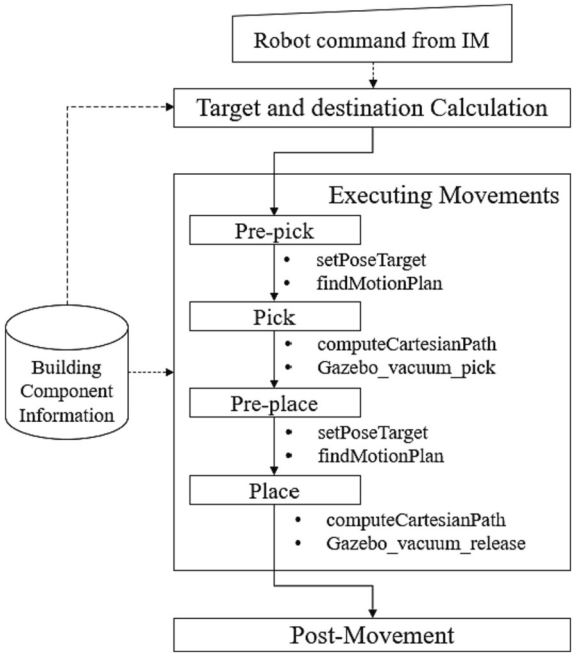


Fig. 6. Process flow for pick-and-place operations.

not adjusted according to the target object's arrangement. Next, a Cartesian path is computed for the robot's end-effector to secure the target object with a gripper (Pick). Then, the robot follows the computed path to move to the target object. Next, reflecting the pre-pick stage, calculated destination and placement method are used to adjust the pose target and motion plan (Pre-Place). The orientation of the end-effector is adjusted for the placement method, with the specific rotation of the sixth link being dictated by whether the placement is vertical or horizontal. Next, the robot follows the determined Cartesian path to place the object at the designated location and releases it (Place). After the placement, the robot arm reverts to its pre-placement stance (Post-Movement).

Throughout these stages, the robotic arm's movement, which is generated by MoveIt [98], has higher priority than the base movement to reduce localization error. This means that the robot's base is only repositioned if the robotic arm fails to devise a feasible motion plan for picking or placing an object. The Open Motion Planning Library (OMPL) [99] and Flexible Collision Library [100] are employed to compute kinematics of each joint in planning movements, ensuring collision-free trajectories. When the robot is carrying a target object, collision checking process is applied while the target is considered as part of the robot, so that the robot and the target object will not collide with their surroundings. Upon successfully completing the installation, a human operator can give the next instructions after target placement is completed.

4. Experimental validation

4.1. Installation of drywall panels

Fig. 7(a) shows a robot operation environment for drywall installation. A KUKA robot is positioned between a stud wall and drywall panels and the base of the robot can move in a straight line as shown in Fig. 7 (a). The stud wall consists of thirteen vertical studs as illustrated in Fig. 7 (b). In this case study, one stud is designated as the final location for place operation and the left edge of a drywall panel is laid on the stud. In general, drywall panels are available in rectangular shapes. Standard panel size is 4 ft wide and 8 ft long and panels of different sizes are cut according to the designed dimensions in construction practice. We use three sizes of panels including the standard ones as well as two unique panel sizes (Fig. 7(c)). The position and dimension information of the building components used in the experiment are shown in Fig. 8.

The drywall panels can be installed in a vertical or horizontal orientation. Fig. 9 shows examples of how to place drywall panels onto the studs. Examples of vertical placement are shown in Fig. 9(a), and the left edge of the panel can be placed on the center line of a stud or the left side of a stud. When the panels are placed horizontally perpendicular to studs, they can be placed on the top or bottom part of the studs as shown in Fig. 9(b). Therefore, natural language instructions for drywall placement should include how (i.e., in what configuration) to place the drywall panels.

4.2. Data generation and labeling

A new dataset of natural language instructions for drywall installation was created and annotated. This study utilized 12 tags that enabled the classification of these three essential categories into more detailed categories as shown in Fig. 10. These tags include six that describe the characteristics of the target object, three that illustrate the final location, and the remaining three for the placement orientation. Each instruction contains these three pieces of information exactly once. To utilize widely used expressions for drywall installation tasks and pick-and-place related language instructions, construction videos about drywall installation 'How To Install Drywall A to Z | DIY Tutorial '(https://www.youtube.com/watch?v=VQIMaR7hWtM) [101] and other studies [28,30] exploring pick-and-place language instructions were considered when generating the new dataset. In these language instructions, drywalls and studs are described by combinations of representations related to ID, dimensions, and relative location.

A drywall panel is represented by its ID, dimension, or position, while a stud is represented by its ID or position (Figs. 8 and 10). Each element ID is represented as a unique 6-digit number in this case study and is tagged with *ID_stud* and *ID_wall* for stud and a drywall panel, respectively. The dimensions of the target drywalls are labeled with *length*, *width*, or *dim*. In this study, we considered three distinct panel size: 4 by 4 ft, 2.7 by 8 ft, and 4 by 8 ft. Notably, 4 by 8 ft panels are

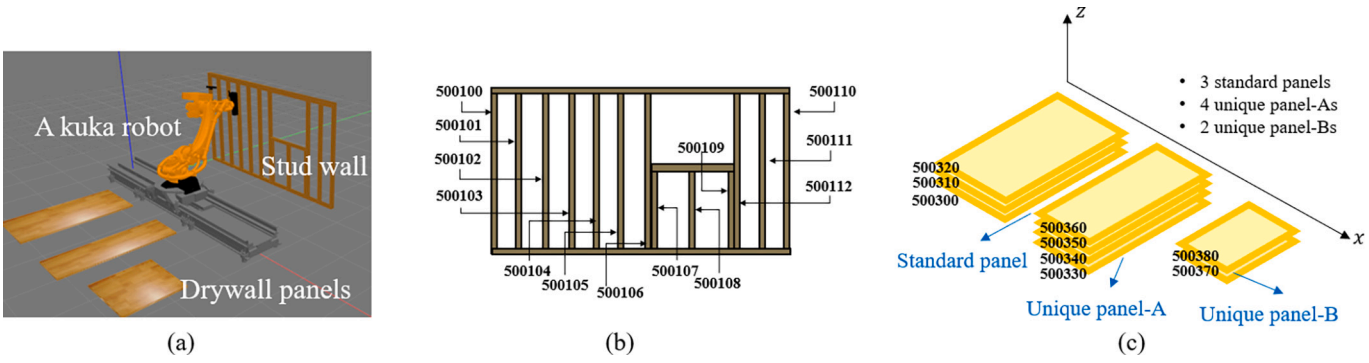
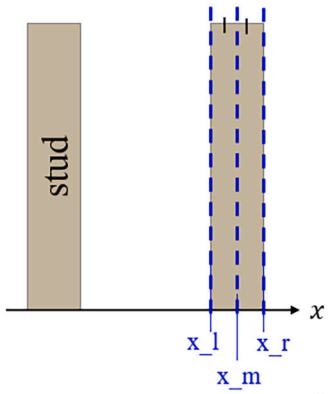
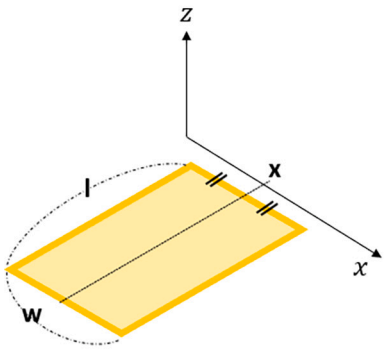


Fig. 7. Case study settings for drywall installation: (a) robot operation environment; (b) a stud wall consisting of 13 studs; (c) 9 drywall panels on the floor.



stud_id	x_l (m)	x_m (m)	x_r (m)
500100	0	0.0254	0.0508
500101	0.381	0.4064	0.4318
500102	0.7874	0.8128	0.8382
500103	1.1938	1.2192	1.2446
500104	1.6002	1.6256	1.651
500105	2.0066	2.032	2.0574
500106	2.3876	2.413	2.4384
500107	2.4384	2.4638	2.4892
500108	3.0226	3.048	3.0734
500109	3.6068	3.6322	3.6576
500110	3.6576	3.683	3.7084
500111	4.0386	4.064	4.0894
500112	4.4196	4.445	4.4704

(a)



dr_id	w (m)	l (m)	w_v (ft)	l_v (ft)	x (m)
500300	1.2192	2.4384	4	8	0.59936
500310	1.2192	2.4384	4	8	0.59936
500320	1.2192	2.4384	4	8	0.59936
500330	0.8128	2.4384	2.7	8	2.59936
500340	0.8128	2.4384	2.7	8	2.59936
500350	0.8128	2.4384	2.7	8	2.59936
500360	0.8128	2.4384	2.7	8	2.59936
500370	1.2192	1.2192	4	4	4.17446
500380	1.2192	1.2192	4	4	4.17446

*w: width value in meters used in ROS
*l : length value in meters used in ROS
*w_v: width value in inches used in language instructions
*l_v: length value in inches used in language instructions
*x: x-coordinate of the panels in meters

(b)

Fig. 8. Stud and drywall information. (a) x-coordinates of the thirteen studs; (b) dimensions and x-coordinates of the nine drywall panels.

considered as the standard panel dimensions.

Both a drywall panel and a stud can be described as their locations using one perspective view in this case study. For example, stud 500,100 is the leftmost stud and drywall sheets 500,300, 500,310, and 500,320 are the leftmost ones as shown in Fig. 6. The words to indicate locations of the stud and drywall panels are labeled as *St_loc1* and *Dw_loc1*. Drawing from the work [28], which explored efficient grounding of abstract spatial concepts for robot interaction, this study incorporates instructions that use both ordinality and relational terms to describe objects. It means that the location changes based on the secondary location. When a final location of stud is described using relative location, both *St_loc1* and *St_loc2* are used together while both *Dw_loc1* and *Dw_loc2* are used together when the target drywall is described. For example, in Fig. 5, the location of the stud 500,101 can be expressed as "second left to the stud 500103" or "right to the stud 500100." In this case, the direction like "second left" or "right" is also annotated as *St_loc1* and the word "500,103" or "500,100", which is corresponding to the secondary location, is annotated as *St_loc2*.

Finally, regarding how to place drywall panels, there are three labels of *Vr_md*, *Hr_top*, and *Hr_btm*. When a panel is vertically placed on the middle line of the stud, the corresponding words like "middle line" or "center line" are labeled as *Vr_md*. When a target object is placed horizontally on the top row of a stud or on the bottom row of a stud, the corresponding words are annotated as *Hr_top* or *Hr_btm*. Terms like "upper part", "upper horizontal row", and "top part" are annotated as *Hr_top* while terms like "lower part" and "bottom row" are annotated as *Hr_btm*. Given this variability, the same words should be annotated as different tags, creating a challenge for language models to correctly interpret the intended context. When a placement method is not mentioned in a language instruction, it means that the panel is installed vertically on the left line of the stud. It is considered default in this study and the language instruction does not have a tag about this placement method.

There are a total of 13 labels, with 12 of them representing either a target drywall, a final location (stud), or a placement method, as shown in Fig. 10. The remaining label, referred to as 'O', is utilized to signify that the corresponding word is not associated with any entity. If a target, a destination, or a placement is mentioned multiple times in a single instruction, words that do not deliver any characteristics of the three information are tagged as 'O.' For example, in a three-sentences instruction "Please move the drywall board and drive it vertically in the center line of the stud. The width is 4 and the length is 8. The stud is laying on the left to the 500103", 'the drywall board' and 'it' in the first sentence refer to a target object but they do not deliver any important characteristic, so they are tagged as 'O.'

