Advanced Engineering Informatics 46 (2020) 101179

Contents lists available at ScienceDirect

Advanced Engineering Informatics

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

# Quantitative similarity assessment of construction projects using WBS-based metrics

Navid Torkanfar [a,*], Ehsan Rezazadeh Azar [b]

[a] Department of Civil Engineering, Lakehead University, Thunder Bay, Canada
[b] Department of Civil Engineering, Lakehead University, Thunder Bay P7B5E1, Canada

ARTICLE INFO

*Keywords:*
Project similarity
Construction
Natural language processing
Work breakdown structure
Knowledge retrieval
Project planning

ABSTRACT

Lessons learned from completed projects are valuable resources for planning of new projects. A quantitative similarity measurement between construction projects can improve knowledge reuse practices. The information and documents of a similar past project can be retrieved to resolve the challenges in a new project. This paper introduces a novel method for measuring the similarity of construction projects based on semantic comparison of their work breakdown structure (WBS). WBS of a project should theoretically encompass a hierarchical decomposition of the total scope of project's works, thus it could be used as an appropriate representative of the projects. The proposed method measures the semantic similarity between WBS of projects by means of natural language processing techniques. This method was implemented based on three metrics: node, structural, and total similarity. Each of these metrics calculate a quantitative similarity score between 0 and 1. The method was assessed using fifteen test samples with promising results in compliance with similarity properties. In addition, precision and recall of the method were evaluated in retrieving similar past projects. The results illustrate that the structural similarity slightly outperforms the other metrics.

## 1. Introduction

Project managers typically consider knowledge gained from previous projects in their decision makings [43,11]. Effective reuse of the gained knowledge not only reduces the time and cost of solving problems, but also improves the quality of solutions [45]. In construction, various studies have investigated different methods to use past information and experiences in new projects. These studies have applied different techniques from other areas such as Knowledge Management (KM) and Artificial Intelligence (AI).

Knowledge management systems are information technology-based systems aiming at improving organizational knowledge processes, including creation, storage/retrieval, transfer, and application of knowledge [2]. One important step in knowledge management systems is to effectively search databases and find the relevant knowledge. The conventional retrieval of relevant knowledge can often be difficult and sometimes results in several irrelevant documents [14]. For instance, in construction, a method was proposed in which the intended information was retrieved through a simple Google™-like search or an advanced search function [44].

Case-Based Reasoning (CBR) is one of the popular techniques to solve a problem by reusing past information. A CBR system recalls a similar past situation to solve a new problem [21]. In construction, CBR has been used in various areas, such as cost estimation [3,35], safety [12], structural design [26], and planning [38,29]. The first step in CBR methods is to measure the similarity of a new case with previously stored cases to retrieve the most similar case(s) [6]. The retrieving process requires some predefined nominal and/or numerical attributes, such as type, size, structural system, and location of the project. In addition, a user-defined weight is considered for each attribute which will be used to calculate the similarity of cases and retrieve the most similar project (s). One of the existing challenges in this step is to find the most relevant attributes and their appropriate weights.

In addition to CBR, some AI methods have been used in construction to model different problems based on the past data and information to predict important project information, such as cost, schedules, and safety plans. Neural networks and linear regression models have been implemented to estimate project costs [20] and facilitate project planning [47].

A quantitative similarity measurement of construction projects can help project managers find similar past projects and extract related information and documents. This can happen at various stages of a project,

* Corresponding author.
*E-mail addresses:* ntorkanf@lakeheadu.ca (N. Torkanfar), eazar@lakeheadu.ca (E. Rezazadeh Azar).

https://doi.org/10.1016/j.aei.2020.101179
Received 27 May 2020; Received in revised form 2 September 2020; Accepted 17 September 2020
Available online 30 September 2020

such as planning and execution phases. Quantitative similarity assessment between projects can potentially improve current CBR and other AI methods by providing more comprehensive attributes that consider the entire project rather than focusing on certain attributes.

The research studies on quantitative measurement of similarity between construction projects, however, are still limited. For example, a recent attempt in the area of project bundling proposed a method to quantify construction projects similarity by vectorizing the projects' pay items and measuring the distance between vectors [34].

Scope management of a construction project requires comprehensive assessment of the project and a main outcome of this assessment, i.e. Work Breakdown Structure (WBS), is used by other project management areas, namely project time and cost management [32]. But there is not any research attempt to use WBS, as a hierarchical breakdown of the scope of a project, for similarity assessment of the projects. The outcome of this assessment can identify similar projects for better development of WBS and project planning of a new project.

The aim of this research study is to develop a method to assess the similarity of construction projects using their WBSs. It has been hypothesised that the tasks and services required during the construction phase can be used to develop metrics to measure the similarity of construction projects. Since the WBS of a project contains hierarchical information about its scope, WBS was considered as a potential representative of construction projects. Natural language processing (NLP) techniques were employed in the proposed method to extract semantic attributes of the work-packages. This method calculates a score between 0 and 1 to determine the semantic similarity of two WBSs.

## 2. Background

### 2.1. Work breakdown structure (WBS)

Project Management Institute (PMI) defines WBS as "*a hierarchical decomposition of the total scope of work to be carried out by the project team to accomplish the project objectives and create the required deliverables. The WBS organizes and defines the total scope of the project and represents the work specified in the current approved project scope statement*" [32]. In another word, the main goal of WBS is to present a complete and proper scope of the entire project work [17].

The highest level of the WBS hierarchy represents the entire project and is decomposed into smaller subjects, each representing tasks that should be performed for the higher-level subject to be completed. The process of subdividing continues until the tasks could not be decomposed any further (or it is not reasonable to do that). The lowest level entries in this structure represent work packages. The responsibility of the performance of each work package is assigned to an individual, unit, or organization [16]. The project management body of knowledge [32] provides generic guideline for creating an appropriate WBS; however, the complex and fragmented nature of construction projects, such as coordination of multiple players (e.g. subcontractors, contract administrators, and suppliers), brings about specific challenges in creating WBSs.

There are research studies on various aspects and application of WBS in construction management, such as proposed methods for automated WBS development [40], WBS-based project documentation [31], combining off-site and on-site WBSs [42], and WBS-based integration of project cost and time [18]. Nonetheless, none of these studies have focused on WBS-based similarity assessment of the construction projects.

### 2.2. Text similarity measurements

Natural language processing (NLP) is a research area that focuses on enabling computers to understand natural language text and speech [7]. Measuring similarity between words, sentences, paragraphs, and documents has been used for a long time in several NLP related fields, such as information retrieval, text classification, document clustering, topic detection, topic tracking, question generation, question answering, essay scoring, short answer scoring, machine translation, and text summarization [13]. One of the first implementations for text similarity measurement aimed at ranking documents in the order of their similarity to the input query [39].

Two words can be similar either semantically or lexically. Lexically similar words contain strings with similar characters in their structures, and this similarity is evaluated through a couple of string-based methods, which are discussed in the following subsection. Semantically similar words, however, are related by means of different relations, such as being synonyms, antonyms, or their utilization in the same context [13]. In other words, semantic similarity determines the relation of words or concepts based on predetermined databases, which include the relations of the words.

There are several studies on semantic analysis of texts and documents in construction management domain focused on information analysis and retrieval, and tested for some key applications, such as text classification and automated regularity compliance checking using NLP methods [50,49]. A method was proposed to extract semantic knowledge from contract documents and to categorize and retrieve information in electronic document management systems using NLP [1]. Another method was proposed to partition multi-topic documents into several passages [24]. The partitioning approach generates passages based on domain ontology. Costa et al. [8] explored a method to enrich the semantic vectors by means of ontology concepts and relations. The semantic vectors were used to represent knowledge sources.

### 2.3. String-based similarity measurement

In order to measure the string similarity between two words, Levenshtein [23] proposed edit distance method which identifies the difference between two strings by the minimum number of changes (insertion, deletion, or substitution) needed to transform one string to another. For example, the distance between the strings "cat" and "hat" is one character (substitution of character "c" with "h"). The edit distance method does not consider the number of strings. In another proposed method for syntactic similarity [25], the number of characters is also considered as shown in Eq. (1).

$$sim_{syn}(c_1, c_2) = max\left(0, \frac{min(|c_1|, |c_2|) - ed(c_1, c_2)}{min(|c_1|, |c_2|)}\right) \quad (1)$$

These two approaches calculate similarity without considering semantics of inputs. Therefore, lexical similarity methods do not reliably provide an accurate similarity measurement. For instance, similarity ($sim_{syn}$) between the concepts "reinforcement" and "rebar" would not return a high score of similarity, even though these two concepts are semantically related to a great degree. It was shown that semantic similarity algorithms outperform simple lexical methods with a 13% error rate reduction [28].

