ITcon
www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

LEVERAGING NATURAL LANGUAGE PROCESSING FOR AUTOMATED INFORMATION INQUIRY FROM BUILDING INFORMATION MODELS

SUBMITTED: February 2023
REVISED: March 2023
PUBLISHED: April 2023
EDITOR: Robert Amor
DOI: 10.36680/j.itcon.2023.013

Armin Nabavi
Graduate Student, Department of Civil Engineering,
K.N. Toosi University of Technology, No. 1346, Valiasr Street, Mirdamad Intersection, Tehran, Iran
E-Mail: armin.nabavi@email.kntu.ac.ir

Issa Ramaji
Associate Professor, School of Engineering, Computing, and Construction Management,
Roger Williams University, One Old Ferry Road, Bristol, RI 02809
E-Mail: iramaj@rwu.edu

Naimeh Sadeghi
Assistant Professor, Department of Civil Engineering,
K.N. Toosi University of Technology, No. 1346, Valiasr Street, Mirdamad Intersection, Tehran, Iran
E-Mail: sadeghi@kntu.ac.ir

Anne Anderson
Associate Professor, School of Engineering, Computing, and Construction Management,
Roger Williams University, One Old Ferry Road, Bristol, RI 02809
E-Mail: akanderson@rwu.edu

SUMMARY: *Building Information Modeling (BIM) is a trending technology in the building industry that can increase efficiency throughout construction. Various practical information can be obtained from BIM models during the project life cycle. However, accessing this information could be tedious and time-consuming for non-technical users, who might have limited or no knowledge of working with BIM software. Automating the information inquiry process can potentially address this need. This research proposes an Artificial Intelligence-based framework to facilitate accessing information in BIM models. First, the framework uses a support vector machine (SVM) algorithm to determine the user's question type. Simultaneously, it employs natural language processing (NLP) for syntactic analysis to find the main keywords of the user's question. Then it utilizes an ontology database such as IfcOWL and an NLP method (latent semantic analysis (LSA)) for a semantic understanding of the question. The keywords are expanded through the semantic relationship in the ontologies, and eventually, a final query is formed based on keywords and their expanded concepts. A Navisworks API is developed that employs the identified question type and its parameters to extract the results from BIM and display them to the users. The proposed platform also includes a speech recognition module for a more user-friendly interface. The results show that the speed of answering the questions on the platform is up to 5 times faster than the manual use by experts while maintaining high accuracy.*

KEYWORDS: *Building Information Modeling (BIM), Natural Language Processing (NLP), Ontology, Support Vector Machine (SVM), Question Answering platform*

REFERENCE: *Armin Nabavi, Issa Ramaji, Naimeh Sadeghi, Anne Anderson (2023). Leveraging Natural Language Processing for Automated Information Inquiry from Building Information Models. Journal of Information Technology in Construction (ITcon), Vol. 28, pg. 266-285, DOI: 10.36680/j.itcon.2023.013*

1. INTRODUCTION

The AEC (Architecture, Engineering, and Construction) industry is still in the initial stages of technology adoption, which presents new challenges and makes it necessary to extend techniques and technologies that facilitate interactions in all parts of the industry (Alizadehsalehi et al., 2020). Building information modeling (BIM) is becoming more widely used in the AEC industry, providing significant benefits such as managing life cycle data, making processes more efficient, and exhibiting geometrical data in an integrated environment (Azhar, 2011). With the emergence of BIM, construction processes have been considerably altered. Different types of data are stored in a BIM model that is practical in various phases of the project life cycle, such as geometrical data of components, scheduled tasks, and costs. Access to this data and information for the project participants and stakeholders can be tedious or time-consuming, especially for those without experience using BIM tools which is one of the main barriers to BIM adoption in the construction industry (Ahmed, 2018, Durdyev et al., 2021, Jamal et al., 2019).

Previous studies have been conducted to facilitate access to information in BIM models. In the area of BIM information retrieval, Zhang and Issa (Zhang and Issa, 2013) proposed an ontology for extracting a partial building information model from an original complete model based on IFC schema specifications. Gao et al. (Gao et al., 2015) proposed a prototype semantic search engine named BIMSeek for retrieving online BIM resources. Liu et al. (Liu et al., 2016) developed an approach to extract construction-oriented quantity take-off information, such as topological relationships among building objects, from a BIM design model. For querying BIM data, Abanda et al. (Abanda et al., 2017) developed an approach to improve cost estimation accuracies and quantity take-off (QTO) for cost estimation. Wu et al. (Wu et al., 2019) developed a retrieval engine using Natural Language Processing techniques and ontology to provide querying BIM object databases. Guo et al. (Guo et al., 2020) developed an automated SPARQL (SPARQL Protocol and RDF Query Language) query generation in which all query keywords can be obtained from the user's question for information extraction. Wang et al. (Wang et al., 2022) used Natural Language Processing (NLP) to create a query-answering system for BIM information extraction to build a virtual assistant for construction project team members. Shin et al. (Shin and Issa, 2021) developed a BIM automatic speech recognition (BIMASR) framework to answer voice-based questions utilizing NLP. Wang et al. (Wang et al., 2021) developed a multi-scale building information retrieval scheme from BIM models employing NLP and International Framework for Dictionaries (IFD).

On the one hand, despite the BIM advancements made in recent years, retrieving information from four-dimensional (4D) BIM models during construction still presents significant challenges. 4D BIM is an integrated information model that combines 3D BIM models with construction schedules for facilities. The use of 4D BIM can greatly improve construction progress, planning, and scheduling (Crowther and Ajayi, 2021). A well-planned schedule enables the project team to monitor the progress of the project and determine whether the work is progressing as planned (Doukari et al., 2022). During the construction phase, stakeholders of a project often require access to schedule information in order to control and manage the project effectively (Tserng et al., 2014). However, not all stakeholders may have expertise in BIM. Therefore, a technical user should always be available to provide the necessary information to the project manager, which requires additional collaboration (Wijayakumar and Jayasena, 2013). Automating the process of accessing schedule information is a solution to this issue. To the authors' knowledge, previous studies did not focus on information retrieval during the construction process using 4D BIM. On the other hand, 4D models include schedule plans connected to building elements and quantities in the model. This information is beneficial for project control and management during the construction phase. Therefore, a framework to easily extract managerial information from 4D BIM models can encourage the use of BIM for construction managers who are not necessarily BIM experts.

Existing approaches for automating information inquiries for BIM models mostly rely on keyword-based searches. However, such searches are not capable of considering the semantic complexities and syntax variations of a user's question. This can be particularly challenging for non-technical users who may not be familiar with the specific terms used in a BIM model. To address this issue, it is necessary to develop an information extraction platform that can semantically understand the user's question. One approach to achieving this is to use Natural Language Processing (NLP) techniques to account for syntax variations. By combining these techniques with open-BIM solutions, it is potentially possible to better understand the semantics of a user's question. Ultimately, the three topics of BIM, semantic understanding, and NLP are closely interconnected, and leveraging NLP to transfer natural language semantics into machine-understandable ontologies may be an effective way to facilitate interoperability between BIM systems. (Locatelli et al., 2021).

This study seeks to address this gap using an Artificial Intelligence (AI)-based framework that can answer essential questions related to 4D BIM use cases in which geometry models are integrated with construction schedules. The objective is to facilitate access to BIM information for non-technical users. Also, this study aims to increase the answering speed of the inquiry process for Quantity Take-Off (QTO) and a manual search in BIM.

2. RELATED WORK

2.1 Natural language processing

Natural language processing (NLP) is an approach for computers to analyze, understand and manipulate human natural language text or speech (Chowdhury, 2003). NLP has several applications, including information retrieval and extraction, question answering, machine translation, summarization, and dialogue systems. NLP processes include semantic analysis, tokenization, named entity recognition, stemming, lemmatizing, automated extraction, and natural language generation. (Maulud et al., 2021)

NLP has been applied to various areas of construction management. For example, Tixier et al. (Tixier et al., 2016) employed an NLP rule-based approach to automatically scan and analyze unstructured injury reports for manual content analysis. Zou et al. (Zou et al., 2017) applied NLP and query expansion to retrieve and return construction accident data from an existing database to avoid similar risks in new situations. They used NLP techniques such as tokenizing, removing stop words, and lemmatizing to process the textual information for risk case contents. Zhang et al. (Zhang et al., 2019) utilized NLP safety and risk management techniques to analyze construction accident reports to prevent similar accidents. Data from these reports were parsed using a part-of-speech (POS) tagger and rule-based chunker and then classified with related techniques. Baker et al. (Baker et al.) used NLP Machine Learning to structure text data collected after on-site safety failures, including near misses and incidents. This structured data allows systematic analysis of these data to improve construction site practices and reduce safety incidents. Kim and Chi (Kim and Chi, 2019) used NLP to develop a knowledge management system for construction accident cases. The system can retrieve appropriate cases according to user intentions and automatically analyze tacit knowledge from construction accident cases. The user's query is expanded using a construction accident case thesaurus and then ranked in this system. For regulatory compliance checking, Zhang and El-Gohary (Zhang and El-Gohary, 2012) used a hybrid syntactic (syntax/grammar-related) and semantic (meaning/context-related) NLP approach to extract information automatically from construction regulatory documents. They applied two NLP approaches, phrase-structure grammar and dependency grammar, to reduce the number of matching patterns. In (Zhang and El-Gohary, 2015), the authors used a rule-based semantic NLP method for compliance checking. Information was extracted from construction regulatory documents and transformed into logic clauses with automated information transformation (ITR) methodology. Moon et al. (Moon et al., 2022) developed a computerized framework for reviewing construction specifications by analyzing the different semantic properties using natural language processing techniques. They developed a semantic thesaurus for construction terms including 208 word-replacement rules, then developed a named entity recognition model, identifying the required keywords from given provisions.

