Semantic Similarity Measurement of Construction Projects using WBS-Based Similarity Metrics

by

Navid Torkanfar

A thesis
submitted to the Faculty of Graduate Studies
in partial fulfilment of the requirements for the
Degree of Master of Science
in

Civil Engineering

Supervisor
Ehsan Rezazadeh Azar
Associate Professor – Dept. of Civil Engineering

Lakehead University
Thunder Bay, Ontario
June 2019

Author’s Declaration Page

I hereby declare that I am the sole author of the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Lessons learned and the knowledge gained from previous projects could save a considerable amount of time and budget in planning and construction of future projects. In the process of knowledge and experiment reuse, finding the most similar case(s) to the current project is critical and therefore, a number of methods have been developed which use different variables to represent each specific sub-area of knowledge and also to measure the similarity of the documented cases to the current project.

It is hypothesized that the hierarchy of project activities, which is represented as Work Breakdown Structure (WBS) of the project, encompasses the entire scope of the project and contains the necessary information to measure the semantic similarity of construction projects. Thus, WBS could be used as an appropriate representative of the projects. In this research project, a novel method is proposed to assess the semantic similarity of projects by means of Natural Language Processing (NLP) techniques. In this method, the current project is compared with the documented as-built projects based on their WBS and the most similar ones to the current project are retrieved.

The proposed WBS similarity measurement is implemented using two metrics, (1) node similarity that compares the semantics of elements in two WBSs; (2) structural similarity which compares the topology of Work Breakdown Structures. The proposed processes to estimate each of these two metrics produce a similarity score between 0 and 1. The average of these two scores provides the final similarity score between two WBSs. The method was tested using nine WBS test samples with promising results in compliance with similarity properties. Finally, the metrics were experimentally evaluated in terms of precision and recall. The results showed that the structural similarity slightly outperformed the other metric.

Acknowledgments

Over the past two years of master study at Lakehead University, I have received support and encouragement from many individuals. First and foremost, I would like to express my deepest appreciation to my supervisor, Prof. Ehsan Rezazadeh Azar, for his patient guidance. During writing, he spent countless hours on proofreading my thesis, providing constructive suggestions, and helping me whenever I had issues on my thesis. I have learned much through his teaching. I could not reach this stage without his supervision. It is such an honor to be his student.

I would also like to thank my friends and faculty members of the civil engineering department at Lakehead University for all the encouragements and supports they provided to me.

Finally, I must express my very profound gratitude to my parents and to my girlfriend Sargol for providing me with unfailing support, love and continuous encouragement through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you

Table of contents

List of Figures

List of Tables

List of Abbreviations

AEC	Architecture, Engineering, Construction
AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
BIM	Building Information Modeling
CAD	Computer Aided Design
CBR	Case-Based Reasoning method
HVAC	Heating, Ventilation, Air Conditioning
IFC	Industry Foundation Classes
KM	Knowledge Management
LSA	Latent Semantic Analysis
LSP	Least Similar Parent
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
PKF	Project Knowledge File
PMI	Project Management Institute
RDF	Resource Description Framework
WBS	Work Breakdown Structure
WSD	Word Sense Disambiguation
XML	Extensible Markup Language

Chapter 1: Introduction

1.1. Background and research motivation

Construction managers and project planners typically make their decisions based on the knowledge gained from previous projects (Tah et al. 1999). Effective reuse of the gained knowledge can improve construction processes and reduce the time and cost of solving problems (Tserng and Lin 2004). If the experience and knowledge gained in projects are electronically captured and shared, then the same issues should not be faced repeatedly. By reusing lessons learned, the need to refer to the similar past projects could be increased, the efforts to solve current issues could be reduced, and effective solutions could be undertaken to address the problems (Tserng and Lin 2004).

Research efforts on reusing the gained knowledge have a long history in construction and project management domains, which are explained in Chapter 2. One of the important approaches in this area is case-based reasoning (CBR), which is a technique for problem solving by recalling similar past problems (Riesbeck and Schank 2013), and the solution of the similar past cases could be adapted to the new problem (Chen et al. 2008). The main limitation of CBR is due to applicability of proposed methods only on retrieving small documents with specific features. Defining the right features to appropriately differentiate the cases has been a main challenge for researchers.

More recently, “knowledge management” and “semantic webs” methods have been used for the retrieval of knowledge in construction domain, but reusing the knowledge was not the main purpose of these research studies and a solid method for knowledge retrieval has not been introduced yet.

Since a comprehensive extraction of domain knowledge is difficult, usually the domain knowledge is broken down into a hierarchy of interrelated problem elements and the research efforts have focused on storage and retrieval of small part of the entire documents and knowledge obtained from past projects, such as the schedules, drawings, designs, and cost estimation data. These approaches do not consider the entire scope of construction projects to estimate the similarity of projects. In other words, the problem is defined as a small part of the bigger problem (which is the entire construction project) in the retrieval process. In this study, however, the definition of the problem is changed from the small sub-areas to the entire scope of the projects.

Therefore, a comprehensive method is needed to compare a representative form of the documented projects with the existing project, and order the documented projects based on their similarity to the current project. Since the Work Breakdown Structure (WBS) of a project should include the total scope of the project work (Project Management Institute 2017), WBS is considered as a potential representative of construction projects. Therefore, a novel method is proposed to measure the semantic similarity of the WBSs of two projects, which calculates a score between 0 and 1 to determine the semantic similarity of two WBSs.

1.2. Research objectives

The main objective of this research project is to develop a method which compares the WBS of a new construction project with the WBSs of documented projects to find the most similar projects. The overall scope of this thesis is to peruse the following objectives:

1. Developing WBS sample tests to evaluate the proposed method.
2. Encoding WBS of projects into a computer readable language.

3. Developing semantic similarity measurements based on Natural Language Processing (NLP) techniques to compare a given WBS with the WBS of all documented projects to find the most similar project(s).
4. Evaluating performance of the retrieval results based on a set of experiments.

1.3. Research methodology

The schematic workflow of this research is presented in Figure 1. It consists of three main phases: 1) Encoding WBS of projects to computer readable language; 2) developing metrics to compare the semantic and structural similarity of WBSs; 3) calculating a similarity score between 0 and 1 which determines the similarity of two WBSs and their corresponding projects. Details of the proposed methods are presented in Chapter 3.

1.4. Thesis organization

This thesis consists of five chapters. Chapter 1 introduces the background, main concepts, and principles of knowledge reuse and documentation of previously completed projects. Then, it explains research motivations and objectives of this thesis, as well as a summary of the methodology. Chapter 2 presents the literature review related to this thesis. It discusses different methods which have been proposed for knowledge reuse in the construction domain. Afterward, the details of the methods used to develop this framework are described in Chapter 3. In this chapter, the details of the proposed metrics to compare two WBSs are explained. Chapter 4 presents development of WBS test samples and the results of experiments to evaluate the performance of this system. Chapter 5 summarizes the findings of this research, highlights its limitations, and provides recommendations for future developments.

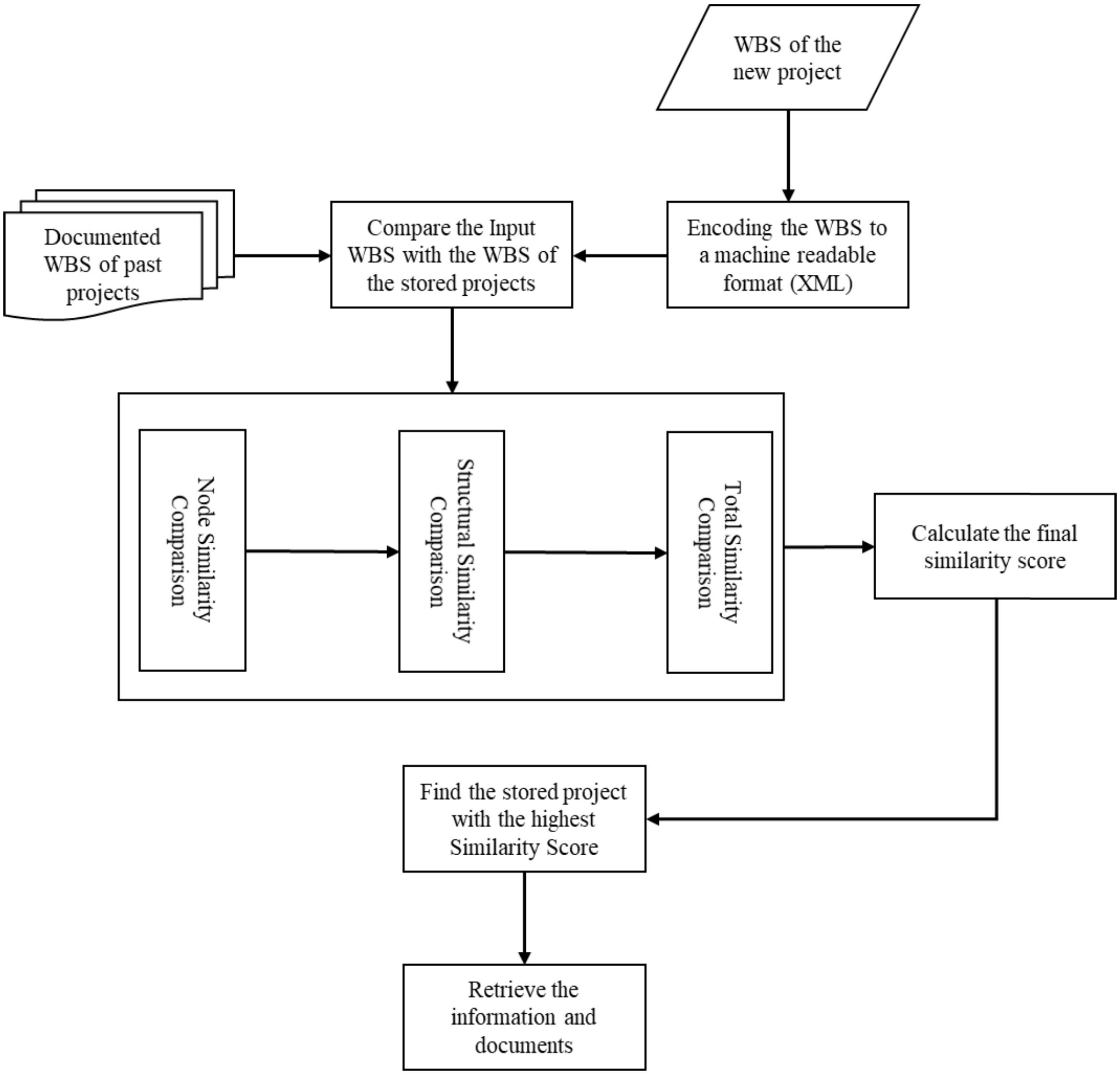


Figure 1. Schematic workflow of the research

Chapter 2: Literature Review

2.1. Introduction

After completion of each construction project, a large amount of knowledge and experience will be gained. They include information and documents which are typically generated during the progress of the project. Forming of the knowledge starts from the very first step, when the idea of the project is created, to the very last step, which is the commissioning of the facility (Kamara et al. 2002).

The knowledge can be classified as tacit or explicit (Addis 2016). The expression of tacit knowledge in a formal language is difficult and this knowledge is stored in the mind of employees and managers involved in a project (Tserng and Lin 2004). Explicit knowledge can be represented, stored, shared, and effectively applied (Tserng and Lin 2004). Explicit knowledge is documented by organizations in different forms, such as reports, articles, manuals, patents, audios, videos, software, and etc. (Tiwana 2000).

Once a project is completed, involved organizations might gradually lose related data and information over time (Tah et al. 1999). Without a systematic approach for storing documented knowledge, this information, including the tacit and explicit knowledge, can be lost after a period of time (Tah et al. 1999).

Although the memory of individuals involved in each project stores the knowledge, human memory might fade away over time. In addition, it is possible that organizations lose the information stored in the memory of their expert employees when they leave the company (Tah et al. 1999; Joe et al. 2013) . As a result, reliable methods are needed to capture the lessons learned and to retrieve them in future similar cases.

The tacit knowledge, which is stored in employees' mind, could result in a loss of valuable project-related experiences and knowledge. Moreover, the information gathered by a construction manager could not be used by others, which would result in repeating mistakes that could have been avoided if the appropriate knowledge was transferred.

This chapter discusses the three important areas in construction research domain which mostly focused on knowledge reuse. These topics include "case-based reasoning", "knowledge management in construction", and more recently the development of "semantic webs" in construction. After clarifying the gap in the literature, the definition of WBS and its features are discussed.

2.2. Case-based Reasoning

Artificial Intelligence (AI) can solve complex problems by attempting to copy human perception, learning, and reasoning (Chen et al. 2008). There are different AI techniques, such as case-based reasoning, rule-based systems (Hayes-Roth 1985), neural networks and deep learning, fuzzy models, genetic algorithms, cellular automata, multi-agent systems, swarm intelligence, reinforcement learning, and hybrid approaches (Chen et al. 2008).

Case-based reasoning (CBR) is a technique for problem solving by recalling similar past problems (Riesbeck and Schank 2013), and the solution of the similar past cases could be adapted to the new problem (Chen et al. 2008). This process involves four steps (Figure 2): 1) retrieving the most similar past case(s) from the database; 2) making a solution for the new problem based on the retrieved case(s); 3) revising of the retrieved solution; and 4) retaining the solution for future applications.

Retrieval process measures the similarity of the new case and the past cases by the means of syntactical (grammatical structure) or semantic (meaning) similarity (Chen et al. 2008). The syntactical similarity is superficial and can be easily applied (Chen et al. 2008). Advanced CBRs use different methods for semantic similarity according to the context of the base. The main retrieval methods are the nearest-neighbor, inductive, and knowledge-guided techniques (Salem 2000).

Nearest-neighbor retrieval method looks for a case with the most common features with the new case and weights them according to their importance. Assigning reasonable weights to the features is the most challenging task. The retrieval time depends on database size and the method could time consuming (Watson and Marir 1994). The inductive method finds the most important features that differentiate the cases (Watson and Marir 1994), where the cases are organized in a decision tree based on the chosen features. As a result, it reduces the retrieval time. Knowledge-guided retrieval chooses important features based on the existing knowledge (Watson and Marir 1994).

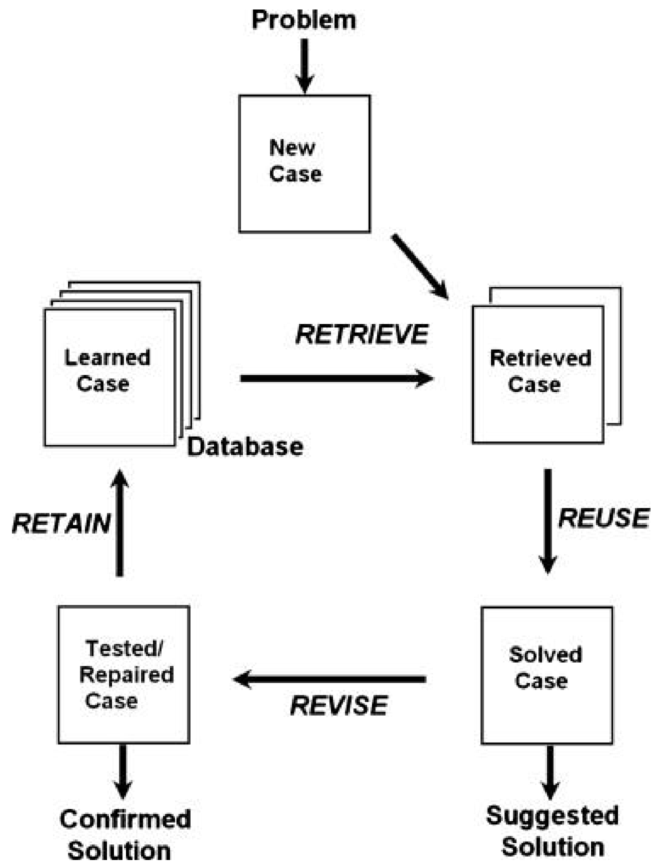


Figure 2. The four steps of the CBR process (Chen et al. 2008)

Rule-based reasoning and neural networks are two other methods in this field, which try to represent the knowledge related to past experiences, but each of these methods has certain limitations. For example, rule-based reasoning systems have a problem in defining an appropriate set of rules for extracting knowledge in the non-experienced domains (Watson 1998), and they lack the capability of self-learning. The major disadvantage of the neural networks is that the knowledge acquisition process is considered as a black box; in other words, the user cannot obtain any information from the software showing the effect of an input variable on the output variable (Yeh 1998).

CBR has been an active research area in the AI research community (Chen et al. 2008). A research done by Marir and Watson discussed a number of prototype applications of CBR developed in various areas, such as knowledge acquisition and refinement, legal reasoning, failure recovery, diagnosis, arbitration, design, general planning, help desk, and teaching and learning. (Marir and Watson 1994)

Case-based planning systems help managers to develop different strategies by reusing past sequence of actions from past plans (Hammond 1989). Juloa (Kolonder 1998), Prodigy/Analogy (Veloso 1992), and CAPlan/CBC (Muñoz-Avila and Huellen 1995) are some examples of such planning approach, in which the main goal of the systems was decomposed into smaller sub-goals, which would lead to a hierarchical representation of plans. This model of plan representation acts as a graphical tree, where each node represents a goal, the child nodes are sub-goals, and the leaf nodes are the actions of the plan. The next section discusses the applications of the CBR in the construction industry.

2.2.1. CBR in the construction industry

The construction industry has been a growing area for various CBR applications. The main areas of interest for these applications include four fields: design-related CBR systems, non-design CBR applications, cost estimation, and CBR in construction planning. SEED software (Flemming et al. 1994) was developed for design applications in which a case-based reasoning system was developed with a large repository of past solutions, and fast derivation of an initial solution for editing or modification. Non-design efforts cover some other building construction fields, such as building regulation interpretation (Yang and Robertson 1994) and contractor prequalification (Ng et al. 1998).

In the cost estimation domain, An et al. (An et al. 2007) proposed a case-based reasoning cost estimation method. As the authors argue in this research, domain knowledge is necessary to answer the following questions: why the retrieved case is similar to the problem, how the similarity measure is calculated, and which attributes have more influence on the similarity measure. Since a comprehensive extraction of the domain knowledge is difficult, this study used Analytic Hierarchy Process (AHP) to elicit the domain knowledge by breaking down the case-based problem into a hierarchy of interrelated problem elements as shown in Figure 3.