### 2.4. Semantic similarity measurement

Semantic similarity measurement methods have been developed using corpus-based and knowledge-based algorithms. A corpus is a large structured set including written or spoken texts for the purpose of language processing. The corpus-based semantic similarity determines the similarity of various words by utilizing a large corpus. Latent semantic analysis (LSA) [22] is one of the most popular methods for obtaining corpus-based similarity. LSA hypothesizes that reoccurring of the same words in similar pieces of texts is an indication for their proximate meaning [22].

The knowledge-based similarity is another type of semantic similarity that measures similarity by using embedded information in semantic networks. A semantic network is a knowledge base which

represents semantic relations of concepts using networks [41].

WordNet is a popular software tool in the field of knowledge-based semantic similarity measurement, which was produced as a result of a comprehensive research program at Princeton University [30] and is utilized as a lexical reference of English language. In WordNet, English nouns, verbs, and adjectives are organized in synonym sets and these sets are related together by means of semantic relations [27]. A variety of semantic relations have been developed in WordNet including (but not limited to) synonymy, autonomy, hyponymy, and membership [30]. By exploiting these relations, semantic hierarchy structures are developed and these hierarchies could be useful in semantic computations.

There are several different methods for semantic similarity measurements based on WordNet, such as path-based, information content-based [36], feature-based [46], and hybrid measurements. This research utilizes a method that calculates the similarity between two concepts based on their depth in the taxonomy [48]. It computes similarity based on the position of concepts $c_1$ and $c_2$, as well as the lowest common subsumer $lso(c_1, c_2)$. In Eq. (2), the function $len(c_1, c_2)$ measures the length of the shortest path from the concept $c_1$ to concept $c_2$, and the *depth* measures the length of the path from each concept to the root element [48].

$$sim_{WP}(c_1, c_2) = \frac{2*depth(lso(c_1, c_2))}{len(c_1, c_2) + 2*depth(lso(c_1, c_2))} \quad (2)$$

## 3. Methodology

This paper proposes a method to quantify similarity of construction projects based on semantic and structural metrics derived from their WBSs. The WBS of a project includes some nodes, which are labelled with tasks required to complete that project. The focus of this research is to quantify the similarity of construction projects based on the tasks required to build a project during its construction phase. A method was developed in Python programming environment to compare documented construction projects with a targeted project, based on their WBSs. Following subsections describe the elements of this method.

### 3.1. WBS encoding

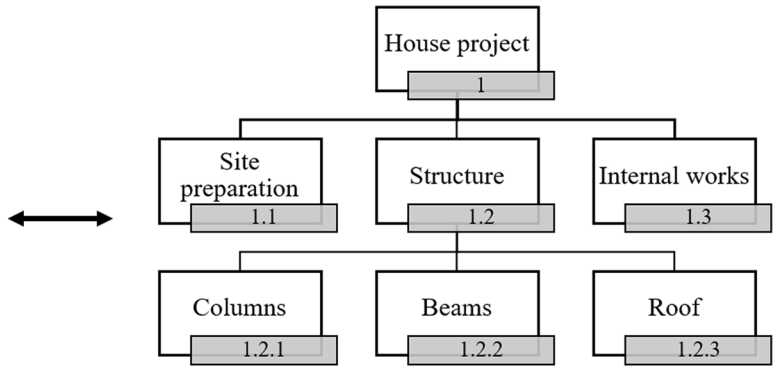
The WBS information are exported from Microsoft Project file of the sample projects to a spreadsheet format file (such as Microsoft Excel). Fig. 1 shows the tasks and WBS codes in spreadsheet format for small parts of two simplified projects (drastically shortened for the purpose of representation) that belong to a "House project" and a "Bridge project".

Each node in the WBS hierarchy contains two main information: the node's task, and the node's code which locates each element in the hierarchy. WBS hierarchies of the projects were written in eXtensible Markup Language (XML) to encode this information into a machine-readable format [4]. Fig. 2 depicts a part of a WBS of a building project which is encoded in the XML format. As shown in Fig. 2, each element contains a text and a numerical attribute, where the text represents the task and is followed with the attribute of the level of the task in the hierarchy. For instance, the XML element in line 8 (in Fig. 2) contains a task which is called "earthworks" and its level is 1.2 (i.e. the second task in the second level of WBS).

### 3.2. Comparison of nodes

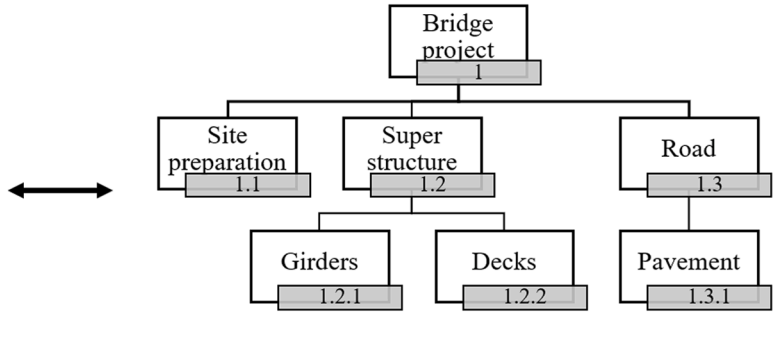
The first step in measuring the similarity of two WBSs is to compare the tasks within the WBS nodes. There are two important issues in measuring the similarity of the tasks. First, their naming is subjective to project managers. For instance, "Rebar placement" and "Reinforcement installation" are not the same strings, but both of them represent the

| Node Code | Tasks |
|---|---|
| 1 | House project |
| 1.1 | Site preparation |
| 1.2 | Structure |
| 1.2.1 | Columns |
| 1.2.2 | Beams |
| 1.2.3 | Roof |
| 1.3 | Internal works |




(a)

| Node Code | Tasks |
|---|---|
| 1 | Bridge project |
| 1.1 | Site preparation |
| 1.2 | Super structure |
| 1.2.1 | Girders |
| 1.2.2 | Decks |
| 1.3 | Road |
| 1.3.1 | Pavement |




(b)

**Fig. 1.** WBS codes and hierarchy of tasks; (a) "House project" (b) "Bridge project".

```
<?xml version="1.0" encoding="Utf-8"?>
<steel_building level="1">
	<site_preparation level="1.1">
		<site_mobilization level="1.1.1"></site_mobilization>
		<surveying level="1.1.2"></surveying>
		<fencing level="1.1.3"></fencing>
	</site_preparation>
	<earthworks level="1.2">
		<stripping_ground level="1.2.1"></stripping_ground>
		<excavation level="1.2.2"></excavation>
	</earthworks>
	<foundation level="1.3">
		<reinforcing_installation level="1.3.1"></reinforcing_installation>
		<form_work level="1.3.2"></form_work>
		<concrete_pouring level="1.3.3"></concrete_pouring>
		<curing_concrete level="1.3.4"></curing_concrete>
		<form_work_removal level="1.3.5"></form_work_removal>
	</foundation>
	<steel_structure level="1.4">
		<installation_of_columns level="1.4.1"></installation_of_columns>
		<installation_of_beams_first_floor level="1.4.2"></installation_of_beams_first_floor>
		<installation_of_beams_roof level="1.4.3"></installation_of_beams_roof>
	</steel_structure>
	<floor_slabs level="1.5">
		<ground_floor_concrete level="1.5.1"></ground_floor_concrete>
		<first_floor_concrete level="1.5.2"></first_floor_concrete>
		<roof_floor_concrete level="1.5.3"></roof_floor_concrete>
	</floor_slabs>
	<ground_floor_Architectural level="1.6">
		<external_walls level="1.6.1"></external_walls>
		<separation_walls level="1.6.2"></separation_walls>
		<partition_walls level="1.6.3"></partition_walls>
		<windows_installation level="1.6.4"></windows_installation>
		<doors_installation level="1.6.5"></doors_installation>
	</ground_floor_Architectural>
```

**Fig. 2.** Segment of the written XML for WBS of a steel structure building.

same task. Thus, two tasks should not contain the exact same texts to be considered similar. This problem can be addressed by including semantic similarity measurements of tasks instead of simple string measurements.

On the other hand, the semantic equivalence of tasks does not necessarily result in similarity of their nodes. For example, there are two nodes with "concrete pouring" as label, but they might represent different tasks, where one can represent concrete pouring for a column (s) and the other one is for a beam(s).

To address the above-mentioned issues, the proposed method determines the similarity of two WBSs through the following three metrics.

(1) Semantic similarity, in which the semantic similarity of the tasks within the compared nodes is measured;
(2) Parent similarity, which measures the semantic similarity of the parents of the compared nodes;
(3) Siblings similarity, which measures the semantic similarity of siblings (nodes from a common parent) of the compared nodes.