Applying NLP to building information modeling, Wu et al. (Wu et al., 2019) developed an intelligent retrieval engine utilizing NLP techniques and ontology to provide querying BIM object databases. Jung and Lee (Jung and Lee, 2019) proposed NLP and unsupervised learning methods to classify BIM case studies. Xie et al. (Xie et al., 2019) proposed a framework that matches real-world facilities to BIM data using NLP to address the information collaboration issue of real-world facilities and BIM models. BIM, semantics, and NLP are all interconnected in these studies, where natural language is translated into a format that machines and computers can understand, such as ontologies, that ultimately support interoperability between BIM systems (Locatelli et al., 2021). The combination of NLP, a semantic technology, and BIM, enriched with semantic information, has the potential to steer the industry toward more efficient processes through digitalization (Pan et al., 2004). Our study contributes to this effort through the development of a platform that leverages NLP to facilitate access to 4D BIM information.

2.2 Ontology

In computer science and information science, the most frequently cited definition of ontology originates from Thomas R. Gruber: ontology is "an explicit specification of a shared conceptualization." (Gruber, 1993).

Conceptualization is about a view of the world, and the specification indicates an official representation. An ontology comprises concepts (or classes), properties, relationships, constraints, axioms, and instances (Park et al., 2013). Ontologies can be roughly divided into two categories: *general ontology* and *domain ontology*. Domain ontology is a hierarchical description of essential concepts in a domain and contains properties for each concept (Lukasiewicz and Straccia, 2008). Concepts in a general ontology are related to the whole world, but domain ontology contains a specification of each domain conceptualization. Some general ontologies like Wordnet have many concepts that may lead to inaccurate descriptions if used in the AEC domain. On the other hand, domain ontology obtains more accurate results for domain-specific retrieval. Domain ontology is crucial for improving domain-specific information retrieval (Gruber, 1993).

Ontology has different forms of representation, such as RDF (Resource description framework), a well-known way of representing data and knowledge, and OWL (Ontology Web Language). OWL, a syntax extension of RDF proposed by W3C as the ontology language for the semantic web, is used for the ontology language. There are different methods, like TOVE (Toronto Virtual Enterprise), SENSUS, Seven-Step, etc., for developing a domain ontology. Among these methods, the Seven-Step, developed by the School of Medicine at Stanford University, is the most widely used. It adapts Gruber's five main principles of ontology construction (Gruber, 1993).

Ontologies are being used in several areas of construction research. Zhang and Issa (Zhang and Issa, 2013) proposed ABox and TBox ontology for extracting a partial building information model from an original complete model based on IFC schema specifications. Zhong et al. (Zhong et al., 2015) proposed a novel ontological and semantic mechanism for reusing plans and their automatic verification in construction. Gao et al. (Gao et al., 2015) proposed an ontology for retrieving online BIM resources. The hierarchical and enumeration relationships of the ontology were utilized for retrieving BIM documents. Ding et al. (Ding et al., 2016) developed an ontology-based construction risk knowledge management method in building information modeling. Liu et al. (Liu et al., 2016) developed an ontology-based semantic approach to extract construction-oriented quantity take-off information from a BIM design model to serve the application needs of participants in the construction field. Abanda et al. (Abanda et al., 2017) developed an ontology based on New Rules of Measurement (NRM) to improve accuracies in cost estimation. Concepts relevant and understandable to professionals in the UK were modeled, and rules were embedded for this ontology. Niknam and Karshenas (Niknam and Karshenas, 2017) developed a shared ontology using RDF/OWL language for the semantic representation of BIM data to facilitate sharing of building information in BIM models. Jiang et al. (Jiang et al., 2018) used BIM and ontology to facilitate the process of green building evaluation. They used BIM data required for green buildings evaluation and created an ontology with OWL description logic (DL) utilizing related terms in the evaluation standards. The studies discussed above show that utilizing ontologies can improve semantic reasoning and understanding, facilitate sharing information, and be effective and practical for information retrieval and extraction. This research leverages the semantic understanding of sentences from ontology.

2.3 Query classification technique

The text classification process has six main parts: a data set, text preprocessing, feature extraction, dimensionality reduction, a classification technique (a classifier), and performance evaluation (Kadhim, 2019). The most critical phase of text classification is to find a text classifier (Kowsari et al., 2019). There are different text classification methods, such as logistic regression, Naïve Bayes, k-nearest neighbour (KNN), support vector machine (SVM), which is broadly used as a classification method, and tree-based classification algorithms such as decision trees and random forests. Deep learning (DL) models such as Deep Neural Networks (DNN), Recurrent Neural Networks (RNN), and Convolutional Neural Networks (CNN) can also be used for text classification (Kowsari et al., 2019). However, they have some disadvantages for use in text classification tasks. One limitation is their "black box" nature, resulting in outcomes that are not readily understandable (Shrikumar et al., 2017). Furthermore, DL methods need much more data than traditional machine learning algorithms which means they cannot be applied for text classification when the dataset is small. SVM and Naïve Bayes, on the other hand, are two of the most commonly used supervised machine learning techniques (Lampinen and McClelland, 2017). SVM is less susceptible to overfitting problems (especially for a text data set) and can model non-linear decision boundaries, but lacks transparency in results due to a high number of dimensions. Naïve Bayes (NB) is easy to implement and works well with text data, but has a strong – and not always accurate – assumption about the shape of the data distribution (Kowsari et al., 2019). In 2015, Desai (Desai, 2015) compared these classification approaches and found SVM's accuracy was 90.21%, while NB's accuracy was only 79.83%. An evaluation was also done in

(Kadhim, 2019) using various methods, including SVM, Naïve Bayes, and KNN. Results indicated that SVM was the most accurate text classifier among them. Another study (Goh and Ubeynarayana, 2017) classified construction accident narratives by applying six text classification techniques, including SVM, Naïve Bayes, and KNN. Results showed that SVM had better performance in classification than the other techniques. Due to its common use and higher accuracy, the authors chose to use SVM for the platform developed in this research.

SVM has been used successfully in several construction-based studies. Dimitrov and Golparvar-Fard (Dimitrov and Golparvar-Fard, 2014) proposed a new vision-based method for material classification from single images taken from an unknown viewpoint. SVM was used to classify the material appearance and semantically label each generated 3D element. Salama and El-Gohary (Salama and El-Gohary, 2016) used SVM to classify various construction documents automatically (e.g., contract clauses) into predefined categories (environmental, safety and health, etc.) for automated regulatory and contractual compliance checking. Zhou et al. (Zhou et al., 2017) utilized the SVM to determine the safety risks that can materialize during the construction of deep pit foundations in subway infrastructure projects. Paudel et al. (Paudel et al., 2017) proposed an artificial intelligence (AI) model to predict the energy consumption of low-energy buildings (LEB). SVM was used to determine the weight of selected climatic conditions and their past days utilizing daily average energy load and suitable wavelet coefficients of climatic conditions and their past days. Koo et al. (Koo et al., 2019) used SVM to check the semantic integrity of mappings between BIM elements and IFC classes. SVM was used to classify elements based on eight IFC classes and then distinguish between the component subtypes within individual IFC classes. The above-mentioned studies represent various applications of support vector machines in the construction industry and text classification. This study builds on previous research by applying SVM algorithms for text classification in the construction industry, using SVM to predict the type of question a user enters into the platform.

2.4 BIM information extraction

Multiple query languages exist for extracting BIM data and information, and, in recent years, accessing data has been facilitated by improvements in query languages. For example, Gao et al. (Gao et al., 2015) developed a prototype semantic search engine, BIMSeek, to retrieve online BIM resources. The input query is expanded with domain and general ontology using the vector space model (VSM), and BIM document resources are ranked and returned to the user. Wu et al. (Wu et al., 2019) developed an intelligent retrieval engine to provide querying BIM object databases. The natural language query is understood syntactically and semantically using NLP and a domain ontology, then the target keywords of the query are mapped to the BIM family database and ranked with similarity value, and objects are returned to the user. Guo et al. (Guo et al., 2020) developed an automatic SPARQL (SPARQL Protocol and RDF Query Language) query generation. Users' requirements are matched with ifcOWL ontology concepts or instances and generate the user-desired SPARQL query. Kang and Hong (Kang and Hong, 2015) proposed a BIM/GIS-based information Extract, Transform, and Load (BG-ETL) architecture that separates geometrical information from that related to the relevant properties. The targeted property is extracted and transformed from BIM with facility management to GIS. In quantity take-off extraction research, Lee et al. (Lee et al., 2014) proposed an ontology-based system for building cost estimation to find tiling work items from work conditions extracted from BIM models. Liu et al. (Liu et al., 2016) developed an ontology-based semantic approach to extract construction-oriented quantity take-off information from a BIM design model. Users semantically query the BIM design model using a domain vocabulary, the SPARQL query. Khosakitchalert et al. (Khosakitchalert et al., 2019) proposed a method for compound elements, such as walls and floors, to improve the extracted quantities' accuracy because they might be incomplete or incorrect in the BIM model.

3. METHODOLOGY

In this research, a vocal assistant using NLP and SVM was developed for non-technical users to query information from a 4D building information model. Developing the platform was a multi-step process described in the following subsections, and the overall platform framework is presented in Fig. 1.