In total, 1584 natural language instructions with the 13 labels for drywall installation were generated and manually annotated. These instructions consist of 3072 sentences and a total word count of 39,841. The dataset was split into three parts: 1268 instructions for training (80%), 158 instructions for validation (10%), and 158 instructions for test (10%). Table 2 shows annotation results of the 1584 instructions. The dataset includes fine-grained details of the target objects, expressed through six tags: *Dw_loc1, Dw_loc2, ID_wall,* dim*, length,* and *width*, which account for a total of 2535 words.

Similarly, the destination details are captured using the tags *ID_stud, St_loc1*, and *St_loc2*, encompassing 4166 words. Additionally, the dataset

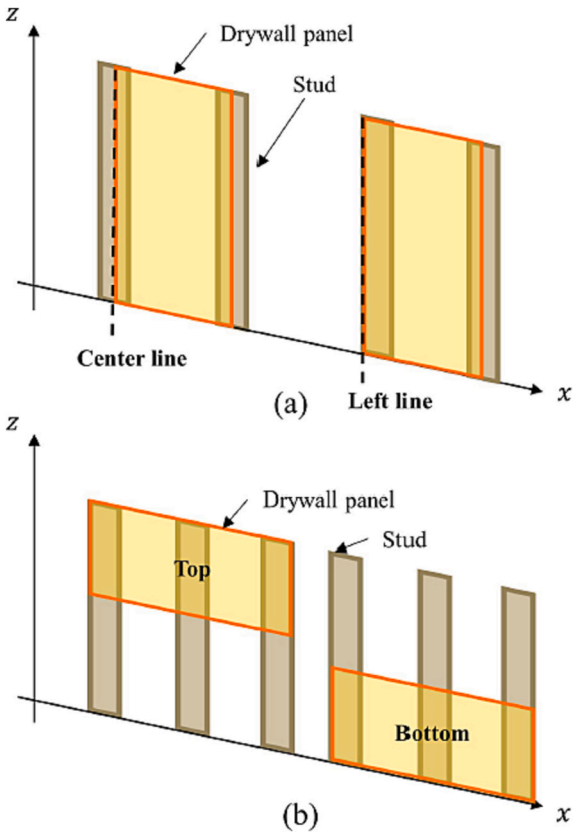


Fig. 9. Two ways of drywall installation: (a) vertical placement of drywall panels; (b) horizontal placement of drywall panels.

incorporates placement orientation information, classified into three distinct classes, and comprising a total of 2060 words. Consider the example instruction: "Can you install the piece 500310 vertically in the stud? The stud is laying third to the left from the stud 500105. Please hang the panel into the middle line." This approach allows for extraction of specific details, such as the *ID_wall* tag for the target, *Dw_loc1* and *Dw_loc2* tags for the destination, and the *Vr_md* tag representing a specific placement orientation rather than simply highlighting three main categories. Such granularity can significantly enhance the richness and precision of the data interpretation.

While the first author performed the initial manual annotation, two other individuals checked the appropriateness of annotation guidelines by annotating the test dataset in two rounds. Appendix A presents the annotation guidelines used in this study. In the first round, the two annotators labeled the dataset based on the annotation guidelines and several examples. The annotators achieved 96.05% and 89.24% accuracy, respectively. They received feedback on the results of the first-round annotation. In the second round, both annotators achieved 98.15% and 98.56% accuracy in annotation, which are almost 100% accuracy. Any errors in the second round were simple human errors. The validation set is used to compare the performance of different models in the NLU module. The model with the best performance on the validation dataset is used to evaluate the test dataset and the results are delivered to the IM.

4.3. Natural Language Understanding (NLU)

The specific parameters of the BiLSTM-CRF model used in this case study are determined based on previous studies [95,96,102] as follows: the number of neural network layers is 2; word embedding size is 50; the number of hidden layer LSTM neurons is 300; batch-size is 16; the dropout is 0.1; the optimizer is set to Adam [103] with a learning rate of 0.001; the Adam optimizer trains 20 epochs. The total number of parameters is about 250,000. In the case of BERT, "BertForTokenClassification" class was used to find-tune the BERT-base-uncased model of the original BERT [94]. The specific parameters are as follows: the number of encoder layers is 12; the number of attention-heads is 12; the number of hidden units: 768; batch-size is 16; the dropout is 0.1; the optimizer is Adam with a learning rate of 3e-5; the number of training epochs is 5. The total number of parameters is 110 million. Fig. 11 shows network architecture diagrams of BiLSTM-CRF and BERT.

This study trained the BiLSTM-CRF model and BERT by varying the number of training data to see the effects of training data size on the performance of the model. With different amounts of training data, four models with the same architecture were trained for both language models. Fig. 12(a) reports the training accuracy of the four BiLSTM-CRF

Table 2
Annotation results of the dataset.

Tags	Number of words
Dw_loc1	702
Dw_loc2	368
Hr_btm	184
Hr_top	210
ID_stud	550
ID_wall	514
O	31,080
St_loc1	2652
St_loc2	964
Vr_md	1666
dim	259
length	346
width	346
SUM	39,841

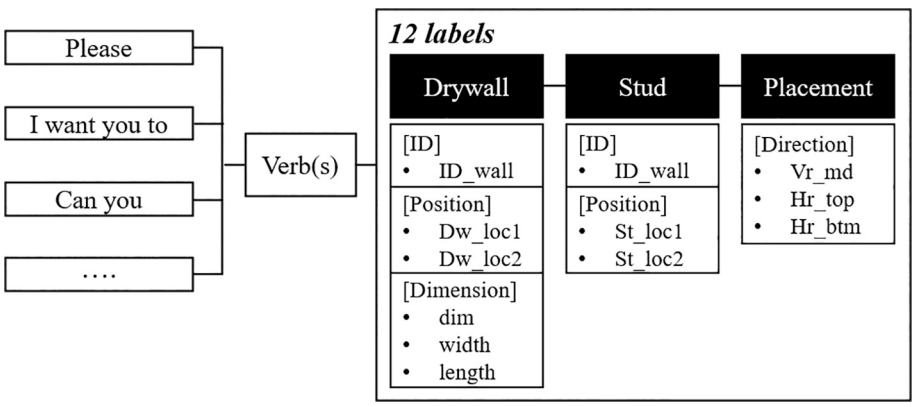


Fig. 10. Dataset generation for drywall installation.

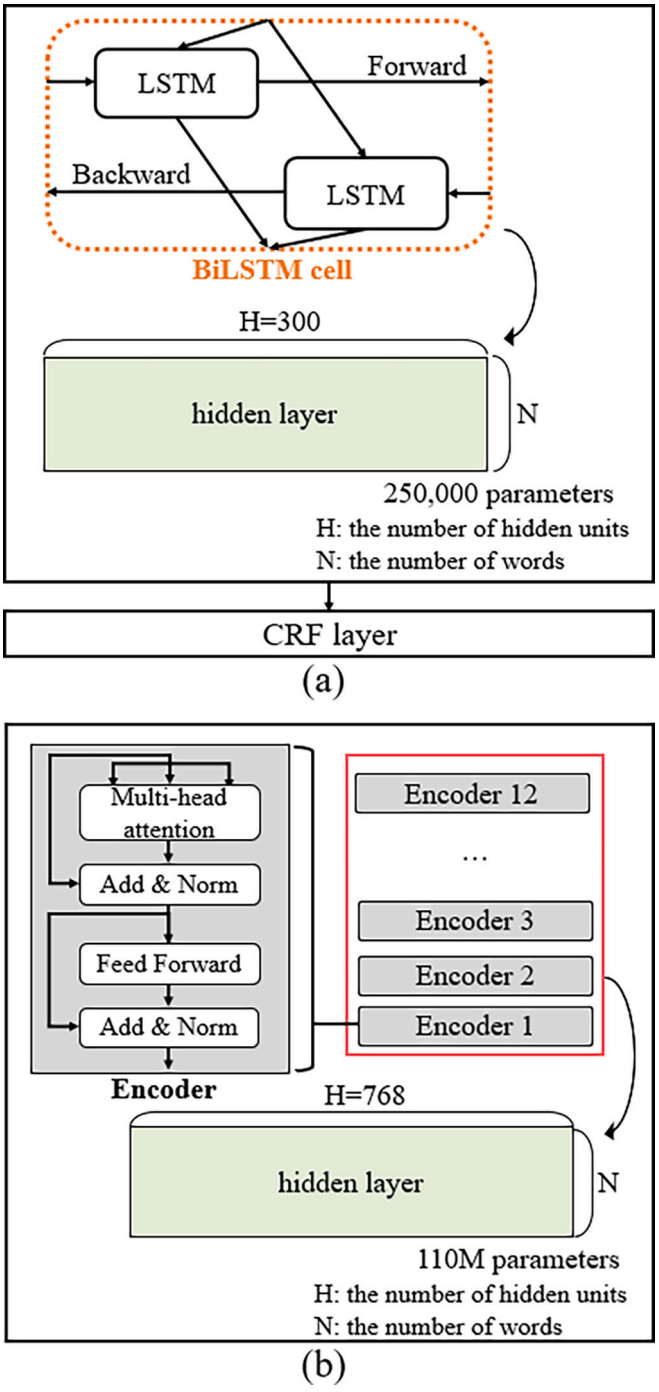


Fig. 11. Network architecture diagrams: (a) BiLSTM-CRF; (b) BERT.

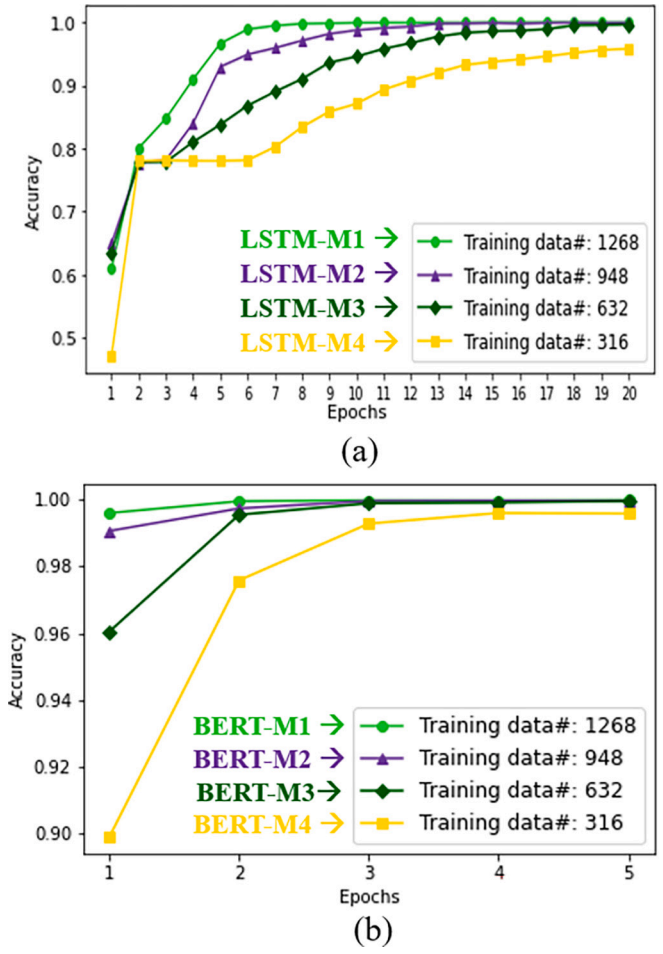


Fig. 12. Comparison of training accuracy: (a) BiLSTM-CRF and (b) BERT.

models across the 20 epochs. The four BERT models were trained across the 5 epochs since they converged quickly as shown in Fig. 12(b). The accuracy of the LSTM-M1 and BERT-M1, which were trained with ample training data, showed a considerably faster increase in the learning progress early in training.