### 3.3. Semantic similarity

WordNet [30] was utilized to measure the semantic similarity of the node's tasks in the proposed method. Tasks are usually expressed as a phrase that contains a few words. There are different methods for measuring the semantic similarity of two sentences or phrases by averaging semantic similarity of their words, such as a method proposed by Mihalcea et al. [28]. To measure the semantic similarity of two text segments $T_1$ and $T_2$, for each word $w$ in the segment $T_1$, this method uses one of the word-to-word similarity measures (previously explained) to find the most semantically similar word from segment $T_2$ ($maxSim(w, T_2)$). The same procedure will determine the most similar word in $T_1$ starting with the words in $T_2$. These similarities are then weighted with corresponding word specificity. The specificity of words *idf(w)* gives higher scores to the specific words compared to the generic concepts such as "get" or "become" [28]. This method measures the semantic similarity of two segments as presented in Eq. (3).

$$sim(T_1, T_2) = \frac{1}{2}\left(\frac{\sum_{w\in\{T_1\}}(maxSim(w, T_2)*idf(w))}{\sum_{w\in\{T_1\}} idf(w)} + \frac{\sum_{w\in\{T_2\}}(maxSim(w, T_1)*idf(w))}{\sum_{w\in\{T_2\}} idf(w)}\right) \quad (3)$$

This method can be adjusted to a more appropriate one by eliminating the word specificity weight. The reason behind this decision is that in this case, the tasks are phrases with a very few component words which are mostly specific to the construction domain rather than being generic concepts.

Using Wu and Palmer method [48] as a word-to-word semantic similarity measurement and considering the above-mentioned assumptions, the semantic similarity between $task_i$ and $task_j$ is calculated using Eq. (4). In this approach for each word $w$ in the $task_i$, the most semantically similar word from $task_j$ ($maxSim(w, task_j)_{wup}$) is found by means of the Wu and Palmer [48] method. The same procedure will determine the most similar word in $task_i$ starting with the words in $task_j$.

$$sim_{semantic}(\text{task}_i, \text{task}_j) = \frac{1}{2}\left(\frac{\sum_{w\in\{\text{task}_i\}}(maxSim(w, \text{task}_j)_{wup})}{\sum_{w\in\{\text{task}_i\}} 1} + \frac{\sum_{w\in\{\text{task}_j\}}(maxSim(w, \text{task}_i)_{wup})}{\sum_{w\in\{\text{task}_j\}} 1}\right) \quad (4)$$

For example, Eq. (4) results in a similarity score of 0.9 for tasks labeled as "reinforcement installation" and "reinforcement placement". This similarity is less than one, because word-to-word similarity of the "installation" and "placement" is lower than one, and therefore it decreases the total semantic similarity to 0.9.

### 3.4. Word-to-word semantic similarity measurements

Since in WordNet the relations between concepts are based on synsets, an algorithm was required to find the similarity between words rather than synsets [19]. WordNet defines synsets as sets of synonyms composed of nouns, verbs, adjectives, or adverbs that each expresses a unique concept [33]. Thus, to compute the semantic similarity of two words by utilizing WordNet, one synset from each word should be selected. The comparison of chosen synsets results in the semantic similarity of two words. In construction domain, however, some of the applied words do not have a special meaning in regular vocabulary resources, such as WordNet. To address this issue, technical words were replaced by meaningful terms defined in the WordNet. For example, the word 'HVAC' was replaced with 'Heating Ventilation and Air Conditioning' or the word "rebar" was replaced with "reinforcement". In addition, the words that are not defined in WordNet were compared lexically by the string-based method which was introduced in Eq. (1).

A simplified method has been used in this study to measure the semantic similarity of two words. In this approach, the system approximates the similarity of two words by using a pair of their synsets that result in maximum similarity, as shown in Eq. (5).

$$word - to - word_{similairty}(w_1, w_2) = max(similarity(C_1, C_2)) \quad (5)$$

$$C_1 \in synsets(w_1), C_2 \in synsets(w_2)$$

Using the word-to-word similarity measure and the proposed method for measuring semantic similarity of two phrases, semantic similarity of two tasks was calculated. Assuming $WBS_{N_1}^{L_1}$ and $WBS_{N_2}^{L_2}$ are two WBSs, in which $L_1$and $L_2$represent the total number of levels that each WBS hierarchy contains (e.g. the WBS illustrated in Fig. 5 has three levels and its L is three). Moreover, $N_1$ and $N_2$ represent the finite sets of WBS's nodes ($N_1 : (n_1, n_2, \cdots, n_N)$ and $N_2 : (m_1, m_2, \cdots, m_M)$). The results of these pairwise comparisons between nodes $n_i$ and $m_j$ from $WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2}$ will form a matrix shown in Eq. (6). This matrix represents the semantic similarity of tasks between nodes $n_i$ and $m_j$.

$$sim_{nodes}\left(WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2}\right) = \begin{bmatrix} sim_{semantic}(n_1, m_1) & \cdots & sim_{semantic}(n_1, m_j) \\ \vdots & \ddots & \vdots \\ sim_{semantic}(n_i, m_1) & \cdots & sim_{semantic}(n_i, m_j) \end{bmatrix} \quad (6)$$

The proposed method considers two nodes semantically similar if they have a semantic similarity more than a user-defined threshold between 0 and 1. In addition, to reduce computation effort, the system only computes the other node similarity metrics (parent similarity and siblings similarity) for the nodes that are semantically similar (more than the threshold). The effects of different thresholds on the accuracy of the system are explored in section 4 "Experimental results".

### 3.5. Comparison of the nodes' parents

In a WBS, except the root element (i.e. the highest level), each node is subdivided from an upper-level element, which is the parent of the node. Also, each parent is generated from an upper-level element which creates a sequence of parents for each node. This metric determines the similarity of the sequence of parents and is calculated by averaging their semantic similarity. Therefore, this method reduces the similarity of the tasks which belong to different parts of compared projects, such as in two matched "concrete pouring" tasks belonging to foundation and shear wall construction accordingly.

Since considering all ancestors of a node requires a large amount of calculations, the least similar parent (LSP) is defined. LSPs are the first pair of parents in the sequence of two given nodes' parents that are not semantically similar (less than the defined threshold). This method only considers the parents up to LSP. Given nodes n and m from $WBS_{N_1}^{L_1}$, $WBS_{N_2}^{L_2}$ respectively, the parent similarity between them ($sim_{parents}(n, m)$) is calculated using Eq. (7). $L_{LSP} - L_n$ is the difference between levels of node n and its LSP, and $sim_{semantic}\left(i^{th}parents\right)$ is the semantic similarity between $i^{th}$ parents of nodes n and m.

$$sim_{parents}(n, m) = \frac{\sum_{i=1}^{L_{LSP}-L_n}(L_{LSP} - L_n - (i-1)) \times sim_{semantic}(ithparents)}{\sum_{i=1}^{L_{LSP}-L_n}(i)} \quad (7)$$

For instance, Fig. 3 shows the first two parents of nodes $n_1$ and $m_1$ with a semantic similarity of 0.8, which is more than an arbitrarily defined threshold of 0.5. In this example, the next two parents have a similarity of 0.2 (i.e. less than threshold of 0.5), and therefore they are defined as LSP. In this case, parent similarity is calculated by the following function.

$$sim_{parents}(n_1, m_1) = \frac{2*0.8 + 1*0.2}{2 + 1} = 0.6$$

The results of parent similarity between nodes from $WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2}$ are presented using the matrix shown in Eq. (8).

$$sim_{parents}\left(WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2}\right) = \begin{bmatrix} sim_{parents}(n_1, m_1) & \cdots & sim_{parents}(n_1, m_j) \\ \vdots & \ddots & \vdots \\ sim_{parents}(n_i, m_1) & \cdots & sim_{parents}(n_i, m_j) \end{bmatrix} \quad (8)$$

### 3.6. Comparison of the nodes' siblings

In a WBS, nodes generated from the same parent are called siblings. Similarity of the nodes' siblings in two WBSs can also enhance the possibility that the nodes' tasks are rather similar. To calculate the

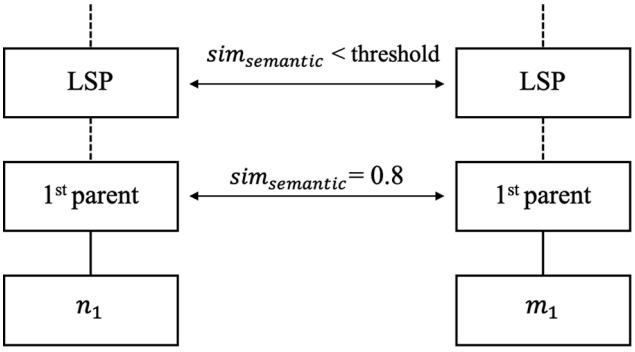



**Fig. 3.** Parent similarity between nodes $n_1$ and $m_1$.

sibling similarity between nodes $n_i$ and $m_i$, their siblings are compared one by one and any two siblings which are semantically similar (i.e. $sim_{semantic} > threshold$) are considered matched together. Thus, $(sibling_{n_i}, sibling_{m_j})$ can be defined as a tuple that includes the pairs of matched siblings from nodes $n_i$ and $m_j$ (Eq. (9)).