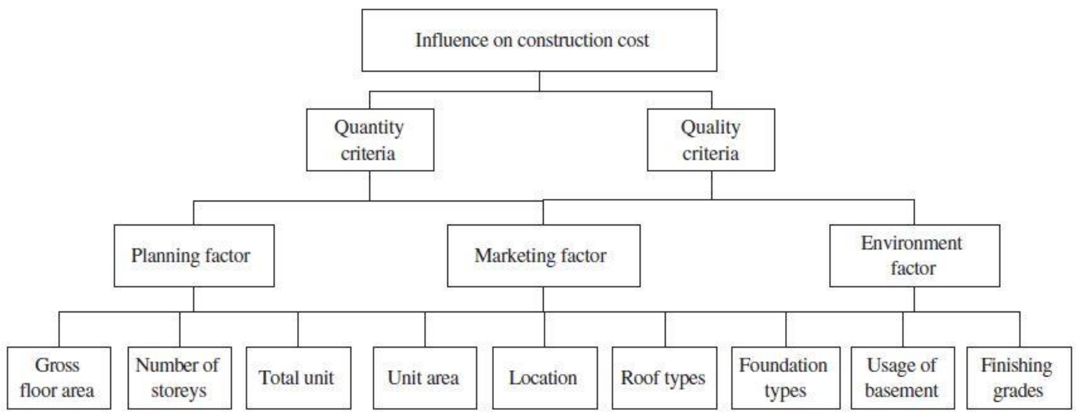


Figure 3. AHP developed for CBR cost estimation (An et al. 2007)

As presented in Figure 3, the authors have considered nine major attributes (gross floor area, number of stories, total unit, unit area, location, roof types, foundation types, usage of the basement, and finish grades), which were obtained from interviews with cost engineers in different construction projects. At the end, by utilizing AHP and pairwise comparison of the attributes to the goal (influence on the construction cost), the influence weight of each attribute was determined. Using these weights, the developed CBR system measures the similarity of the new project with the documented cases and retrieves a similar case(s).

In the planning domain, Tah et al. (Tah et al. 1999) proposed a system to apply CBR to facilitate construction planning. The proposed system contains two levels of information: A database containing information of the past projects and a case-base which contains only essential elements of database information that were filtered into a concentrated form for the purpose of case retrieval, adaption, and reuse.

The case-base is a hierarchical representation of basic components. This tree-like representation is consistent with the project's WBS. For each component node of this hierarchy, several local design features were defined as indices. The function, behavior, and structural design attributes, which were defined by Highway Agency, were adopted to cover design level information. The case retrieval is performed on the case-base, using the defined design level information as a match.

2.2.2. Conclusion

As explained earlier, most of the studies in this area were focused on how different types of construction knowledge, such as cost estimation, planning, and design, are represented, documented, and could be reused. In most of the studies, the retrieval process of CBR was not completely explained. The selected attributes to retrieve similar cases focused on specific features without considering all aspects of the project. The other problem with these methods is that they cannot be extended to different kind of construction projects, for example, a building and highway construction project cannot be distinguished from each other.

As a result, a more comprehensive method is needed to expand the problem in case-based to extract all of the document and knowledge related to the projects. However, comparison of projects with huge amount of data and documents is not practical, and projects can be expressed using a representative and methods could be developed to semantically compare these representatives. This system should find the most semantically matched solution to the input problem. For the semantic comparison, a semantic understanding ability should be provided for the computer. However, one of the challenges is to develop a complete knowledge domain to represent the knowledge of construction projects and to provide the computer with a semantic understanding ability. Next section explores more sophisticated methods of knowledge representation in Knowledge Management (KM) systems and the retrieval methods in these domains.

2.3. Knowledge Management

Knowledge is one of the most important assets in the field of marketing-oriented organization, and its integration across departments and disciplines should be emphasized (Carneiro 2001). Success of an organization highly depends on the availability of the knowledge that it uses in its activities. As Manasco explained, knowledge is a valuable asset that needs to be managed in order to improve organizational business performance (Manasco 1996).

KPMG (Parlby 1998) describes the Knowledge Management (KM) as an organized system to use the knowledge inside an organization to improve the performance of the organization by utilizing the KM's ability to store and use the knowledge. By considering the growing importance of KM in the success and survival of an organization, the necessity of an organized knowledge management is more than before. (Parlby 1998)

Since, the exact differences among knowledge, information, and data could be debatable, there are considerable efforts in the KM literature to define knowledge, information, and data. One of the most popular arguments defines data as raw numbers and facts, information as processed data, and knowledge as validated information (Dretske 1981).

Koenig discusses that the first generation of KM systems explored IT driven knowledge sharing with a focus on "best practices" and "lesson learned" (Koenig 2002), while Rezgui et al. defined the knowledge sharing as "supply-side KM" in which people can reach to the supplied information by means of KM systems (Rezgui et al. 2010). The second generation of KM is knowledge creation, which is defined as "demand-side KM" with the main goal of increasing a capacity to satisfy the growing demands for new knowledge that has not been already captured and stored in a knowledge repository (Turk 2006). Next section explains recent knowledge management innovation and practices in the construction industry.

2.3.1. Knowledge Management in construction industry

Based on a survey from project-based organizations in the construction industry in U.K., more than 50% of participants mentioned that the organizations can take advantage of new technologies developed in knowledge management (Egbu 2002). Another survey in construction industry in U.K. revealed that about 40% of the organizations who participated in the survey already had a KM strategy and another 41% of them had plans to implement this strategy within a year (Carrillo et al. 2003). Although the industry has been aware of the importance of KM, there are a number of limitations in the current practices for capturing and reusing of knowledge.

The Architecture, Engineering, and Construction (AEC) industry is a project-based environment and the knowledge of construction industry is mostly obtained through the delivery of projects based on the client's requirements and objectives (Kamara et al. 2002). Figure 4 shows a simplified model of the construction processes. A construction project starts with establishing a need by the client with a set of requirements. The requirements are converted to appropriate designs and these designs are transformed into a facility during the construction phase.

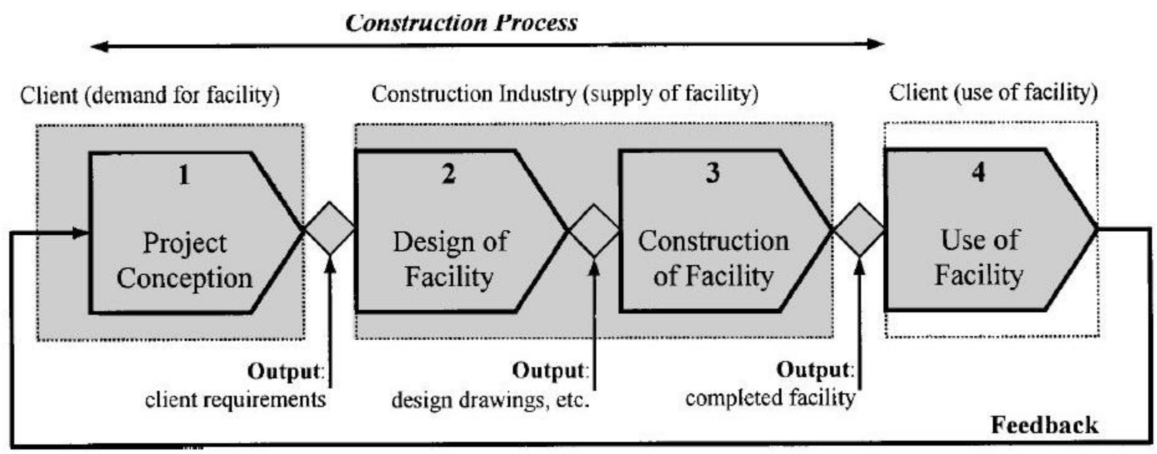


Figure 4. Simplified model of the construction processes (Kamara et al. 2002)

The AEC is a knowledge intensive industry and it is difficult to communicate effectively and efficiently among partners, which would result in a fragmented industry (Rezgui et al. 2010). This fragmented nature causes low efficiency in project delivery which could result in dissatisfaction of clients and low profitability for the construction firms (Carrillo et al. 2000). An effective project knowledge management can improve this issue through effective application of the lessons learned.

In addition, effective management of the obtained knowledge could initiate many innovations in this industry. Tan et al. argued (Tan et al. 2007) that effective management of knowledge in an organization can help prevent the “reinvention of wheel” and repetition of the same mistakes. An organization can focus on lessons learned to improve problem solving which provide a basis for future innovations and better solutions (Tan et al. 2007).

2.3.2. Knowledge Management Stages

According to various studies, Knowledge Management systems consist of several stages. Bergmann proposes that knowledge and experiences should be preserved and managed through the following steps: capturing, modeling, storing, retrieving, adapting, evaluating, and maintaining

(Bergmann 2002). On the other hand, Kululanga and McCaffer considers the issues of knowledge obsolescence in KM and proposed four main KM processes within the context of construction: capturing the knowledge, sharing knowledge, reusing knowledge, and maintaining knowledge (Kululanga and McCaffer 2001).

Each of these processes has several steps in common, such as knowledge capturing, knowledge sharing, and knowledge reuse. Knowledge capturing step, which is common in all the proposed procedures, is composed of three sub-processes: identifying and locating the knowledge, representing and storing knowledge, and validating it (Tan et al. 2007). The knowledge sharing step is about submitting the right knowledge to the right parties (Heisig et al. 2001). The Knowledge reusing step focuses on reusing stored knowledge in future cases with the purpose of reapplication of best practice (Szulanski 2000), or innovation with necessary adaptions (Majchrzak et al. 2004).

To keep the knowledge in each KM system updated, maintaining the knowledge is necessary. The knowledge can become expired due to the employment of new information, rules, and theories (Bhatt 2001). Maintaining the knowledge stage aims to review, correct, update, and refine the knowledge to keep it updated (Rollett 2003).

2.3.3. Knowledge reuse in Knowledge Management

The reuse of knowledge in KM means the reapplication of gained knowledge, such as reapplication of best practices (Szulanski 2000), and the reuse of knowledge for innovation with required adjustments (Majchrzak et al. 2004). This adoption of reused knowledge includes reconceptualization of the problem and searching for new ideas, scanning and evaluating reusable ideas, analyzing and choosing the best ideas, and developing the most innovative idea.

A method on reusing of the knowledge in the construction domain was proposed by Tan et al. (Tan et al. 2007). In this method, details of project information are saved in a file named PKF (Project Knowledge File) for future access. The users have an option to search for intended information through a simple Google™-like search or an advanced search function. Within this advanced search, the knowledge can be searched based on keywords, rating given by others in validating the knowledge, project details (such as project title, type of contract, location), and any combination of these methods.

Studies in the last decade found several methods for the implementation of KM in the construction domain and for taking advantage of a comprehensive project Knowledge Management for improving projects' efficiency. However, none of them was able to achieve a complete knowledge reuse method. By reviewing current literature, it was noticed that Knowledge Management systems are moving toward implementation of semantic webs in the construction industry. Thus, it is important to consider current studies in semantic webs and explore the methods for knowledge recovery and reuse in this domain. A summary of recent works in semantic web field is provided in the next section.

2.4. Semantic web and Ontology

In the traditional World Wide Web (W3c 2004), most of the contents were linked together by hyperlinks that reference one data to another and users could move from one file to another by clicking on the link. In other words, the traditional World Wide Web was a massive repository of information with trillions of links connecting them together (Berners-Lee et al. 2001).

Most of the web's contents were not designed to be meaningfully manipulated by computers and designed to be read by humans (Berners-Lee et al. 2001). As a result, computer programs could not reliably analyze information using the traditional word wide web. The semantic web has been developed to resolve the gap between human and computer understanding and to create a document repository which is understandable to both human and computers. The initial idea of a machine-understandable web was utilized in 1989 (Pan et al. 2004) and it was called semantic web, for the first time by Tim Berners-Lee in early 2000s (Berners-Lee et al. 2001).

Several researchers have been studying semantic web in different fields of computer science, such as database management, artificial intelligence, and library science. In all of these fields, they are trying to reach a global goal of "*bringing structure to the meaningful content of web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users*" (Berners-Lee et al. 2001).

The main problem is to define a structured collection of information and a set of rules for automated reasoning. The provided information will define the architecture of knowledge representation for the semantic web. Good structured knowledge can increase efficiency and functionality of the semantic web.

Extensible Markup Language (XML) and Resource Description Framework (RDF) are two considerable technologies utilized in developing semantic web, both recommended by W3C. XML allows users to add arbitrary structures to their document, but the meaning of the aforementioned structure is not defined in XML. Meaning is expressed by RDF, which encodes it in a set of triples, each triple could be subject, verb, or object of an elementary sentence. In RDF, a document that explores a certain thing (such as people, Web pages or others) have properties (such as "is a sister of," "is the author of") with certain values (another person, another Web page) (Berners-Lee et al.

2001).

Two databases may use the same concept in different ways. For example, “postal codes” and “zip codes” both represent the location of a place by using two different terms. To compare and combine information between two different databases, a semantic program needs to understand the similarity of each concept and relates them with the corresponding meaning. This problem has been solved through the use of ontologies.

Ontology is a theory about the nature of existence, to study what type of things exist (Berners-Lee et al. 2001). Artificial intelligence and web researchers have adopted this term to define the relationship among the terms in a document. This is represented by sets of concept hierarchies and relationships that link these concepts together, and axioms for reasoning in a semantic way. In other words, semantic webs utilize ontologies to encapsulate and manage relative knowledge and to give information a human-relevant meaning (Lima et al. 2005).

Stab and Stuber define ontology through a tuple as shown in Equation 1 (Staab and Studer 2004).

Eq. 1

$$\boldsymbol{O} \coloneqq (\boldsymbol{C}, \boldsymbol{H_C}, \boldsymbol{P_C}, \boldsymbol{I}. \boldsymbol{A})$$

In which, H_C hierarchy and concepts C are arranged in a sub-sumption hierarchy. Each concept is defined by its properties P_C. Each concept has a set of instances I and axioms A in the hierarchy.

Bodenreider classified ontologies in two major types, domain ontologies and upper-level ontologies (Bodenreider 2003). Domain ontology is a conceptualized representation of vocabulary of the special domain. For example, in engineering design, one ontology could be defined for construction planning domain and another one could be defined for chemical plant design domain.

Possible conceptual elements of the ontology for construction planning field might contain the words “manager”, “contract”, “schedule”, and all other concepts that are commonly used in the construction domain. These concepts are related by means of the ontology; for example, the “manager” develops a “schedule” for each “contract”.

2.4.1. Semantic webs in the AEC industry

2.4.1.1. The emergence of Semantic Web technologies in the AEC industry

Development of Building Information Modeling (BIM) in the last two decades have resulted in a broad implementation of BIM application tools in AEC industries (Eastman et al. 2011). A research conducted by Yalcinkaya and Singh introduced 12 most recent research areas in the AEC domain by means of Natural Language Processing in the papers with BIM as topic and implementation and adoption, information exchange and interoperability, as-is and as-built data, design codes, and code compliance were highlighted. BIM and all the mentioned research efforts focus on information and making the information available for addressing existing problems (Yalcinkaya and Singh 2015).

The AEC industry is facing an increasing demand for information and exchange of this information among project parties. Inefficient understanding of project information and ineffective communication among different partners are major obstacles for the success of construction projects (Pan et al. 2004). The information is usually managed using current web-based technologies, which typically enable data sharing and processing by human and not by computers. Thus, the partners have difficulties in interacting with others, communicating with different languages, or in using alternative wording to refer to the same concept (Pan et al. 2004).

The thriving focus on this area has attracted more attention to data and information exchange across different application areas of the building life-cycle. This emerging change in the AEC industry could be addressed by allowing the web content to be understandable for both human and computers, and incorporating the semantic web into the systems (Pauwels et al. 2017). Semantic description of project resources can improve construction collaboration by enhancing the understanding of the information content (Zeeshan et al. 2004).

2.4.1.2. Semantic web's research in the AEC industry

One of the earliest research projects about implementation of semantic web technologies in AEC industry was done by Pet et al. (Pan et al. 2004), in which they determined key concepts and advantages of semantic webs, such as improvement of the information exchange in the construction industry and discussed how semantic web can be applied in design, communication, and change and claim management.

Studies have been continued through the last decade about various applications of semantic web in the construction industry. In the following section, a complete framework of the topics and challenges that researchers are facing today will be discussed. These topics could be classified into one of the following topics: interoperability, linking between different domains, and logical inference and proofs (Pauwels et al. 2017).

Interoperability is the implementation of importing and using the same content in different computer applications; hence, it can provide a rich data exchange environment in the construction industry. The data exchange has been enhanced by utilizing the semantic web as an opportunity to represent information in a computer-understandable manner.

BuildingSMART international (Buldingsmart), an organization with the mandate of information exchange among various applications in AEC industry, has developed Industry Foundation Classes

(IFC) data model for describing building and construction industry data. A methodology for interoperability among construction industry's stakeholders was proposed which was implemented using IFC language (Venugopal et al. 2015). Since IFC lacks semantic clarity, especially in mapping semantic information among different federated models (Venugopal et al. 2015), this methodology was developed as an ontology-based framework to make the IFC definitions more formal, consistent, and unambiguous.

Linking different domains enables access the information stemming from domains such as BIM, GIS, and simulation data with the universal web. One of the organizations who has been investing on this area is Linked Building Data Community Group, a group of experts in BIM and web technologies, that have been able to define the requirements to link database applications across the life cycle of a building (W3C 2014).

Developing ontologies for collaborative information management purpose is an interesting topic, which requires tremendous information coming from different applications to be managed within a single platform. The key area of interest is to create an ontology that represents domain concepts and identifies how these concepts are linked together. In other words, creating domain ontology is the first step of semantic Knowledge Management procedure (Pauwels et al. 2017). In the next section, the application of the ontology to link different domains in a semantic base is explained.

To demonstrate the difference between interoperability and linking across domains, it should be noted that interoperability is mapping the same concepts which are defined differently in different languages, while linking across domains focuses on mapping different languages which are used for modeling the same object (Pauwels et al. 2017).

Logical inference and proofs explore the underlying logical foundations of the languages utilized in semantic technologies (Pauwels et al. 2017). The formal logical basis of semantic web languages can provide proofs for the results inferred from the reasoning process, which can be used by the semantic web application to make a trust around their results. One of the research topics in this area is regulations compliance checking (Pauwels et al. 2017), which is beyond the scope of this research.

2.4.1.3. Semantic application in Knowledge Management

The construction industry can be divided into several different disciplines ranging from design, construction, operation, and maintenance. As mentioned before, the management of information coming from different sources and close collaboration among parties are required for the current state of the construction industry. Ontology-based information management can be a solution to this problem.

Various semantic systems have been developed in building and construction industry which can be ranged from domain dictionaries to specialized taxonomies (Costa et al. 2016). Among them are BS61 (Glossary of Building and Civil Engineering terms produced by the British Standard Institution), bcXML (an XML vocabulary developed by the eConstruct IST project for the construction industry), IFD (BuildingSmart. 2012); OCCS (OCCS 2013), BARBi (Norwegian Building and Construction Reference Data Library), and e-COGNOS (Consistent knowledge management across project and between enterprises in the construction domain IST-2000-28671).

e-COGNOS platform proposed by Lim et al. was the first comprehensive ontology-based portal for knowledge management in the construction industry (Lima et al. 2005). This platform consists of different ontologies to encapsulate human knowledge and a set of web services to support the management of these ontologies, user management, and knowledge management.

e-COGNOS is composed of three elements: the e-COGNOS ontology that is the domain knowledge representor, the ontology server (e-COSer) to manage the ontologies, and e-Construction Knowledge Management Interface (e-CKMI) to handle the main knowledge management services.

e-COGNOS provides four major knowledge management services to resolve challenges encountered in the industry, including user profiling, indexing, searching, and knowledge discovery. User profiling allows the system administrator to identify the skill sets of various individuals, their requirements, roles, and preferences. Development of these profiles helps the system for ordering proper knowledge to individuals during mass e-mailing.

Indexing tool rates, the most relative concepts to the input keywords. Searching module recovers related knowledge to the semantic concepts, which are provided by the indexing tool. In the knowledge discovery stage, the user can specify a set of related web sites or set up a profile for the knowledge of interest. The system provides ontology-based lists of relevant new knowledge. This stage provides users an updated knowledge based on the user's profile and requests (Lima et al. 2005).