The performance of the eight models were evaluated on the validation set and compared in Table 3. In this study, two types of accuracy are computed to measure performance. Word-level accuracy (Acc_{word}) was computed based on the number of all the words in the dataset, which provides the proportion of words that are correctly predicted.

The eight models achieved high Acc_{word} over 96%. However, even one tag incorrectly predicted in a language command can affect the IM module that derives the final robot command, causing disruptions in the robot's performance. To address this problem, Instruction-level accuracy (Acc_{inst}) considers whether all words in each instruction are correctly predicted or not, thus providing the proportion of language instructions in which all words are correctly predicted. For example, as shown in Table 3, Acc_{word} of LSTM-M4 was measured as high as 96.13%, but Acc_{inst} of LSTM-M4 showed an accuracy of 48.73%. This means that the robot can accurately perform 48% of the given language instructions.

Out of all eight models, BERT-M1 achieved the highest accuracy, with 100.00% accuracy at both the word-level and instruction-level. This accuracy, manifesting as 100% on the validation set and nearly as high during training might initially seem indicative of overfitting.

Table 3
Comparison of model performance on validation dataset.

Model	Result 1		Result 2	
	N_w	Acc_{word}	N_l	Acc_{inst}
LSTM-M1	2	99.95%	1	99.37%
LSTM-M2	2	99.95%	2	98.73%
LSTM-M3	11	99.73%	9	94.30%
LSTM-M4	144	96.13%	81	48.73%
BERT-M1	0	100.00%	0	100.00%
BERT-M2	1	99.97%	1	99.36%
BERT-M3	6	99.85%	6	96.20%
BERT-M4	43	98.90%	33	79.11%

N_w= the number of incorrect prediction of words.
N_l = the number of language instructions including incorrect prediction.
$Acc_{word}=\frac{3,895-N\_w}{3895}$, $Acc_{inst}=\frac{158-N\_l}{158.}$

However, it is noteworthy that deep learning models, as documented in previous research [104], can often attain zero training error. This phenomenon, where models effectively memorize the training set, does not necessarily compromise their ability to generalize. In the experiment, model performance generally increased with larger amounts of training data. BERT models, including BERT-M1, outperformed the BiLSTM-CRF model when trained on equivalent amounts of data. This is in conformity with previous study to test Name Entity Recognition (NER) dataset in the AEC domain [105]. Even with a small dataset (BERT-M4), the model achieved an instruction-level accuracy of 79.11%, demonstrating the effectiveness of fine-tuning pre-trained models in such cases. The study also confirmed that training with a minimal amount of data (equivalent to twice the validation set) resulted in a rapid decline in accuracy compared to the other models.

The number of false predictions for the 13 tags is compared in Table 4. LSTM-M1 and LSTM-M2 had two wrong predictions for *Dw_loc1* and *Vr_md*, respectively. As in the example in Fig. 13(a), 'most left' was incorrectly predicted as *St_loc1* representing a stud instead of *Dw_loc1* representing a drywall panel. Within our dataset, the word 'middle' is contextually labeled as Vr_md or Dw_loc1, which can occasionally increase the complexity of predictions. Fig. 13(b) shows that the word 'middle' was predicted as *Dw_loc1* instead of *Vr_md* indicating the placement method. BERT-M2 also had one error, the word 'middle' corresponding to *Dw_loc1* was predicted as *Vr_md* (Fig. 13(c)). These results may be due to the similarity of the words referring to the position and the placement method. Such issues tend to be mitigated when language models are trained with a large amount of data as shown in the previous deep learning-based studies [106,107].

LSTM-M4 and BERT-M4, which were trained with a limited amount of data, had 144 and 43 incorrect predictions, respectively. Most incorrect predictions occurred in the *Dw_loc1* category. LSTM-M4 displayed a high number of prediction errors for the *Dw_loc1, St_loc1, Vr_md,* and *width* labels. In contrast, BERT-M4 had far fewer prediction errors in these categories, which is attributed to its token-level classification approach and pre-trained BERT original version. However, unlike other models, BERT-M4 exhibited a high error rate in predicting *Hr_btm*, with all corresponding words being incorrectly predicted as *Hr_top*. This suggests that when BERT models are trained with small datasets, placement methods may be mispredicted, leading to incorrect positioning of the target panel on the stud by the robot. In the test dataset, BERT-M1, which exhibited the best performance, achieved a word-level accuracy of 99.95% with two incorrect predictions and an instruction-level accuracy of 99.37% with one error. The error occurred when the values corresponding to width and length were incorrectly predicted as length and width, respectively.

In the test using the BERT-M1 on the Google Colab platform, which offers the use of free GPU, the results showed that the average prediction time for one instruction was about 0.025 s. The 158 test data can be categorized into four groups based on the number of sentences: 46 one-sentence instructions, 74 two-sentences instructions, 27 three-sentences instructions, and 11 four-sentences instructions. The average prediction time of each group was 0.0224 s, 0.0176 s, 0.0324 s, and 0.0606 s, respectively. As the number of sentences in a single instruction increased, the analysis time tended to increase as well. In other words, time performance is better when the number of sentences is smaller. However, the absolute value was negligible across all sentence groups, showing the effectiveness of the NLU module.

4.4. Information Mapping (IM)

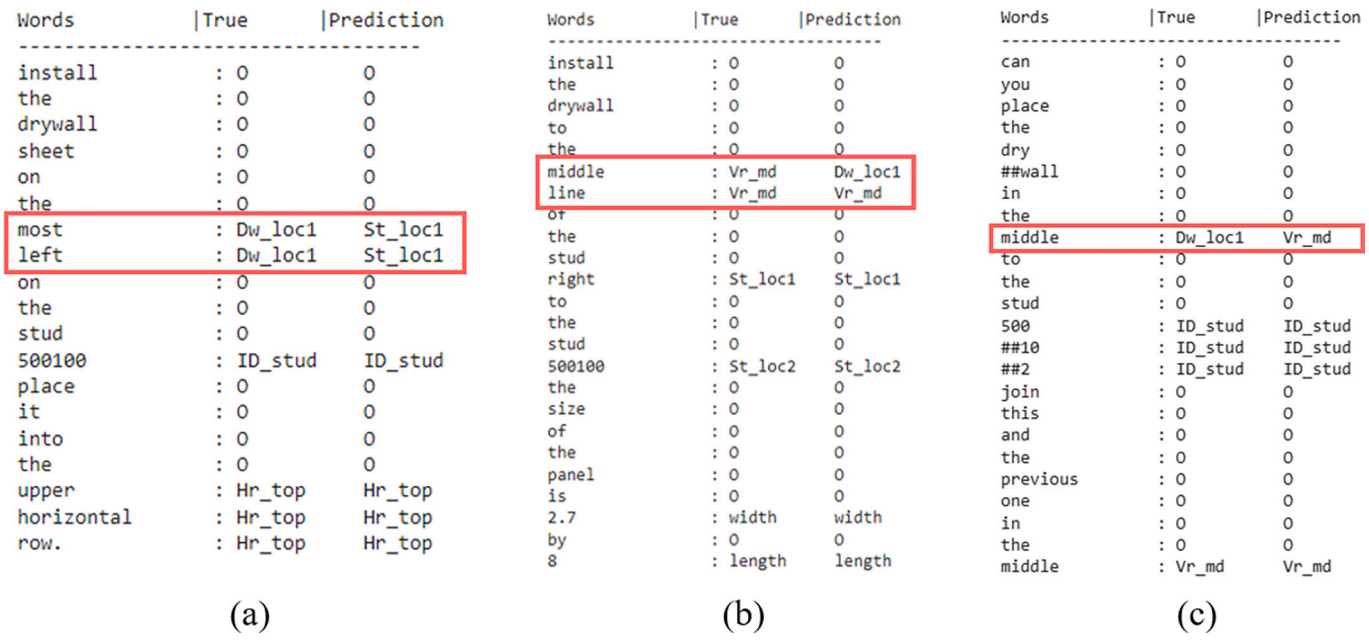
The IM module utilized several rules to extract final information about a target panel, a stud as destination, and a placement method based on the output of the NLU module and building component information (Fig. 10). The output of this module is recorded in an action history table as nine types of values: *stud_id* (ID of the stud), *installed_x_left* (x coordinate of the left side of the installed panel), *installed_x_right* (x coordinate of the right side of the installed panel), *left_cent* (if the panel is installed on the left side of the stud or the center line of the stud), *ver_hor* (if the panel is installed vertically or horizontally), *top_btm* (if the panel is installed on the top row or the bottom row), *drywall_id* (ID of the drywall panel), *w* (width of the drywall panel), and *l* (length of the drywall panel). The records in the action history table can be used to extract the final command for the robot control.

The rules of the IM module about drywall panels are shown in Figs. 14 and 15. The pseudocode in Fig. 14 can be used when a target of pick-and-place operation is described as its dimension. If the dimension of the target drywall panel is described by its length and width values or words like 'standard' and 'full-size', the target features are extracted by its length and width values in the drywall information table in Fig. 8(b), which is marked as *TableD* in Fig. 14. When an expression for a previously performed operation is used, such as "previously installed", the target of the last performed operation is retrieved from the action history table *ActHist* and the panel with the same characteristics is determined as the target of the current operation.

Fig. 15 shows pseudocode for the process used when drywall panels are labeled as their IDs or position. When the tag of *ID_wall* is included in the output of the NLU, the information of the panel corresponding to that tag is returned. If only *Dw_loc1* refers to a workpiece at the output of the NLU module, the target is determined by the x coordinate value for the initial position of drywall panels and the word tag to *Dw_loc1*. In the case that both of *Dw_loc1* and *Dw_loc2* are included in the output of NLU, a target panel is explained by its relative location that changes based on the secondary location. The x coordinate of the target panel's initial position, which is finally used to extract the target information, is determined from the secondary place and the direction tagged with *Dw_loc2* and *Dw_loc1*, respectively.

Table 4
Comparison of incorrect prediction of each class for the four models.