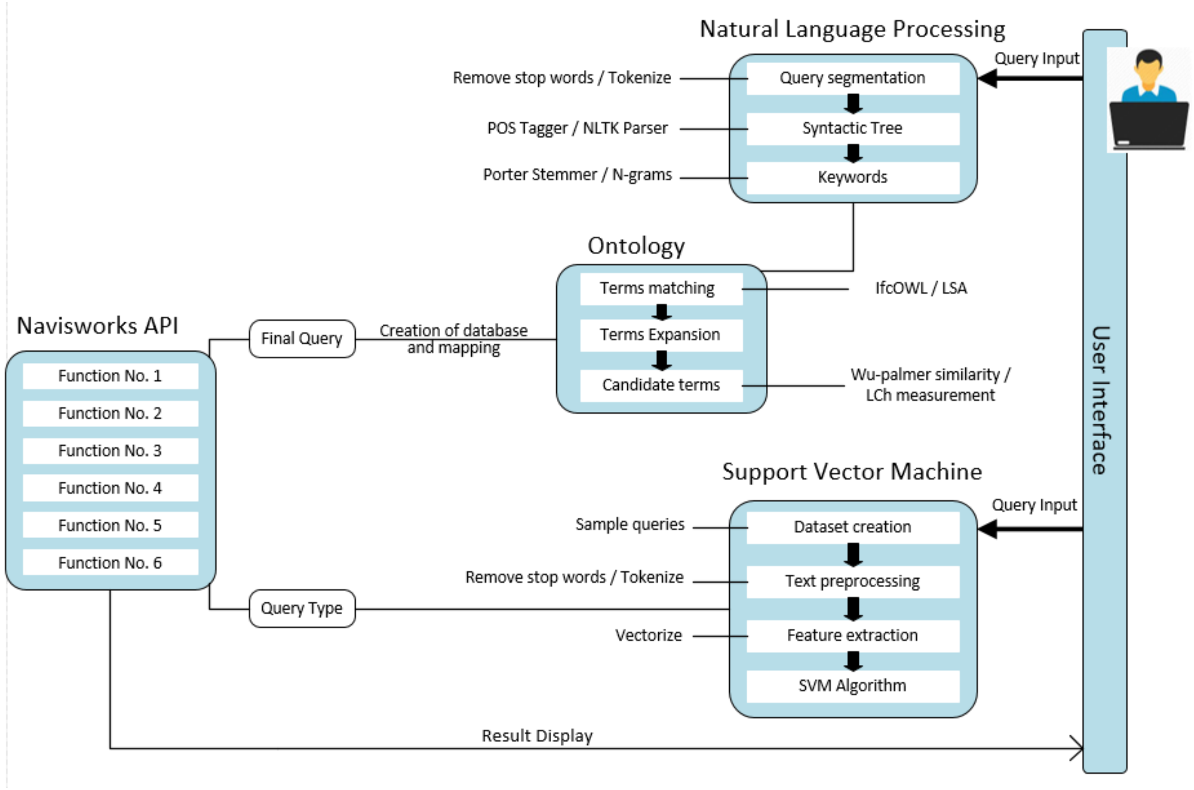


FIG. 1: An overview of the framework structure

3.1 Finding questions and determining question types

3.1.1 Finding common questions through a questionnaire

Different kinds of information can be extracted from a BIM model, and various questions can be asked from it. Therefore, the first step in this research was to identify questions commonly asked during the construction phase. The authors created a survey with a set of twelve questions and asked participants to rate them based on their level of importance. Participants were also able to add questions if they did not see their highest-priority questions in the set provided. The survey was emailed to 200 people involved in the construction industry in Iran. 44 out of 200 completed the survey for a response rate of 22 per cent. 73 per cent of the 44 participants were contractors or subcontractors, and the rest were consultants or owner representatives. 88 per cent of them were not an expert in BIM.

Participants were asked to evaluate the twelve questions using four parameters: 1) significance of the question, 2) difficulty and time-consumption of the answer search, 3) how many times the question may be repeated in the life cycle of the project and, 4) which group needs this information the most – expert or non-expert BIM users?

Since the primary purpose of this research is to facilitate accessing BIM model information for non-expert users in BIM, parameter number 4 was assigned more weight than the others. The weight of the three other parameters was equal. Four primary categories of BIM information are quantification, timeline, task, and location. By combining these categories, the researchers created two question types. The first type was the quantification of material in a location in the model (for instance, the volume of concrete on the first and second floor). The second type was the quantification of 4D BIM information (for instance, the volume of concrete needed between November 1 and December 1 or the time progress percentage of the project until November 1). After evaluating the survey results, the top six questions having the highest weights were chosen for proof of concept of the voice assistant. The first question was from the first type (quantification of material in a location in the model), which was the highest-ranked question in the questionnaire. The other five questions were from the second type, which was the quantification of 4D BIM information. The questions are listed in Table 1.

TABLE 1: Predefined query types

Label	Question type
1	Quantity of material (of an object) in a specific location
2	Quantity of material (of an object) needed for a particular task
3	Quantity of material (of an object) required in a specific timeline
4	A list of tasks that must be finished within a particular timeline
5	Delayed tasks of the project (if any exist)
6	The percentage of time progress of the project

3.1.2 Determining question types through an SVM classification model

Since the six questions developed for this study were different and each required specific information from the BIM model, an exclusive function for each question type was written to search for an answer. The platform must determine the user's question type so it can activate the proper function to answer the question. To accomplish this, text classification techniques were applied.

For the first step in text classification, an Excel file with a list of example questions was each labeled with a query type to create a dataset. Question types and their labels are shown in Table 1. For step two, NLP techniques (e.g. stemming and removing stop words), were applied to remove blank spaces and special characters from the questions. These NLP techniques are described in the next section of this paper. Several methods exist for the subsequent steps (feature extraction and feature weighting), such as TF-IDF, Bag of Words (BOW), and Vectorize. Most of them consider the maximum term frequency for each text and weigh them in various ways. For evaluating these techniques, 80% of the dataset was used for training, and 20% acted as the test set. TF-IDF received the highest score in question type prediction and was applied for training among the mentioned techniques. In step four, dimensionality reduction (DR) methods are mostly used for documents because the text classifier must be able to deal with a large dataset, so it is vital to decrease the calculation budget. Since the dataset in this research was small and contained specific and short questions, this step was eliminated to avoid errors in determining the user's query type.

Next, an SVM algorithm was applied to train the dataset and predict the user's question type. There are several parameters for an SVM algorithm. The first parameter is Kernel which has a few types. The first type is the linear kernel which works fine if the dataset is linearly separable. The second one is a non-linear kernel such as the Radial Basis Function (RBF) kernel for non-linear problems. RBF works appropriately with smaller data, but it is a universal kernel, and using it on smaller datasets might increase the chances of overfitting. Therefore, "linear" is used as a kernel for the algorithm. Another parameter is the regularization parameter which is mostly termed "C." It tells the SVM optimization how much is wanted to avoid misclassifying each training example. The "C" value in this research is 1 to avoid overfitting (being excellent at classifying training data but very bad at classifying unseen testing data). The final step was to validate the trained classifier model. The validation was done for the SVM model with different types of feature extraction methods. The highest score in question type prediction was the model with TF-IDF as the feature extraction method. The results of validation with feature extraction methods are represented in Table 2. The whole process of text classification is displayed in Fig. 2.

TABLE 2: Validation of the SVM model with various feature extraction methods

Feature Extraction method	TF-IDF			Vectorizer			BOW		
	accuracy	macro avg	weighted avg	accuracy	macro avg	weighted avg	accuracy	macro avg	weighted avg
precision		0.91	0.92		0.97	0.97		0.8	0.84
recall		0.92	92		0.96	0.96		0.83	0.84
f1-score	0.9	0.89	0.9	0.96	0.97	0.96	0.8	0.81	0.84

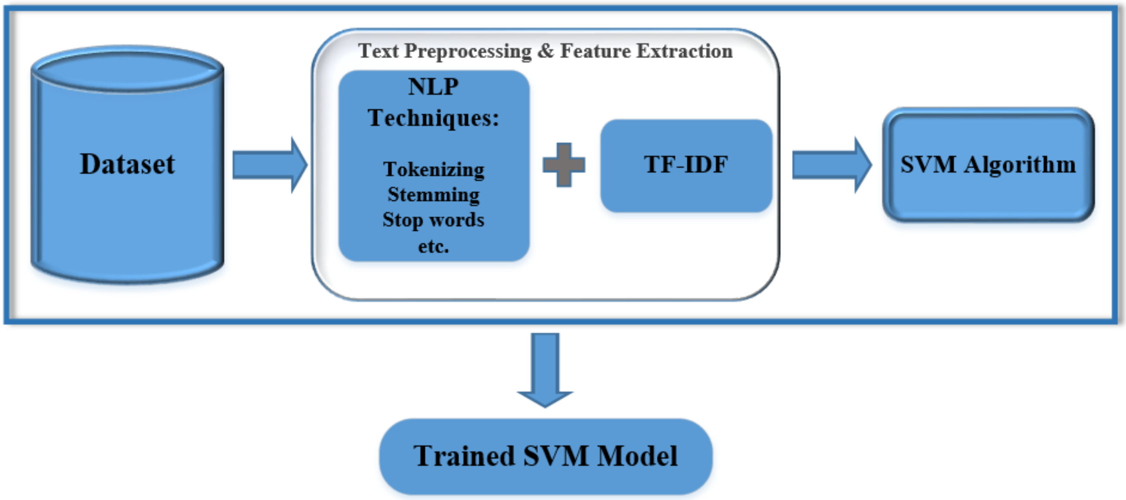


FIG. 2: The query classification model

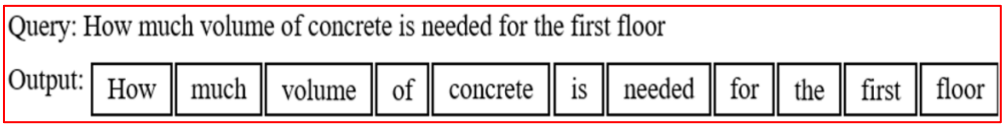
After the algorithm predicts the query type, the user will be asked whether the determined question type is correct or incorrect. If it is incorrect, the user is prompted to choose the correct type. Each question that a user asks, with its correct question type, is automatically added to the dataset. As the number of asked questions from the platform increases over time, the dataset becomes more robust, resulting in higher accuracy in determining question types.