Domain ontology and its' applications are the most important part to achieve semantic knowledge management. The ontology developed by Lima et al. for implementation in e-COGNOS was the first attempt to deploy a domain ontology in the construction industry (Lima et al. 2005). The ontology developed for e-COGNOS is composed of four major elements: actors, resources, processes, and products. The relationship among these elements is always defined as follows: the actors use the resources to construct products which are made using certain processes. The e-COGNOS ontology is composed of taxonomy of concepts and taxonomy of relationships, both based on the IFC entities.

One of the important findings revealed during implementation and use of e-COGNOS was the dependency of organizations on re-using of successful strategies from previous projects, especially in the design domain, even though this was a recognized factor in improving the system.

In a recent project (Costa et al. 2016), a domain ontology was developed based on several resources already available in building and construction industry such as: the OmniClass Standards for the Construction Environment (OCCS 2013), the Building Smart IFD (BuildingSmart. 2012), and the Construction Information and Knowledge Portal ontology (Zhang 2010). This ontology was utilized for enrichment of traditional knowledge representations by incorporation of implicit information derived from the ontology with the information presented in documents, thereby a baseline was provided for facilitating knowledge sharing between humans and machines (Costa et al. 2016).

2.5. Statement of the gap in the literature

As mentioned earlier, the methods of knowledge re-using have not been fully incorporated in the previous studies and were only partially implemented using simplistic methods. In conclusion, reviewing relevant studies have illustrated the importance of previous knowledge and experiments in the construction industry. Thus, the development of a comprehensive method to measure the similarity of projects is needed. Such a method can compare different projects semantically to retrieve the best matched project. Related information and knowledge linked to that project can be extracted and used to improve the efficiency of the current project.

In previous studies, the knowledge re-using was implemented using some simple methods. Tah et al. proposed a CBR searching method for similar cases based on some limited attributes, which were defined by the user (Tah et al. 1999). Since these attributes are identified manually by the user, the choices of attributes are subjective and might not appropriately find the most similar project(s).

Another approach proposed a method to search in the databases of the past project for the most similar project using some keywords (for example, the project title, type of contract, and location of the project) (Tan et al. 2007). This method would encounter some shortcomings, because keywords cannot consider all aspects of the project and cannot be a reliable variable for finding the similarity between non-identical activities. Moreover, keywords are also subjective.

As a result, this research investigates the possibility of using project activities as a potential representation for each project. This means that a project can be recognized by the types of activities. Work Breakdown Structure (WBS), including all activities required to complete a project, is a hierarchical structure of a construction project activities and could be a good representative for each construction project. In the next section, a brief description of WBS is provided and its applications in finding the similarity among projects are studied.

2.6. Work Breakdown Structure

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 currently approved project scope statement*" (PMBOK Guide 2017). In other word, the main goal of WBS is to present a complete and proper scope of the entire project work (Ibrahim et al. 2007).

The highest level of the WBS tree represents the entire project, and it is then decomposed into smaller subjects, each representing tasks that should be done 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 meaningful). The lowest level entries in this structure represent work packages. The responsibility of the performance of each work package is assigned to an individual or organization (Haugan 2001).

In this research, the hypothesis is that the WBS hierarchy can be used as an identifier for construction projects, because it includes a sufficient amount of information about project requirements and all activities required for that construction project. The idea of comparing different projects based on their WBS and measuring the similarity of two WBS is novel in the project management domain. Considering the hierarchical structure of WBS, the next section discusses similar studies which were conducted in other domains to measure the similarity between hierarchies.

2.7. Hierarchies and similarity measurement

Measuring the similarity between two physical objects is one of the areas where the hierarchical similarity measurement methods were employed. To achieve this, each object will be represented by a hierarchy of categories, and then the similarity score is computed to determine the similarity between two objects (Shukla et al. 2016). This similarity score is grounded on the effect of categories of two hierarchies on each other.

The effect of each category may be computed by measuring distance and sibling factors; the distance factor represents the effect of depth of each category and the sibling factor measures the influence of each category by means of the number of siblings existed in that category. The effect of categories can be presented in a vector for each object, which could be utilized to compute the similarity of the aforementioned category (Shukla et al. 2016).

The similarity of semantic business process models is another related field in which some methods were introduced with regards to the ontology structure in two business process models (Ehrig et al. 2007). One of these methods proposed an approach for semi-automatic detection of synonyms and homonyms of the process element names and the other method considers three similarity measures, including syntactic, linguistic, and structural, in order to determine the similarity of the business processes. By combining these similarity measures to obtain a combined similarity measure, semantic processes will be given a score between 0 and 1 to represent the similarity between them.

Unlike work breakdown structures, which only represent the name of each task, business process models may have some additional attributes attached to their nodes. These methods consider the name of each node, their attributes, siblings (successor), and the value of each node, and then use a weighted average of these variables to compute the structural similarity between nodes (Ehrig et al. 2007).

To compute the syntactic similarity, the proposed method (Ehrig et al. 2007) compares the number of characters in the concept's names in each process's nodes. Linguistic similarity measurement is implemented by utilizing a dictionary to determine the synonyms of each concept's name and the method computes the similarity based on whether the names are listed as synonyms. They used WordNet as the source dictionary in this study. More information about the WordNet is provided in the upcoming sections.

Syntactic and linguistic measures alone cannot compare two business process models, as they do not consider the context of the processes. The structural measure has been proposed to solve this issue. A context of a term is defined as the set of all elements which are more effective on the similarity of the term.

In the proposed method in this study, the differences between hierarchy structures of the two processes were not considered. To overcome this shortcoming, an structural similarity measurement method was proposed in another research to define graph-edit-distance of business process model as part of the calculations (Dijkman et al. 2011) (Hart et al. 1968). The graph-edit-distance between two graphs is the minimum number of graph-edit actions needed for altering one graph to the other. This is achieved by a variety of edit operations, such as node deletion or insertion, node substitution (a node from one graph is mapped to a node in the other graph), and edge deletion or insertion.

Since there have not been any studies on comparing the semantic similarity of work breakdown structures, this study proposes to employ and alter ideas from other domains and apply them to the construction domain by adding new variables and measures. Thus, a novel method for comparing the semantic similarity of two work breakdown structures is proposed. The details and hypothesis used in this research are explained in Chapter 3

2.8. Text similarity

2.8.1. Introduction

Measuring similarity between words, sentences, paragraphs, and documents has been used for a long time in several Natural Language Processing (NLP) fields, such as information retrieval, text classification, document clustering, topic detection, topic tracking, question generation, question answering, essay scoring, short answer scoring, machine translation, text summarization, and many other areas (Gomaa and Fahmy 2013).

One of the first implementations for text similarity carried out by Salton and Lesk, in which the most similar documents to an input query were determined by the ranking of the documents in the order of their similarity to the input query (Salton and Lesk 1968). The text similarity has also been used recently in the construction domain for automated regulatory compliance checking (Zhang and El-Gohary 2013).

Finding similarity between words is one of the most important steps in the text similarity measure. Work Breakdown Structure's entries mostly contain short phrases, which include a couple of words. Thereby, the focus of this research is on measuring word-to-word similarity measure for measuring the similarity of very short text segments. Details of this method are explained in the next sections.

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. Whereas semantically similar words are related by means of different relations, such as being synonyms or antonyms and being used in the same way or in the same context. In fact, semantic similarity determines the relation of words or concepts based on databases which are used and the relations that are defined for the words in those databases. The following subsections provide a brief description of different methods for measuring the semantic and lexical similarities.

2.8.2. String-based similarity measure

In order to measure the similarity between two concept strings, Levenshtein proposed edit distance method in which the difference between two strings is defined by the minimum number of changes (insertion, deletion or substitution) needed to transform one string to another (Levenshtein 1966). For example, the distance between the strings "cat" and "hat" is one character (substitution of "c" for "h"). Edit distance method does not consider the number of strings of compared concepts. In another proposed method for syntactic similarity (Maedche and Staab 2002), the number of characters is also being considered as shown in Equation 2.

Eq. 2

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

Lexical similarity method does 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. Mihalcea et al. showed that semantic similarity algorithms outperform simple lexical methods with a 13% error rate reduction (Mihalcea et al. 2006).

2.8.3. Semantic similarity measure

Semantic similarity measurement methods were introduced using Corpus-based and Knowledge-based algorithms. A Corpus is a large database for collecting written or spoken texts for the purpose of language processing. The corpus-based similarity is a semantic similarity by the exploitation of a large corpus which determines the similarity of various words. Latent Semantic Analysis (LSA) (Landauer and Dumais 1997) is one of the most popular methods for obtaining corpus-based similarity. LSA believes that reoccurring of the same words in a similar piece of texts is an indication for their proximate meaning (Landauer and Dumais 1997).

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 relation of concepts using networks (Sowa 2012).

WordNet is the most popular semantic network in the field of Knowledge-based semantic similarity measurement, which was produced as a result of a comprehensive research program at Princeton University (Miller 1995). It is utilized as a lexical reference of English. In WordNet, English nouns, verbs, and adjectives are organized in synonym sets and these sets are related together by means of semantic relations. A variety of semantic relations have been developed in WordNet including (but not limited to) synonymy, autonomy, hyponymy, membership, similarity, domain, and cause-and-effect relationships (Meng et al. 2013). By exploiting these relations, semantic hierarchy structures are developed and these hierarchies could be useful in semantic computations.

Since nouns have a significant role in language semantics, most of these studies focused on nouns for semantic similarity calculations. Four common semantic relations for nouns are hyponym/hypernym (is-a), part meronym/part holonym (part-of), member meronym/member holonym (member-of), and substance meronym/substance holonym (substance-of). For example, “car” is a “vehicle” and a “wheel” is part of a “car”.

There are a number of different methods based on WordNet for semantic similarity measurements such as path-based measure, information content-based measure, feature-based measure, and hybrid measure. In this research the method proposed by Wu and Palmer (path-based measure) will be used for the semantic comparison of tasks between WBSs.

2.8.3.1. Path-based measure

This method measures the similarity between two concepts based on the length of the path linking them in the taxonomy and their position in the taxonomy. Wu and Palmer (1994) proposed a method to compute similarity based on the position of concepts c_1 and c_2, as well as the position of the lowest common subsumer $lso\ (c_1, c_2)$. As it can be seen in the following equation, the $len(c_1, c_2)$ measures the length of the shortest path from the synset c_1 to synset c_2, and the $depth$ measures the length of the path from each synset to the root element (Wu and Palmer 1994).

Eq. 3

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

2.8.3.2. Information content-based measure

This measure is based on the information content of each concept in WordNet. In this method, a higher similarity of information between two concepts will result in a higher similarity in value of concepts. (Resnik 1995) It was proposed that the similarity of two given concepts depends on the information content that subsumes them in the taxonomy. This similarity could be calculated using Equation 4, that for two given concepts c1 and c2, similarity depends on the information content (IC) that subsumes them in the taxonomy.

Eq. 4

$$\boldsymbol{sim_{Resnik}(c_1, c_2) = -\log p\left(lso\ (c_1, c_2)\right) = IC\left(lso\ (c_1, c_2)\right)}$$

2.8.3.3. Feature-based measure

This method has a different approach from path-based and information content-based methods. It's not affected by the taxonomy and subsumer of concepts, and the similarity value is the function of properties of each concept. This method is based on the set of words indicating the properties and features of them, such as their definitions or "glosses" in WordNet. More common characteristics between two concepts result in greater similarity values between them. Tversky (Tversky 1977) defined this similarity as follows :

Eq. 5

$$sim_{Tversky}(c_1, c_1) = \frac{|C_1 \cap C_2|}{|C_1 \cap C_2| + k|C_1 \backslash C_2| + (k-1)|C_1 / C_2|}$$

Chapter 3: Methodology

3.1. Introduction

This chapter presents the proposed method to compare two projects similarity by measuring semantic and structural similarity of their WBSs. It was explained that each WBS comprises different elements, which are labeled with some tasks and services required to complete the project. In the following sections, the Work Breakdown Structure's elements are mentioned as nodes of WBS and each element's task is called as node's label.

WBS similarity measurement is implemented using two metrics: 1) node similarity that compares the semantics of elements in two WBSs; 2) structural similarity which compares the topology of Work Breakdown Structures. The proposed processes to calculate each of these two metrics result in a similarity score between 0 and 1. The average of these two scores provides the final similarity score between two WBSs.

This chapter explains the executed programming and the implemented methods for reading and importing WBS's data to the developed program. In the subsequent sections, the proposed methods for calculating node similarity score, structural similarity scores, and the total similarity score are discussed.

3.2. Reading and parsing WBS data

A program was developed in Python programming language ('Python programming Language') to read WBSs from the database for further processes. The following sections explains how the WBS information is read and parsed using the developed program.

3.2.1. WBS encoding

The WBS for each project can be developed in different formats. In this study, WBS information is extracted for projects that were developed in the Microsoft Project application (Microsoft). This software product is one of the most popular project management applications, which can implement WBS of projects along with their schedules. The Microsoft Project also provides WBS codes, which specify the exact location of each node in the WBS's hierarchy. These data can be exported from Microsoft Project to a spreadsheet format file (such as Excel). Figure 6 and Figure 7 show the tasks names and WBS codes in Excel format for small parts of two simplified projects that belong to a "House project" and a "Bridge project".

The labels and codes of WBS's nodes need to be converted from a simple note in Excel to a machine-readable format. To address this issue, WBS hierarchies were encoded in eXtensible Markup Language (XML). XML is a markup language, which encodes documents by defining a set of rules and makes them readable for both human and computers (Bray et al. 1997).

XML documents are formatted in a tree structure which starts with a root element and branches from the root to child elements. The tree structure format of XML makes it compatible with WBS which has a hierarchy structure. Each element is represented by a tag name in XML, which contains other information such as text, attributes, other elements or a mix of these. For the purpose of encoding of WBS, the labels are encoded as XML tags. These tags also have an attribute of "level" which presents WBS's codes. Given the WBS of the mentioned "House project", the XML file can be written in the .xml format as shown in Figure 5. For example, the task "internal_works" is defined as an XML tag name with the "level" attribute of "1.3". Since space character in a tag name of an XML element is not allowed, the words in each tag are separated by "_" character instead.

```xml
<?xml version="1.0" encoding="Utf-8"?>
<house_project level="1">
<site_preparation level="1.1"></site_preparation>
<structure level="1.2">
<columns level="1.2.1"></columns>
<beams level="1.2.2"></beams>
<roof level="1.2.3"></roof>
</structure>
<internal_works level="1.3"></internal_works>
</house_project>
```

Figure 5. XML code of the "House project"

WBS	Task name
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

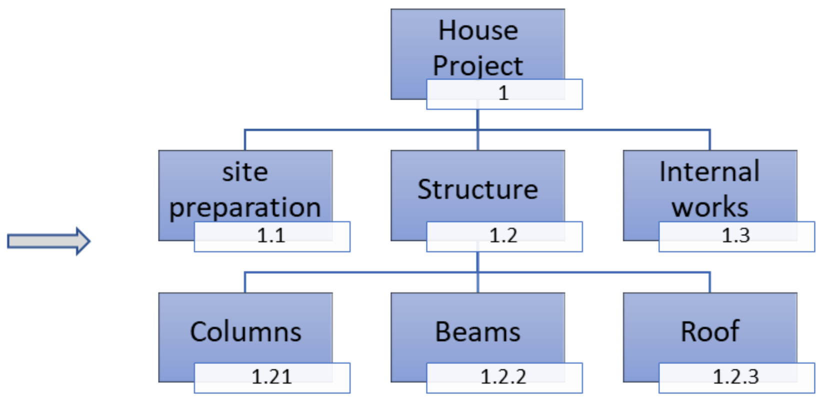


Figure 6. WBS codes and hierarchy of a "House project"

WBS	Task name
1	Bridge project
1.1	Design
1.2	Superstructure
1.2.1	Girders
1.2.2	Decks
1.3	Road
1.3.1	Pavement

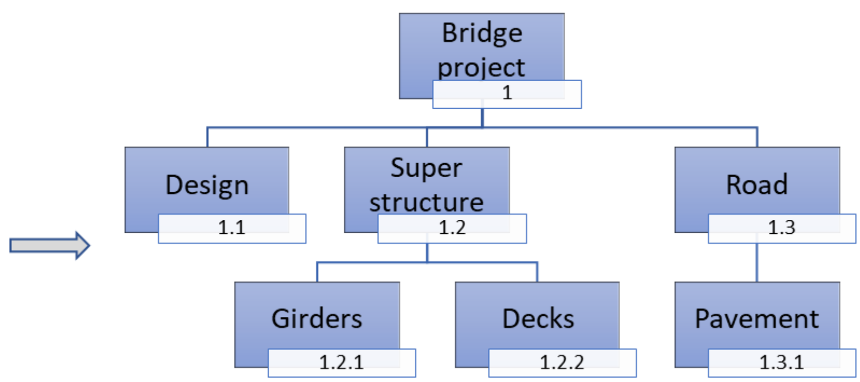


Figure 7. WBS codes and hierarchy of a "Bridge project"

3.2.2. Importing encoded WBS into Python programming language

The proposed method for semantic comparison of two WBS is developed using Python programming language ('Python programming Language'). Each WBS written in XML language is imported to the Python environment. For example, Wbs1.xml and Wbs2.xml are the XML files for the previously mentioned “House project” and “Bridge project”. They are imported to the Python environment as File1 and File2 as follows.

```
>>> File1='C:/Users/admin/Wbs1.xml'
>>> File2='C:/Users/admin/Wbs2.xml'
```

3.2.3. Parsing imported documents

The imported information from .xml files into the program environment can be parsed into two separate lists, including elements tags (WBS’s tasks) in one list and attribute of each element (WBS’s codes) in another list. Given Wbs1, the task list and code list can be developed by means of the following codes.

```
>>> from xml.etree.ElementTree import iterparse
>>> Wbs1= []
>>> Level1= []
>>> for (event, node) in iterparse (File1, ['start', 'end', 'start-ns', 'end-ns']):
>>> if event=='start':
>>>     Wbs1.append(node.tag)
>>>     Level1.append(node.get('level'))
>>> Wbs1
['house project', 'site preparation', 'structure', 'columns', 'beams', 'roof', 'internal
works']
>>> Level1
['1', '1.1', '1.2', '1.2.1', '1.2.2', '1.2.3', '1.3']
```

3.3. Node similarity

The first step to measure the similarity of two WBSs is a pairwise comparison of two WBS’s nodes. In the proposed method, the semantic similarity of two nodes is measured by considering the semantic similarity of labels, parents, and siblings. These three metrics are discussed in following sections.

There are two important issues in measuring the semantic similarity of nodes. First, the labeling of tasks in a project are subjective to the project managers. For instance, “Reinforcing placement” and “Reinforcement installation” are not exactly the same with regards to their strings, but a project manager considers them the same tasks. Thus, two labels to be considered similar must not contain exactly the same concepts. This problem can be solved by semantic similarity measurements of labels instead of simple string measurements.

On the other hand, the semantic equivalence of two labels does not necessarily mean similarity of two nodes. Assuming that there are two nodes with “concrete pouring” as a label for both. Although they are semantically similar, they might represent different tasks, for example, one can represent concrete pouring for a column and the other one is for a beam.

To address the mentioned issues, the proposed method determines the similarity of two WBS through three metrics as follows.