Tags	# of words (Ground truth)	Incorrect prediction							
		LSTM-M1	LSTM-M2	LSTM-M3	LSTM-M4	BERT-M1	BERT-M2	BERT-M3	BERT-M4
Dw_loc1	83	2	–	1	38	–	1	5	12
Dw_loc2	37	–	–	–	5	–	–	–	3
Hr_btm	16	–	–	1	–	–	–	–	11
Hr_top	26	–	–	–	–	–	–	–	–
ID_stud	56	–	–	–	3	–	–	–	–
ID_wall	45	–	–	–	3	–	–	–	–
O	3021	–	–	1	3	–	–	1	–
St_loc1	259	–	–	3	39	–	–	–	4
St_loc2	94	–	–	–	2	–	–	–	2
Vr_md	171	–	2	4	16	–	–	–	–
dim	19	–	–	–	4	–	–	–	1
length	34	–	–	1	–	–	–	–	1
width	34	–	–	–	31	–	–	–	9
TOTAL	3895	2	2	11	144	–	1	6	43



Words	True	Prediction
install	O	O
the	O	O
drywall	O	O
sheet	O	O
on	O	O
the	O	O
most	Dw_loc1	St_loc1
left	Dw_loc1	St_loc1
on	O	O
the	O	O
stud	O	O
500100	ID_stud	ID_stud
place	O	O
it	O	O
into	O	O
the	O	O
upper	Hr_top	Hr_top
horizontal	Hr_top	Hr_top
row.	Hr_top	Hr_top

(a)

Words	True	Prediction
install	O	O
the	O	O
drywall	O	O
to	O	O
the	O	O
middle	Vr_md	Dw_loc1
line	Vr_md	Vr_md
of	O	O
the	O	O
stud	O	O
right	St_loc1	St_loc1
to	O	O
the	O	O
stud	O	O
500100	St_loc2	St_loc2
the	O	O
size	O	O
of	O	O
the	O	O
panel	O	O
is	O	O
2.7	width	width
by	O	O
8	length	length

(b)

Words	True	Prediction
can	O	O
you	O	O
place	O	O
the	O	O
dry	O	O
##wall	O	O
in	O	O
the	O	O
middle	Dw_loc1	Vr_md
to	O	O
the	O	O
stud	O	O
500	ID_stud	ID_stud
##10	ID_stud	ID_stud
##2	ID_stud	ID_stud
join	O	O
this	O	O
and	O	O
the	O	O
previous	O	O
one	O	O
in	O	O
the	O	O
middle	Vr_md	Vr_md

(c)

Fig. 13. Examples of errors in: (a) LSTM-M1, (b) LSTM-M2, and (c) BERT-M2.

Definition
- Tags for drywalls: $T_{dim}, T_{length}, T_{width}$
- *Find_w* (tag): to return a word corresponding to the input tag in $[O_w, O_t]$ where O_w is a word and O_t is a tag in the [word, tag] pair.
- *Find_row* (key, value): to return n-th row for the input value in the key column of the drywall information table ***TableD.***

Input: [word, tag] pair set of NLU output $[O_w, O_t]$
Drywall information table ***TableD.***
Drywall id list ***DwIdList***
Action history table ***ActHist.***
* heads [w] and [l] of ***TableD*** and ***ActHist*** refer to width and length of the panels, respectively.

```
def DimDw([O_w, O_t ], TableD, ActHist):
  if T_dim in O_t :
    for i in range (len(O_t)):
      if O_t(i) == T_dim and O_w(i) ∈ {'standard', 'full', 'fullsize',
                                       'full-size', 'full-sized'}:
        wid_v = 4; leng_v = 8
      elif O_t(i) == T_dim and (O_w(i) ∈ {'previous', 'previously'}:
        wid_v = ActHist.iloc[-1][w]; leng_v = ActHist.iloc[-1][l];
  if T_width in O_t and T_length in O_t :
    for i in range (len(O_t)):
      wid_v = O_w(i) if O_t(i) == T_width:
      leng_v = O_w(i) if O_t(i) == T_length:

  DwInfo = Find_row (w, wid_v) ∩ Find_row (l, leng_v)
return DwInfo
```

Fig. 14. Pseudocode for information extraction about drywall panels using dimension-related tags.

Definition
- Tags for drywalls: $T_{ID\_dw}, T_{Dw\_loc1}, T_{Dw\_loc2}, T_{dim}, T_{length}, T_{width}$
- *Find_w* (tag): to return a word corresponding to the input tag in $[O_w, O_t]$ where O_w is a word and O_t is a tag in the [word, tag] pair.
- *Find_row* (key, value): to return n-th row for the input value in the key column of the drywall information table ***TableD.***
- *DimDw* ($[O_w, O_t]$, ***TableD, ActHist***): a function to extract drywall information

Input: [word, tag] pair set of NLU output $[O_w, O_t]$
Drywall information table ***TableD.***
Drywall id list ***DwIdList***
Action history table ***ActHist.***
* A head [x] of ***TableD*** and ***ActHist*** refers x-coordinate of the panels

```
if T_ID_dw in O_t :
  for i in range (len(O_t)) if O_t (i) == T_ID_dw :
    DwInfo = Find_row (id, Find_w(O_t(i))
if T_Dw_loc1 and in O_t and T_Dw_loc2 and not in O_t :
  for i in range (len(O_t)):
    if O_t(i) == T_Dw_loc1 and (O_w(i) ∈ {'leftmost', 'mostleft', 'left'}:
      DwInfo = Find_row (x, min ('x' column in TableD))
    if O_t(i) == T_Dw_loc1 and (O_w(i) ∈ {'rightmost', 'mostright', 'right'}:
      DwInfo = Find_row (x, max ('x' column in TableD))
    if O_t(i) == T_Dw_loc1 and (O_w(i) ∈ {'center', 'middle'}:
      DwInfo = Find_row (x, median ('x' column in TableD))
if T_Dw_loc1 and T_Dw_loc2 in O_t :
  SecondLoc = DimDw ([O_w, O_t], TableD, ActHist)
  for i in range (len(O_t)):
    for j in range (len(DwIdList):
      if O_t(i) == T_Dw_loc1 and O_w(i) =='left'
                             and TableD[x][j] < SecondLoc[x];
        DwInfo = Find_row (x, TableD [x][j])
return DwInfo
```

Fig. 15. Pseudocode for information extraction about drywall panels using tags of ID and positions.

Fig. 16 shows how to extract information for a stud that is a final location for pick-and-place operations. When the tag of *ID_stud* is included in the output of the NLU, the information of the stud corresponding to that tag is returned. Otherwise, the output of NLU includes *St_loc1* or *St_loc2*, so that the stud is described by its location. When *St_loc2* is not included, the stud is either the leftmost one or rightmost one. When both *St_loc1* and *St_loc2* are extracted, the stud as final location is determined by the spatial relationship described by words tagged by *St_loc1* and *St_loc2*.

To start a pick-and-place operation for drywall installation, it is essential to know the placement method as well as the target and final location. Three types of placement methods are used in this study: *Vr_md, Hr_top,* and *Hr_btm*. If the output of the NLU module does not contain these three tags, the left edge of the drywall panel is set to be placed vertically to the left of the stud. The three pieces of information about the current job are recorded in the action history table. The *installed_x_left* value in the action history table is determined according to the combination of the placement method and the final location, and the *installed_x_right* value is calculated based on the placement method, the target, and the *installed_x_left* value.

Definition

- Tags for studs: T_{ID_stud}, T_{St_loc1}, T_{St_loc2}
- *Find_row* (key, value): to return n-th row for the input value in the key column of the stud information table ***TableS***.

Input: [word, tag] pair set of NLU output $[O_w, O_t]$ where O_w is a word and O_t is a tag
Stud information table ***TableS***

```
if T_ID_stud in O_t :
    for i in range (len(O_t)):
        if O_t(i) == T_ID_stud :
            StudInfo = Find_row (O_w(i))
if T_St_loc1 and in O_t :
    for i in range (len(O_t)) if O_t(i) == T_St_loc1:
        n=1; n=2 if 'second' in O_w (i);
        n=3 if 'third' in O_w (i)
    for i in range (len(O_t)) if O_t(i) == T_St_loc1 :
        StudInfo = Find_row (id, O_w) – n if 'left' in O_w(i)
        StudInfo = Find_row (id, O_w) + n if 'right' in O_w(i)
if T_St_loc1 in O_t and not in O_t:
    for i in range (len(O_t)):
        if O_t(i) == T_St_loc1 and O_w(i)=='leftmost':
            StudInfo = Find_row (min(StudIdList))
        if O_t(i) == T_St_loc1 and O_w(i)=='rightmost':
            StudInfo = Find_row (max(StudIdList))
return StudInfo
```

Fig. 16. Pseudocode for information extraction about studs.

4.5. Robot Control (RC)

Using studs and drywall panels introduced in the case study, drywall panels can be placed in three different types as shown in Fig. 17. The layouts in Fig. 17(a) and Fig. 17(b) use one unique panel A and one unique panel B, and two standard panels installed vertically and horizontally, respectively. In the layout in Fig. 17(c), two types of distinct panels are placed vertically. Drywall installation is demonstrated based on the outputs of the NLU module and the IM module for three drywall layouts. The input data of the NLU module were selected from the test dataset.

Demonstration results for the layout 1 are shown in the Fig. 18. Figs. 17(a)-(d) show a pair of a natural language instruction and how the KUKA robot successfully placed a panel for each instruction. As a result of IM for the instruction in Fig. 18(a), the drywall panel 500,320 and the stud 500,100 were determined as the target and the final location, respectively. The target panel was installed perpendicular to the left line of the stud. The first row of the action history table in Fig. 18(c) shows this result.

As shown in Fig. 18(b), the drywall panel was installed vertically on the center line of the stud because *Vr_md* was predicted as a result of the NLU module for the second sentence of the language instruction. The second row of the fourth and fifth columns in Fig. 18(e) shows this result. In Fig. 18(c) and Fig. 18(d), "second to the left" and "left" were tagged as *St_loc1*, and "500,109" and "500,111" were tagged as *St_loc2* in the NLU module. The rules of the IM module shown in Fig. 15 determined the stud 500,107 and the stud 500,110 as the final location for the third and fourth instructions, respectively. According to the action history table about the output of the IM, the robot installed drywall panels onto the stud walls.

Fig. 19 and Fig. 20 show the natural language instructions and demonstration results for layout 2 and layout 3. As shown in both figures, the robot successfully installed drywall panels by extracting correct information for pick-and-place operations from the NLU and IM modules.

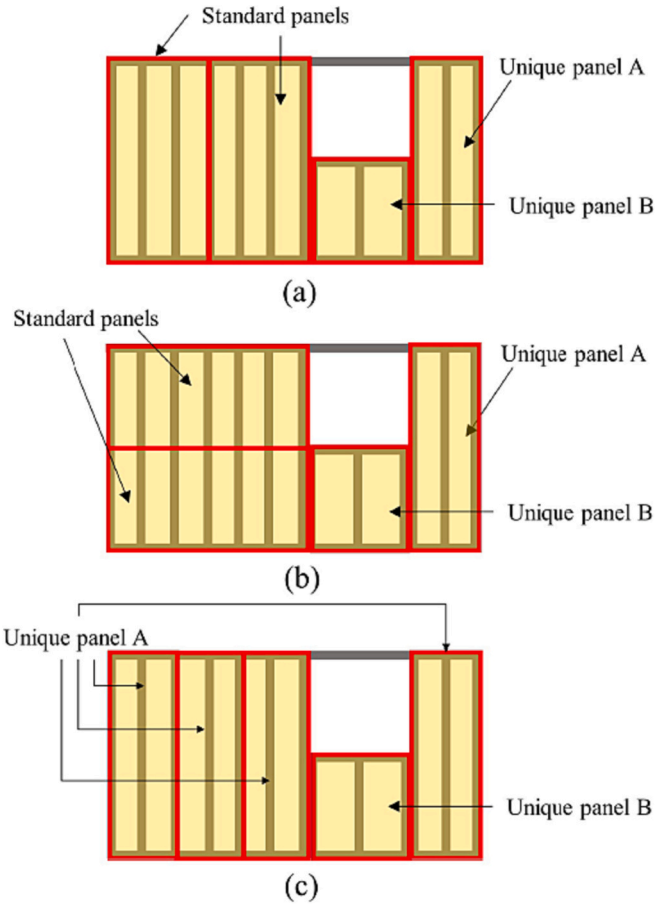


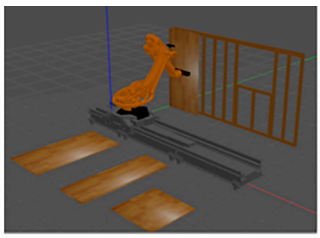
Fig. 17. Three drywall layouts: (a) layout 1; (b) layout 2; (c) layout 3.