3.2 Keyword extraction

Next, the keywords of the user's question are extracted. NLP is employed to capture and analyze a user's natural language question. NLP is used for the syntactic understanding of the user. First, stop words such as “a”, “an”, “in”, or “so” are removed from the query. Then, using Tokenization, the entire text is split into separate words. Using part-of-speech tagging, every word is tagged by its part of speech (POS), such as noun, verb, adjective, etc. Through a parser, a syntactic tree of the query is created, showing the relationship between words. The user's purpose, which is considered to lie in noun phrases, is extracted and saved. Before the expansion and further steps, the stem of each keyword is found.

3.2.1 Tokenization

Tokenization is the process of splitting a text into separate words called tokens and simultaneously removing punctuation. Since POS tagging works with tokens, it is needed to tokenize the query. A graphical sample of tokenization is shown in Fig. 3.



Query: How much volume of concrete is needed for the first floor

Output:	How	much	volume	of	concrete	is	needed	for	the	first	floor

FIG. 3: A sample of tokenization

3.2.2 Part-of-speech tagging and parsing

In this step, using a part-of-speech tagger, the words are classified and labeled into their parts of speech. After POS tagging, a chunker is used to parse the text by extracting phrases, such as noun phrases (NP), to preserve the purpose of the user’s text. Tagging and parsing results in a representative syntactic tree structure of the query. An example of a parse tree is shown in Fig. 4.

3.2.3 Stemming

Words are often derivations of other words. For information extraction, it is essential to use the root form of the word. For instance, the root form of the words "program," "programs," "programming," and "programmer" is "program." Stemming is the process of removing morphological affixes from words and leaving the stem of the word. The keywords from the query will be mapped with the lists of objects, materials, etc., of the BIM model.

Since the words in these lists are all in the root form, the keywords must also be in their root shape. "Porter stemmer" was used for this task.

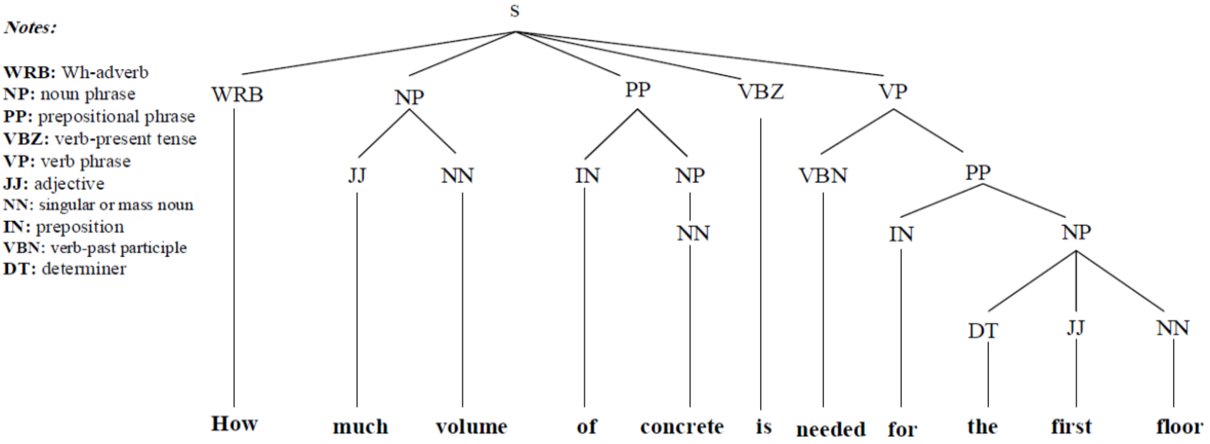


FIG. 4: A sample of a syntactic tree

3.2.4 N-grams

N-gram is a set of co-occurring or continuous sequences of n items in a text. To illustrate this process, the query "How much volume of concrete is needed in the task of constructing the shear wall story 2?" will be used as an example. To tokenize and POS tag this query, the task's name (constructing the shear wall story 2) would not entirely be known as a noun phrase. For instance, "constructing" is tagged as a single noun, and "the shear wall story 2" is tagged as a noun phrase. When they must be searched in the task list of the BIM model, the platform will search "constructing" and "the shear wall story 2" separately and won't find a match for them. Therefore, no tasks will be found from this query which is incorrect. Additionally, the task's names mostly rely on the BIM experts who enter these names, and sometimes they are not in a standard form, so this mistake may be repeated over and over again by the platform. To address this problem, N-grams were used in the platform. If N equals 2, the query is partitioned like: ″How much″, ″much volume″, ″volume of″, ″of concrete″, ″concrete is″, and so on. Then, each of these parts will separately be searched in a task list of the BIM model. For this example, if N equals 6, one of the created parts will be "constructing the shear wall story 2" and be searched in the list of tasks of the model and saved as a task name that the user mentioned. We applied N-grams from N=2 to N=7 to ensure all possible noun phrases were covered.

3.2.5 Implementation

For implementing the NLP and analyzing the user's question type, the natural language toolkit (NLTK) (Bird et al., 2009) was used. The NLTK is a leading platform for building Python programs to work with human language data. This platform was used for syntactic analysis of the user's query for two reasons. First, it provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet. Second, it has myriad text processing libraries such as classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

3.3 Keyword expansion

Understanding the purpose of the user is a critical part of the question-answering platform. Keywords of the query alone cannot achieve the user's purpose and are not ready to be directly searched because a keyword may be just one form of a different term representing the same meaning. There are additional terms with the same meaning and various forms of a term, such as an abbreviation. Therefore, corresponding synonyms and other forms of keywords must be collected and used to search in the database. This process is called query expansion. To improve the quality of the question-answering platform, we used query expansion methods using Latent Semantic Analysis (LSA) and domain ontology. For this process, three steps were required: 1) providing LSA and a domain ontology to obtain corresponding similar terms, 2) providing a query expansion method to measure the semantic similarity of obtained terms for a more accurate search, and 3) searching the candidate terms in the database to find what the user wants in the BIM model.

LSA and ontology in this research were used to add semantic terms to the user's question. Since this study's questions were about the construction phase and BIM in the construction industry, the domain of the ontology must be the objects, materials, etc. In this regard, various concepts in the construction industry was collected to form the ontology. The second step was to select and reuse existing resources. Several semantic resources have been developed, such as ISO 12006-2, Uniclass, OmniClass, and Industrial Foundation Classes (IFC). In this research, IFC specification, a semantic foundation, was used for ontology establishment due to its wide range of definitions of concepts. Instead of creating a new ontology, this research used ifcOWL, which provides a web ontology language representation of the IFC schema. IfcOWL ontology has the same status as the EXPRESS and XSD schemas of IFC.

OWLready2, a package for ontology-oriented programming in Python, provides access to IfcOWL, so semantic concepts related to keywords of the user's query can be easily extracted. The users might not ask their questions using the standardized concepts in the IFC specification and in the BIM model – for example, the user may use the word "girder" in their question, while another word with a similar meaning (e.g., "beam") exists in the BIM model. Therefore, we need to expand the user's query to understand their purpose for an accurate and semantic search. The expansion method has two parts: 1) using a method by which the query's keywords are expanded with their synonyms, and 2) using IfcOWL, semantic concepts are added to the user's query. For the first step, we used LSA to find synonyms for each keyword.

LSA is a method that represents the meaning of words by statistical computations applied to a large corpus of text (Landauer et al., 1998). LSA makes latent semantic dimensions that represent the semantic content in the text as a set of vectors (Evangelopoulos et al., 2012). LSA reduces the corpus to a series of eigenvectors and utilizes the subsequent steps. First, a document-term matrix (DTM) containing word counts per paragraph is produced. LSA then leverages a mathematical method known as singular value decomposition (SVD) to reduce this matrix. Through SVD, eigenvectors for each word are approximated and appraised to specify how close/distal the words in their multidimensional vector space are (Dumais, 2004).

LSA was subsequently used for several tasks such as synonym detection (Landauer et al., 1998), document clustering (Song and Park, 2007), vocabulary acquisition simulation (Landauer and Dumais, 1997), Word Sense Discrimination (Pino and Eskenazi, 2009), etc. In this research, LSA was used to provide the most relevant and semantic synonyms for the words in the user's question. Using the earlier example, if one of the keywords from the user's question is "girder," its synonyms such as "beam", "joist", "rafter", "balk," Probe, "bar," "purlin," and "ridge" are added to this word in this step.

The user's query is related to the construction phase, and LSA alone cannot expand the question and prepare it for searching in the database of the BIM model. Therefore, we used IfcOWL to expand the query by utilizing the relationships between concepts in IfcOWL. First, using OWLready, access to the ontology is provided. Then, the expanded concepts will be replaced with their counterpart synonym that exists in the ontology. For example, "girder" with synonyms such as "beam," "ridge," "joist," "rafter," "probe," and "purlin" will be replaced with the standard concept "beam" defined in the ontology. Next, semantically adjacent concepts in the ontology that matches the keywords (e.g., "beam") are extracted from the ontology based on their semantic relatedness value named similarity value. As a result, the keyword "beam" and its semantically expanded concepts are ready to be searched in the databases of the model. However, there might be an overexpansion problem. A threshold is defined manually for similarity value to avoid overexpansion. If the similarity value of an expanded concept falls short of the threshold, that concept is uncorrelated with the user's purpose and is removed from the expanded concepts list. For measuring semantic similarity value, studies are mainly divided into corpus-based methods such as LSA and PMI (Pointwise Mutual Information) and knowledge-based methods such as Wu–Palmer measure (Wu and Palmer, 1994), Jiang–Conrath measure (Jiang and Conrath, 1997), Resnik measure (Resnik, 1995), Lin measure (Lin, 1998), Leacock–Chodorow measure (Leacock and Chodorow, 1998). Considering the ontology structure and the features, the Leacock–Chodorow measure (Leacock and Chodorow, 1998), based on path length, was found more effective for this research. The formula is:

$$R(C, C_i) = -\log\left(\left(Len(C, C_i)\right)/2Depth\right)$$

Eq. 1: Leacock–Chodorow semantic relationship measurement equation

C is the query concept in IfcOWL, C_i is the i-th concept in the neighbourhood of C, and Depth is the maximal tree depth of the ontology. The factor of 2 in the denominator is used to represent the possible maximum length between two concepts. $Len(\mathrm{C}, C_i)$ is the shortest path between these two concepts. So, the similarity value between two concepts won't be less than 0.