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

This way, other related tasks are examined to have a more comprehensive semantic similarity measure between two nodes. In this part, the semantic similarity measurements are carried for all the nodes of two WBSs. The results of these pairwise comparisons are presented by a matrix.

Assuming $WBS_{N_1}^{L_1}$ and $WBS_{N_2}^{L_2}$ are two Work Breakdown Structures, in which L_1 and L_2 represent the total number of levels that each WBS hierarchy contains, and N_1 and N_2 represent the finite sets of WBS's nodes. The semantic similarity results matrix can be developed as bellow.

Eq. 6

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

$N_1: (n_1, n_2, \dots, n_i)$

$N_2: \left(m_1, m_2, \dots, m_j\right)$

3.3.1. Semantic similarity

WordNet (Miller 1995) was utilized to measure the semantic similarity of labels in this system. WBS labels are usually expressed as a phrase which contain 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. (Mihalcea et al. 2006). In order 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 finds the most semantically similar word from segment T_2 ($maxSim(w, T_2)$), using one of the word-to-words similarity measures explained previously in Chapter 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" (Mihalcea et al. 2006). This method measures the semantic similarity of two segments by means of the equation bellow.

Eq. 7

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

This method can be adjusted to a more appropriate method for this study by eliminating the word specificity weight. The reason behind this decision is that in this case, the labels are phrases with a very few (one or two) generic concepts and most of the component words are specific to the construction domain.

Using Wu and Palmer method (Wu and Palmer 1994) as a word-to-word semantic similarity measurement and taking the mentioned assumptions, for given nodes n_i and m_j from $WBS_{N_1}^{L_1}$ and $WBS_{N_2}^{L_2}$, the semantic similarity of their labels is obtained by means of the following equation. In this equation for each word as w in the task node n_i, the most semantically similar word from task node m_j ($maxSim(w, m_j)_{wup}$) is found by means of the Wu and Palmer (1994) method which was explained previously in Chapter 2. The same procedure will determine the most similar word in n_i starting with the words in m_j (a sample calculation is presented in Appendix D).

Eq. 8

$$sim_{semantic}(n_i, m_j) = \frac{1}{2}\left(\frac{\sum_{w\in\{n_i\}}(maxSim(w, m_j)_{wup})}{\sum_{w\in\{n_i\}} 1} + \frac{\sum_{w\in\{m_j\}}(maxSim(w, n_i)_{wup})}{\sum_{w\in\{m_j\}} 1}\right)$$

To execute the above-mentioned method on two labels, each label should be decomposed to its components, and then a method is needed to compute word-to-words similarity.

To implement natural language processing to measure the semantic similarity of two WBS's labels, this study has benefited from Natural Language Toolkit (NLTK) (Bird and Loper 2004). NLTK is a suite of libraries and programs written in Python programming language which enables the human natural language programming. NLTK provides more than 50 corpora and lexical recourses which have easy-to-use interfaces such as WordNet. NLTK also provides a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning (Toolkit 2018). NLTK library and WordNet interface can be imported in Python as follows.

```
>>> from nltk.corpus import wordnet as wn
```

3.3.1.1. Label tokenization

NLTK is implemented by a large collection of small modules. `nltk.toenizer` is a *task module* which decomposes a sentence or phrase into its components (Bird and Loper 2004). Given "concrete pouring" as a task, it will be tokenized into words "concrete" and "pouring" by executing the code below.

```
>>> from nltk import word_tokenize

>>> task="concrete pouring"
>>> print(word_tokenize(task))
['concrete', 'pouring']
```

3.3.1.2. word-to-word semantic similarity measurements

As it was explained in the literature review, *synset* is a set of synonyms which share a common meaning. `Wn.synset` module in NLTK develops a list of synsets with regards to the input word. In the developed list, *synsets* are ordered by their frequency of occurrence in the corpus texts. For example, the synsets of the word "concrete" are developed as it is shown in Figure 8. As it's shown in this Figure, the definition of the first noun synset is different from the definition of the first verb synset of this word.

```
>>> print(wn.synsets('concrete'))
[Synset('concrete.n.01'), Synset('concrete.v.01'), Synset('concrete.v.02'),…]

>>> print(wn.synsets('concrete,n,01').definition())
a strong hard building material composed of sand and gravel and cement and water

>>> print(wn.synsets('concrete,v,01').definition())
cover with cement
```

Figure 8. Synsets of the word "concrete"

Each synset is labeled with three parts, <word>, <part of speech of word> and <index of the synset>. For instance, the first synset is <concerete>, with a noun part of speech <n> and index of <1>. The other part of speeches of a word can be verb <v>, adjective <a> adverb <r> or adjective satellite <s>.

Since the relations between concepts are based on synsets in WordNet, an algorithm is needed to find the similarity between words rather than synsets (Jurafsky and Martin 2014). 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 the process of choosing the right synset for a given word, a large amount of ambiguity might happen. First, the part of speech of the word needs to be clarified and then the intended synset with that specific part of speech should be specified. There are different methods for disambiguation of part of speech and sense of words, such as **part-of-speech tagging** and **word sense disambiguation** (Jurafsky and Martin 2014).

Word Sense Disambiguation or WSD techniques find the correct sense of input word by means of some databases, such as sense-tagged databases, which contain context sentences, labeled with a correct sense for the targeted word. But such database has not been developed for the construction domain yet and implementation of these techniques needs more expert knowledge about natural language processing, which is out of the scopes of this research.

Due to all above-mentioned reasons, some simplified methods have been used in this study to measure the similarity of two words. One approach was targeting only the first noun synset of each word. The key idea of this hypothesis was that most of the words used in tasks are a noun, and the first synset is usually the most common synset.

Focusing only on noun aspect of words can disregard words without any noun synset. This problem can cause zero semantic similarity for any pairwise comparison containing such a word. For example, as it is shown below in the developed list of synsets for the word "reinforcing", all the synsets are a verb.

```
>>> print(wn.synsets('reinforcing'))
[Synset ('reinforce.v.01'), Synset ('reinforce.v.02')]
```

On the other hand, always the first synsets are not necessarily the intended sense of the word, especially in the construction domain. For example, the first synset of the word "reinforcement" is not obviously the intended sense (see the Figure 9), which is used as a task in construction domain and the relevant sense is C.4. The definition of different senses of the word "reinforcement" is presented in Figure 9.

```
>>> for synset in wn.synsets('reinforcement'):
>>>    print(synset.definition())

(c.1) a military operation (often involving new supplies of men and materiel) to
strengthen a military force or aid in the performance of its mission.
(C.2) information that makes more forcible or convincing
(C.3) (psychology) a stimulus that strengthens or weakens the behavior that
produced it
(C.4) a device designed to provide additional strength
(C.5) an act performed to strengthen approved behavior
```

Figure 9. The definitions of different synsets for the word "reinforcement"

Another approach was to use *pos_tag* module to disambiguate part of speech of each word and only target the synsets with that selected part of speech, but this approach also failed, because for example in the task "concrete pouring", *pos_tag* module tagged the word "pouring" as a noun ('NN'), however as its presented below, this word does not have any synsets as a noun.

```
>>> print(pos_tag(word_tokenize('concrete pouring')))
[('concrete', 'NN'), ('pouring', 'NN')]

>>> print(wn.synsets('pouring'))
[Synset('pour.v.01'), Synset('pour.v.02'), Synset('decant.v.01'),
Synset('pour.v.04'), Synset('pour.v.05'), Synset('pour.v.06'),
Synset('gushing.s.01')]
```

In the final approach, the system approximates the similarity by using the pair of synsets for two words that result in maximum sense similarity.

Eq. 9

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

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

The word-to-word similarity is executed by the following code.

```
>>> print(wn.synsets('concrete', 'n')+ wn.synsets('concrete', 'v'))
[Synset('concrete.n.01'), Synset('concrete.v.01'), Synset('concrete.v.02')]

>>> def word_to_word_similarity(word1, word2):
    similarity=0
    if wn.synsets(word1)==[] or wn.synsets(word2)==[]:
       edit_distance=nltk.edit_distance(word1,word2)
       max_len=max(len(word1),len(word2))
       syntactit_similarity=(1-(edit_distance/max_len))
       similarity=syntactit_similarity
    else:
          for synset1 in wn.synsets(word1,'n')+wn.synsets(word1,'v'):
              for synset2 in wn.synsets(word2,'n')+wn.synsets(word2,'v'):
                  if wn.wup_similarity(synset1,synset2)==None:
                      sim=0
                  else:
                      sim=wn.wup_similarity(synset1,synset2)
                  if sim>similarity:
                      similarity=sim
          return similarity
```

As it can be seen in the above code, before semantic similarity of two words, a string base similarity, which was explained in Chapter 2, is measured for the words that provoke in two conditions. In the first condition, it prevents giving the similarity of zero for the words which are wrongly misspelled (to avoid possible typos). The second condition considers the cases in which the words are not defined in the WordNet and do not have any defined synsets, and therefore it gives the string similarity of them instead of zero. For example, the similarity of the word "HVAC" by itself can be increased from the semantic similarity of zero to the correct similarity of one by means of the string similarity.

Using the word-to-word similarity measure and the proposed method for measuring the semantic similarity of two phrases, the semantic similarity of two labels is calculated. 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 the following matrix. This matrix represents the semantic similarity of labels of nodes n_i and m_j.

Eq. 10

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

The proposed method considers two nodes semantically similar if they have a semantic similarity more than a threshold which can be set to a number in the range of 0 to 1. In addition, 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 the next Chapter.

3.3.2. Parent similarity

In a Work Breakdown Structure, except the root element (highest level), each node is subdivided from an upper-level element, which is the node's parent. Also, each parent is generated from another upper-level element which creates a sequence of parents for each node. In this step, the semantic similarity of the parents of two given tasks are considered. This metric determines the similarity of parents and is calculated by a weighted average of semantic similarity of them.

Since considering all ancestors of a node requires an extremely large amount of calculations, the Least Similar Parent is defined (LSP). LSPs are the first pair of parents in the sequence of two given node's 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 the equation bellow. $L_{LSP} - L_n$ is the difference between levels of node n and its LSP, and $sim_{semantic}(i^{th}\ parents)$ is the semantic similarity between i^{th} parents of nodes n and m.

Eq. 11

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

For instance, assuming the threshold as 0.5 (an arbitrary number between 0 and 1), Figure 10 shows the first two parents of nodes n and m with a semantic similarity of 0.8, which is more than the defined threshold (i.e. 0.5 in this example) and the next two parents have similarity of 0.0, and therefore they are defined as Least Similar Parents. Parent similarity is calculated by the following function.

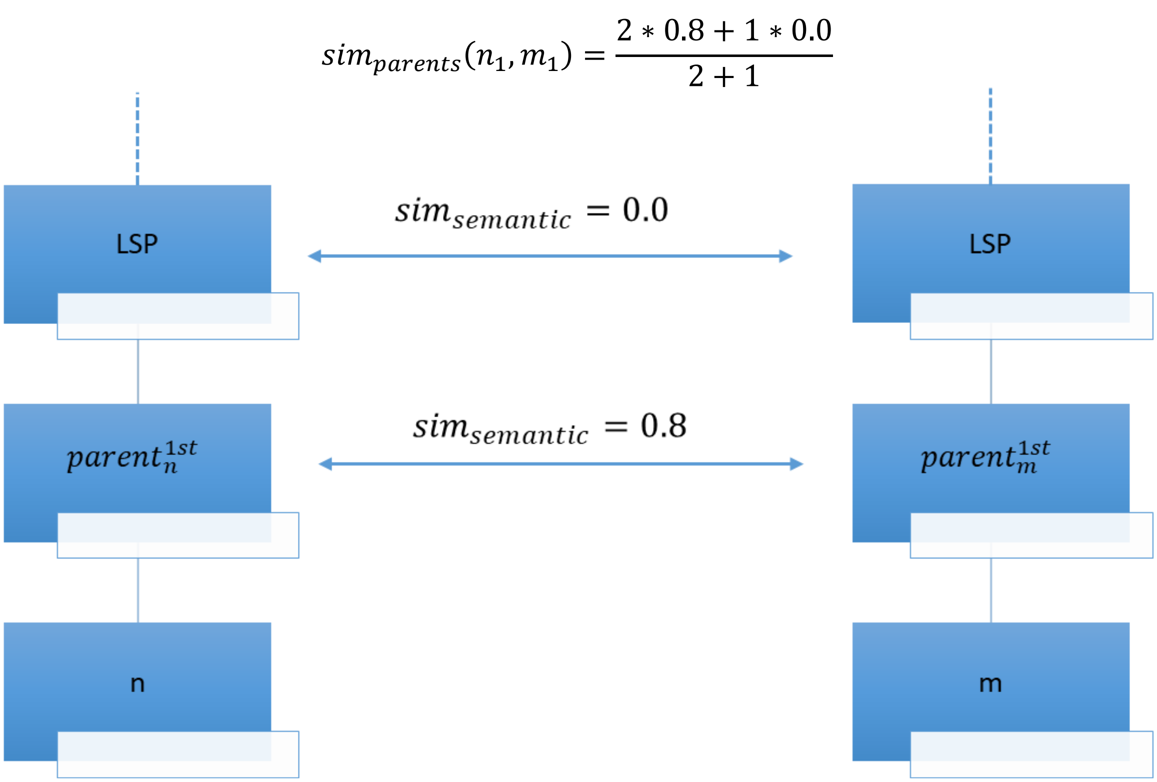


Figure 10. Parent similarity between n and m

The results of parent similarity between nodes from $WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2}$ are presented using the following matrix.

Eq. 12

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

3.3.3. Siblings similarity

In a Work Breakdown Structure, the nodes generated from the same parent are called siblings. Semantic similarity of siblings of two nodes is another metric that can increase the confidence in similarity degree. For example, the chance for a task like “reinforcing” to have a sibling task of “concrete pouring” is more than an unrelated task such as “marking of the road”.

To find the matched siblings of two nodes, the siblings should be semantically compared one by one using the $sim_{semantic}\left(WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2}\right)$ matrix, that was developed in the previous step. Any two siblings which are semantically similar ($sim_{semantic} > threshold$) 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, m_j.

Eq. 13

$$matched_{siblings}(n_i, m_j) = (sibling_{n_i}, sibling_{m_j})$$

As a result, the sibling similarity score for nodes n_i , m_i is calculated by means of equation bellow, which is obtained by dividing the total number of matched nodes by the total number of siblings.

Eq. 14

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

For example, the sibling similarity between nodes n_1 and m_1 with only one pair of matched siblings in Figure 11 is calculated by the function bellow:

$$sim_{siblings}(n_1, m_1) = \frac{2 * 1}{2 + 2}$$

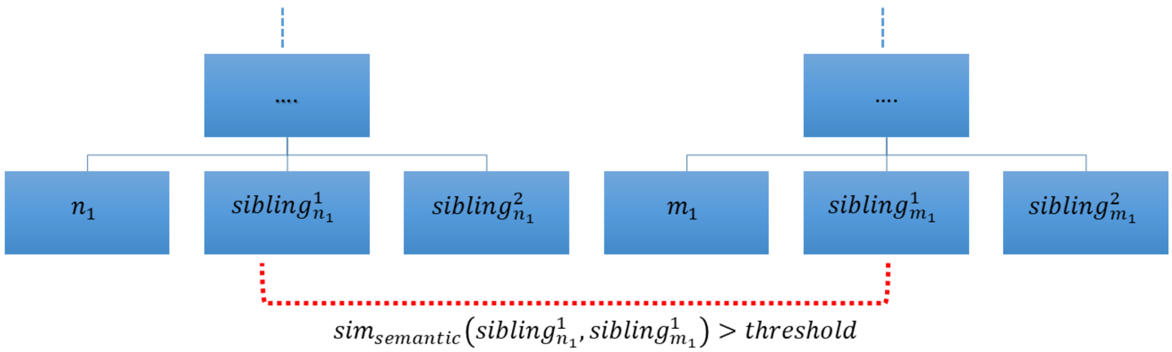


Figure 11. Sibling similarity between nodes $\mathbf{n_1}$ and $\mathbf{m_1}$

A sibling similarity matrix, which contains a pairwise comparison of nodes n_i, m_i from $WBS^{L_1}_{N_1}, WBS^{L_2}_{N_2}$, can be developed which can be expressed as the following.

Eq. 15

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

3.3.4. Average similarity

Three different metrics for the node to node comparison between two WBSs were proposed. In the next step, an average between these three measurements is calculated, which is defined as the average similarity score. The average similarity matrix represents the average node to node similarity between nodes of $WBS^{L_1}_{N_1}$ and $WBS^{L_2}_{N_2}$. The effect of the weight of the $sim_{semantic}$ will be discussed in Chapter 4.

Eq. 16

$$sim_{average} = \frac{w \times sim_{semantic} + sim_{parents} + sim_{siblings}}{2 + w}$$

Eq. 17

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

3.3.5. Nodes mapping

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 defined threshold. This threshold is considered to prevent mapping of irrelevant nodes which have a semantic similarity score below the threshold.

In some cases, there is more than one node with the highest $sim_{average}$. In these cases, the system prefers the task nodes with a closer level of details. The level of details of the nodes depends on their level in the WBS hierarchy. The details of task nodes in the hierarchy decreases from the lowest to the highest level. Since usually the lowest level of WBS contains the task with the highest level of details, the 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 the distance between the level of details of two nodes

The $bottom_{level}$ is the level of nodes which its numbering starts from the lowest level. For example, for the Wbs1 (“house project”) the $bottom_{level}$ and regular $level$ of nodes are indicated in Figure 13 and Figure 12, respectively.

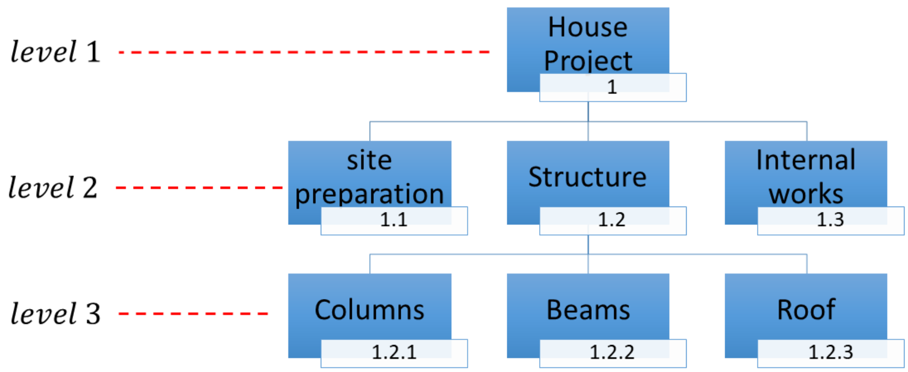


Figure 12. WBS regular levels

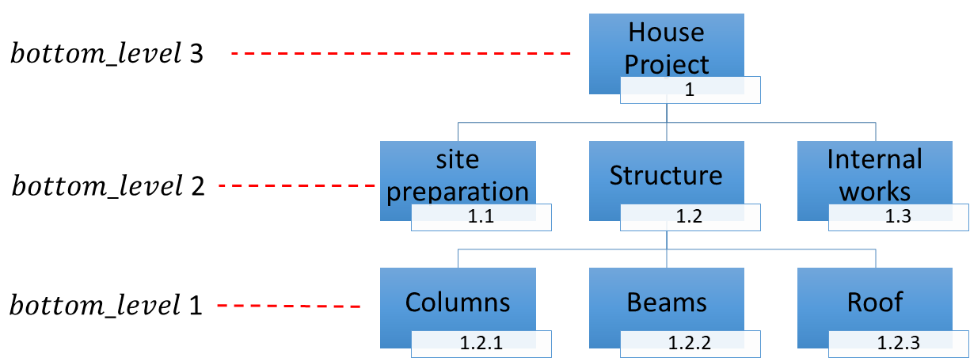


Figure 13. WBS bottom_levels

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. 18

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

For example, this score for nodes "columns" and "road" from the "House project" and "Bridge project" which were introduced in Figure 6 and Figure 7 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,

$$level_{scores}(\text{“columns”}, \text{“Girders”}) = \left|\frac{(1-1)}{3} - 1\right| = 1$$

This score will increase 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 nodes of two WBSs.