$$matched_{siblings}(n_i, m_j) = \left(sibling_{n_i}, sibling_{m_j}\right) \tag{9}$$

As a result, the sibling similarity score between nodes$n_i$ and $m_i$ is calculated using Eq. (10), which is obtained by dividing the total number of matched siblings by the total number of siblings.

$$sim_{siblings}(n_i, m_j) = \frac{\left|matched_{siblings}(n_i, m_j)\right|}{\left|siblings_{n_i}\right| + \left|siblings_{m_j}\right|} \tag{10}$$

For example, the sibling similarity between nodes $n_i$ and $m_j$ with only one pair of matched siblings in Fig. 4 is calculated by the function bellow.

$$sim_{siblings}(n_i, m_j) = \frac{2*1}{2+2} = 0.5$$

A sibling similarity matrix, which contains a pairwise comparison of nodes $n_i$ and $m_j$ from $WBS_{N_1}^{L_1}$ and $WBS_{N_2}^{L_2}$, can be expressed as shown in Eq. (11).

$$sim_{siblings}\left(WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2}\right) = \begin{bmatrix} sim_{siblings}(n_1, m_1) & \cdots & sim_{siblings}(n_1, m_j) \\ \vdots & \ddots & \vdots \\ sim_{siblings}(n_i, m_1) & \cdots & sim_{siblings}(n_i, m_j) \end{bmatrix} \tag{11}$$

### 3.7. Average similarity of compared nodes

The average similarity matrix represents the average node to node similarity between nodes of $WBS_{N_1}^{L_1}$ and $WBS_{N_2}^{L_2}$, which is calculated by means of Eq. (12) and presented as Eq. (13).

$$sim_{average} = \frac{sim_{nodes} + sim_{parents} + sim_{siblings}}{3} \tag{12}$$

$$sim_{average}\left(WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2}\right) = \begin{bmatrix} sim_{average}(n_1, m_1) & \cdots & sim_{average}(n_1, m_j) \\ \vdots & \ddots & \vdots \\ sim_{average}(n_i, m_1) & \cdots & sim_{average}(n_i, m_j) \end{bmatrix} \tag{13}$$

### 3.8. Mapping of nodes

Each node from the first WBS will be mapped to a node from the second WBS with the highest average similarity. The highest average similarity must be more than the determined threshold. This threshold is considered to prevent mapping of irrelevant nodes which have a semantic similarity score below the threshold.

In some cases, there could be more than one node with the same highest $sim_{average}$. In these cases, the system prefers the nodes with a closer level of details. Level of details of the nodes depends on their level in the WBS hierarchy. Details in the hierarchy decreases from the lowest to the highest level. Since the lowest level of WBS usually contains the task with the highest level of details, level of details of each node is assessed based on the distance between its level and the lowest level in the WBS hierarchy. For this purpose, the system defines a weight between 0 and 1, which determines distance between the level of details of two nodes.

$Bottom_{level}$ is the level of nodes which is numbered starting from the lowest level in the hierarchy. For example, $bottom_{level}$ and regular $level$ of nodes in WBS ("House project") are indicated in Fig. 5.

This weight is calculated by the absolute difference between $bottom_{level}$ of two nodes, divided by the maximum number of levels that two WBS have (Eq. (14)).

$$level_{scores}(n_i, m_j) = \left|\frac{\left(bottom_{level}(n_i) - bottom_{level}(m_j)\right)}{max(L_1, L_2)} - 1\right| \tag{14}$$

For example, $level_{scores}$ for nodes "Columns" and "Road" from the "House project" and "Bridge project" in Fig. 1 is calculated as,

$$level_{scores}(\text{"Columns"}, \text{"Road"}) = \left|\frac{(1-2)}{3} - 1\right| = 0.66$$

and for "Columns", "Girders" is calculated as,

$$level_{scores}(\text{"}Columns\text{"}, \text{"}Girders\text{"}) = \left|\frac{(1-1)}{3} - 1\right| = 1$$

This score increases the chance of node "Columns" to be mapped to "Girders" instead of the node "Road" with a lower $level_{scores}$. The following matrix is used to contain node to node $level_{scores}$ for the nodes of two WBSs (see Eq. (15)).

$$level_{scores}\left(WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2}\right) = \begin{bmatrix} level_{scores}(n_1, m_1) & \cdots & level_{scores}(n_1, m_j) \\ \vdots & \ddots & \vdots \\ level_{scores}(n_i, m_1) & \cdots & level_{scores}(n_i, m_j) \end{bmatrix} \tag{15}$$

By multiplying matrixes $level_{scores}$ and $sim_{average}$, a matrix is formed which contains the required scores that can be used to find the mapped nodes (see Eq. (16)).

$$mapping_{scores}\left(WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2}\right) = \begin{bmatrix} mapping_{score}(n_1, m_1) & \cdots & mapping_{score}(n_1, m_j) \\ \vdots & \ddots & \vdots \\ mapping_{score}(n_i, m_1) & \cdots & mapping_{score}(n_i, m_j) \end{bmatrix} \tag{16}$$

The system searches through the $mapping_{scores}$ matrix to find the highest mapping score and when the highest score is found, the system will use that for mapping corresponding nodes and removes them for finding the other matched paired in the next runs. The system continues this procedure until all the possible nodes are mapped.

$Mappednodes$ is a list of tuples $(n_i, m_j, sim_{average})$, in which $n_i$ and $m_j$ are mapped together with the average similarity of $sim_{average}$ (Eq. (17)).