The relationship between two concepts varies in ontology. Relationships such as subclass, superclass, type enumeration, etc., represent the distance between two concepts. For instance, the shortest distance between two concepts happens when C_i is the subclass or superclass of C, in this case, the shortest path ($Len(\mathrm{C}, C_i)$) equals one, and the similarity value between them reaches maximum. On the other hand, the semantic similarity value between two concepts reaches its minimum when they have the maximum distance. It happens when they are in the leaf nodes, and their only common ancestor is the root node.

All semantic similarity values with the query expansion algorithm are obtained, and concepts whose value are below the predefined threshold are removed from the expansion list. Eventually, the final query – the standard concepts acquired from LSA and domain ontology and the expansion method, namely candidate terms – are utilized to map with the database.

3.4 Creation of database and mapping

In our question-answering platform, six-question types can be answered. Some of the question `, all of the BIM model's objects, materials, tasks, and layers must be extracted and saved in separate lists. Access to these lists is accomplished through the Navisworks application programming interface (API) with C# language, and then they are called in Python for mapping. Saving lists of every model takes approximately fifteen to thirty seconds which is considered time-consuming. For this reason, efforts were made to save the lists of a BIM model on the user's system, the first time the user utilizes the platform. If the model is changed, the user must click an "Update" button to update the lists.

After testing various BIM models, it became clear that some of the terms were shared in multiple lists; for example, the word "Wall" was in both the material list and object list, or the word "Level 1" was found in both the layer list and object list. Therefore, the next step was to automatically remove irrelevant terms from each list.

After constructing the database with lists of the BIM model, the candidate terms are searched separately in all lists. If any candidate term matches with any term within a list, it will be selected as input to the platform. For instance, the word "concrete" as a candidate term will be searched in all lists; once that is matched with the term "concrete" in the material list, this word will be selected as a material input for the platform. There is no limitation on the number of inputs; for example, the user can ask for material in as many objects as needed within one question.

3.5 Finding the answer

In this step, through the Navisworks API, functions are written separately for each question type. Based on the question type and the parameters, the related function is activated and searched for in the whole model. The answer will be displayed to the user if it exists; otherwise, the user will receive a message that there is nothing in the model related to the query.

The question "How much volume of concrete is needed for the basic wall on the first floor?" will be used as an example to illustrate this step. First, with the link between .Net and Python, the query is sent to Python to determine the query type and the user's purpose. The query type is recognized as type 1, which is related to the quantity of material in a location. Then, the inputs are determined as follows: "Concrete" as the material and "Basic wall" as the object, "First floor" as the layer, and "Volume" as the quantity. Next, these recognized inputs with the query type are called in the Navisworks API. Based on the question type, which is 1, function number one will be activated. This function takes quantity, material, object (if any exists), a location (if no location is found in the query, it is assumed the user wants the quantity of the material in the whole project), and a unit (e.g. cubic meter, cubic feet, cubic yards – if no unit is found in the query, the unit is assumed to be cubic feet). Then it searches the first floor for objects named Basic Wall, and the material concrete. Based on the quantity (volume in this example), all volumes of the materials are summed using the units requested. The final answer is then displayed to the user and spoken by a voice.

4. NAVISWORKS PLUG-IN

Because working with the Navisworks API requires expertise and involves manual processes, we propose the prototype question answering platform be developed as a plug-in for Navisworks. Using this plug-in, users can query their requirements through natural language questions by a voice from the platform in six types and receive answers quickly. Details of the plug-in development are discussed in this section.

4.1 Question answering platform for Navisworks

The Navisworks application programming interface (API) allows for external development in the Navisworks application. The plug-in development is based on the C# language in Visual Studio 2019, linked with Python for machine learning processes. There are four windows overall in the platform, and each window's functionality will be explored. The overview of the question-answering platform is shown in Fig. 5.

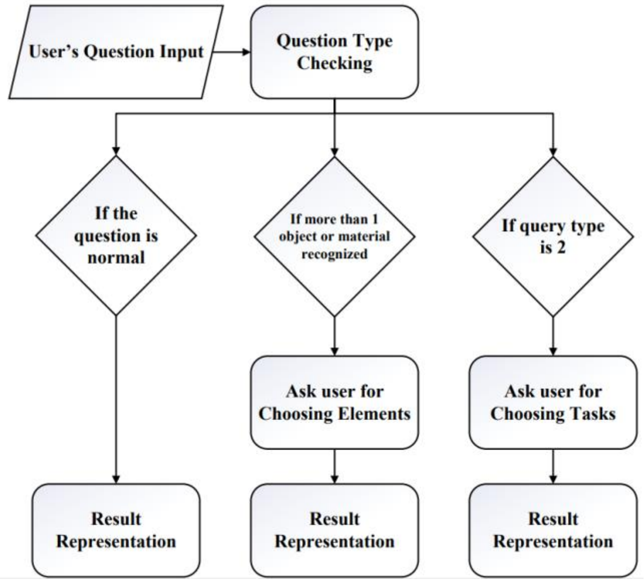

User's Question Input
Question Type Checking
If the question is normal
If more than 1 object or material recognized
If query type is 2
Ask user for Choosing Elements
Ask user for Choosing Tasks
Result Representation
Result Representation
Result Representation

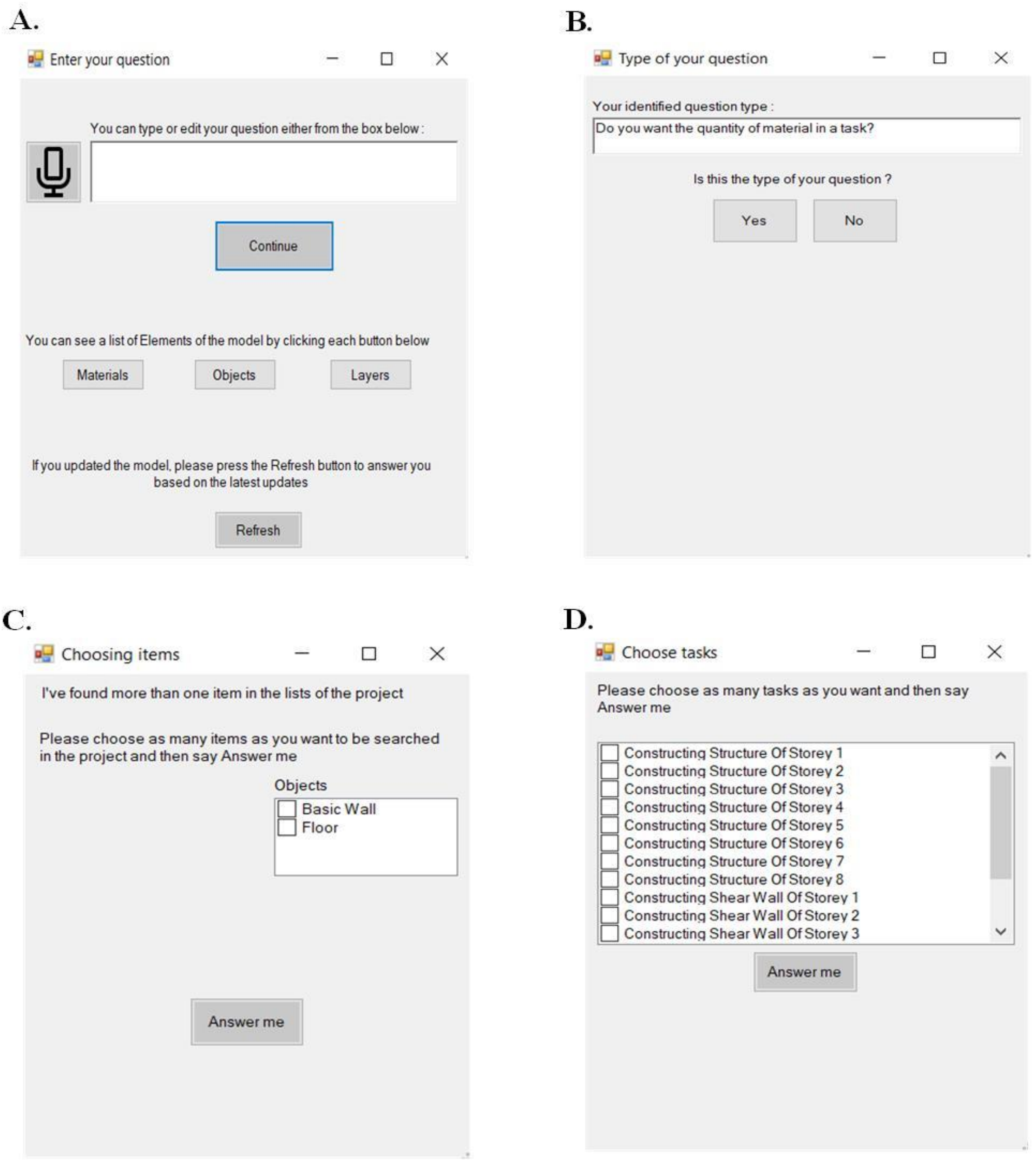
FIG. 5: Overview of the question-answering platform

The first window named "Enter your question" is displayed so the user can ask a question using voice or text. (Refer to Fig. 6-part A.) Google Web Speech API, a speech recognizer, recognizes the user's query and converts it to text. Speech recognition can turn BIM software from an export-oriented to a user-oriented environment. The user can also see lists of all model materials, objects, and levels if a specific item is needed and the user either doesn't remember the name or wants to see all items that exist in the project. In addition, a "Refresh" button is provided so that if the model is updated, the database will be updated as well by clicking this button. In this way, the answers will be based on the project's latest version. In the second window named "Type of your question" (Refer to Fig. 6-part B), the user is asked whether the recognized question type is correct or not. If it's correct, the user must say yes or click the "Yes" button. If the query type is wrong, the user must select the correct type from the list presented or say the correct type using voice. It will automatically understand what the user says and select the correct query type, then the answer will appear in the same window. There are two exceptions considered in the questions:

1) if more than one item is determined within the question (for example, three objects), the user will be asked to choose one or more items to be searched in the model, appearing in the third window named "Choosing items" (Refer to Fig. 6-part C), and
2) if the question type is number two (about searching material in tasks), the user will be shown the list of all tasks and asked to choose one or more tasks for searching in the second window.