Eq. 19

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

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

Eq. 20

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

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

$mapped\ nodes$ 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. 21

$$mapped\ nodes = \{(n_i, m_j, sim_{average})\}$$
$$n_i \in N_1$$
$$m_j \in N_2$$

3.3.6. Node similarity score

The 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 following equation.

Eq. 22

$$Node\ similarity\ (WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2}) = \frac{2 \times \sum_{(n_i,m_j)\ \in\ mapped\ nodes} \left(sim_{avg}(n_i, m_j)\right)}{|N_1| + |N_2|}$$

3.4. Structural similarity

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 (Hart et al. 1968) of two WBSs. The graph-edit-distance measures the minimum required operations which are needed 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 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 distance 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 $WBS_{N_2}^{L_2}$.

Eq. 23

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

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 $mapped\ nodes$ list, the *SE* is calculated by one minus their similarity.

Eq. 24

$$for\ (n_i, m_j) \in mapped\ nodes, \qquad SE(n_i, m_j) = 1 - sim_{\text{average}}(n_i, m_j)$$

And, the total SE effort between $WBS_{N_1}^{L_1}$ and $WBS_{N_2}^{L_2}$ over all the mapped nodes can be calculated by following function.

Eq. 25

$$SE(WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2}) = \frac{2 * \sum_{(n_i, m_j) \in mapped\ nodes}(1 - sim_{\text{average}}(n_i, m_j))}{|N_1| + |N_2| - |UN|}$$

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*). Less effort that is required to transfer structure of the first WBS to second one, which results in higher structural similarity and vice versa.

Eq. 26

$$Structural\ similarity\ (WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2}) = 1 - \frac{DE(WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2}) + SE(WBS_{N_1}^{L_1}, WBS_{N_2}^{L_2})}{2}$$

3.5. 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 of *Node similarity* (Eq. 22) and *Structural similarity* (Eq. 26) scores using the equation bellow. This final measurement produces a score between 0 and 1, which 0 is hypothetically resulted from the comparison of two completely different projects, and 1.0 is resulted for two exact similar projects.

Eq. 27

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

Chapter 4: Results and Discussion

4.1. Introduction

This chapter presents a set of experiments to evaluate the performance of the proposed system in two major parts: First the precision of the node to node similarity measurements in mapping the related tasks is discussed, and second, the precision of the overall similarity scores in measuring the similarity of WBS samples and retrieving most similar samples to the given samples are examined.

The experiments are carried out on nine different WBS test samples, which the method of their development is presented in the following section. The obtained results from each experiment are presented and discussed in each part.

4.2. Test Samples

Three different construction project models were given to three experts in the construction management area to develop the WBS samples. The main reason for choosing different experts was to reduce the subjectivity of tasks in each WBS. The experts were asked to develop one WBS corresponding to each given project model. In the following sections, the project models and corresponding WBS are presented and the encoding of the samples in XML language is explained.

4.2.1. Construction project models

Three different construction projects were chosen to have WBS samples from different types of projects. Two of these projects were building construction and one was a bridge construction project. The bridge project had twelve spans with ten concrete piers and two concrete abutments, and was constructed with steel girders and reinforced concrete decks. The first building project was a four-story reinforced concrete-framed residential building and the other one was a two-story steel-framed commercial building.

3D building information model (BIM) of these projects were provided to each expert to ensure that all participant have the same level of detailed information about the projects. These models were developed in Revit software product to not only visualize the projects, but also to provide object-oriented information about the elements of the structures to develop the WBSs. Figure 14, Figure 15 and Figure 16 show the developed 3D BIM models for the bridge, concrete-framed building, and steel-framed building, respectively.

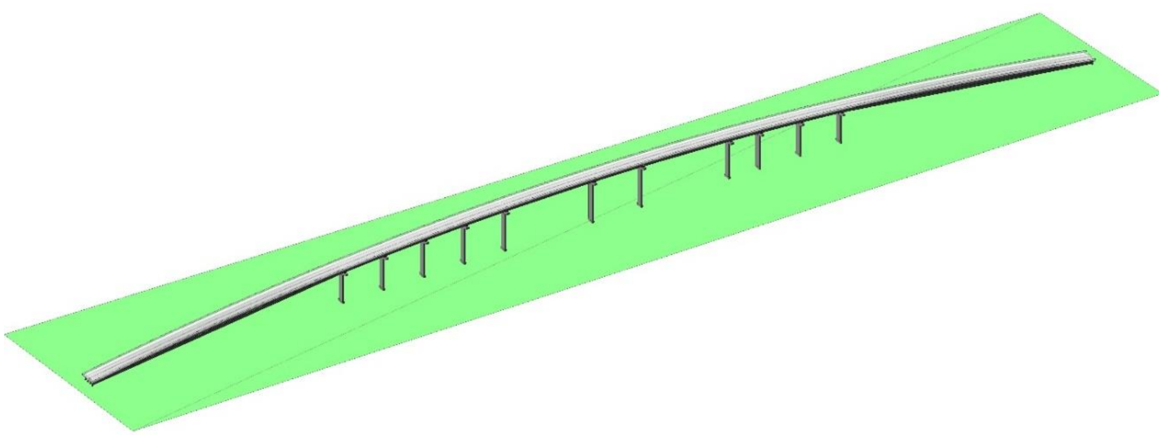


Figure 14. The 3D model of the bridge project

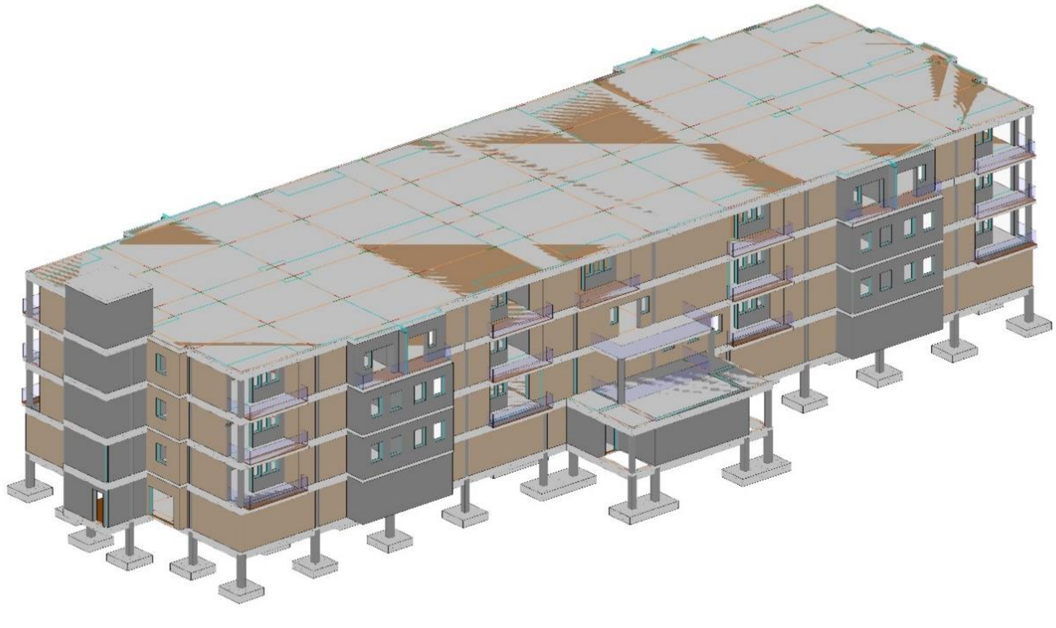


Figure 15. The 3D model of the concrete structure building project

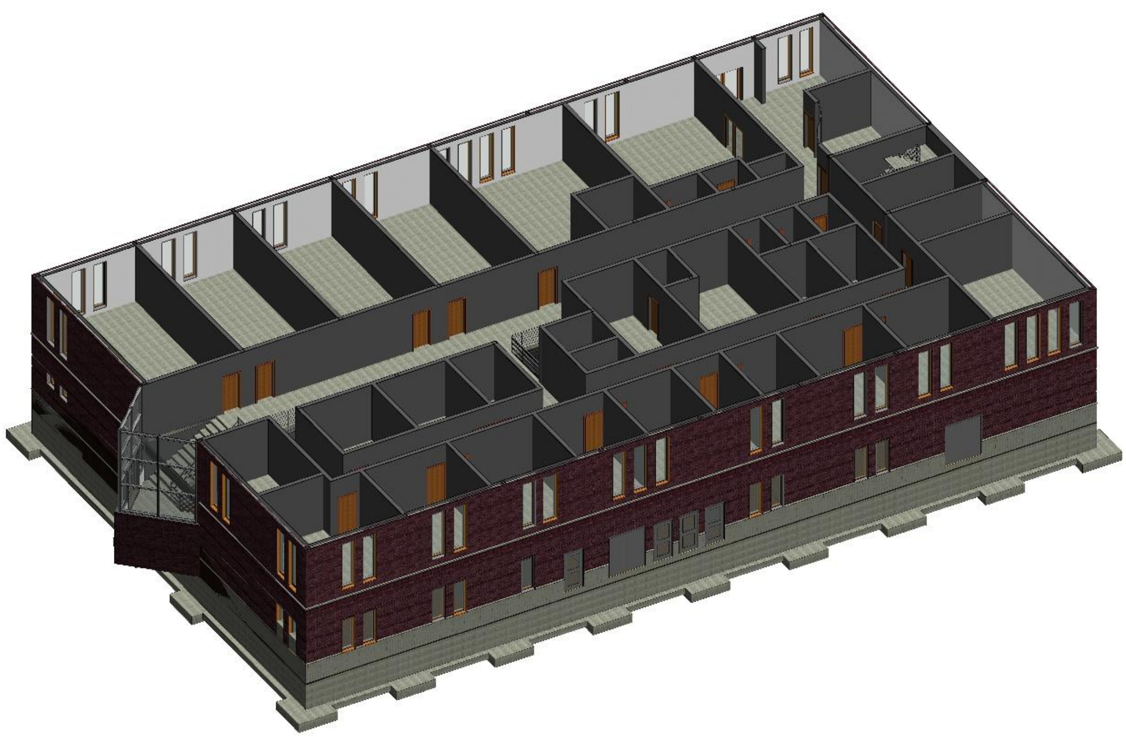


Figure 16. The 3D model of the steel structure building project (roof was sectioned to provide internal details)

4.2.2. Developing WBS of the test samples

The prepared project models were given to the experts to develop the corresponding WBSs. The experts were chosen from qualified civil engineers who were experienced in construction management and were academically trained about the principles of project planning and development of work breakdown structures.

As it is shown in Table 1, the developed samples are represented by B, C and S, which refer to $Bridge\ construction$, $concrete\ structure\ building$ and $steel\ structure\ building$, respectively. The developed samples needed some small adjustments to become compatible with the proposed system. In the following subsection, these changes are explained.

Table 1. The developed samples by the experts.

Experts	Developed samples	Represented by
Expert 1	$Bridge\ construction_1$ $concrete\ structure\ building_1$ $steel\ structure\ building_1$	B_1 C_1 S_1
Expert 2	$Bridge\ construction_2$ $concrete\ structure\ building_2$ $steel\ structure\ building_2$	B_2 C_2 S_2
Expert 3	$Bridge\ construction_3$ $concrete\ structure\ building_3$ $steel\ structure\ building_3$	B_3 C_3 S_3

4.2.3. WBS sample adjustments

Due to the sensitivity of the semantic similarity measurements to the correct spelling of the words within the task nodes, any mistake could cause undesirable results. For instance, the similarity measurement of the words "concret" and "form" results in 0.28 which is scored by the string similarity measurement. By changing the "concret" to its correct spelling of "concrete", the semantic similarity measurement score increases to 0.625. Thus, the first adjustment was to correct the misspelled words in the WBS samples.

Since the WordNet is developed for general English language, some technical words of the construction domain are not defined in this lexical database. The second adjustment was to change this kind of words to their synonyms which are defined in the WordNet. For instance, the word “rebar”, which in construction stands for the “reinforcement rebar”, is not defined in WordNet, therefore, it was changed to the “reinforcing”.

A Similar problem was found with the words “HVAC”, “formwork” and “elastomeric” which were used in tasks like “HVAC installation”, “formwork installation” and “elastomeric pads”. The word “HCAV” was changed to the “heating ventilation and air conditioning”, the word “formwork” was altered to “form work”, and the word “bearing” was used instead of the “elastomeric”.

These modified samples were in B_3, C_1 and S_1, and these three adjusted samples together with the rest of the samples are presented in Table 10 to Table 18 in Appendix A.

4.2.4. Encoding WBS samples in XML language

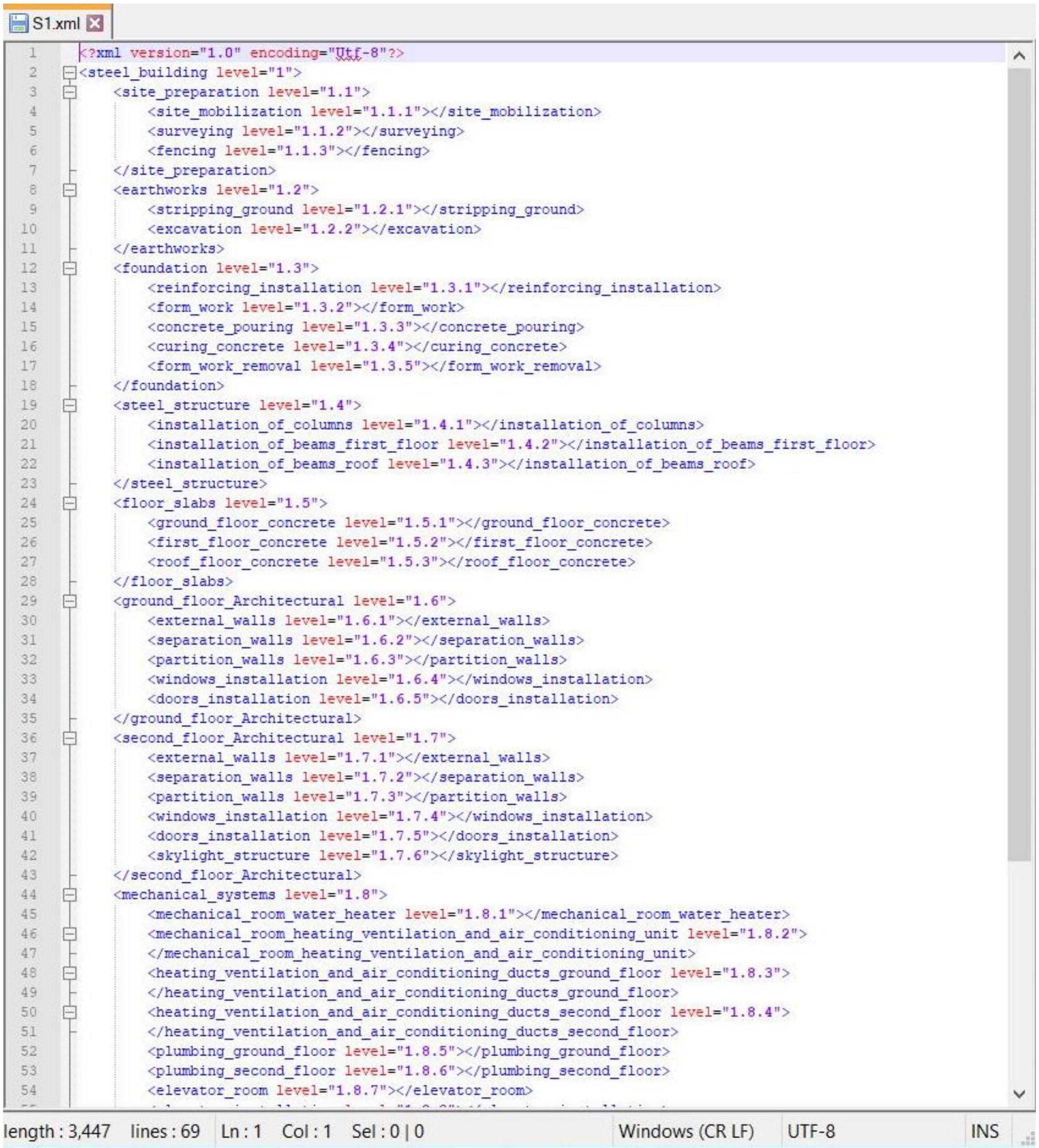
The WBS samples were initially developed in MS project software by the experts, which were then exported to spreadsheet format to apply the required adjustments, and then were encoded in XML language. The XML files were saved in the documented samples file for further experiments. Figure 17 and Figure 18 illustrate a part of XML documents which were written for the B_1 and S_1 samples.