4.6. Co-reference issue

This study focused on words distinctly characterizing targets and destinations when establishing annotation rules, rather than all words denoting the targets and destinations. This annotation strategy was chosen due to insufficiency of generic words like drywall, stud or pronouns in clearly distinguishing among multiple panels or studs. However, co-reference issues are crucial for robots to thoroughly interpret human instructions. Thus, additional experiments addressing co-reference issues were conducted using BERT to evaluate the impacts of the co-reference issues in this study.

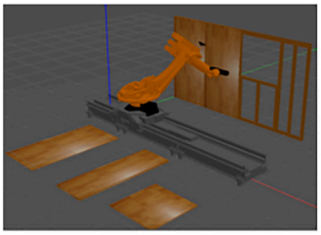
The dataset was re-annotated with two additional labels: *Trg* and *Dst*, representing a target and destination, respectively. For instance, in a three-sentences instruction "Please move the wall panel and move it on the stud 500100. Place it to the upper horizontal row. The dimension of the drywall is 4 by 8", 'wall panel' in the first sentence, 'it' in the second sentence, and 'drywall' in the third sentence were annotated as *Trg* while 'stud' in the first sentence was annotated as *Dst*. BERT was trained following the same procedure as the prior experiments with variations in the volume of training data. Fig. 21 presents the training accuracy for the re-annotated datasets comprising 316, 632, 948, and 1268 instructions.

The insights from Fig. 12(b) and Fig. 21 reveal that the impact of the co-reference issue on training accuracy is not significant in this study. Initially, in epoch 1, the BERT-C models exhibited lower accuracy in comparison to the BERT-M models. However, as training progressed up to epoch 5, the training accuracy of both BERT-C and BERT-M models converged and became similar. Table 5 presents a comprehensive summary of the performance of the trained models on the validation dataset. It can be observed that BERT-C models, which considered co-



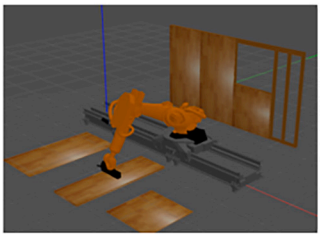
"please pick up the 4 by 8 drywall board and hang it into the 500100 vertically"

(a)



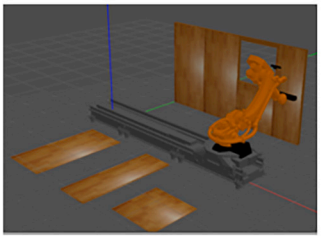
"please pick up the full-size panel and hang it vertically on the stud 500103. join this panel and the previously installed one on the stud in the middle"

(b)



"install the rightmost panel vertically in the stud second to the left of the stud 500109"

(c)



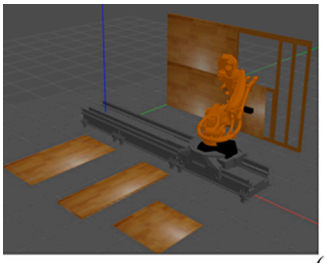
"move the sheet vertically on the stud the width of the drywall is 2.7 and the length is 8. the stud is laying on the left to the 500111"

(d)

stud_id	installed_x_left	installed_x_right	left_cent	ver_hor	top_btm	drywall_id	w	l
500100	0	1.2192	left	vertical	0	500320	1.2192	2.4384
500103	1.2192	2.4384	center	vertical	0	500310	1.2192	2.4384
500107	2.4384	3.6576	left	vertical	0	500380	1.2192	1.2192
500110	3.6576	4.4704	left	vertical	0	500360	0.8128	2.4384

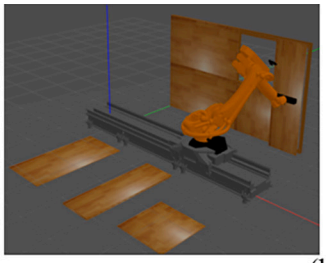
(e)

Fig. 18. Examples of drywall installation for the layout 1: (a)-(d) show a robot installing drywall panels based on natural language instructions; (e) is the action history table.



"can you install the rightmost drywall panel in the stud which is left to the stud 500108"

(a)



"can you place the drywall piece vertically in the stud that is second left from the stud 500112. the dimension of the panel is 2.7 by 8"

(b)

stud_id	installed_x_left (m)	installed_x_right (m)	left_cent	ver_hor	top_btm	drywall_id	w (m)	l (m)
500100	0	1.2192	left	horizontal	top	500320	1.2192	2.4384
500100	0	1.2192	left	horizontal	bottom	500310	1.2192	2.4384
500107	2.4384	3.6576	left	vertical	0	500380	1.2192	1.2192
500110	3.6576	4.4704	left	vertical	0	500360	0.8128	2.4384

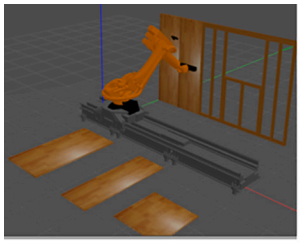
(c)

Fig. 19. Examples of drywall installation for the layout 2: (a) and (b) are corresponding to the third and fourth placement, respectively; (c) is the recorded action history.

reference issues, displayed slightly lower performance compared to the BERT-M models, which did not consider co-reference. However, with a large amount of training data, both BERT-C1 and BERT-C2 achieved accuracy close to 100%. These findings indicate that while co-reference issues may have a minor impact on performance, the BERT models trained with co-reference consideration can still achieve high accuracy when provided with a large amount of training data.

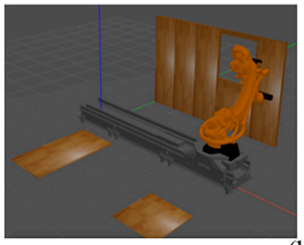
5. Discussion

This paper presented a framework of a natural language-enabled HRC system that consists of three steps: natural language understanding, information mapping, and robot control. The proposed approach enables human workers to interact with construction robots using natural language instructions and building component information. The proposed system was validated through a case study on drywall installation and BERT-M1 achieved a highest accuracy of 99.37% at instruction-level for the 158 test data in the NLU module. Even with a small amount of training data, BERT achieved an instruction-level accuracy close to 80%, suggesting that it is an effective approach for analyzing natural language instructions in the context of construction robotics.



"install the custom-cut drywall panel vertically on the stud 500102. join this panel and the previously installed one on the stud in the middle. the width is 2.7 foot, and the length is 8 foot"

(a)



"please pick up the sheet 500330 and position it in the stud 500110"

(b)

stud_id	installed_x_left (m)	installed_x_right (m)	left_cent	ver_hor	top_btm	drywall_id	w (m)	l (m)
500100	0	0.8128	left	vertical	0	500360	0.8128	2.4384
500102	0.8128	1.6256	center	vertical	0	500350	0.8128	2.4384
500104	1.6256	2.4384	center	vertical	0	500340	0.8128	2.4384
500107	2.4384	3.6576	left	vertical	0	500380	1.2192	1.2192
500110	3.6576	4.4704	left	vertical	0	500330	0.8128	2.4384

(c)

Fig. 20. Examples of drywall installation for the layout 3: (a) and (b) are corresponding to the second and fifth placement, respectively; (c) is the recorded action history.

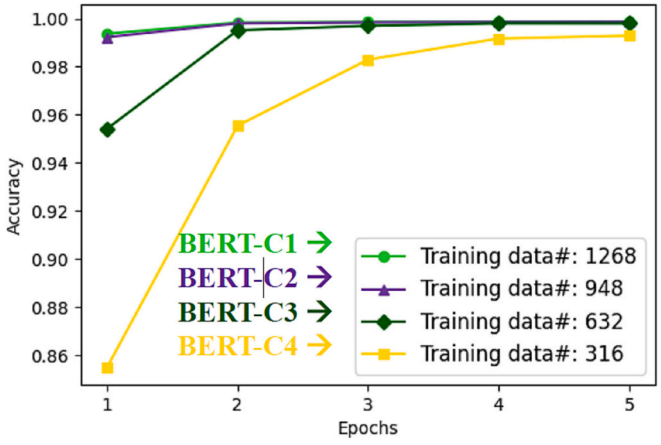


Fig. 21. Training accuracy on the re-annotated dataset.

Table 5
Model performance on validation dataset with co-reference issues.

Model	Result 1		Result 2	
	N_w	Acc_{word}	N_l	Acc_{inst}
BERT-C1	2	99.95%	2	98.73%
BERT-C2	2	99.95%	2	98.73%
BERT-C3	14	99.64%	11	93.04%
BERT-C4	62	98.41%	44	72.15%

N_w= the number of incorrect prediction of words.
N_l = the number of language instructions including incorrect prediction.
$Acc_{word}= \frac{3,895 - N\_w}{3895}$, $Acc_{inst}= \frac{158 - N\_l}{158}$

However, it should be noted that BERT-based models may require more training time compared to BiLSTM-based models [108]. Therefore, if the amount of available data is sufficient, it may be worthwhile to consider using the BiLSTM-CRF model, which has shown similar performance to BERT for tagging tasks in this study. In the IM and RC module, it is observed that drywall installation tasks were performed successfully through natural interaction using language instructions. This study clearly demonstrates that the proposed system has significant potential for field implementation to achieve natural interaction with robots in construction.

Even though the proposed method achieved high performance on the given datasets, there are still some challenges that must be addressed. First, the conducted experiments did not consider the potential influence of background noise typical on construction sites, which could affect the voice data processing. However, the recent advancements in noise-robust speech recognition techniques [109,110] suggest a promising outlook for the implementation of voice commands in noisy construction environments. Additionally, with the increasing integration of digital twins in construction and the potential for remote interaction system could significantly reduce the adverse effects of on-site noise, ensuring clear communication with the construction robots.

Second, the proposed framework relies entirely on the output of the NLU module to generate the final command in the IM module to accurately interpret contextual and historical data with language instructions. However, the proposed system has dependency of the IM module on the NLU module's accuracy. Park et al. [111] attempted to address this by exploring the combination of these two modules using a single language model. While this approach showed potential, it encountered limitations in considering historical data due to its reliance on single language instructions as inputs. Future studies can explore the development of a more integrated language model that leverages natural language instructions, building component information, and historical work data as input. Such an approach could potentially simplify the translation process and enhance the overall accuracy and robustness of the system, moving closer to a more streamlined natural language to robot language translation.

Thirdly, there is a data generation rule requiring key information to be mentioned only once in a single instruction. In future work, this limitation could be mitigated by expanding the dataset in the NLU module and incorporating additional conditional statements in the IM module. Additionally, the current dataset was never intended to replicate human-to-human communication prevalent among field practitioners, which means the ways in which objects are described in the commands may differ from colloquial on-site language between humans. Future studies could further solidify the practicality of the interaction system by sourcing or validating data directly from construction workers.