The user can ask the platform to explore with the same recognized task(s) or request to see all of the project tasks. The first case will be answered in the same second window. For subsequent tasks, a new window called "Choose tasks" will be opened (Refer to Fig. 6-part D), and the user can say the name of the tasks to be searched. The platform will pick them and answer the question in this new window. The first exception is designed for a case where if the platform misinterprets an item from a user's query or if an additional item is mistakenly identified, that is not the user's purpose. The user will be asked beforehand to ensure the answer will provide what was requested. The second exception is added to the platform because many non-expert users don't know the exact name of tasks they want to query. The name of the tasks in the model is directly related to the BIM expert who entered them; therefore, these names can vary model by model. So, if the query type is determined as number two, the user is provided with a list of all project tasks to select as many as needed among them.

All details recognized from the user's question are displayed as a table in the window that answers the question based on the query type, recognized elements consist of materials, objects, layers, tasks, dates, quantity, and units determined from the question. To make the platform more user-friendly, the user can speak the button labels and item names or state the button's purpose instead of clicking on them. The platform is designed to understand the user's purpose in each step. The platform also speaks to the user in some steps, such as asking whether the recognized query type is right or wrong and providing the answer by voice. This work was accomplished through Microsoft Speech Recognition, inspired by similar platforms such as Google Assistant, Siri, and Alexa. To improve the answering process, if the user asks for the material of objects, the found objects will be highlighted in the model so the user can easily see them.



A.

Enter your question

You can type or edit your question either from the box below :

You can see a list of Elements of the model by clicking each button below

If you updated the model, please press the Refresh button to answer you based on the latest updates

B.

Type of your question

Your identified question type :

Do you want the quantity of material in a task?

Is this the type of your question ?

C.

Choosing items

I've found more than one item in the lists of the project

Please choose as many items as you want to be searched in the project and then say Answer me

Objects

☐ Basic Wall
☐ Floor

D.

Choose tasks

Please choose as many tasks as you want and then say Answer me

☐ Constructing Structure Of Storey 1
☐ Constructing Structure Of Storey 2
☐ Constructing Structure Of Storey 3
☐ Constructing Structure Of Storey 4
☐ Constructing Structure Of Storey 5
☐ Constructing Structure Of Storey 6
☐ Constructing Structure Of Storey 7
☐ Constructing Structure Of Storey 8
☐ Constructing Shear Wall Of Storey 1
☐ Constructing Shear Wall Of Storey 2
☐ Constructing Shear Wall Of Storey 3

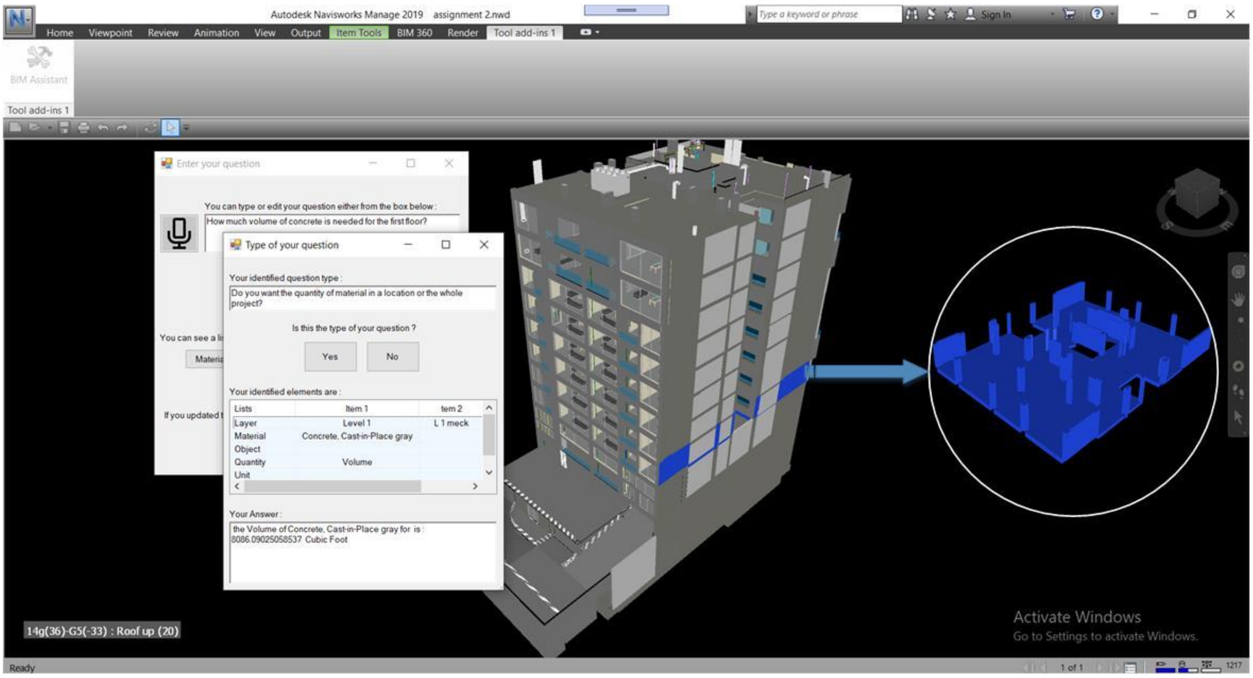
FIG. 6: Overview of windows

4.2 Experimental case

As shown in Fig. 7, the question-answering platform window is located at the top left of the Navisworks interface. After clicking on the plug-in button, the first window prompts the user for a question, for example, "How much volume of concrete is needed for the first floor?". After pressing the "Continue" button in the first window, the next window appears over the previous one and asks the user if the query type determined by the platform is correct; for example, it asks if the user wants the quantity of material in a specific location. If the user says yes or clicks the “Yes” button, the answer containing a total volume of concrete for the first floor is displayed. In addition, all objects containing concrete as the material in level 1 will be highlighted so the user can find them in the model (as shown in Fig. 7).

Additional functionalities are described in this next example. For the question "How much volume of concrete is needed in the task constructing the shear wall of story 2?", after verifying the type of question, the user is given a choice to see a list of all tasks with the ability to select more tasks if desired. If so, a list of all tasks in a new window will appear (Fig. 6-part D). After picking desired tasks and pressing the "Answer" button or requesting the answer using voice, the answer will be displayed.

As a final example, the question "How much volume of concrete is needed for the floor and the basic wall in the whole project?" is asked from the platform. After verifying the type of question, in a new window (Fig. 6-part C), the user is asked to select one or both items to be searched in the model between objects "Floor" and "Basic wall." This window will provide the answer after the user voices the request or clicks the "Answer" button.



You can type or edit your question either from the box below :

How much volume of concrete is needed for the first floor?

Your identified question type :

Do you want the quantity of material in a location or the whole project?

Is this the type of your question ?

Your identified elements are :

Lists	Item 1	tem 2
Layer	Level 1	L 1 meck
Material	Concrete, Cast-in-Place gray	
Object		
Quantity	Volume	
Unit		

Your Answer :

the Volume of Concrete, Cast-in-Place gray for is :
8086.09025058537 Cubic Foot

FIG. 7: Overview *of the interface*

5. EVALUATION

A standard BIM model was chosen to evaluate the platform, and ten participants – five BIM experts and five non-experts – were recruited to ask the platform some questions from each query type. Four principles were considered to assess the performance of this platform:

Query classification, in which the ratio of the number of times the platform recognized the type of question correctly to the number of the total questions is calculated.

Parameters recognition (purpose of the user), where the number of times users' needs were met in each question compared to a total number of questions, was calculated.

Answer checking to determine whether the platform could or could not return correct answers to users.

Response speed, comparing the time of finding an answer in Navisworks QTO versus the platform to find out which method is faster.

Using these principles, the platform was evaluated using an eight-story building model as an experimental case for users to ask questions. The model had standard names for most objects and materials and included a timeline with sixteen various tasks from January 1, 2021, to August 15, 2021. Building elements were also assigned to their related tasks. The participants were asked to query the platform in each query type.

First, a document that included the name of most essential objects and materials of the project, information about the project timeline, and materials included in every task was given to the users so they would be familiar with the project and ask their questions as if they were a member of the project. After reading the document, we requested each user ask two questions from every query type (twelve questions total from six query types). At the end of all interviews, the ten users had asked twenty questions from every query type.

After a user asked a natural language question from the platform, it first determined the query type and asked if the user if it was correct or not. This was used to evaluate principle number 1 (query classification). Then, the answer and a table containing every recognized parameter from the question were displayed so the user could see if the request was correctly identified from the question or not. The parameters shown in the table vary depending on the query type. For example, for question type two (quantity of material of an object in a task), parameters are material, object, task, unit, and quantity (volume, area, etc.). From the information in the table, principle number 2 (purpose of the user) is asked from the user and compared to the results. For principle number 3 (answer checking), users' questions were given to a BIM expert to find every answer through the "Quantification" part of the Navisworks application so the value of answers could be checked. Finally, to evaluate principle number 4 (response speed), the speed of answering in both methods (this platform and quantity take-off (QTO) of Navisworks) was measured through available records. Interviews with users, platform interactions, and response searching by the BIM expert were video-recorded for this purpose. For each question type, the average response speed of the two methods was calculated to illustrate how much faster the platform's speed was than the Navisworks QTO.