B1.xml


```xml
<?xml version="1.0" encoding="UTF-8"?>
<bridge_project level="1">
    <site_preparation level="1.1">
        <site_mobilization level="1.1.1"></site_mobilization>
        <transport_equipment level="1.1.2"></transport_equipment>
        <surveying level="1.1.3"></surveying>
        <securing_the_site level="1.1.4"></securing_the_site>
        <temporary_buildings level="1.1.5"></temporary_buildings>
    </site_preparation>
    <earthworks level="1.2">
        <vegetation_removal level="1.2.1"></vegetation_removal>
        <stripping_ground level="1.2.2"></stripping_ground>
        <excavation level="1.2.3"></excavation>
    </earthworks>
    <substructure level="1.3">
        <pile_driving level="1.3.1"></pile_driving>
        <pile_caps level="1.3.2">
            <reinforcement_installation level="1.3.2.1"></reinforcement_installation>
            <form_work level="1.3.2.2"></form_work>
            <concrete_pouring level="1.3.2.3"></concrete_pouring>
            <curing_concrete level="1.3.2.4"></curing_concrete>
            <form_work_removal level="1.3.2.5"></form_work_removal>
        </pile_caps>
        <left_abutment level="1.3.3">
            <reinforcement_installation level="1.3.3.1"></reinforcement_installation>
            <form_work level="1.3.3.2"></form_work>
            <concrete_pouring level="1.3.3.3"></concrete_pouring>
            <curing_concrete level="1.3.3.4"></curing_concrete>
            <form_work_removal level="1.3.3.5"></form_work_removal>
            <backfilling level="1.3.3.6"></backfilling>
        </left_abutment>
        <right_abutment level="1.3.4">
            <reinforcement_installation level="1.3.4.1"></reinforcement_installation>
            <form_work level="1.3.4.2"></form_work>
            <concrete_pouring level="1.3.4.3"></concrete_pouring>
            <curing_concrete level="1.3.4.4"></curing_concrete>
            <form_work_removal level="1.3.4.5"></form_work_removal>
            <backfilling level="1.3.4.6"></backfilling>
        </right_abutment>
    </substructure>
    <super_structure level="1.4">
        <piers level="1.4.1">
            <reinforcement_installation level="1.4.1.1"></reinforcement_installation>
            <form_work level="1.4.1.2"></form_work>
            <concrete_pouring level="1.4.1.3"></concrete_pouring>
            <form_work_removal level="1.4.1.4"></form_work_removal>
        </piers>
        <column_caps level="1.4.2">
            <reinforcement_installation level="1.4.2.1"></reinforcement_installation>
            <form_work level="1.4.2.2"></form_work>
            <concrete_pouring level="1.4.2.3"></concrete_pouring>
            <form_work_removal level="1.4.2.4"></form_work_removal>
            <bearing_pads level="1.4.2.5"></bearing_pads>
        </column_caps>
```


length : 3,411 lines : 78 Ln : 1 Col : 1 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

Figure 17. A part of the written XML for the test sample B1



```xml
<?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>
    <second_floor_Architectural level="1.7">
        <external_walls level="1.7.1"></external_walls>
        <separation_walls level="1.7.2"></separation_walls>
        <partition_walls level="1.7.3"></partition_walls>
        <windows_installation level="1.7.4"></windows_installation>
        <doors_installation level="1.7.5"></doors_installation>
        <skylight_structure level="1.7.6"></skylight_structure>
    </second_floor_Architectural>
    <mechanical_systems level="1.8">
        <mechanical_room_water_heater level="1.8.1"></mechanical_room_water_heater>
        <mechanical_room_heating_ventilation_and_air_conditioning_unit level="1.8.2">
        </mechanical_room_heating_ventilation_and_air_conditioning_unit>
        <heating_ventilation_and_air_conditioning_ducts_ground_floor level="1.8.3">
        </heating_ventilation_and_air_conditioning_ducts_ground_floor>
        <heating_ventilation_and_air_conditioning_ducts_second_floor level="1.8.4">
        </heating_ventilation_and_air_conditioning_ducts_second_floor>
        <plumbing_ground_floor level="1.8.5"></plumbing_ground_floor>
        <plumbing_second_floor level="1.8.6"></plumbing_second_floor>
        <elevator_room level="1.8.7"></elevator_room>
```


Figure 18. A part of the written XML for the test sample S1

4.3. Results

The results of this study can be organized into two sections. First, precision of the node to node similarity measurements in mapping the relative task nodes between WBS test samples is discussed, and second, the precision of the overall similarity scores in comparing the WBS test samples and retrieving the similar ones, in terms of precision and recall measures is examined. In each part, the effects of two important variables with a significant impact on the precision of the measurements are studied.

The first variable is the threshold used in measuring node to node similarity metrics, which was introduced in Chapter 3. This threshold was determined to measure the $sim_{parents}$ and $sim_{siblings}$ and finally, map the nodes with a similarity higher than the threshold. The other variable is the weight of the $sim_{semantic}$ in measuring the $sim_{average}$ (Eq. 16). The $sim_{semantic}$ has a higher impact on the node to node similarity scores than the other two measurements. For this reason, only the effect of this metric is investigated.

Before discussing the results, it is important to explain the precision and recall measures. The definition of these two terms can be defined based on the binary relevance judgment in which every retrievable item is recognizably "relevant" or "not relevant" (Buckland and Gey 1994). Hence, in a search result, each item is placed in only one of the four groups in which the items are "relevant" or "not relevant" and "retrieved" or not "retrieved" (Buckland and Gey 1994).

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 can be 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 (Buckland and Gey 1994).

4.3.1. Node mapping precision

When two WBSs are compared, the system only maps the nodes with a similarity score higher than the defined threshold. In this section, the terms "mapped" and "not mapped" are used instead of the terms "retrieved" and "not retrieved" to evaluate the precision. Therefore, as it can be seen in the following equation that the mapping precision can be measured by dividing the number of mapped relevant nodes to the total number of mapped nodes.

$$Mapping\ precision\ = \frac{|\{Mapped\ task\ nodes\} \cap \{Relevant\ task\ nodes\}|}{|Mapped\ task\ nodes|}$$

Since human judgment for assessing the relevance of the mapped nodes is a costly and time consuming task (Samimi and Ravana 2014), two sets of rules were considered which helped the relevance assessment of the mapped tasks: First, the nodes which represent the same tasks, for instance, in Table 2 showing the mapped nodes for the pair (S_1, S_2), the nodes "fencing" and "fences installation" are relevant. Second, the nodes which are subdivided from the same tasks, although they do not represent the same tasks, they are still relevant since their completion fulfilled finishing of the same tasks. For example, the pairs such as "external walls" and "painting" are not referring to the same task, but they are somehow related since both tasks are carried out in walls finishing jobs.

The nodes that do not fulfill the overmentioned condition are categorized as not relevant nodes. Some tasks, such as "elevator installation" and "form work removal", are highlighted in red which means that they are not relevant.

To examine the precision of the system in mapping relevant task nodes, the pairs (S_1, S_2) and (B_1, C_2) are randomly chosen to be investigated. The precision is explored in four different scenarios as follow.

1. $threshold = 0.5\ ,\ sim_{average} = \frac{sim_{semantic}+sim_{parents}+sim_{siblings}}{3}$

2. $threshold = 0.7$, $sim_{average} = \frac{sim_{semantic}+sim_{parents}+sim_{siblings}}{3}$

3. $threshold = 0.5$, $sim_{average} = \frac{2*sim_{semantic}+sim_{parents}+sim_{siblings}}{4}$

4. $threshold = 0.7$, $sim_{average} = \frac{2*sim_{semantic}+sim_{parents}+sim_{siblings}}{4}$

In the first and second scenarios, the weight of $sim_{semantic}$ was considered as one and the threshold values of 0.5 and 0.7 were used. In the third and fourth scenarios, the weight of the $sim_{semantic}$ was increased to two and the threshold values were the same as the first two cases.

Table 2 shows the results of the mapped nodes for the projects (S_1 , S_2) in the first scenario. The results of the second to the fourth scenarios for this pair and all the scenarios for the pair (B_1 , C_2) are provided in the Table 19 to Table 25 (Appendix B). Table 3 provides a summary of these results.

Each table presents a list of all mapped nodes accompanied by the node to node similarity measurements scores, including $sim_{semantic}$, $sim_{parents}$, $sim_{siblings}$ and the $sim_{average}$. The pairs of task nodes, which are mapped and not relevant, are highlighted in red. At the end of each table, the overall measurements between two WBS, such as similarity scores, the total number of mapped nodes and unmapped nodes as well as the number of irrelevant mapped nodes (highlighted in red) are provided.

Table 2. Mapped nodes for the pair (S_1 , S_2), *(First scenario)*

Row	$steel\ building_1$		$steel\ building_2$		$sim_{semantic}$	$sim_{parents}$	$sim_{siblings}$	$sim_{average}$
	WBS code	**Task node**	**WBS code**	**Task node**				
1	1	steel building	1	steel building	1.000	1.000	0.000	0.667
2	1.1	site preparation	1.1	Site preparation	1.000	1.000	0.533	0.844

Row	$steel\ building_1$		$steel\ building_2$		$sim_{semantic}$	$sim_{parents}$	$sim_{siblings}$	$sim_{average}$
	WBS code	Task node	WBS code	Task node				
3	1.1.1	site mobilization	1.1.2	equipment mobilization	0.853	1.000	0.971	0.941
4	1.1.2	surveying	1.1.1	surveying	1.000	1.000	0.898	0.966
5	1.1.3	fencing	1.1.3	fences installation	0.942	1.000	0.926	0.956
6	1.2	earthworks	1.2	earthworks	1.000	1.000	0.533	0.844
7	1.2.1	stripping ground	1.2.1	stripping	0.850	1.000	1.000	0.950
8	1.2.2	excavation	1.2.2	excavation	1.000	1.000	0.850	0.950
9	1.3	foundation	1.5	electrical works	0.529	1.000	0.601	0.710
10	1.3.1	reinforcing installation	1.3.1.2	reinforcing placement	0.816	0.786	0.946	0.850
11	1.3.2	form work	1.6.2	second floor finishing	0.788	0.899	0.568	0.752
12	1.3.3	concrete pouring	1.3.1.3	concrete pouring	1.000	0.786	0.900	0.896
13	1.3.4	curing concrete	1.3.1.4	curing	0.833	0.786	0.942	0.854
14	1.3.5	form work removal	1.6.1	ground floor finishing	0.840	0.899	0.600	0.780
15	1.4	steel structure	1.3	structure	0.917	1.000	0.565	0.827
16	1.4.1	installation of columns	1.3.1	foundation	0.742	0.967	0.685	0.798
17	1.4.2	installation of beams first floor	1.4.3	elevator	0.657	0.845	0.725	0.743
18	1.4.3	installation of beams roof	1.3.3	slabs	0.647	0.967	0.732	0.782
19	1.5	floor slabs	1.4	mechanical works	0.567	1.000	0.595	0.721
20	1.5.1	ground floor concrete	1.4.1	ground floor	0.952	0.827	0.765	0.848

Row	$steel\ building_1$		$steel\ building_2$		$sim_{semantic}$	$sim_{parents}$	$sim_{siblings}$	$sim_{average}$
	WBS code	Task node	WBS code	Task node				
21	1.5.2	first floor concrete	1.4.2	second floor	0.911	0.827	0.786	0.841
22	1.5.3	roof floor concrete	1.3.2	steel structure	0.788	0.954	0.758	0.833
23	1.6	ground floor Architectural	1.3.2.1.3	joints	0.667	0.588	0.258	0.504
24	1.6.1	external walls	1.6.1.6	painting	0.594	0.617	0.823	0.678
25	1.6.2	separation walls	1.6.1.2	dry walls installation	0.808	0.617	0.758	0.728
26	1.6.3	partition walls	1.6.1.1	exterior walls installation	0.897	0.617	0.738	0.751
27	1.6.4	windows installation	1.6.1.3	windows installation	1.000	0.617	0.715	0.778
28	1.6.5	doors installation	1.6.1.4	doors installation	1.000	0.617	0.683	0.767
29	1.7.1	external walls	1.6.2.6	painting	0.594	0.614	0.886	0.698
30	1.7.2	separation walls	1.6.2.2	dry walls installation	0.808	0.614	0.809	0.743
31	1.7.3	partition walls	1.6.2.1	exterior walls installation	0.897	0.614	0.791	0.767
32	1.7.4	windows installation	1.6.2.3	windows installation	1.000	0.614	0.770	0.795
33	1.7.5	doors installation	1.6.2.4	doors installation	1.000	0.614	0.741	0.785
34	1.7.6	skylight structure	1.6.2.5	flooring	0.724	0.614	0.824	0.721
35	1.8.1	mechanical room water heater	1.3.2.1.1	steel columns installation	0.729	0.610	0.319	0.553
36	1.8.2	mechanical room heating ventilation and air conditioning unit	1.4.2.2	heating ventilation and air conditioning installation	0.908	0.462	0.164	0.511

Row	$steel\ building_1$		$steel\ building_2$		$sim_{semantic}$	$sim_{parents}$	$sim_{siblings}$	$sim_{average}$
	WBS code	Task node	WBS code	Task node				
37	1.8.3	heating ventilation and air conditioning ducts ground floor	1.3.2.1	ground floor steel installation	0.833	0.607	0.189	0.543
38	1.8.4	heating ventilation and air conditioning ducts second floor	1.4.1.2	heating ventilation and air conditioning installation	0.935	0.484	0.164	0.528
39	1.8.5	plumbing ground floor	1.6.1.5	flooring	0.911	0.510	0.627	0.683
40	1.8.6	plumbing second floor	1.3.2.2	second floor steel installation	0.921	0.607	0.189	0.572
41	1.8.7	elevator room	1.5.2.4	fire alarm systems	0.775	0.462	0.371	0.536
42	1.8.8	elevator installation	1.3.1.5	form work removal	0.714	0.474	0.472	0.553
43	1.9.1	distribution boards	1.5.1.4	fire alarm systems	0.839	0.484	0.571	0.631
44	1.9.2	electrical wiring ground floor	1.3.1.1	form work installation	0.728	0.474	0.623	0.608
45	1.9.3	electrical wiring second floor	1.5.2	second floor	0.838	0.375	0.305	0.506
46	1.9.4	smoke detectors	1.5.1.3	install light fixtures	0.700	0.484	0.620	0.602
47	1.9.5	outlets and switches	1.5.1.2	install outlets switches	0.774	0.484	0.623	0.627
48	1.10	finishing works	1.6	finishing works	1.000	1.000	0.568	0.856
49	1.10.1	landscaping	1.5.1	ground floor	0.464	0.855	0.466	0.595

Row	$steel\ building_1$		$steel\ building_2$		$sim_{semantic}$	$sim_{parents}$	$sim_{siblings}$	$sim_{average}$
	WBS code	Task node	WBS code	Task node				
50	1.10.2	testing systems	1.6.4	testing systems	1	1	0.4	0.8
51	1.10.3	cleaning	1.6.3	cleaning	1	1	0.4	0.8

Similarity scores and measurements					
mapped nodes	Unmapped nodes	Mapped not relevant	Node similarity	Structural similarity	Total similarity
102	19	32	0.621	0.806	0.713
Mapping precision			0.69		

Table 3. Summary of mapping precision results for (S_1 , S_2) and (B_1 , C_2)

Pair of samples	Scenario	Total similarity score	Total number of nodes	Number of mapped nodes	Mapped relevant	Mapping precision
(S_1 , S_2)	1st	0.713	121	102	70	0.69
	2nd	0.607	121	64	62	0.97
	3rd	0.741	121	108	72	0.67
	4th	0.628	121	68	62	0.91
(B_1 , C_2)	1st	0.576	196	126	44	0.35
	2nd	0.505	196	90	64	0.71
	3rd	0.587	196	126	54	0.43
	4th	0.532	196	96	64	0.67

4.3.1.1. Discussion

The results in Table 2 shows that the system had a promising performance in mapping relevant nodes with an overall 0.69 mapping precision. From 121 nodes of the WBSs of samples S_1 and S_2, 102 nodes were mapped together, in which 32 were not relevant. Manual investigation of these non-relevant mapped nodes revealed that the $sim_{semantic}$ had a high score in most cases. The main reason of this limitation is the lexical database which was utilized in this study. Some words are semantically similar in the English language, but they are less similar in the construction domain. This problem can be improved by employing a customized lexical database developed for the construction domain.

The results in Table 3 show that the increase of the threshold from 0.5 to 0.7 increased the mapping precision as well (0.5 and 0.7 were two thresholds which were changed from first to second and third to fourth scenarios). A higher threshold only maps the nodes with a higher degree of similarity, and this will increase the chance of relevance of the mapped nodes.

In addition, the results in Table 3 indicate that increasing the weight of semantic similarity (second variable) does not significantly affect the mapping precision for these two cases. It is interesting to note that, although the precision is not affected considerably, the overall total similarity score is increased for the pairs (S_1 , S_2) and (B_1 , C_2). This increase is about 3% for the (S_1 , S_2) and 5% for the (B_1 , C_2). It is speculated that increasing the weight of the $sim_{semantic}$ may increase the similarity of non-similar projects more than the similarity of similar projects, which is not a desirable result. If this hypothesis is true, the retrieving of non-similar cases may decrease retrieving precision. This hypothesis is further investigated in the next section by experimenting with all the samples.

4.3.2. Overall similarity scores

In this section, the overall similarity scores, which are measured by three metrics of node similarity, structural, and total similarity, are discussed. These results are explored in two parts: First, the compliance of the results with similarity measure properties is examined. In the second part, the performance of the method is evaluated in the retrieving process in which the method searches through the stored WBS samples to find the similar ones to a sample which is given to the system. This evaluation is performed based on two measures: precision and recall.

The results in this section are obtained through experiments in two main scenarios. In both scenarios, the threshold varies in the range of 0.50 to 0.80 with 0.05 intervals. In the first scenario, the weight of the $sim_{semantic}$ was one and in the second scenario, it was increased to two.

1)

Threshold = {0.50, 0.55, 0.60, 0.65, 070, 0.75, 0.80}

$$sim_{average} = \frac{sim_{semanti\ c} + sim_{parents} + sim_{siblings}}{3}$$

2)

Threshold = {0.50, 0.55, 0.60, 0.65, 070, 0.75, 0.80}

$$sim_{average} = \frac{2 * sim_{semanti\ c} + sim_{parents} + sim_{siblings}}{4}$$

4.3.2.1. Properties of similarity measures

The overall similarity measurements must fulfill the properties of symmetry and reflexivity (Richter 1993). A similarity function S: S×S→ [0,1] on a set S measuring the degree of similarity between two elements, is called similarity measure if, $\forall X, Y \in S$:

Eq. 28

$$Sim\ (X, Y) = Sim\ (Y, X)\ (Symmetry)$$

$$Sim\ (X, X) = 1\ (Reflexivity)$$

Symmetry

In this system to determine the symmetry fulfillment, the symmetry error for two WBSs such as A and B is computed by using the following equation. In this equation, the sim can be one of the three overall similarity measurements (total similarity, node similarity or structural similarity).

Eq. 29

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

The symmetry errors of all the possible pairwise comparisons from the test samples were measured and averaged for different overall similarity metrics. Table 4 and Table 5 present the average of the symmetry errors for the first and the second scenario. Also, these results are illustrated in Figure 19 and Figure 20.