$$mappednodes = \left\{\left(n_i, m_j, sim_{average}\right)\right\} \tag{17}$$

$$n_i \in N_1$$

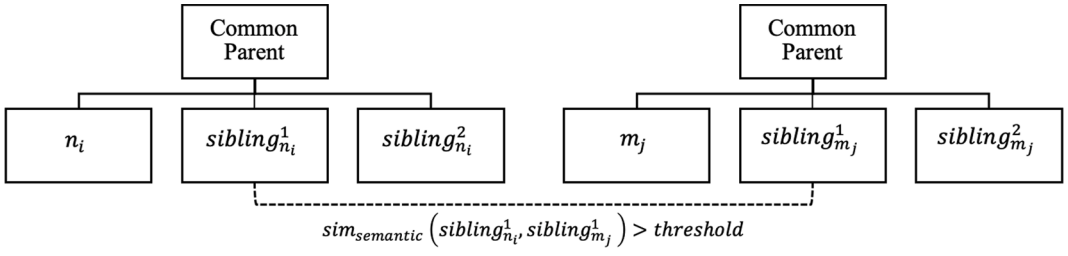



**Fig. 4.** Sibling similarity between nodes $\mathbf{n}_i$ and $\mathbf{m}_j$.

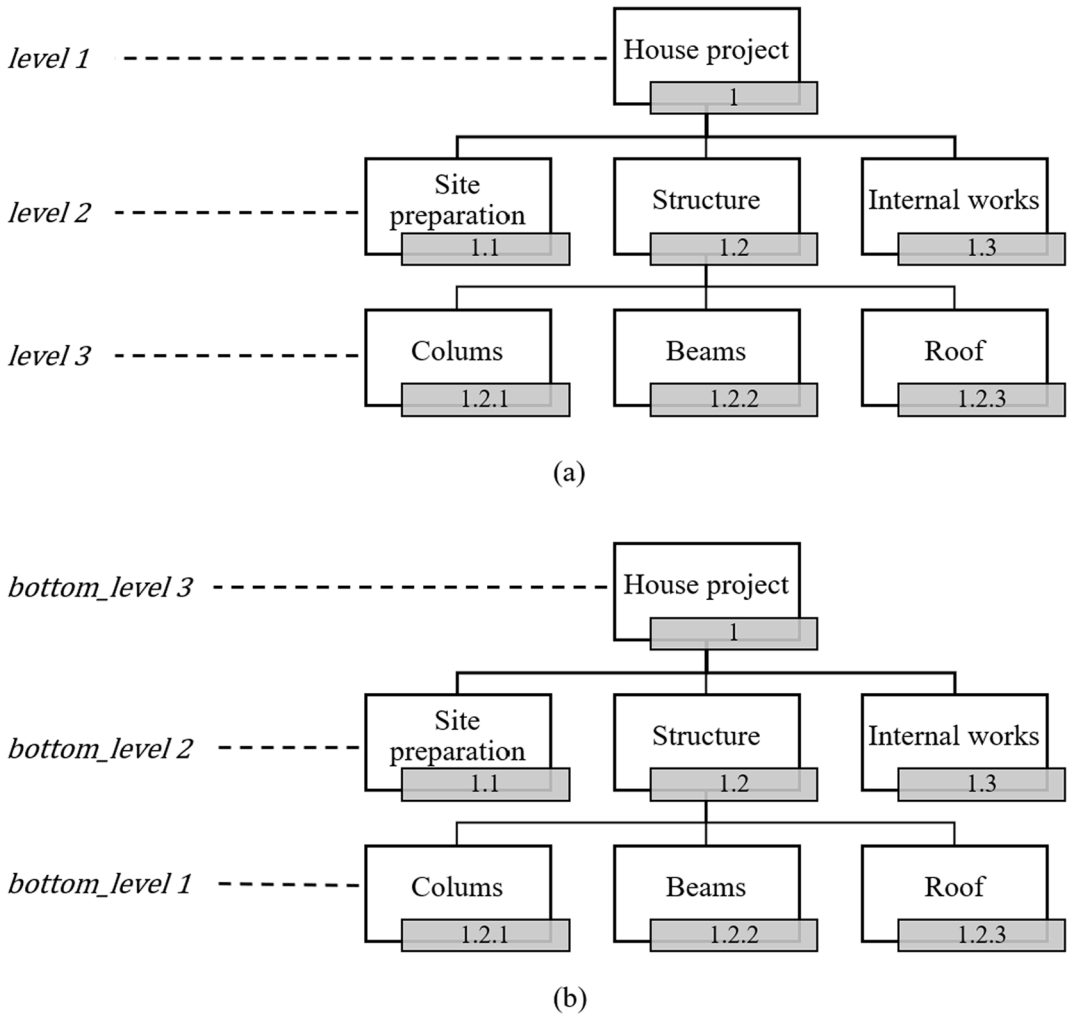



**Fig. 5.** WBS of the "House project"; (a): Regular level, (b): Bottom level.

$m_j \in N_2$

### 3.9. Node similarity score

Overall *Node similarity* score between $WBS_{N_1}^{L_1}$ and $WBS_{N_2}^{L_2}$ is the average of $sim_{average}$ of all the mapped nodes. This score is calculated by means of Eq. (18).

$$Nodesimilarity\left(WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2}\right) = \frac{2 \times \sum_{(n_i,m_j)\in mappednodes}\left(sim_{avg}\left(n_i, m_j\right)\right)}{|N_1| + |N_2|} \tag{18}$$

### 3.10. Structural similarity score

The second similarity measurement is the structural similarity, which examines the hierarchy structure of two WBSs. This metric is defined based on graph-edit-distance method [15,9] for the structure of two WBSs. The graph-edit-distance measures the minimum required operations to change the structure of one WBS to another. There are different graph-edit operations which can be used here. Node deletion or insertion, and node substitution were considered in this study.

The structural similarity measurements [9] start with the mapped nodes that were found in the previous stage. The node deletion or insertion cost (or effort) can be defined as the required operations to delete unmapped nodes. This cost was defined as Deletion Effort (*DE*), and can be computed by the total number of Unmapped Nodes ($|UN|$) divided by the total number of nodes in $WBS_{N_1}^{L_1}$ and $WBS_{N_2}^{L_2}$ (see Eq. (19)).

$$DE\left(WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2}\right) = \frac{|UN|}{|N_1| + |N_2|} \tag{19}$$

The Substitution Effort (*SE*) can be explained as the required effort to map the nodes. In other words, the required effort to map two similar nodes is lower than the required effort to map two less similar nodes. Therefore, for each pair of mapped nodes in *mappednodes* list, the *SE* is calculated by one minus their similarity (see Eq. (20)).

$$for\left(n_i, m_j\right) \in mappednodes, SE\left(n_i, m_j\right) = 1 - sim_{average}\left(n_i, m_j\right) \tag{20}$$

And, the total *SE* effort [9] between $WBS_{N_1}^{L_1}$ and $WBS_{N_2}^{L_2}$ over all the

$$Structuralsimilarity\left(WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2}\right) = 1 - \frac{DE\left(WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2}\right) + SE\left(WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2}\right)}{2} \tag{22}$$

mapped nodes can be calculated by Eq. (21).

$$SE\left(WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2}\right) = \frac{2^{*}\sum_{(n_i,m_j)\in mappednodes}\left(1 - sim_{average}\left(n_i, m_j\right)\right)}{|N_1| + |N_2| - |UN|} \quad (21)$$

The structural similarity between $WBS_{N_1}^{L_1}$ and $WBS_{N_2}^{L_2}$ is defined by 1 minus average of two over mentioned efforts (*DE* and *SE*), as shown in Eq. (22). Smaller amount of required effort to transfer structure of the first WBS to second one results in higher structural similarity and vice versa.

### 3.11. Total similarity score

The final score determines the *Total similarity* between $WBS_{N_1}^{L_1}$ *and* $WBS_{N_2}^{L_2}$, which is calculated by the average (see Eq. (23)) of *Node similarity* (Eq. (18)) and *Structural similarity* (Eq. (22)) scores. This final measurement produces a score between 0 and 1, in which 0 is hypothetically resulted from the comparison of two completely different projects, and 1.0 is resulted for two exact similar projects.

$$Total\ \ similarity\ \left(WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2}\right) = \frac{Node\ \ similarity\ \left(WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2}\right) + structural\ \ similarity\ \left(WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2}\right)}{2} \quad (23)$$

## 4. Experimental results

This section presents a set of experiments to evaluate the performance of the defined WBS similarity metrics in distinguishing construction projects and retrieving relevant samples. The experiments were carried out on fifteen different construction projects test samples.

These WBSs were developed for the construction phase of the projects, and other phases, such as feasibility study and design phases, were excluded. Three-dimensional models of five different construction projects were given to three experts in construction management industry to develop their WBS samples. To assure that the experts follow the same basic standards for creating their WBS samples, they were provided with some guidelines. The guideline includes the 100% rule, level of details, and the coding scheme. The 100% rule specifies that the total amount of work covered by the child elements have to be exactly same as the work content of the parent element. The experts were also asked to decompose each task to a level that it is not reasonable to further decompose the work package. In addition, the coding criteria, discussed in "3.1 WBS encoding" subsection, was defined for experts to follow.

Sample projects consisted of a bridge construction (steel girder with

**Table 1**
The developed samples by the experts.

| Experts | Developed samples | Represented by |
|---|---|---|
| Expert 1 | $Bridgeconstruction_1$ | $B_1$ |
| | $concretestructurebuilding_1$ | $C_1$ |
| | $steelstructurebuilding_1$ | $S_1$ |
| | $Roadmaintenance_1$ | $M_1$ |
| | $hotelbuilding_1$ | $H_1$ |
| Expert 2 | $Bridgeconstruction_2$ | $B_2$ |
| | $concretestructurebuilding_2$ | $C_2$ |
| | $steelstructurebuilding_2$ | $S_2$ |
| | $Roadmaintenance_2$ | $M_2$ |
| | $hotelbuilding_2$ | $H_2$ |
| Expert 3 | $Bridgeconstruction_3$ | $B_3$ |
| | $concretestructurebuilding_3$ | $C_3$ |
| | $steelstructurebuilding_3$ | $S_3$ |
| | $Roadmaintenance_3$ | $M_3$ |
| | $hotelbuilding_3$ | $H_3$ |

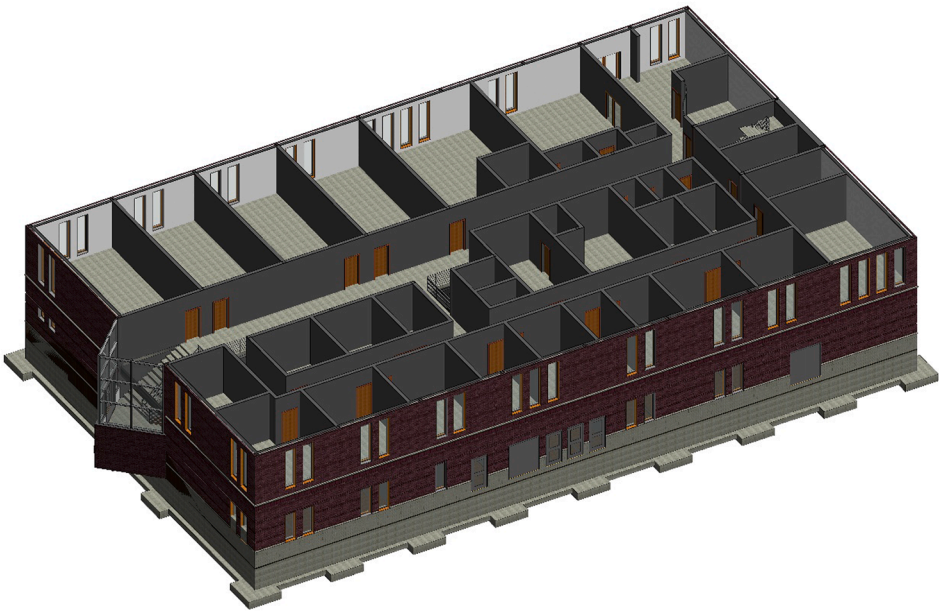


**Fig. 6.** The 3D model of the steel structure building project (roof was sectioned to provide internal details).

composite concrete slab), a steel-framed office building, a reinforced concrete-framed residential building, a road widening project, and a steel-framed hotel building. For instance, Fig. 6 shows the 3D model for the steel-framed building. Table 1 shows developed samples which are represented by B, C, S, M and H, respectively.