Despite these limitations, it is important to note that the goal of this study is to improve the interaction between human operators and robots in future work environments. These environments, where both humans

and robots access databases similar to BIM, necessitate a shift in language from traditional site commands. To test the practicality of the approach, a supplementary study with 12 construction workers was conducted in a subsequent study using speech-based commands with a robot for panel installation tasks. These workers effectively communicated with the robot using specific IDs or location data, with commands like "Okay, robot, please pick up panel 504 and place it at the center of the stud 606" and "Put 503 on the rightmost section." A survey using a five-point Likert scale (1 being Strongly Disagree and 5 being Strongly Agree) on usefulness and ease of use for the interaction yielded an average score above 4. This implies that while the current dataset may differ from authentic language commands, it remains an acceptable and viable command form for construction workers.

Fourth, the case study was conducted in a single stud structure with a fixed perspective for identifying locations of panels and studs. In future work, the proposed approach can be improved by extending the system with more complex structures and building materials, along with considering diverse perspective of human workers. Such advancements would require both an expansion of the instruction dataset and refinement of the motion planning process. As Wang et al. [14] note, calculating collision-free trajectories in pick-and-place operations becomes challenging with large objects and in complex workspace. Future study could incorporate operator intervention on the robot's trajectories, as proposed by Wang et al. [14]. This would allow operators to actively participate in directing the robot by suggesting specific intermediate positions, thereby facilitating the generation of optimal path plans.

Finally, bidirectional communication was not considered in the proposed system. It implies that human workers are unable to intervene in robot tasks or provide new plans when the robot encounters difficulties for higher level of HRC. Additionally, the system does not verify whether the instructions from workers are accurate or not, as there is no built-in filter to assess this. These limitations highlight the need for more complicated communication protocols that require a deeper understanding of human-robot interaction. To address this, the authors will consider bidirectional communication in a future study to improve the proposed system and increase the level of natural interaction with construction robots.

6. Conclusion

This study made several contributions: the research laid the foundation for natural interaction with robots by using natural language instructions in pick-and-place construction operations. To our best knowledge, it is the first study to propose a framework for interaction with construction robots using natural language instructions, building component information, and working history. It effectively handles complex data such as target object, destination, and placement method, facilitating natural and intuitive human-robot interactions in pick-and-place operations. This integration of three modules – NLU, IM, and RC – marks a significant stride in enabling efficient verbal communication with construction robots.

Second, we demonstrated interaction with construction robots using natural language instructions. A demonstration of the proposed system in drywall installation tasks showed the potential of HRC through speech channels in construction. We extracted information about target objects, destinations, and placement orientation that can be applied to other pick-and-place operations in construction tasks, such as ceiling tile installation, wall tile installation, or bricklaying. Even though the application of the framework we proposed was demonstrated through a drywall installation, the framework itself is generalizable and adaptable to any pick-and-place construction task making this technical contribution broadly applicable.

Third, to address the lack of an existing dataset suitable for drywall installation, a natural language instruction dataset was created based on human interactions and work observed in construction videos and related studies. The dataset stands out due to its fine-grained annotation as it was meticulously annotated to deal with the necessary information for pick-and-place operations including unique characteristics such as IDs, dimensions, or locations. This annotation process enhanced the quality and depth of the labeled data, making our dataset a valuable resource for advancing research in the field of construction-related natural language processing. Furthermore, the dataset labeling approach can be adapted to create datasets for other pick-and-place operations.

Fourth, the proposed system facilitates interaction with the robot by using the information available in the construction projects. The data mapping process interprets building component information and previous working records as well as information from analyzed language instructions. This empowers human operators to give language instructions to a robot in a shorter or more intuitive way. We believe that this approach significantly contributes to the development of a practical and efficient human-robot collaboration system on construction sites.

Finally, two different language models, which are BiLSTM-CRF and BERT, were trained by labels reflecting characteristics of construction activities. Our comparative analysis of these models with the newly generated dataset revealed their effectiveness in a construction setting. In addition, BERT proved to be highly accurate, even with limited data, achieving a 96% instruction-level accuracy in the validation set. This has important implications for the construction industry, where there is a lack of data for natural language instructions. Our study demonstrates that leveraging and fine-tuning pre-trained models like BERT can address this challenge, enabling high accuracy in interpreting construction-related instructions.

CRediT authorship contribution statement

Somin Park: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Conceptualization. **Xi Wang:** Visualization, Validation, Software. **Carol C. Menassa:** Writing – review & editing, Validation, Supervision, Resources, Funding acquisition, Conceptualization. **Vineet R. Kamat:** Writing – review & editing, Validation, Supervision, Resources, Funding acquisition, Conceptualization. **Joyce Y. Chai:** Writing – review & editing.

Declaration of competing interest

Carol C. Menassa reports financial support was provided by National Science Foundation. One of the authors Vineet R. Kamat is a member of the Editorial Board for the journal Automation in Construction.

Data availability

Data will be made available on request.

Acknowledgments

The work presented in this paper was supported financially by two United States National Science Foundation (NSF) Awards: 2025805 and 2128623. The support of the NSF is gratefully acknowledged.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.autcon.2024.105345.

References

[1] J.M. Davila Delgado, L. Oyedele, A. Ajayi, L. Akanbi, O. Akinade, M. Bilal, A. Owolabi, Robotics and automated systems in construction: understanding industry-specific challenges for adoption, J. Build. Eng. 26 (2019), https://doi.org/10.1016/j.jobe.2019.100868, pp. 100868.

[2] X. Wang, S. Wang, C.C. Menassa, V.R. Kamat, W. McGee, Automatic high-level motion sequencing methods for enabling multi-tasking construction robots,