Table 4. The average of the symmetry errors (first scenario)

Threshold	Symmetry error obtained by		
	Total similarity score	**Node similarity score**	**Structural similarity score**
0.5	0.008	0.015	0.004
0.55	0.009	0.018	0.003
0.6	0.009	0.019	0.003
0.65	0.013	0.035	0.004
0.7	0.018	0.057	0.004
0.75	0.022	0.108	0.003
0.8	0.004	0.026	0.000
Average		0.018	

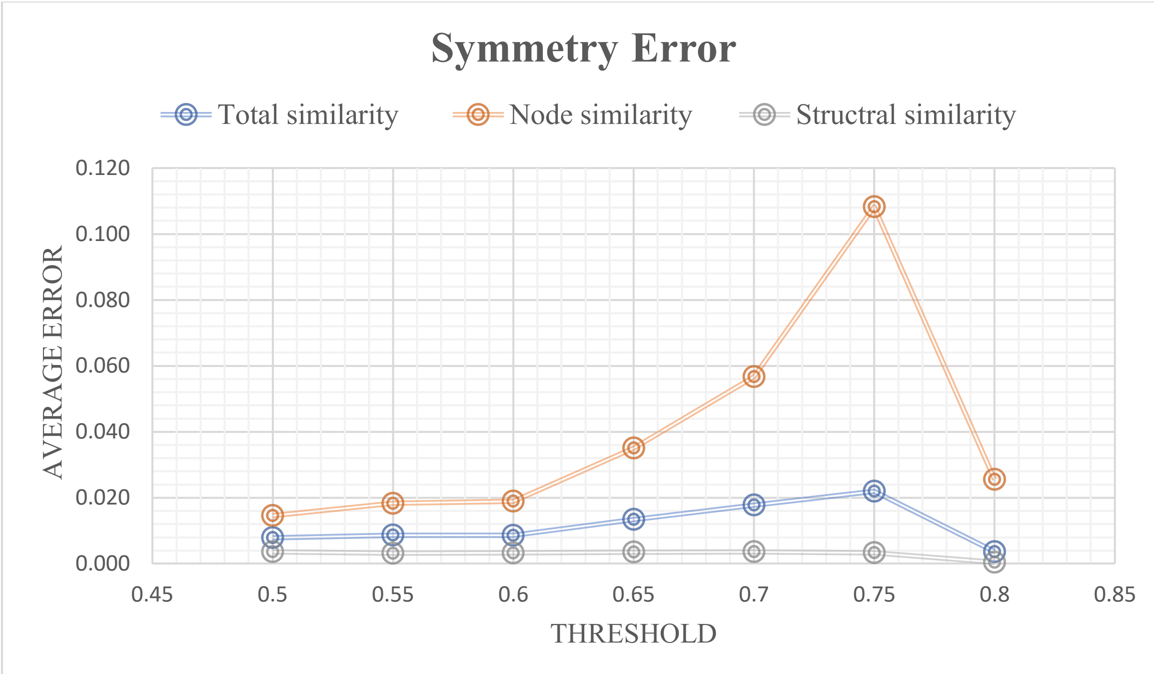


Figure 19. The average of the symmetry errors (first scenario)

Table 5. The average of the symmetry errors (second scenario)

Threshold	Symmetry error obtained by		
	Total similarity score	**Node similarity score**	**Structural similarity score**
0.5	0.006	0.009	0.003
0.55	0.007	0.013	0.003
0.6	0.008	0.016	0.003
0.65	0.009	0.019	0.003
0.7	0.011	0.029	0.003
0.75	0.014	0.046	0.003
0.8	0.015	0.071	0.003
Average		0.014	

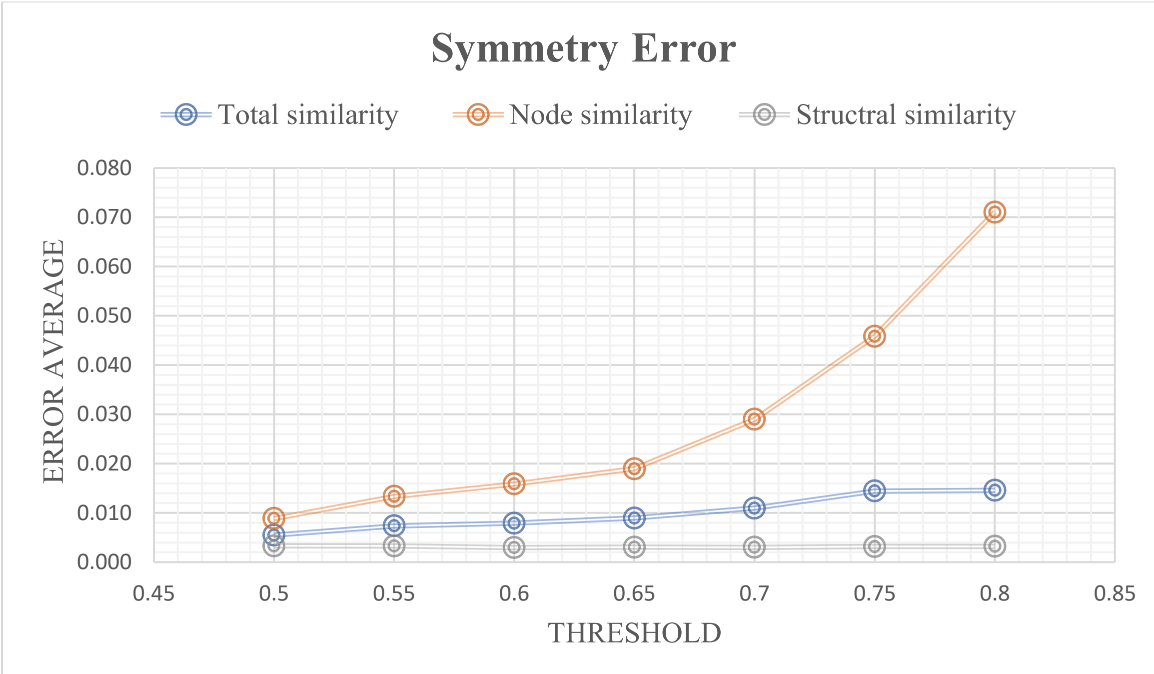


Figure 20. The average of the symmetry errors (second scenario)

Discussion

In both scenarios, the node similarity had larger errors, specifically as the threshold was increased, and the structural similarity measurement performed better than the other two metrics in the symmetry analysis.

As it can be seen in Figure 19, the symmetry errors for all three metrics are increasing and for thresholds higher than 0.75, it starts to decline. We speculate that declining of the symmetry errors might be due to increasing of the threshold in which only the nodes with a high similarity score are mapped together and this increases the chance of being relevant for the mapped nodes. When most of the nodes are relevant, the symmetry effects will be decreased and as a result, the symmetry error declines.

To sum up, the symmetry property of this method performed rather well with the average symmetry error of 0.018 and 0.014 for the first and second scenarios, respectively. However, when comparing the result of the first with the second scenario, it must be pointed out that the system is performing better in the first scenario, especially with Structural and Total similarity measurements, because both are declining to zero for the threshold more than 0.75.

Reflexivity

The reflexivity error (e.g. for the sample B_1) is calculated using the following equation (Richter 1993):

$$Reflexivity\ error = 1 - sim(B_1, B_1)$$

The reflexivity errors were obtained by comparing the test samples with themselves. The average of the results are presented in Table 6 and Table 7 for the first and the second scenarios, respectively. The summarized results are graphically presented in Figure 21 and Figure 22.

Table 6. The average of the reflexivity errors (first scenario)

Threshold	Reflexivity error obtained by		
	Total similarity	**Node similarity**	**Structural similarity**
0.5	0.005	0.007	0.003
0.55	0.005	0.007	0.003
0.6	0.005	0.007	0.003
0.65	0.005	0.007	0.003
0.7	0.012	0.020	0.005
0.75	0.012	0.020	0.005
0.8	0.012	0.020	0.005
Average		0.008	

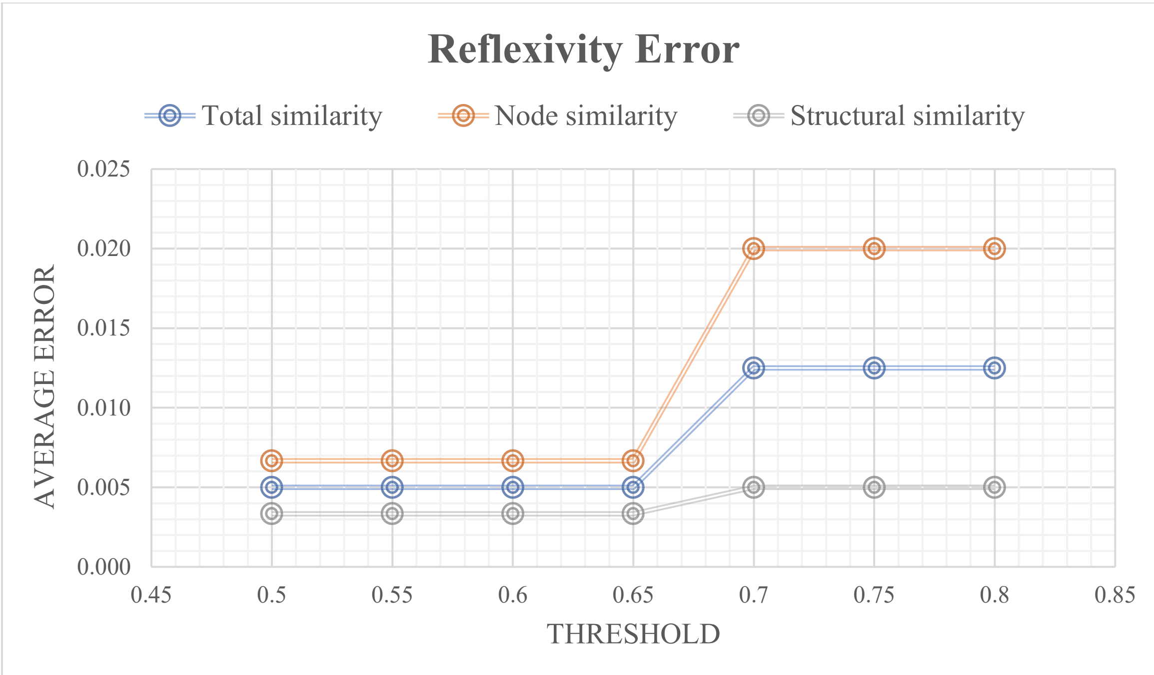


Figure 21. The average of the reflexivity errors (first scenario)

Table 7. The average of the reflexivity errors (second scenario)

Threshold	Reflexivity error obtained by		
	Total similarity	Node similarity	Structural similarity
0.5	0.004	0.005	0.002
0.55	0.004	0.005	0.002
0.6	0.004	0.005	0.002
0.65	0.004	0.005	0.002
0.7	0.004	0.005	0.002
0.75	0.004	0.005	0.002
0.8	0.012	0.020	0.005
Average		0.005	

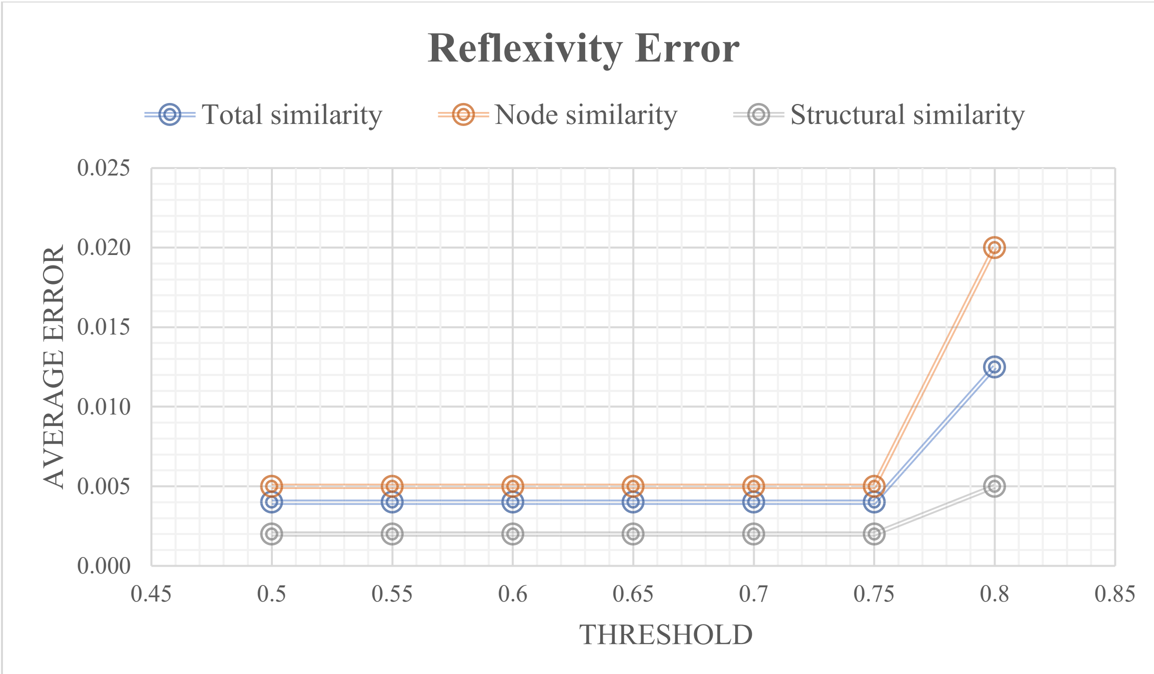


Figure 22. The average of the reflexivity errors (second scenario)

Discussion

The average reflexivity errors of 0.008 and 0.005 for two scenarios confirm that the system generated promising results and the reflexivity errors were negligible across the various thresholds. However, it should be mentioned that the structural similarity metric performed better than two other metrics in both scenarios.

4.3.2.2. Retrieval precision and recall

This section discusses the performance of the three overall similarity metrics in retrieving the similar stored samples to the query sample (given sample). For this purpose, the samples B_1 , C_1 and S_1 were chosen as the query samples and the rest of samples were considered as the stored samples. The reason for choosing different types of samples as query samples was to study the effect of various types of test samples in task labelling and structure.

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 and S_1) was compared with all the stored samples and the results were ranked from the highest to the lowest similarity score. For instance, Table 8 and Table 9 show the results in the first and the second scenarios and thresholds of 0.65 and 0.75 respectively. In these two tables, the B_1 is set as the query sample and the results are ordered by the Total similarity score.

Table 8. Comparing B1with stored samples with the threshold of 0.65 (First scenario)

Query sample	Documented sample	Total similarity score	Node similarity score	Structural similarity score
B_1	B_2	0.719	0.635	0.803
B_1	S_2	0.688	0.561	0.815
B_1	B_3	0.634	0.479	0.789
B_1	C_1	0.599	0.467	0.731
B_1	S_1	0.589	0.398	0.779
B_1	S_3	0.557	0.351	0.762
B_1	C_2	0.531	0.392	0.670
B_1	C_3	0.508	0.336	0.680

Table 9. Comparing B1with the stored samples with the threshold of 0.75 (second scenario)

Query sample	Documented sample	Total similarity score	Node similarity score	Structural similarity score
B_1	B_2	0.665	0.530	0.799
B_1	B_3	0.539	0.305	0.774
B_1	S_1	0.533	0.291	0.776
B_1	C_1	0.527	0.322	0.731
B_1	S_2	0.517	0.260	0.774
B_1	S_3	0.515	0.268	0.762
B_1	C_2	0.462	0.248	0.676
B_1	C_3	0.457	0.228	0.685

The retrieving precision is calculated by the following equation that calculates the number of retrieved relevant samples as a proportion of retrieved samples. In this part, the relevance arguments are not challenging, because the sample tests were developed for the same project by different experts and are relative. 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.

$$Retrieving\ precision = \frac{|\{Relevant\ samples\} \cap \{Retrived\ samples\}|}{|\{Retrieved\ samples\}|}$$

The number of retrieved samples are determined by the recall score, in other words the retrieving procedures continue until the number of retrieved relevant samples fulfill the recall score. Because of the small number of stored WBS samples and therefore small number of relevant samples, only two recall scores are considered in this study. In the recall score of 0.5, the retrieving of stored samples continues until one of the two relevant samples among the stored samples is retrieved. The second recall score is 1 in which both relevant samples are retrieved.

For instance, from Table 8, the precision scores for retrieving the B_1 were obtained as follow: As it can be seen, for the recall score of 1.0, the retrieving process continues until both relevant samples to B_1 are retrieved. This results in two retrieved relative samples out of three retrieved samples and 0.66 precision. For the recall score of 0.5 only one of the relative samples to the B_1 must be retrieved which is achieved by retrieving only the first sample from Table 8 and it results in a precision that equals to one.

1. $Recall = 0.5$ $\quad$ $Retrieving\ precision = \frac{|\{B_2\} \cap \{B_2\}|}{|\{B_2\}|} = \frac{1}{1} = 1.0$

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

The averages of the precision scores in each scenario are presented in Table 26 and

Table 27 (Appendix C) which are summarized in Figure 23 and Figure 24.

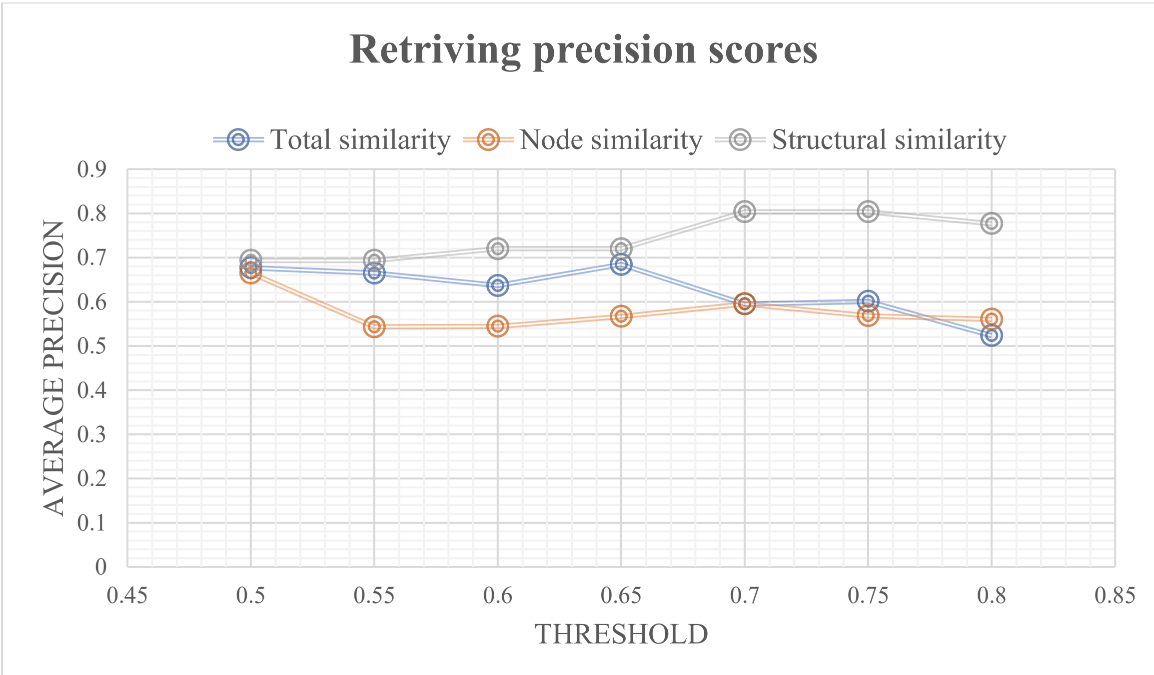


Figure 23. Average precision for the first scenario

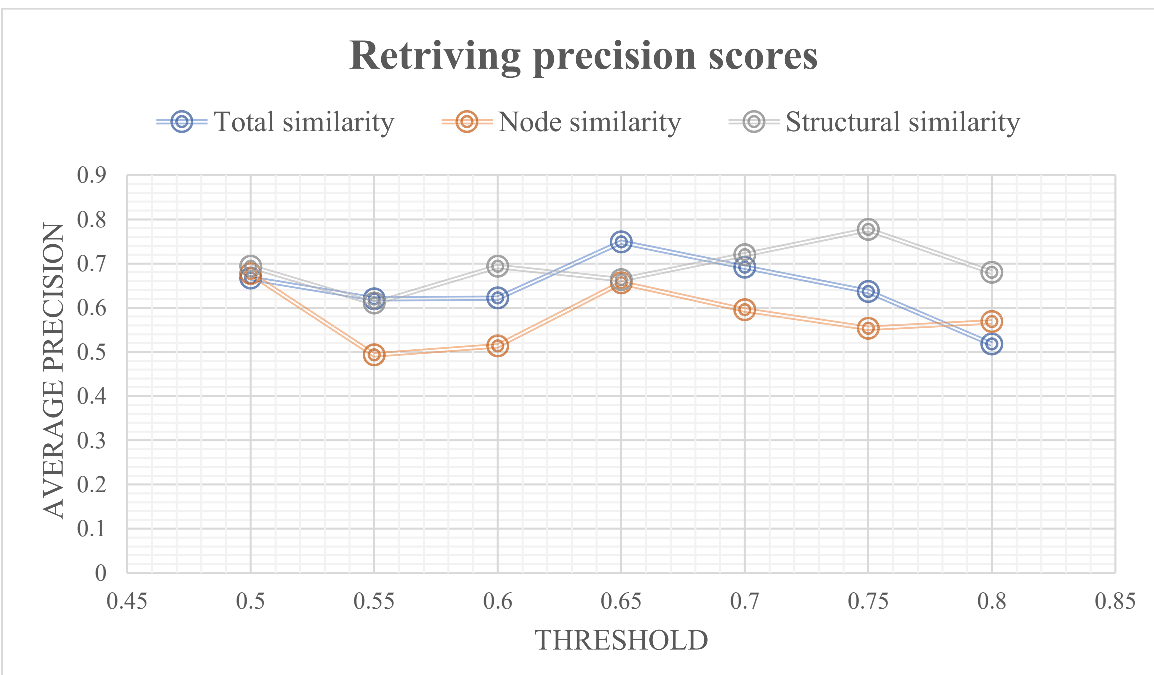


Figure 24. Average precision for the second scenario

Discussion

From Figure 23 and Figure 24, it is clear that the Structural similarity measurement is delivering higher precision scores than the Total and Node similarity metrics especially in the first scenario with an average precision of 0.8. In both figures, the retrieval precision (for structural similarity) increases with the increase of threshold until 0.75, and after that starts to decline, especially with a higher rate for the second scenario. The higher rate of declining for the second scenario might be due to the higher weight for the $sim_{semantic}$ which increases the impact of changing the threshold on the semantic similarity metrics and therefore on the retrieving precision.