In addition to the factors mentioned above, to test the comprehensive performance of the platform, different questions for each query type were generated by changing the subject, predicate, object, and its modifiers. This allowed researchers to determine if the platform was able to recognize the question type and parameters, despite its various forms. A sample of the query generation method for query type 1 is shown in Fig. 8.

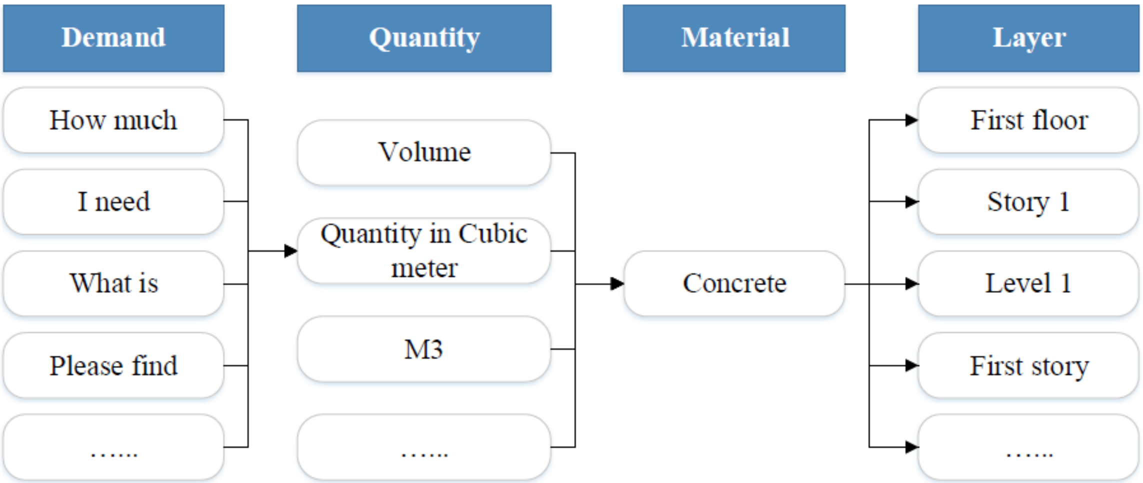


FIG. 8. A sample method of query generation

Post-interviews and data collection, researchers analyzed the data using the four principles described above. Each of the ten users asked two questions for each of the six query types for a total of 120 questions (20 for each query type). The results for each principle are as follows:

Query classification: 116 of the 120 question types were correctly recognized by the platform, resulting in 96.66% accuracy in the case of query classifying.

Parameters recognition (purpose of the user): most of the parameters found from users' queries were the exact purpose of them, and the platform considerably recognized their needs and found them from the model. Therefore, 91.66% of accuracy was also achieved in this case.

Answer checking: every answer found by the platform was equal to the answer that Navisworks QTO achieved. It should be noted that for some elements in the model, the value of properties was not defined. For instance, the area of "Travertine for Ramp" was not specified. This platform found and highlighted this element but returned zero as the value, which was reasonable. Therefore, for this principle, an accuracy of 100% was achieved.

Response speed: although this factor depends on the size of the BIM model and the number of tasks it has, for this study's eight-story building model, the results were as follows:

Query type 1 (quantity of a material or an object in a task in a level): the average speed of the platform for this question type was **2.89** times faster than manual QTO in Navisworks.

Query type 2 (quantity of a material or an object in a task in a task): this platform was **2.15** times faster than manual QTO in Navisworks QTO for this question type. Of course, this number is strongly related to the number of tasks the user mentions in the query. The more tasks in a query, the more efficient the platform will be than Navisworks QTO because increasing the number of tasks doesn't take too much time for the platform to calculate the answer, but in Navisworks QTO, it might take twice the time to find the answer. Most of the users mentioned only one task in their questions.

Query type 3 (quantity of a material or an object in a task in a timeline): for this question type, the speed of the platform was on average **4.28** times manual QTO in Navisworks, and that's because users primarily mentioned timelines that included more than one task, so it took more time to answer them through Navisworks QTO.

Query type 4 (list of tasks must be finished in a specific timeline): since users can answer their questions by seeing the dates of the timeline in Navisworks, the platform's speed almost equaled the rate of a BIM expert user to find the answer of this question type. But still, answering this question needs a BIM expert user, while the platform can efficiently respond to non-expert BIM users without any required skill.

Query type 5 (list of delayed tasks): for this question type, the platform worked significantly faster than that of a BIM expert, and the time it needed to find the answer was **3.28** times less than the time of a BIM expert.

Query type 6 (time progress percentage of the project): the platform had the best function for this question type in the matter of speed, answering **5.15** times faster than manual calculation.

An overview of the full results is presented in Table 3. To further test the platform, one sample question was chosen for each query type, and five alternate questions were generated from the primary question by changing the various parts of the sentence (subject, predicate, object, and its modifiers). Then, the five generated questions were asked to check the comprehensive performance of the platform. The platform could understand every question, return the same value as the answer for all of the various forms of the question, and meet all the principles mentioned above.

TABLE 3: Overview of the results in this method and Navisworks QTO and manual calculation

Question types	query classification (SVM algorithm)	understand the purpose of the user (NLP and Ontology)	Accuracy of answer	Response speed percentage (the platform compared to Navisworks QTO)
type 1	95%	85%	100%	289%
type 2	90%	75%	100%	215%
type 3	95%	90%	100%	428%
type 4	100%	100%	100%	97%
type 5	100%	100%	100%	328%
type 6	100%	100%	100%	515%

6. CONCLUSIONS AND FUTURE WORK

The BIM model often contains essential information needed by project stakeholders during the construction phase. Accessing this information is tedious and time-consuming for non-expert BIM users which could lead to resisting BIM adoption in construction projects. To address these issues, this study has introduced a question-answering platform that understands the vocal natural language question of the user and returns the answers through an automated information retrieval from 4D BIM. The proposed framework facilitates and accelerates accessing information in BIM models during the construction phase of the building project by automating the information query process. Two primary goals were to remove tedious and time-consuming manual searches, coordination, and communications for accessing BIM data, and to increase the answer speed of the inquiry process to Navisworks Quantity Take-Off (QTO) and manual searching. Furthermore, the platform includes a voice recognition module; thus, it works with both text and voice to create a more user-friendly environment for BIM tools.

The evaluation of the platform indicates that the voice assistant facilitates information retrieval from the BIM model, outperforming manual methods in five out of six questions investigated in the case study. The results illustrate that the platform can correctly recognize the type of users' questions. It can also understand the user's exact purpose and return the related elements or requirements from the model. Its accuracy and precision in answering are considerably high. Another goal of the platform was to display answers quickly. The speed was two to five times faster than manual QTO in Navisworks in five of six question types. The speed of answering for the remaining question type was equivalent to manual searching. One significant advantage of this platform is that anyone can ask their questions and achieve responses directly, regardless of the users' BIM skills. The platform enables non-technical users to benefit from BIM without intermediaries.

The platform can be extended for the automation of other types of information retrieval from BIM models. Additionally, enriching ontologies for a better understanding of the user's purpose will enhance the performance of the proposed voice assistant. Implementation of the platform on the web will increase the accessibility and versatility of the platform.

The platform has some limitations to be addressed in future studies: 1) IfcOWL, which is used as a domain ontology, does not have enough material and object terms, so it needs to be combined with other existing AEC ontologies such as Uniclass or OmniClass, 2) the user can't ask for two different quantities (such as Area and Volume) in one question, 3) grammar used for analyzing questions and finding noun phrases is not specialized for each question type. Using specific grammar for each question type can improve understanding of the user's purpose.

REFERENCES

ABANDA, F. H., KAMSU-FOGUEM, B. & TAH, J. 2017. BIM–New rules of measurement ontology for construction cost estimation. Engineering Science and Technology, an International Journal, 20, 443-459.

AHMED, S. 2018. Barriers to implementation of building information modeling (BIM) to the construction industry: a review. Journal of civil engineering and construction, 7, 107-113.

ALIZADEHSALEHI, S., HADAVI, A. & HUANG, J. C. 2020. From BIM to extended reality in AEC industry. Automation in Construction, 116.

AZHAR, S. 2011. Building information modeling (BIM): Trends, benefits, risks, and challenges for the AEC industry. Leadership and management in engineering, 11, 241-252.

BAKER, H., SMITH, S., MASTERTON, G. & HEWLETT, B. DATA-LED LEARNING: USING NATURAL LANGUAGE PROCESSING (NLP) AND MACHINE LEARNING TO LEARN FROM CONSTRUCTION SITE SAFETY FAILURES. Management, 356, 365.

BIRD, S., KLEIN, E. & LOPER, E. 2009. Natural language processing with Python: analyzing text with the natural language toolkit, " O'Reilly Media, Inc.".

CHOWDHURY, G. G. 2003. Natural language processing. Annual review of information science and technology, 37, 51-89.

CROWTHER, J. & AJAYI, S. O. 2021. Impacts of 4D BIM on construction project performance. International Journal of Construction Management, 21, 724-737.

DESAI, A. 2015. A review on knowledge discovery using text classification techniques in text mining. International Journal of Computer Applications, 111.

DIMITROV, A. & GOLPARVAR-FARD, M. 2014. Vision-based material recognition for automated monitoring of construction progress and generating building information modeling from unordered site image collections. Advanced Engineering Informatics, 28, 37-49.

DING, L., ZHONG, B., WU, S. & LUO, H. 2016. Construction risk knowledge management in BIM using ontology and semantic web technology. Safety science, 87, 202-213.