### 4.1. Results

This section discusses the overall similarity scores measured by three metrics of node, structural, and total similarity. These results are explored in two parts: compliance of the results with similarity measure properties and performance of the method in the retrieval process. In the later part, the WBS samples were sorted based on their calculated similarity scores to find the most similar projects to the queried sample. This evaluation was performed based on precision and recall measures. In addition, the effect of the variations of threshold in measuring node to node similarity metrics was examined. The results were obtained through experiments in which the threshold varied in the range of 0.50 to 0.80 with 0.05 intervals.

Before discussing the results, it is important to explain the precision and recall measures. These two terms can be defined based on the binary relevance judgment in which every retrievable sample is recognizably "relevant" or "not relevant" [5]. Hence, in a search result, each sample is placed in only one of the pair of groups, in which the samples are "relevant" or "not relevant" and "retrieved" or not "retrieved" [5].

For any given retrieved set of items, recall is defined as the number of retrieved relevant items as a proportion of all relevant items. In other words, recall is a measure of performance in including relevant items in the retrieved set. Precision is defined as the number of retrieved relevant items as a proportion of retrieved items. Therefore, precision is a measure of excluding the nonrelevant items from the retrieved set [5].

### 4.2. Properties of similarity measures

The similarity measurements must fulfil the properties of symmetry and reflexivity [37,10]. A similarity function S: $S \times S \rightarrow [0,1]$ on a set S measuring the degree of similarity between two elements, is called similarity measure if, $\forall X, Y \in S$ (see Eqs. (24) and (25)).

$$Sim(X,Y) = Sim(Y,X) \ \ [\text{symmetry}] \tag{24}$$

$$Sim(X,X) = 1 \ \ [\text{reflexivity}] \tag{25}$$

#### 4.2.1. Symmetry

To determine the symmetry fulfilment, the symmetry error for two WBSs, such as A and B, is computed using Eq. (26). In this equation, the *sim* can be one of the three overall similarity measurements (total similarity, node similarity or structural similarity).

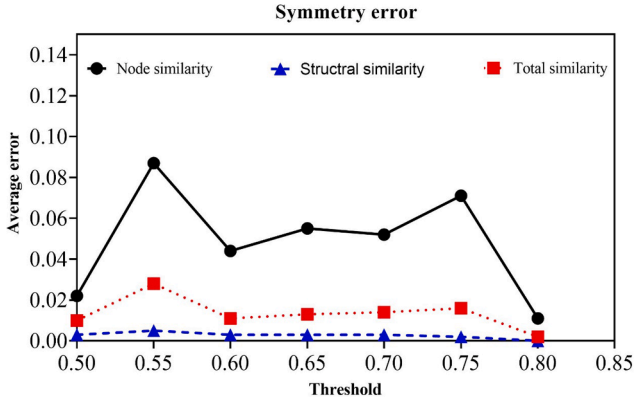



**Fig. 7.** The average of the symmetry errors.

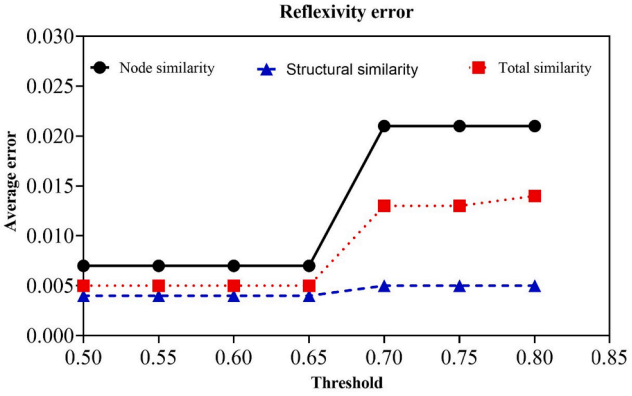



**Fig. 8.** The average of the reflexivity errors.

$$Symmetry \ \ error \ \ (A,B) = \frac{|sim(A,B) - sim(B,A)|}{average[sim(A,B), sim(B,A)]} \tag{26}$$

The symmetry errors of all the possible pairwise comparisons from the test samples were measured and averaged for different overall similarity metrics. Fig. 7 presents the averages of the symmetry errors. As it can be observed in Fig. 7, the node similarity had larger errors, and the structural similarity measurement performed better than the other two metrics in the symmetry analysis.

#### 4.2.2. Reflexivity

The reflexivity error (e.g. for the sample $B_1$) is calculated using Eq. (27) [37]:

$$Reflexivity \ \ error = 1 - sim(B_1, B_1) \tag{27}$$

The reflexivity errors were obtained by comparing the test samples with themselves. The average values of the reflexivity errors are presented in Fig. 8. The results confirm that the system generated promising results and the reflexivity errors were negligible across the various thresholds.

### 4.3. Retrieval precision and recall

This section discusses the performance of the similarity metrics in retrieving the similar stored cases to the query sample (given sample). For this purpose, the samples $B_1$, $C_1$, $S_1$, $M_1$, and $H_1$were chosen as the query samples and the rest of samples were considered as the stored

**Table 2**
Comparing $B_1$ with the stored samples using the threshold of 0.65.

| Query sample | Documented sample | Total similarity score | Node similarity score | Structural similarity score |
|---|---|---|---|---|
| $B_1$ | $B_2$ | 0.72 | 0.64 | 0.80 |
| $B_1$ | $S_2$ | 0.69 | 0.56 | 0.81 |
| $B_1$ | $B_3$ | 0.63 | 0.48 | 0.79 |
| $B_1$ | $C_1$ | 0.60 | 0.47 | 0.73 |
| $B_1$ | $S_1$ | 0.59 | 0.40 | 0.78 |
| $B_1$ | $H_3$ | 0.56 | 0.39 | 0.73 |
| $B_1$ | $S_3$ | 0.56 | 0.35 | 0.76 |
| $B_1$ | $H_2$ | 0.56 | 0.42 | 0.69 |
| $B_1$ | $H_1$ | 0.54 | 0.34 | 0.74 |
| $B_1$ | $C_2$ | 0.53 | 0.39 | 0.67 |
| $B_1$ | $C_3$ | 0.51 | 0.34 | 0.68 |
| $B_1$ | $M_2$ | 0.44 | 0.13 | 0.75 |
| $B_1$ | $M_3$ | 0.43 | 0.13 | 0.74 |
| $B_1$ | $M_1$ | 0.40 | 0.05 | 0.74 |

samples. The reason for choosing different types of samples as query samples was to study the effect of various types of test samples in manual task labelling and structuring of WBSs.

The performance of the three overall similarity metrics was evaluated by the precision score in retrieving the relative stored samples. For this purpose, each query sample ($B_1$, $C_1$, $S_1$, $M_1$, and $H_1$) was compared with all the stored samples and the results were ranked from the highest to the lowest similarity score. For instance, Table 2 shows the results with the thresholds equals to 0.65. In this table, the $B_1$ was the query sample and the results are ordered by the Total similarity score.

The highest similarity score belongs to a relevant sample ($B_2$); however, the second score in this list belongs to a non-relevant sample ($S_2$). The limited number of relevant samples in our test case causes the retrieving process highly competitive. One of the two relevant samples received the highest ($B_2$) and the other one had the third ($B_3$) similarity score.

The retrieving precision is calculated using Eq. (28) which measures the number of retrieved relevant samples as a proportion of retrieved samples. In this part, the relevance argument is not challenging, as the sample tests were identified as relevant if they were developed for the same project by different experts. Therefore, for each query sample, two relevant samples exist among the stored samples. For example, the samples $B_2$ and $B_3$ are the relevant samples to the query sample $B_1$.

$$Retrieving\ \ precision = \frac{|\{Relevant\ \ samples\} \cap \{Retrived\ \ samples\}|}{|\{Retrieved\ \ samples\}|} \quad (28)$$

The number of retrieved samples in each query is determined by the recall score, where the retrieval process continues until the number of retrieved relevant samples fulfil the recall score. Since there was a small set of stored WBS samples in this study, only two recall thresholds (i.e. 0.5 and 1) were considered. In the recall score of 0.5, retrieval of stored samples continues until one of the two relevant samples is retrieved. The other tested recall score was 1.0, in which both relevant samples should be retrieved.