Autom. Constr. 155 (2023) 105071, https://doi.org/10.1016/j.autcon.2023.105071.
[3] J. Cai, A. Du, X. Liang, S. Li, Prediction-based path planning for safe and efficient human–robot collaboration in construction via deep reinforcement learning, J. Comput. Civ. Eng. 37 (1) (2023) 1–10, https://doi.org/10.1061/(ASCE)CP.1943-5487.0001056.
[4] C. Feng, Y. Xiao, A. Willette, W. McGee, V.R. Kamat, Vision guided autonomous robotic assembly and as-built scanning on unstructured construction sites, Autom. Constr. 59 (2015) 128–138, https://doi.org/10.1016/j.autcon.2015.06.002.
[5] K.M. Lundeen, V.R. Kamat, C.C. Menassa, W. McGee, Scene understanding for adaptive manipulation in robotized construction work, Autom. Constr. 82 (2017) 16–30, https://doi.org/10.1016/j.autcon.2017.06.022.
[6] M. Pan, T. Linner, W. Pan, H.-M. Cheng, T. Bock, Influencing factors of the future utilisation of construction robots for buildings: a Hong Kong perspective, J. Build. Eng. 30 (2020), https://doi.org/10.1016/j. jobe.2020.101220 pp.101220.
[7] C.J. Liang, V.R. Kamat, C.C. Menassa, Teaching robots to perform quasi-repetitive construction tasks through human demonstration, Autom. Constr. 120 (2020), https://doi.org/10.1016/J.AUTCON.2020.103370 pp.103370.
[8] G. Michalos, S. Makris, J. Spiliotopoulos, I. Misios, P. Tsarouchi, G. Chryssolouris, ROBO-PARTNER: Seamless human-robot cooperation for intelligent, flexible and safe operations in the assembly factories of the future, in: 5th CATS 2014 - CIRP Conference on Assembly Technologies and Systems, Procedia CIRP, Vol. 23, 2014, pp. 71–76, https://doi.org/10.1016/j.procir.2014.10.079.
[9] F. Sherwani, M.M. Asad, B.S.K.K. Ibrahim, Collaborative robots and industrial revolution 4.0 (ir 4.0), in: 2020 International Conference on Emerging Trends in Smart Technologies (ICETST), 2020, pp. 1–5, https://doi.org/10.1109/ICETST49965.2020.9080724.
[10] X. Su, S. Talmaki, H. Cai, V.R. Kamat, Uncertainty-aware visualization and proximity monitoring in urban excavation: a geospatial augmented reality approach, Visual. Eng. (2013) 1–13, https://doi.org/10.1186/2213-7459-1-2.
[11] F. Pini, F. Leali, M. Ansaloni, A systematic approach to the engineering design of a HRC workcell for bio-medical product assembly, in: 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), 2015, pp. 1–8, https://doi.org/10.1109/ETFA.2015.7301655.
[12] G. Cupido, The role of production and teamwork practices in construction safety: a cognitive model and an empirical case study, J. Saf. Res. 40 (2009) 265–275, https://doi.org/10.1016/j.jsr.2009.05.002.
[13] C.-J. Liang, X. Wang, V.R. Kamat, C.C. Menassa, Human–robot collaboration in construction: classification and research trends, J. Constr. Eng. Manag. 147 (10) (2021), https://doi.org/10.1061/(ASCE)CO.1943-7862. 0002154, pp. 03121006.
[14] X. Wang, C.-J. Liang, C.C. Menassa, V.R. Kamat, Interactive and immersive process-level digital twin for collaborative human–robot construction work, J. Comput. Civ. Eng. 35 (2021), https://doi.org/10.1061/(asce)cp.1943-5487.0000988, pp. 04021023.
[15] V. Villani, F. Pini, F. Leali, C. Secchi, Survey on human–robot collaboration in industrial settings: safety, intuitive interfaces and applications, Mechatronics 55 (2018) 248–266, https://doi.org/10.1016/j.mechatronics.2018.02.009.
[16] I. Maurtua, I. Fernandez, A. Tellaeche, J. Kildal, L. Susperregi, A. Ibarguren, B. Sierra, Natural multimodal communication for human–robot collaboration, Int. J. Adv. Robot. Syst. 14 (2017) 1–12, https://doi.org/10.1177/1729881417716043.
[17] S. Park, H. Yu, C.C. Menassa, V.R. Kamat, A comprehensive evaluation of factors influencing acceptance of robotic assistants in field construction work, J. Manag. Eng. 39 (3) (2023), https://doi.org/10.1061/JMENEA.MEENG-5227, 04023010.
[18] M. Tanzini, J.M. Jacinto-Villegas, A. Filippeschi, M. Niccolini, M. Ragaglia, New interaction metaphors to control a hydraulic working machine's arm, in: Proc IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), 2016, pp. 297–303, https://doi.org/10.1109/SSRR.2016.7784319.
[19] P. Adami, B. Rodrigues Patrick, J. Woods Peter, B. Becerik-Gerber, L. Soibelman, Y. Copur-Gencturk, G. Lucas, Impact of VR-based training on human–robot interaction for remote operating construction robots, J. Comput. Civ. Eng. 36 (3) (2022), https://doi.org/10.1061/(ASCE)CP.1943-5487.0001016 pp.04022006.
[20] Y. Liu, M. Habibnezhad, H. Jebelli, Brain-computer interface for hands-free teleoperation of construction robots, Autom. Constr. 123 (2021), https://doi.org/10.1016/j.autcon.2020.103523 pp.103523.
[21] C. Follini, A.L. Cheng, G. Latorre, L.F. Amores, Design and development of a novel robotic gripper for automated scaffolding assembly, in: Proc IEEE Third Ecuador Technical Chapters Meeting, 2018, pp. 1–6, https://doi.org/10.1109/ETCM.2018.8580276.
[22] S. Karpagavalli, E. Chandra, A review on automatic speech recognition architecture and approaches, Int. J. Signal Proc. Image Proc. Pattern Recognit. 9 (4) (2016) 393–404, https://doi.org/10.14257/ijsip.2016.9.4.3.
[23] P. Tsarouchi, S. Makris, G. Chryssolouris, Human–robot interaction review and challenges on task planning and programming, Int. J. Comput. Integr. Manuf. 29 (8) (2016) 916–931, https://doi.org/10.1080/0951192X.2015.1130251.
[24] Z. Peng, X. Li, Z. Zhu, M. Unoki, J. Dang, M. Akagi, Speech emotion recognition using 3D convolutions and attention-based sliding recurrent networks with auditory front-end, IEEE Access 8 (2020) 16560–16572, https://doi.org/10.1109/ACCESS.2020.2967791.
[25] D. Mukherjee, K. Gupta, L.H. Chang, H. Najjaran, A survey of robot learning strategies for human-robot collaboration in industrial settings, Robot. Comput. Integr. Manuf. 73 (2022), https://doi.org/10.1016/j.rcim.2021.102231 pp.102231.
[26] R. Liu, X. Zhang, Systems of natural-language-facilitated human-robot cooperation: a review, arXiv (2017) 1–21, https://doi.org/10.48550/arXiv.1701.08269.
[27] J.R. Lin, Z.Z. Hu, J.P. Zhang, F.Q. Yu, A natural-language-based approach to intelligent data retrieval and representation for cloud BIM, Comput. Aided Civ. Inf. Eng. 31 (1) (2016) 18–33, https://doi.org/10.1111/mice.12151.
[28] R. Paul, J. Arkin, N. Roy, T.M. Howard, Efficient grounding of abstract spatial concepts for natural language interaction with robot manipulators, Robot. Sci. Syst. Found. (2016) 1–9, https://doi.org/10.15607/RSS.2016.XII.037.
[29] Y. Bisk, D. Yuret, D. Marcu, Natural language communication with robots, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 751–761, https://doi.org/10.18653/v1/N16-1089.
[30] J. Hatori, Y. Kikuchi, S. Kobayashi, K. Takahashi, Y. Tsuboi, Y. Unno, W. Ko, J. Tan, Interactively picking real-world objects with unconstrained spoken language instructions, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 3774–3781, https://doi.org/10.1109/ICRA.2018.8460699.
[31] A. Magassouba, K. Sugiura, A.T. Quoc, H. Kawai, Understanding natural language instructions for fetching daily objects using gan-based multimodal target–source classification, IEEE Robot. Autom. Lett. 4 (4) (2019) 3884–3891, https://doi.org/10.1109/LRA.2019.2926223.
[32] G. Albeaino, M. Gheisari, R.R. Issa, Human-drone interaction (HDI): opportunities and considerations in construction automation and robotics in the architecture, Engi. Constr. Ind. (2022) 111–142, https://doi.org/10.1007/978-3-030-77163-8_6.
[33] X. Wang, Z. Zhu, Vision–based framework for automatic interpretation of construction workers' hand gestures, Autom. Constr. 130 (2021), https://doi.org/10.1016/j.autcon.2021.103872 pp.103872.
[34] J. Von Tiesenhausen, U. Artan, J.A. Marshall, Q. Li, Hand gesture-based control of a front-end loader, in: 2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 2020, pp. 1–4, https://doi.org/10.1109/CCECE47787.2020.9255828.
[35] M. Tölgyessy, M. Dekan, F. Duchň, J. Rodina, P. Hu-binský, L. Chovanec, Foundations of visual linear human–robot interaction via pointing gesture navigation, Int. J. Soc. Robot. 9 (4) (2017) 509–523, https://doi.org/10.1007/s12369-017-0408-9t.
[36] V.R. Kamat, J.C. Martinez, Scene graph and frame update algorithms for smooth and scalable 3D visualization of simulated construction operations, Comput. Aided Civ. Inf. Eng. 17 (4) (2002) 228–245, https://doi.org/10.1111/1467-8667.00272.
[37] V.R. Kamat, J.C. Martinez, Automated generation of dynamic, operations level virtual construction scenarios electronic, J. Inf. Technol. Constr. 8 (2003) 65–84. http://www.itcon.org/2003/6.
[38] S. Dong, C. Feng, V.R. Kamat, Sensitivity analysis of augmented reality-assisted building damage reconnaissance using virtual prototyping, Autom. Constr. 33 (2013) 24–36, https://doi.org/10.1016/j.autcon.2012.09.005.
[39] S. Ahmed, M.M. Hossain, M.I. Hoque, A brief discussion on augmented reality and virtual reality in construction industry, J. Syst. Manag. Sci. 7 (3) (2017) 1–33. http://www.aasmr.org/jsms/Archives/Vol-7/Vol-7-3/.
[40] L. Pérez, E. Diez, R. Usamentiaga, D.F. García, Industrial robot control and operator training using virtual reality interfaces, Comput. Ind. 109 (2019) 114–120, https://doi.org/10.1016/j.compind.2019.05.001.
[41] T. Zhou, Q. Zhu, J. Du, Intuitive robot teleoperation for civil engineering operations with virtual reality and deep learning scene reconstruction, Adv. Eng. Inform. 46 (2020), https://doi.org/10.1016/j. aei.2020.101170 pp.101170.
[42] A.H. Behzadan, V.R. Kamat, Integrated information modeling and visual simulation of engineering operations using dynamic augmented reality scene graphs, J. Inf. Technol. Constr. 16 (2011) 259–278. https://www.itcon.org/paper/2011/16.
[43] M. Dalle Mura, G. Dini, Augmented reality in assembly systems: state of the art and future perspectives, I, in: Smart Technologies for Precision Assembly: 9th IFIP WG 5.5 International Precision Assembly Seminar, IPAS 2020, Virtual Event, December 14–15, 2020, pp. 3–22, https://doi.org/10.1007/978-3-030-72632-4_1.
[44] M. Dianatfar, J. Latokartano, M. Lanz, Review on existing VR/AR solutions in human–robot collaboration, Procedia CIRP 97 (2021) 407–411, https://doi.org/10.1016/j.procir.2020.05.259.
[45] Z. Ji, Q. Liu, W. Xu, B. Yao, J. Liu, Z. Zhou, A closed-loop brain-computer interface with augmented reality feedback for industrial human-robot collaboration, Int. J. Adv. Manuf. Technol. (2021) 1–16, https://doi.org/10.1007/s00170-021-07937-z.
[46] Y. Liu, M. Habibnezhad, H. Jebelli, Brainwave-driven human-robot collaboration in construction, Autom. Constr. 124 (2021), https://doi.org/10.1016/j.autcon.2021.103556 pp.103556.
[47] M. Aljalal, S. Ibrahim, R. Djemal, W. Ko, Comprehensive review on brain-controlled mobile robots and robotic arms based on electroencephalography signals, Intell. Serv. Robot. 13 (4) (2020) pp. 539–563, https://doi.org/10.1007/s11370-020-00328-5.
[48] S.O. Abioye, L.O. Oyedele, L. Akanbi, A. Ajayi, J.M. Davila Delgado, M. Bilal, O. O. Akinade, A. Ahmed, Artificial intelligence in the construction industry: a review of present status, opportunities and future challenges, journal of building, Engineering 44 (2021), https://doi.org/10.1016/j.jobe.2021.103299 pp.103299.
[49] M. Beetz, M. Scheutz, F. Yazdani, Guidelines for improving task-based natural language understanding in human-robot rescue teams, in: 2017 8th IEEE