DOUKARI, O., SECK, B. & GREENWOOD, D. 2022. The Creation of Construction Schedules in 4D BIM: A Comparison of Conventional and Automated Approaches. Buildings, 12, 1145.

DUMAIS, S. T. 2004. Latent semantic analysis. Annu. Rev. Inf. Sci. Technol., 38, 188-230.

DURDYEV, S., MBACHU, J., THURNELL, D., ZHAO, L. & HOSSEINI, M. R. 2021. BIM adoption in the Cambodian construction industry: key drivers and barriers. ISPRS International Journal of Geo-Information, 10, 215.

EVANGELOPOULOS, N., ZHANG, X. & PRYBUTOK, V. R. 2012. Latent semantic analysis: five methodological recommendations. European Journal of Information Systems, 21, 70-86.

GAO, G., LIU, Y.-S., WANG, M., GU, M. & YONG, J.-H. 2015. A query expansion method for retrieving online BIM resources based on Industry Foundation Classes. Automation in construction, 56, 14-25.

GOH, Y. M. & UBEYNARAYANA, C. 2017. Construction accident narrative classification: An evaluation of text mining techniques. Accident Analysis & Prevention, 108, 122-130.

GRUBER, T. R. 1993. A translation approach to portable ontology specifications. Knowledge acquisition, 5, 199-220.

GUO, D., ONSTEIN, E. & ROSA, A. D. L. 2020. An Approach of Automatic SPARQL Generation for BIM Data Extraction. Applied Sciences, 10, 8794.

JAMAL, K. A. A., MOHAMMAD, M. F., HASHIM, N., MOHAMED, M. R. & RAMLI, M. A. Challenges of Building Information Modelling (BIM) from the Malaysian architect's perspective. MATEC web of conferences, 2019. EDP Sciences, 05003.

JIANG, J. J. & CONRATH, D. W. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. arXiv preprint cmp-lg/9709008.

JIANG, S., WANG, N. & WU, J. 2018. Combining BIM and ontology to facilitate intelligent green building evaluation. Journal of Computing in Civil Engineering, 32, 04018039.

JUNG, N. & LEE, G. 2019. Automated classification of building information modeling (BIM) case studies by BIM use based on natural language processing (NLP) and unsupervised learning. Advanced Engineering Informatics, 41, 100917.

KADHIM, A. I. 2019. Survey on supervised machine learning techniques for automatic text classification. Artificial Intelligence Review, 52, 273-292.

KANG, T. W. & HONG, C. H. 2015. A study on software architecture for effective BIM/GIS-based facility management data integration. Automation in construction, 54, 25-38.

KHOSAKITCHALERT, C., YABUKI, N. & FUKUDA, T. 2019. Improving the accuracy of BIM-based quantity takeoff for compound elements. Automation in Construction, 106, 102891.

KIM, T. & CHI, S. 2019. Accident case retrieval and analyses: using natural language processing in the construction industry. Journal of Construction Engineering and Management, 145, 04019004.

KOO, B., LA, S., CHO, N.-W. & YU, Y. 2019. Using support vector machines to classify building elements for checking the semantic integrity of building information models. Automation in Construction, 98, 183-194.

KOWSARI, K., JAFARI MEIMANDI, K., HEIDARYSAFA, M., MENDU, S., BARNES, L. & BROWN, D. 2019. Text classification algorithms: A survey. Information, 10, 150.

LAMPINEN, A. K. & MCCLELLAND, J. L. 2017. One-shot and few-shot learning of word embeddings. arXiv preprint arXiv:1710.10280.

LANDAUER, T. K. & DUMAIS, S. T. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological review, 104, 211.

LANDAUER, T. K., FOLTZ, P. W. & LAHAM, D. 1998. An introduction to latent semantic analysis. Discourse processes, 25, 259-284.

LEACOCK, C. & CHODOROW, M. 1998. Combining local context and WordNet similarity for word sense identification. WordNet: An electronic lexical database, 49, 265-283.

LEE, S.-K., KIM, K.-R. & YU, J.-H. 2014. BIM and ontology-based approach for building cost estimation. Automation in construction, 41, 96-105.

LIN, D. An information-theoretic definition of similarity. Icml, 1998. 296-304.

LIU, H., LU, M. & AL-HUSSEIN, M. 2016. Ontology-based semantic approach for construction-oriented quantity take-off from BIM models in the light-frame building industry. Advanced Engineering Informatics, 30, 190-207.

LOCATELLI, M., SEGHEZZI, E., PELLEGRINI, L., TAGLIABUE, L. C. & DI GIUDA, G. M. 2021. Exploring natural language processing in construction and integration with building information modeling: A scientometric analysis. Buildings, 11, 583.

LUKASIEWICZ, T. & STRACCIA, U. 2008. Managing uncertainty and vagueness in description logics for the semantic web. Journal of Web Semantics, 6, 291-308.

MAULUD, D. H., ZEEBAREE, S. R., JACKSI, K., SADEEQ, M. A. M. & SHARIF, K. H. 2021. State of art for semantic analysis of natural language processing. Qubahan Academic Journal, 1, 21-28.

MOON, S., LEE, G. & CHI, S. 2022. Automated system for construction specification review using natural language processing. Advanced Engineering Informatics, 51, 101495.

NIKNAM, M. & KARSHENAS, S. 2017. A shared ontology approach to semantic representation of BIM data. Automation in Construction, 80, 22-36.

PAN, J., ANUMBA, C. & REN, Z. Potential application of the semantic web in construction. Proceedings of the Twentieth Annual ARCOM Conference. Edinburgh, UK, 2004.

PARK, M., LEE, K.-W., LEE, H.-S., JIAYI, P. & YU, J. 2013. Ontology-based construction knowledge retrieval system. KSCE Journal of Civil Engineering, 17, 1654-1663.

PAUDEL, S., ELMITRI, M., COUTURIER, S., NGUYEN, P. H., KAMPHUIS, R., LACARRIÈRE, B. & LE CORRE, O. 2017. A relevant data selection method for energy consumption prediction of low energy building based on support vector machine. Energy and Buildings, 138, 240-256.

PINO, J. & ESKENAZI, M. Measuring Hint Level in Open Cloze Questions. FLAIRS Conference, 2009. Citeseer.

RESNIK, P. 1995. Using information content to evaluate semantic similarity in a taxonomy. arXiv preprint cmp-lg/9511007.

SALAMA, D. M. & EL-GOHARY, N. M. 2016. Semantic text classification for supporting automated compliance checking in construction. Journal of Computing in Civil Engineering, 30, 04014106.

SHIN, S. & ISSA, R. R. 2021. BIMASR: framework for voice-based BIM information retrieval. Journal of Construction Engineering and Management, 147, 04021124.

SHRIKUMAR, A., GREENSIDE, P. & KUNDAJE, A. Learning important features through propagating activation differences. International conference on machine learning, 2017. PMLR, 3145-3153.

SONG, W. & PARK, S. C. A novel document clustering model based on latent semantic analysis. Third International Conference on Semantics, Knowledge and Grid (SKG 2007), 2007. IEEE, 539-542.

TIXIER, A. J.-P., HALLOWELL, M. R., RAJAGOPALAN, B. & BOWMAN, D. 2016. Automated content analysis for construction safety: A natural language processing system to extract precursors and outcomes from unstructured injury reports. Automation in Construction, 62, 45-56.

TSERNG, H.-P., HO, S.-P. & JAN, S.-H. 2014. Developing BIM-assisted as-built schedule management system for general contractors. Journal of Civil Engineering and Management, 20, 47-58.

WANG, J., GAO, X., ZHOU, X. & XIE, Q. 2021. Multi-scale Information Retrieval for BIM using Hierarchical Structure Modelling and Natural Language Processing. J. Inf. Technol. Constr., 26, 409-426.

WANG, N., ISSA, R. R. & ANUMBA, C. J. 2022. NLP-Based Query-Answering System for Information Extraction from Building Information Models. Journal of Computing in Civil Engineering, 36, 04022004.

WIJAYAKUMAR, M. & JAYASENA, H. S. 2013. Automation of BIM quantity take-off to suit QS's requirements.

WU, S., SHEN, Q., DENG, Y. & CHENG, J. 2019. Natural-language-based intelligent retrieval engine for BIM object database. Computers in Industry, 108, 73-88.

WU, Z. & PALMER, M. 1994. Verb semantics and lexical selection. arXiv preprint cmp-lg/9406033.

XIE, Q., ZHOU, X., WANG, J., GAO, X., CHEN, X. & LIU, C. 2019. Matching real-world facilities to building information modeling data using natural language processing. Ieee Access, 7, 119465-119475.

ZHANG, F., FLEYEH, H., WANG, X. & LU, M. 2019. Construction site accident analysis using text mining and natural language processing techniques. Automation in Construction, 99, 238-248.

ZHANG, J. & EL-GOHARY, N. 2012. Extraction of construction regulatory requirements from textual documents using natural language processing techniques. Computing in Civil Engineering (2012).

ZHANG, J. & EL-GOHARY, N. M. 2015. Automated information transformation for automated regulatory compliance checking in construction. Journal of Computing in Civil Engineering, 29, B4015001.

ZHANG, L. & ISSA, R. R. 2013. Ontology-based partial building information model extraction. Journal of Computing in Civil Engineering, 27, 576-584.

ZHONG, B., DING, L., LOVE, P. E. & LUO, H. 2015. An ontological approach for technical plan definition and verification in construction. Automation in Construction, 55, 47-57.

ZHOU, Y., SU, W., DING, L., LUO, H. & LOVE, P. E. 2017. Predicting safety risks in deep foundation pits in subway infrastructure projects: support vector machine approach. Journal of Computing in Civil Engineering, 31, 04017052.

ZOU, Y., KIVINIEMI, A. & JONES, S. W. 2017. Retrieving similar cases for construction project risk management using Natural Language Processing techniques. Automation in construction, 80, 66-76.