For example, the precision scores of retrieving $B_1$ were obtained as follow: For the recall score of 1.0, the retrieval process continues until both relevant samples ($B_2$ and $B_3$) to $B_1$ were retrieved. As we can see in Table 2, this results in two retrieved relative samples out of three retrieved samples ($B_2$, $S_2$, and $B_3$) and the precision is 0.66 (see Eq. (30)). For the recall score of 0.5, only one of the relative samples to the $B_1$ must be retrieved, which was achieved by retrieving only the first sample from Table 2 and it results in a precision score of one (see Eq. (29)). The average values of the precision scores are presented in Fig. 9, in which it is evident that the Structural similarity measurement provides higher precision scores than the Total and Node similarity metrics. In particular, it provides the highest precision in the thresholds between 0.7 and 0.75. The reason is that the structural similarity considers the way that the WBSs are structured in addition to the semantics of the

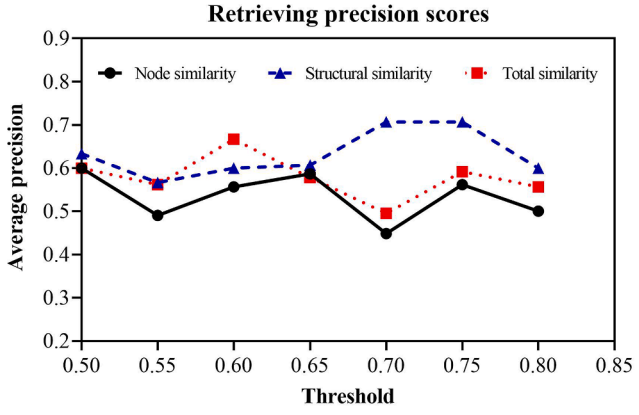



**Fig. 9.** Average precision scores.

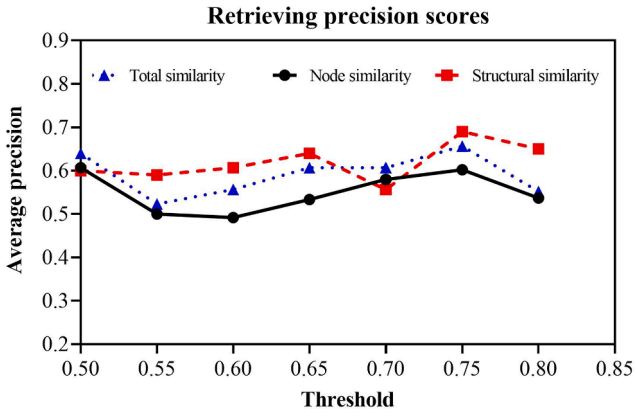



**Fig. 10.** Average precision scores with a higher weight for semantic similarity.

tasks within the WBSs. As a result, it provides a more comprehensive metric. The average precision in this range was about 0.7, which means that the two relevant samples were mostly among the top three retrieved cases. This approach provides the opportunity to the users to assess the top-ranked retrieved items and use the most relevant one(s).

$$Recall = 0.5, Retrieving\ \ precision = \frac{|\{B_2\} \cap \{B_2\}|}{|\{B_2\}|} = \frac{1}{1} = 1.0 \quad (29)$$

$$Recall = 1.0, Retrieving\ \ precision = \frac{|\{B_2, B_3\} \cap \{B_2, S_2, B_3\}|}{|\{B_2, S_2, B_3\}|} = \frac{2}{3} = 0.66 \quad (30)$$

The weight of semantic similarity measurement (i.e. $sim_{nodes}$) was increased from one to two in Eq. (12) to investigate the effect of the weights in this equation on the average retrieving precision. Fig. 10 shows the results of average retrieving precision with the increased weight of semantic similarity in Eq. (12). As illustrated in Fig. 10, the precision scores were affected (reduced in lower thresholds), and the structural similarity with thresholds around 0.75 had the highest precision scores.

These WBS samples were developed according to the Project Management Institute's guidelines to encompass the entire scope of the project [32]; however, this might not be followed in all actual construction projects which can hinder the performance of the proposed method. In addition, performance of the developed system heavily depends on the employed semantic network, which was WordNet in this research. Since this vocabulary source is a generic database for English language, it might occasionally fail to provide desirable results in technical domains such as construction. Thereby, it would be valuable to develop and use a customized semantic network for construction, which would be able to match the related concepts in this domain.

### 4.4. Sample application

The proposed method can be used to develop a comprehensive WBS for construction projects during the planning stage. For example, a preliminary WBS can be developed by junior schedulers for a construction project, then the proposed method runs a query to find WBS of similar past project(s) to complete the preliminary WBS. To evaluate the performance of the proposed method in this context, the sample $S_3$ was altered in which 16 tasks were omitted (out of 53) and six of the remining were reworded. The omitted tasks were mostly included operations that might not be visible in the project outcome, such as surveying and concrete curing, and can be missed by the novice schedulers. This experiment was performed to represent a scenario where an incomplete WBS is provided to find the similar projects. Table 3 shows the results of comparing altered $S_3$ ($S_3$_a) with the stored samples. The results for four different thresholds are sorted from the

**Table 3**
Comparing $S_3$_a with the stored samples using different thresholds.

| Quired sample | Thresholds | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0.60 | | 0.65 | | 0.70 | | 0.75 | |
| | Retrieved sample | Total similarity | Retrieved sample | Total similarity | Retrieved sample | Total similarity | Retrieved sample | Total similarity |
| S3_a | $S_1$ | 0.69 | $S_1$ | 0.66 | $S_1$ | 0.64 | $H_3$ | 0.62 |
| | $S_2$ | 0.64 | $S_2$ | 0.63 | $H_3$ | 0.63 | $S_1$ | 0.56 |
| | $B_2$ | 0.63 | $H_3$ | 0.63 | $S_2$ | 0.62 | $S_2$ | 0.52 |
| | $H_3$ | 0.63 | $B_2$ | 0.60 | $B_1$ | 0.51 | $B_1$ | 0.48 |
| | $B_1$ | 0.60 | $B_1$ | 0.55 | $B_2$ | 0.51 | $H_1$ | 0.48 |
| | $B_3$ | 0.58 | $H_1$ | 0.55 | $H_1$ | 0.50 | $B_3$ | 0.47 |
| | $H_1$ | 0.56 | $B_3$ | 0.53 | $B_3$ | 0.50 | $B_2$ | 0.46 |
| | $M_3$ | 0.53 | $C_1$ | 0.49 | $C_1$ | 0.48 | $C_3$ | 0.45 |
| | $M_2$ | 0.51 | $C_3$ | 0.48 | $C_3$ | 0.48 | $H_2$ | 0.42 |
| | $C_1$ | 0.50 | $H_2$ | 0.47 | $H_2$ | 0.46 | $C_1$ | 0.41 |
| | $C_3$ | 0.48 | $M_3$ | 0.44 | $C_2$ | 0.43 | $C_2$ | 0.38 |
| | $H_2$ | 0.48 | $C_2$ | 0.44 | $M_2$ | 0.39 | $M_2$ | 0.37 |
| | $C_2$ | 0.45 | $M_2$ | 0.42 | $M_1$ | 0.36 | $M_1$ | 0.34 |
| | $M_1$ | 0.42 | $M_1$ | 0.39 | $M_3$ | 0.35 | $M_3$ | 0.34 |

largest to the lowest similarity based on the total similarity metric. The results show that $S_1$ and $S_2$ were among the top three retrieved cases in all thresholds, which indicates that the method was able to retrieve the similar projects to an incomplete WBS with a precision score between 0.66 and 1.00, depending on the applied threshold.

## 5. Conclusion

Reuse of the knowledge and experiences gained from completed construction projects can improve planning of the new projects. In order to reuse knowledge, finding similar past projects is critical. This research was undertaken to develop quantitative similarity metrics, to measure the similarity of construction projects using the WBS as their representative. These metrics were implemented using NLP techniques written in Python programming language. The similarity metrics were evaluated based on two sets of experiments: First the metrics were tested for the similarity properties fulfilment, including symmetry and reflexivity; second, the metrics were tested to search among test samples and to find the relevant cases to the given samples.

The results show promising outcomes in compliance with similarity properties (i.e. symmetry and reflexivity) with small errors. The results on the second part of the experiments, which were the main focus of this research, revealed that the structural similarity metric had the best performance in retrieval of similar projects with thresholds in the range of 0.7 to 0.75.

## 6. Future works

The proposed method could identify similar projects using their WBSs. But the future research can investigate inclusion of major quantitative attributes, such as work quantity of the tasks and their duration, to enhance the similarity assessment of the construction projects. The vocabulary source in this research (i.e. WordNet) is a general and comprehensive source, and might not be able to provide flawless similarity assessments for some technical terms. Developing a specialized resource for construction technical words is a valuable opportunity for future research. Lastly, this study focused on development of a method to quantify the similarity of construction projects using their WBS and did not explore retrieval of the information and documents of projects. The future work can investigate integration of the proposed method with the existing knowledge retrieval systems, namely case-based reasoning.