International Conference on Cognitive Infocommunications, 2017, pp. 203–208, https://doi.org/10.1109/CogInfoCom.2017.8268243.
[50] O. Mees, A. Emek, J. Vertens, W. Burgard, Learning object placements for relational instructions by hallucinating scene representations, in: 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 94–100, https://doi.org/10.48550/arxiv.2001.08481.
[51] S. Ishikawa, K. Sugiura, Target-dependent UNITER: a transformer-based multimodal language comprehension model for domestic service robots, IEEE Robot. Autom. Lett. 6 (4) (2021) 8401–8408, https://doi.org/10.1109/LRA.2021.3108500.
[52] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Proceedings of the 31st International Conference on Neural Information Processing Systems, Neural Information Processing Systems Foundation, 2017, pp. 6000–6010, https://doi.org/10.48550/arXiv.1706.03762.
[53] D. Guo, H. Liu, F. Sun, Audio–visual language instruction understanding for robotic sorting, Robot. Auton. Syst. 159 (2023), https://doi.org/10.1016/j.robot.2022.104271 pp.104271.
[54] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, Hierarchical attention networks for document classification, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics (ACL), 2016, pp. 1480–1489, https://doi.org/10.18653/V1/N16-1174.
[55] D. Nyga, S. Roy, R. Paul, D. Park, M. Pomarlan, M. Beetz, N. Roy, Grounding robot plans from natural language instructions with incomplete world knowledge, in: Conference on Robot Learning, 2018, pp. 714–723, in: https://proceedings.mlr.press/v87/nyga18a.html.
[56] H. Chen, H. Tan, A. Kuntz, M. Bansal, R. Alterovitz, Enabling robots to understand incomplete natural language instructions using commonsense reasoning, in: 2020 IEEE International Conference on Robotics and Automation, 2020, pp. 1963–1969, https://doi.org/10.48550/arxiv.1904.12907.
[57] J. Brawer, O. Mangin, A. Roncone, S. Widder, B. Scassellati, Situated Human-Robot Collaboration: predicting intent from grounded natural language, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2018, pp. 827–833, https://doi.org/10.1109/IROS.2018.8593942.
[58] Y. Kang, Z. Cai, C.W. Tan, Q. Huang, H. Liu, Natural language processing (NLP) in management research: a literature review, J. Manag. Anal. 7 (2) (2020) 139–172, https://doi.org/10.1080/23270012.2020.1756939.
[59] Y. Ding, J. Ma, X. Luo, Applications of natural language processing in construction, Autom. Constr. 136 (2022), https://doi.org/10.1016/j.autcon.2022.104169 pp.104169.
[60] H. Fan, H. Li, Retrieving similar cases for alternative dispute resolution in construction accidents using text mining techniques, Autom. Constr. 34 (2013) 85–91, https://doi.org/10.1016/j.autcon.2012.10.014.
[61] T. Kim, S. Chi, Accident case retrieval and analyses: using natural language processing in the construction industry, J. Constr. Eng. Manag. 145 (3) (2019), https://doi.org/10.1061/(ASCE)CO.1943-7862.0001625 pp.04019004.
[62] A.J.-P. Tixier, M.R. Hallowell, B. Rajagopalan, D. Bowman, Automated content analysis for construction safety: a natural language processing system to extract precursors and outcomes from unstructured injury reports, Autom. Constr. 62 (2016) 45–56, https://doi.org/10.1016/j.autcon.2015.11.001.
[63] J. Zhang, N.M. El-Gohary, Extending building information models Semiautomatically using semantic natural language processing techniques, J. Comput. Civ. Eng. 30 (2016), https://doi.org/10.1061/(ASCE)CP.1943-5487.0000536 pp.C4016004.
[64] L.J. McGibbney, B. Kumar, An intelligent authoring model for subsidiary legislation and regulatory instrument drafting within construction and engineering industry, Autom. Constr. 35 (2013) 121–130, https://doi.org/10.1016/j.autcon.2013.04.005.
[65] J. Lee, Y. Ham, J.-S. Yi, J. Son, Effective risk positioning through automated identification of missing contract conditions from the contractor's perspective based on FIDIC contract cases, J. Manag. Eng. 36 (3) (2020), https://doi.org/10.1061/(ASCE)ME.1943-5479.0000757 pp.05020003.
[66] B. Kosovac, D.J. Vanier, T.M. Froese, Use of keyphrase extraction software for creation of an AEC/FM thesaurus, J. Inf. Technol. Constr. 5 (2) (2002) 25–36. http://www.itcon.org/2002/5.
[67] H. Liu, V. Kwigizile, W.C. Huang, Holistic framework for highway construction cost index development based on inconsistent pay items, J. Constr. Eng. Manag. 147 (7) (2021), https://doi.org/10.1061/(ASCE)CO.1943-7862.0002080 pp.04021052.
[68] K.C. Roy, S. Hasan, P. Mozumder, A multilabel classification approach to identify hurricane-induced infrastructure disruptions using social media data, Comput. Aided Civ. Inf. Eng. 35 (12) (2020) 1387–1402, https://doi.org/10.1111/mice.12573.
[69] B. Zhong, X. Xing, H. Luo, Q. Zhou, H. Li, T. Rose, W. Fang, Deep learning-based extraction of construction procedural constraints from construction regulations, Adv. Eng. Inform. 43 (2020), https://doi.org/10.1016/j. aei.2019.101003 pp.101003.
[70] C. Wu, P. Wu, J. Wang, R. Jiang, M. Chen, X. Wang, Developing a hybrid approach to extract constraints related information for constraint management, Autom. Constr. 124 (2021), https://doi.org/10.1016/j. autcon.2021.103563 pp.103563.
[71] R. Liu, J. Webb, X. Zhang, Natural-language-instructed industrial task execution, in: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2016, pp. 1–7, https://doi.org/10.1115/DETC2016-60063, 50084.
[72] S. Shin, R.R. Issa, BIMASR: framework for voice-based BIM information retrieval, J. Constr. Eng. Manag. 147 (10) (2021), https://doi.org/10.1061/(ASCE)CO.1943-7862.0002138 pp.04021124.
[73] N. Wang, R.R. Issa, C.J. Anumba, NLP-based query answering system for information extraction from building information models, J. Comput. Civ. Eng. 36 (2022) 04022004, https://doi.org/10.1061/(ASCE)CP.1943-5487.0001019.
[74] F. Xu, T. Nguyen, J. Du, Augmented reality for maintenance tasks with ChatGPT for automated text-to-action, arXiv (2023), https://doi.org/10.48550/arXiv.2307.03351.
[75] Y. Ye, H. You, J. Du, Improved trust in human-robot collaboration with ChatGPT, IEEE Access (2023) 55748–55754, https://doi.org/10.1109/ACCESS.2023.3282111.
[76] M. Eppe, S. Trott, J. Feldman, Exploiting deep semantics and compositionality of natural language for human-robot-interaction, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2016, pp. 731–738, https://doi.org/10.1109/IROS.2016.7759133.
[77] M. Ralph, M.A. Moussa, Toward a natural language interface for transferring grasping skills to robots, IEEE Trans. Robot. 24 (2) (2008) 468–475, https://doi.org/10.1109/TRO.2008.915445.
[78] C. Matuszek, E. Herbst, L. Zettlemoyer, D. Fox, Learning to parse natural language commands to a robot control system, in: Experimental Robotics: the 13th International Symposium on Experimental Robotics, 2013, pp. 403–415, https://doi.org/10.1007/978-3-319-00065-7_28.
[79] L. She, J. Chai, Interactive learning of grounded verb semantics towards human-robot communication, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics 1, 2017, pp. 1634–1644, https://doi.org/10.18653/v1/P17-1150.
[80] K.M. Lundeen, V.R. Kamat, C.C. Menassa, W. McGee, Autonomous motion planning and task execution in geometrically adaptive robotized construction work, Autom. Constr. 100 (2019) 24–45, https://doi.org/10.1016/j.autcon.2018.12.020.
[81] T.D. Oesterreich, F. Teuteberg, Understanding the implications of digitisation and automation in the context of Industry 4.0: a triangulation approach and elements of a research agenda for the construction industry, Comput. Ind. 83 (2016) 121–139, https://doi.org/10.1016/j.compind.2016.09.006.
[82] Y. Chen, J.M. Kamara, A framework for using mobile computing for information management on construction sites, Autom. Constr. 20 (7) (2011) 776–788, https://doi.org/10.1016/j.autcon.2011.01.002.
[83] H. Liu, M. Al-Hussein, M. Lu, BIM-based integrated approach for detailed construction scheduling under resource constraints, Autom. Constr. 53 (2015) 29–43, https://doi.org/10.1016/j.autcon.2015.03.008.
[84] D. Heigermoser, B. García de Soto, E.L.S. Abbott, D.K.H. Chua, BIM-based Last Planner system tool for improving construction project management, Autom. Constr. 104 (2019) 246–254, https://doi.org/10.1016/j. autcon.2019.03.019.
[85] A. Fazeli, M.S. Dashti, F. Jalaei, M. Khanzadi, An integrated BIM-based approach for cost estimation in construction projects, Eng. Constr. Archit. Manag. 28 (9) (2020) 2828–2854, https://doi.org/10.1108/ECAM-01-2020-0027.
[86] S.M. Berger, T.D. Ludwig, Reducing warehouse employee errors using voice-assisted technology that provided immediate feedback, J. Organ. Behav. Manag. 27 (1) (2007) 1–31, https://doi.org/10.1300/J075v27n01_01.
[87] D. Goomas, P.H. Yeow, Ergonomics improvement in a harsh environment using an audio feedback system, Int. J. Ind. Ergon. 40 (6) (2010) 767–774, https://doi.org/10.1016/j.ergon.2010.08.005.
[88] D. Goomas, Increasing warehouse worker performance using voice technology that provided immediate feedback: personal performance productivity prompt, J. Organ. Behav. Manag. 43 (2022) 1–10, https://doi.org/10.1080/01608061.2022.2113588.
[89] K. Kim, M. Peavy, BIM-based semantic building world modeling for robot task planning and execution in built environments, Autom. Constr. 138 (2022), https://doi.org/10.1016/j.autcon.2022.104247 pp.104247.
[90] O. Chong, J. Zhang, R.M. Voyles, B.C. Min, BIM-based simulation of construction robotics in the assembly process of wood frames, Autom. Constr. 137 (2022), https://doi.org/10.1016/j.autcon.2022.104194 pp.104194.
[91] A. Benayas, R. Hashempour, D. Rumble, S. Jameel, R.C. De Amorim, Unified transformer multi-task learning for intent classification with entity recognition, IEEE Access. 9 (2021) 147306–147314, https://doi.org/10.1109/ACCESS.2021.3124268.
[92] A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, Neural Netw. 18 (2005) 602–610, https://doi.org/10.1016/j.neunet.2005.06.042.
[93] J. Lafferty, A. McCallum, F.C. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: Proceeding ICML '01 Proceedings of the Eighteenth International Conference on Machine Learning, MA, USA, Jun 28 – Jul 01, 2001, pp. 282–289. https://repository.upenn.edu/handle/20.500.14332/6188.
[94] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 1, 2019, pp. 4171–4186, https://doi.org/10.18653/v1/N19-1423.
[95] Z. Huang, W. Xu, K. Yu, Bidirectional LSTM-CRF models for sequence tagging, arXiv Preprint (2015) 1–10, https://doi.org/10.48550/arxiv.1508.01991, arXiv: 1508.01991.

[96] N. Reimers, I. Gurevych, Optimal hyperparameters for deep lstm-networks for sequence labeling tasks, arXiv preprint (2017) 1–34, https://doi.org/10.48550/arXiv.1707.06799, arXiv:1707.06799.
[97] J. Kong, Y. Cai, D. Ren, Z. Li, Deep multi-task learning with cross connected layer for slot filling, in: CCF International Conference on Natural Language Processing and Chinese Computing, 2019, pp. 308–317, https://doi.org/10.1007/978-3-030-32236-6_27.
[98] S. Chitta, I. Sucan, S. Cousins, MoveIt! [ROS Topics], IEEE Robot. Autom. Mag. 19 (1) (2012) 18–19, https://doi.org/10.1109/MRA.2011.2181749.
[99] I.A. Sucan, M. Moll, L.E. Kavraki, The open motion planning library, IEEE Robot. Autom. Mag. 19 (4) (2012) 72–82, https://doi.org/10.1109/MRA.2012.2205651.
[100] J. Pan, S. Chitta, D. Manocha, FCL: a general purpose library for collision and proximity queries, in: 2012 IEEE International Conference on Robotics and Automation, 2012, pp. 3859–3866, https://doi.org/10.1109/ICRA.2012.6225337.
[101] Home RenoVision DIY, How To Install Drywall A to Z | DIY Tutorial, Accessed November 20, 2023, https://www.youtube.com/watch?v=VQIMaR7hWtM, 2020.
[102] P. Tang, P. Yang, Y. Shi, Y. Zhou, F. Lin, Y. Wang, Recognizing Chinese judicial named entity using BiLSTM-CRF, J. Phys. Conf. Ser. 1592 (1) (2020), https://doi.org/10.1088/1742-6596/1592/1/012040 pp.012040.
[103] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv (2014) 1–15, https://doi.org/10.48550/arXiv.1412.6980.
[104] L. Rice, E. Wong, Z. Kolter, Overfitting in adversarially robust deep learning, in: International Conference on Machine Learning, 2020, pp. 8093–8104, in: http://proceedings.mlr.press/v119/rice20a.
[105] Z. Zheng, X.Z. Lu, K.Y. Chen, Y.C. Zhou, J.R. Lin, Pretrained domain-specific language model for natural language processing tasks in the AEC domain, Comput. Ind. 142 (2022), https://doi.org/10.1016/j.compind.2022.103733 pp.103733.
[106] F. Wei, H. Qin, S. Ye, H. Zhao, Empirical study of deep learning for text classification in legal document review, in: Proceedings – 2018 IEEE International Conference on Big Data 2018, Big Data, 2019, pp. 3317–3320, https://doi.org/10.1109/BigData.2018.8622157.
[107] A. Mathew, P. Amudha, S. Sivakumari, Deep learning techniques: an overview, Proc. AMLTA (2021) 599–608, https://doi.org/10.1007/978-981-15-3383-9_54.
[108] A. Ezen-Can, A comparison of LSTM and BERT for small corpus, arXiv (2020) 1–12, https://doi.org/10.48550/arxiv.2009.05451.
[109] T. Fukumori, C. Cai, Y. Zhang, L. El Hafi, Y. Hagiwara, T. Nishiura, T. Taniguchi, Optical laser microphone for human-robot interaction: speech recognition in extremely noisy service environments, Adv. Robot. 36 (2022) 304–317, https://doi.org/10.1080/01691864.2021.2023629.
[110] Y. Qian, T. Tan, H. Hu, Q. Liu, Noise robust speech recognition on aurora4 by humans and machines, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 5604–5608, https://doi.org/10.1109/ICASSP.2018.8462629.
[111] S. Park, C.C. Menassa, V.R. Kamat, Joint BERT model for intent classification and slot filling analysis of natural language instructions in co-robotic field construction work, in: Proceedings of the 2023 ASCE International Conference on Computing in Civil Engineering, June 25–28, Corvallis, Oregon, 2024, pp. 453–460, https://doi.org/10.1061/9780784485224.