## 7. Data availability

The source code of the developed program (in Python programming language) is publicly available and can be found at: https://osf.io/b8qvy/ Data generated or analysed during the experiments are available from the corresponding author upon reasonable request.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgement

This research project was funded by Discovery grant RGPIN-2015-03812 from Natural Sciences and Engineering Research Council of Canada.

## References

[1] M. Al Qady, A. Kandil, Concept relation extraction from construction documents using natural language processing, J. Construct. Eng. Manage. 136 (3) (2010) 294–302.
[2] M. Alavi, D.E. Leidner, Knowledge management and knowledge management systems: conceptual foundations and research issues, MIS Quarterly (2001) 107–136.
[3] S.H. An, G.H. Kim, K.I. Kang, A case-based reasoning cost estimating model using experience by analytic hierarchy process, Build. Environ. 42 (7) (2007) 2573–2579.
[4] T. Bray, J. Paoli, C.M. Sperberg-McQueen, E. Maler, 2000. Extensible Markup Language (XML) 1.0. W3C Recommendation 6 October 2000. Available via the World Wide Web at http://www.w3.org/TR/1998/REC-xml-19980210.
[5] M. Buckland, F. Gey, The relationship between recall and precision, J. Am. Soc. Inform. Sci. 45 (1) (1994) 12–19.
[6] S.H. Chen, A.J. Jakeman, J.P. Norton, Artificial intelligence techniques: an introduction to their use for modelling environmental systems, Math. Comput. Simul 78 (2–3) (2008) 379–400.
[7] G.G. Chowdhury, Natural language processing, Ann. Rev. Inform. Sci. Technol. 37 (2003) 51–89.
[8] R. Costa, C. Lima, J. Sarraipa, R. Jardim-Gonçalves, Facilitating knowledge sharing and reuse in building and construction domain: an ontology-based approach, J. Intell. Manuf. 27 (1) (2016) 263–282.
[9] R. Dijkman, M. Dumas, B. Van Dongen, R. Käärik, J. Mendling, Similarity of business process models: metrics and evaluation, Inform. Syst. 36 (2) (2011) 498–516.
[10] M. Ehrig, A. Koschmider, A. Oberweis, Measuring similarity between semantic business process models, in: Proceedings of the fourth Asia-Pacific Conference on Comceptual modelling, vol. 67, 2007, pp. 71–80.
[11] S. Gasik, A model of project knowledge management, Project Manage. J. 42 (3) (2011) 23–44.

[12] Y.M. Goh, D.K.H. Chua, Case-based reasoning approach to construction safety hazard identification: adaptation and utilization, J. Construct. Eng. Manage. 136 (2) (2010) 170–178.
[13] W.H. Gomaa, A.A. Fahmy, A survey of text similarity approaches, Int. J. Comput. Appl. 68 (13) (2013) 13–18.
[14] J. Hahn, M. Subramani, A framework of knowledge management systems: issues and challenges for theory and practice, ICIS 2000 Proc. 28 (2000).
[15] P.E. Hart, N.J. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE Trans. Syst. Sci. Cybernetics 4 (2) (1968) 100–107.
[16] G.T. Haugan, Effective Work Breakdown Structures, Berrett-Koehler Publishers, 2001.
[17] Y.M. Ibrahim, A.P. Kaka, E. Trucco, M. Kagioglou, A. Ghassan, Semi-automatic development of the work breakdown structure (WBS) for construction projects. In: Proceedings of the 4th International SCRI Research Symposium, Salford, UK, 2007.
[18] Y. Jung, S. Woo, Flexible work breakdown structure for integrated cost and schedule control, J. Construct. Eng. Manage. 130 (5) (2004) 616–625.
[19] D. Jurafsky, J.H. Martin, 2014. Speech and Language Processing, vol. 3.
[20] G.H. Kim, S.H. An, K.I. Kang, Comparison of construction cost estimating models based on regression analysis, neural networks, and case-based reasoning, Build. Environ. 39 (10) (2004) 1235–1242.
[21] J.L. Kolodner, An introduction to case-based reasoning, Artif. Intell. Rev. 6 (1) (1992) 3–34.
[22] T.K. Landauer, S.T. Dumais, A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge, Psychol. Rev. 104 (2) (1997) 211.
[23] V.I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet Physics Doklady, vol. 10, no. 8, 1966, pp. 707–710.
[24] H.T. Lin, N.W. Chi, S.H. Hsieh, A concept-based information retrieval approach for engineering domain-specific technical documents, Adv. Eng. Inf. 26 (2) (2012) 349–360.
[25] A. Maedche, S. Staab, Measuring similarity between ontologies. In: International conference on knowledge engineering and knowledge management, Springer, Berlin, Heidelberg, 2002, pp. 251–263.
[26] M.L. Maher, A.G. de Silva Garza, Developing case-based reasoning for structural design, IEEE Expert 11 (3) (1996) 42–52.
[27] L. Meng, R. Huang, J. Gu, A review of semantic similarity measures in wordnet, Int. J. Hybrid Inform. Technol. 6 (1) (2013) 1–12.
[28] R. Mihalcea, C. Corley, C. Strapparava, July). Corpus-based and knowledge-based measures of text semantic similarity, Aaai 6 (2006) (2006) 775–780.
[29] E. Mikulakova, M. König, E. Tauscher, K. Beucke, Knowledge-based schedule generation and evaluation, Adv. Eng. Inf. 24 (4) (2010) 389–403.
[30] G.A. Miller, WordNet: a lexical database for English, Commun. ACM 38 (11) (1995) 39–41.
[31] J. Park, H. Cai, WBS-based dynamic multi-dimensional BIM database for total construction as-built documentation, Autom. Constr. 77 (2017) 15–23.
[32] PMBOK® Guide, 2017. Sixth edition, Project Management Institute.
[33] Princeton University "About WordNet." WordNet. Princeton University, 2010.
[34] Y. Qiao, J.D. Fricker, S. Labi, Quantifying the similarity between different project types based on their pay item compositions: application to bundling, J. Construct. Eng. Manage. 145 (9) (2019) 04019053.
[35] B. Raphael, B. Domer, S. Saitta, I.F. Smith, Incremental development of CBR strategies for computing project cost probabilities, Adv. Eng. Inf. 21 (3) (2007) 311–321.
[36] P. Resnik, Using information content to evaluate semantic similarity in a taxonomy. arXiv preprint cmp-lg/9511007, 1995.
[37] M.M. Richter, Classification and learning of similarity measures, in: Information and Classification, Springer, Berlin, Heidelberg, 1993, pp. 323–334.
[38] H.G. Ryu, H.S. Lee, M. Park, Construction planning method using case-based reasoning (CONPLA-CBR), J. Comput. Civil Eng. 21 (6) (2007) 410–422.
[39] G. Salton, M.E. Lesk, Computer evaluation of indexing and text processing, J. ACM (JACM) 15 (1) (1968) 8–36.
[40] E. Siami-Irdemoosa, S.R. Dindarloo, M. Sharifzadeh, Work breakdown structure (WBS) development for underground construction, Autom. Constr. 58 (2015) 85–94.
[41] J.F. Sowa, Semantic networks. John_Florian_Sowa isi [2012-04-20 16: 51]> Author [2012-04-20 16: 51], 2012.
[42] M. Sutrisna, C.D. Ramanayaka, J.S. Goulding, Developing work breakdown structure matrix for managing offsite construction projects, Arch. Eng. Des. Manage. 14 (5) (2018) 381–397.
[43] J.H.M. Tah, V. Carr, R. Howes, Information modelling for case-based construction planning of highway bridge projects, Adv. Eng. Softw. 30 (7) (1999) 495–509.
[44] H.C. Tan, P.M. Carrillo, C.J. Anumba, N. Bouchlaghem, J.M. Kamara, C.E. Udeaja, Development of a methodology for live capture and reuse of project knowledge in construction, J. Manage. Eng. 23 (1) (2007) 18–26.
[45] H.P. Tserng, Y.C. Lin, Developing an activity-based knowledge management system for contractors, Autom. Constr. 13 (6) (2004) 781–802.
[46] A. Tversky, Features of similarity, Psychol. Rev. 84 (4) (1977) 327.
[47] Y.R. Wang, G.E. Gibson Jr, A study of preproject planning and project success using ANNs and regression models, Autom. Constr. 19 (3) (2010) 341–346.
[48] Z. Wu, M. Palmer, 1994. Verb semantics and lexical selection. arXiv preprint cmp-lg/9406033.
[49] J. Zhang, N.M. El-Gohary, 2013. Information transformation and automated reasoning for automated compliance checking in construction. In: Computing in civil engineering, 2013, pp. 701–708.
[50] J. Zhang, N.M. El-Gohary, Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking, J. Comput. Civil Eng. 30 (2) (2016) 04015014.