Overall, these two figures demonstrate the ability of the system in retrieving relative models with an average precision of 70% for the thresholds in the range of 0.7 to 0.75. However, when comparing the results in both figures, it can be concluded that the first scenario is giving more stable results among various values of thresholds, namely in the Structural similarity metric and the threshold values in the range of 0.7 to 0.75.

4.3.3. Conclusion

In this chapter, a set of experiments was conducted to explore the precision of the method in finding relative tasks and mapping them together in comparing two WBSs. Determination of the tasks' relativeness in this part was challenging and two rules were defined to differentiate the relative and not relative arguments. A better approach to produce more promising results on this experiment could be to ask different experts to argue the relativity of the mapped nodes which was not possible in limited timeframe of this research.

Three similarity metrics were proposed to measure the similarity of WBSs which were tested by two set of experiments. The first experiment evaluated the similarity properties of the proposed method. The experiment revealed promising results with very low errors in the symmetry and reflexivity properties (average error of 0.016 for both properties). The most important goal of this chapter was the experiments on retrieving precision. The experiments on this part show that the system can retrieve the relative samples with an average precision of 0.7 for thresholds in the range of 0.70 to 0.75.

The retrieval system with Structural similarity metric outperforms in the first scenario with the thresholds in the range of 0.7 to 0.75. A major limitation on the experiments was the lack of enough test samples in the database. With access to a larger database containing different types of construction projects, the method could be tested more comprehensively and the results could be more reliable.

4.3.4. Run time efficiency

The tests were conducted on a computer with a 3.6 GHz quad-core intel processor, 16GB RAM, and using Python version 3.6 compiler. The execution time for comparing a pair of WBS with 150 nodes took around two minutes. The following adjustments can be considered to reduce the computation time of this method.

1. For each two words comparison, the system searches through a huge lexical database used in this research (WordNet), which includes all English words developed for professional language processing. In this research, application of a smaller database, customized for construction-related words, can considerably decrease the running time of the system.

2. Another reason that increases the computation time is the issue which was explained in Chapter 3, which occurs in the word-to-word semantic similarity measurements. In this

method, all the existed synset of two given words are compared and the highest score is chosen. Having a database in which each word is tagged only with the construction related synsets can reduce the running time.

3. The developed system executes all the similarity metrics, which can be limited only to measurements with the best performances. This can reduce the running time since the amount of calculation can be reduced dramatically.

Chapter 5 Conclusion

5.1. Summary and Conclusion

Reuse of the knowledge and experiences gained from previously completed construction projects can improve the time and cost efficiency of the new projects. In order to reuse knowledge, finding similar past projects is crucial. This research was undertaken to develop three semantic based similarity metrics to measure the similarity of construction projects using their Work Breakdown Structure (WBS) as their representative, which include:

1. Node similarity measurement comparing the semantic of elements of two WBSs;
2. Structural similarity measurement comparing the topology and semantic of two WBSs;
3. Total similarity measurement which is the average of Node and Structural similarity measurements.

These metrics were performed by utilizing NLP techniques written in Python programing language. The similarity metrics were evaluated based on two sets of experiments: First the metrics were tested for the similarity properties fulfillment, including symmetry and reflexivity; second, the metrics were tested to search between WBS test samples and to find the similar ones to the given samples. Subsequently, two information retrieval metrics, including precision and recall, were used to evaluate the retrieval performance.

The results show promising results 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 method can retrieve similar projects with an average precision of 0.65 among all of the threshold. It should be noted that the structural similarity in the first scenario outperforms the other two metrics in the retrieval process with an average precision of 0.8 for the thresholds in the range of 0.7 to 0.75.

5.2. Limitations

The findings of this study must be seen in the light of some limitations, and the main limitations of this study can be noted as follow:

1. A major limitation of this research project was an insufficient test sample size. The experimental tests should be carried out on a larger sample size to further investigate performance of this system.

2. Lack of previous research efforts on this topic was another limitation of this study, because there is no benchmark to compare the performance of this method.

3. An apparent limitation of this system is the utilized lexical database (WordNet). It is a massive database of words which most of them are not used in the construction domain. The second problem of this source is that each word contains all the available synsets and synonyms, which most of them are not related to construction. Therefore, a database, containing technical construction words which are tagged with intended synsets, can improve the efficiency of the method.

4. Running time of the developed program is another issue, because the system has to search various scenarios in a generic lexical database. A specialized lexical database, which is developed for construction industry, can significantly reduce the computation

efforts and also increase the accuracy of semantic matching.

5. Another limitation of the proposed method is reusing of the information and documents of projects from different geographic locations. Although the system can compare the construction projects from different geographic locations, the effect of geographical situations such as weather and economy on the construction documents (e.g. schedules and cost estimation) should be considered.

5.3. Recommendations for Future Work

There are many opportunities for further investigation on this topic to advance this method, which can benefit intelligent knowledge management systems in construction industry:

1. Investigating the performance of the method by implementing the method on a larger test samples, including different types of construction projects.

2. Developing a platform to store all the information and documents related to past projects in a server and link it to the corresponding WBS. The WBS of the new project is given to the system and the platform can retrieve the corresponding information of the most similar project for further planning and development of the new project.

3. Although some vocabulary recourses have been developed in construction researches, they do not contain different senses of the technical words with complex relationships between them same as WordNet. So, they can not be used for word sense disambiguation (WSD) and measuring the similarities between words. There is a good opportunity in developing a semantic lexical database specialized for the construction technical words

for future semantic research studies in the construction management.

4. Developing a system based on the proposed method in which the effects of the geographic differences of construction projects in knowledge reusing is considered.

5.4. Developed Program Source Code

The developed program in Python programming language can be accessed by the following link:

https://osf.io/b8qvy/

References

Addis, Mark. 2016. 'Tacit and explicit knowledge in construction management', *Construction management and economics*, 34: 439-45.

An, Sung-Hoon, Gwang-Hee Kim, and Kyung-In Kang. 2007. 'A case-based reasoning cost estimating model using experience by analytic hierarchy process', *Building and Environment*, 42: 2573-79.

Bergmann, Ralph. 2002. *Experience management: foundations, development methodology, and internet-based applications* (Springer-Verlag).

Berners-Lee, Tim, James Hendler, and Ora Lassila. 2001. 'The semantic web', *Scientific american*, 284: 34-43.

Bhatt, Ganesh D. 2001. 'Knowledge management in organizations: examining the interaction between technologies, techniques, and people', *Journal of knowledge management*, 5: 68-75.

Bird, Steven, and Edward Loper. 2004. "NLTK: the natural language toolkit." In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, 31. Association for Computational Linguistics.

Bodenreider, O. 2003. "Biomedical Ontologies: Session Introduction O. Bodenreider, JA Mitchell, and AT McCray Pacific Symposium on Biocomputing 8: 562-564 (2003)." In *Pacific Symposium on Biocomputing*, 562-64.

Bray, Tim, Jean Paoli, C Michael Sperberg-McQueen, Eve Maler, and François Yergeau. 1997. 'Extensible markup language (XML)', *World Wide Web Journal*, 2: 27-66.

Buckland, Michael, and Fredric Gey. 1994. 'The relationship between recall and precision', *Journal of the American society for information science*, 45: 12-19.

BuildingSmart. 2012. 'IFD Library for BuildingSmart'. http://www.ifd-library.org/index.php?title=Home_Page.

Buldingsmart. https://www.buildingsmart.org.

Carneiro, Alberto. 2001. 'The role of intelligent resources in knowledge management', *Journal of knowledge management*, 5: 358-67.

Carrillo, Patricia M, Chimay J Anumba, and John M Kamara. 2000. 'Knowledge management strategy for construction: key IT and contextual issues', *Proceedings of CIT*, 2000: 28-30.

Carrillo, Patricia M, Herbert S Robinson, Chimay J Anumba, and Ahmed M Al-Ghassani. 2003. 'IMPaKT: A framework for linking knowledge management to business performance', *Electronic Journal of Knowledge Management*, 1: 1-12.

Chen, Serena H, Anthony J Jakeman, and John P Norton. 2008. 'Artificial intelligence techniques: an introduction to their use for modelling environmental systems', *Mathematics and computers in simulation*, 78: 379-400.

Costa, Ruben, Celson Lima, João Sarraipa, and Ricardo Jardim-Gonçalves. 2016. 'Facilitating knowledge sharing and reuse in building and construction domain: an ontology-based approach', *Journal of Intelligent Manufacturing*, 27: 263-82.

Dijkman, Remco, Marlon Dumas, Boudewijn Van Dongen, Reina Käärik, and Jan Mendling. 2011. 'Similarity of business process models: Metrics and evaluation', *Information Systems*, 36: 498-516.

Dretske, Fred. 1981. 'Knowledge and the flow of information MIT Press Cambridge', *MA Google Scholar*.

Eastman, Chuck, Paul Teicholz, Rafael Sacks, and Kathleen Liston. 2011. *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors* (John Wiley & Sons).

Egbu, CO. 2002. "Knowledge management, intellectual capital and innovation: their association, benefits and challenges for construction organisations." In *Proceedings of the 10th International Symposium on Construction Innovation and Global Competitiveness, W65*, 9-13.

Ehrig, Marc, Agnes Koschmider, and Andreas Oberweis. 2007. "Measuring similarity between semantic business process models." In *Proceedings of the fourth Asia-Pacific conference on Comceptual modelling-Volume 67*, 71-80. Australian Computer Society, Inc.

Flemming, Ulrich, Robert Coyne, and James Snyder. 1994. 'Case-based design in the SEED system', *Automation in construction*, 3: 123-34.

Gomaa, Wael H, and Aly A Fahmy. 2013. 'A survey of text similarity approaches', *International Journal of Computer Applications*, 68: 13-18.

Hammond, Kristian J. 1989. "Viewing Planning as a Memory Task, Aca." In.: demic Press.

Hart, Peter E, Nils J Nilsson, and Bertram Raphael. 1968. 'A formal basis for the heuristic determination of minimum cost paths', *IEEE transactions on Systems Science and Cybernetics*, 4: 100-07.

Haugan, Gregory T. 2001. *Effective work breakdown structures* (Berrett-Koehler Publishers).

Hayes-Roth, Frederick. 1985. 'Rule-based systems', *Communications of the ACM*, 28: 921-32.

Heisig, Peter, K Mertins, and Jens Vorbeck. 2001. "Knowledge Management. Concepts and Best practices in Europe." In.: ISBN 3-540-00490-4. Springer-Verlog. Berlin Heidelberg. New York.

Ibrahim, YM, AP Kaka, E Trucco, M Kagioglou, and A Ghassan. 2007. "Semi-automatic development of the work breakdown structure (WBS) for construction projects." In *Proceedings of the 4th International SCRI Research Symposium, Salford, UK*.

Joe, Carmel, Pak Yoong, and Kapila Patel. 2013. 'Knowledge loss when older experts leave knowledge-intensive organisations', *Journal of knowledge management*, 17: 913-27.

Jurafsky, Dan, and James H Martin. 2014. *Speech and language processing* (Pearson London).

Kamara, John M, Godfried Augenbroe, Chimay J Anumba, and Patricia M Carrillo. 2002. 'Knowledge management in the architecture, engineering and construction industry', *Construction innovation*, 2: 53-67.

Koenig, Michael ED. 2002. 'The third stage of KM emerges', *Km World*, 11: 20-20.

Kolonder. 1998. 'Judging which is the 'best' case for a case base reasoner. ', *In: Case-based reasoning Proc. Workshop in Case-Based Reasoning , Florida, Morgan-Kaufmann*: 221-28.

Kululanga, GK, and R McCaffer. 2001. 'Measuring knowledge management for construction organizations', *Engineering, construction and architectural management*, 8: 346-54.

Landauer, Thomas K, and Susan T Dumais. 1997. 'A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge', *Psychological review*, 104: 211.

Levenshtein, Vladimir I. 1966. "Binary codes capable of correcting deletions, insertions, and reversals." In *Soviet physics doklady*, 707-10.

Lima, Celson, Tamer El-Diraby, and Jeff Stephens. 2005. 'Ontology-based optimisation of knowledge management in e-Construction', *Journal of Information Technology in Construction (ITcon)*, 10: 305-27.

Maedche, Alexander, and Steffen Staab. 2002. "Measuring similarity between ontologies." In *International Conference on Knowledge Engineering and Knowledge Management*, 251-63. Springer.
Majchrzak, Ann, Lynne P Cooper, and Olivia E Neece. 2004. 'Knowledge reuse for innovation', *Management science*, 50: 174-88.
Manasco, Britton. 1996. 'Leading firms develop knowledge strategies', *Knowledge Inc*, 1: 26-9.
Marir, Farhi, and Ian Watson. 1994. 'Case-based reasoning: a categorized bibliography', *The knowledge engineering review*, 9: 355-81.
Meng, Lingling, Runqing Huang, and Junzhong Gu. 2013. 'A review of semantic similarity measures in wordnet', *International Journal of Hybrid Information Technology*, 6: 1-12.
Microsoft. 'Microsoft Project management '. https://products.office.com/en-ca/project/project-management.
Mihalcea, Rada, Courtney Corley, and Carlo Strapparava. 2006. "Corpus-based and knowledge-based measures of text semantic similarity." In *AAAI*, 775-80.
Miller, George A. 1995. 'WordNet: a lexical database for English', *Communications of the ACM*, 38: 39-41.
Muñoz-Avila, Hector, and Jochem Huellen. 1995. "Retrieving cases in structured domains by using goal dependencies." In *International Conference on Case-Based Reasoning*, 241-52. Springer.
Ng, S Thomas, Nigel J Smith, and Martin Skitmore. 1998. 'A case based reasoning model for contractor prequalification', *International Journal of Construction Information Technology*, 6: 47-61.
OCCS. 2013. 'OmniClass -
A Strategy for Classifying the Built Environment', Accessed September 3. http://www.omniclass.org/.
Pan, Jiayi, CJ Anumba, and Z Ren. 2004. "Potential application of the semantic web in construction." In *Proceedings of the Twentieth Annual ARCOM Conference. Edinburgh, UK*.
Parlby, David. 1998. 'Knowledge Management Research Report 1998', *KPMG Management Consulting*.
Pauwels, Pieter, Sijie Zhang, and Yong-Cheol Lee. 2017. 'Semantic web technologies in AEC industry: A literature overview', *Automation in construction*, 73: 145-65.
PMBOK Guide. 2017. (project management institution).
'Python programming Language'. https://www.python.org/.
Resnik, Philip. 1995. 'Using information content to evaluate semantic similarity in a taxonomy', *arXiv preprint cmp-lg/9511007*.
Rezgui, Yacine, Christina J Hopfe, and Chalee Vorakulpipat. 2010. 'Generations of knowledge management in the architecture, engineering and construction industry: An evolutionary perspective', *Advanced Engineering Informatics*, 24: 219-28.
Richter, Michael M. 1993. 'Classification and learning of similarity measures.' in, *Information and Classification* (Springer).
Riesbeck, Christopher K, and Roger C Schank. 2013. *Inside case-based reasoning* (Psychology Press).
Rollett, Herwig. 2003. *Knowledge Management: Processes and Technologies* (Springer Science & Business Media).
Salem, AB. 2000. "Intelligent Learning Systems: A Case-Based Reasoning Approach." In.

Salton, Gerard, and Michael E Lesk. 1968. 'Computer evaluation of indexing and text processing', *Journal of the ACM (JACM)*, 15: 8-36.
Samimi, Parnia, and Sri Devi Ravana. 2014. 'Creation of reliable relevance judgments in information retrieval systems evaluation experimentation through crowdsourcing: a review', *The Scientific World Journal*, 2014.
Shukla, Saurabh, Vinayak Agarawal, Rajan Bhargava, and Shivam Handa. 2016. "Hierarchy similarity measure." In.: Google Patents.
Sowa, John F. 2012. 'Semantic networks', *John_Florian_Sowa isi [2012-04-20 16: 51]> Author [2012-04-20 16: 51]*.
Staab, Steffen, and Rudi Studer. 2004. 'Handbook on Ontologies'.
Szulanski, Gabriel. 2000. 'The process of knowledge transfer: A diachronic analysis of stickiness', *Organizational behavior and human decision processes*, 82: 9-27.
Tah, JHM, V Carr, and R Howes. 1999. 'Information modelling for case-based construction planning of highway bridge projects', *Advances in Engineering Software*, 30: 495-509.
Tan, Hai Chen, Patricia M Carrillo, Chimay J Anumba, Nasreddine Bouchlaghem, John M Kamara, and Chika E Udeaja. 2007. 'Development of a methodology for live capture and reuse of project knowledge in construction', *Journal of management in engineering*, 23: 18-26.
Tiwana, Amrit. 2000. *The knowledge management toolkit: practical techniques for building a knowledge management system* (Prentice Hall PTR).
Toolkit, Natural Language. 2018. 'NLTK 3.4 documentation'. https://www.nltk.org/.
Tserng, H Ping, and Yu-Cheng Lin. 2004. 'Developing an activity-based knowledge management system for contractors', *Automation in construction*, 13: 781-802.
Turk, Žiga. 2006. 'Construction informatics: Definition and ontology', *Advanced Engineering Informatics*, 20: 187-99.
Tversky, Amos. 1977. 'Features of similarity', *Psychological review*, 84: 327.
Veloso, Manuela M. 1992. "Learning by analogical reasoning in general problem solving." In.: CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE.
Venugopal, Manu, Charles M Eastman, and Jochen Teizer. 2015. 'An ontology-based analysis of the industry foundation class schema for building information model exchanges', *Advanced Engineering Informatics*, 29: 940-57.
W3c. 2004. 'Architecture of the World Wide Web'. http://www.w3.org/TR/webarch/.
———. 2014. 'Linked Building Dta Community Group'. https://www.w3.org/community/lbd/.
Watson, Ian. 1998. *Applying case-based reasoning: techniques for enterprise systems* (Morgan Kaufmann Publishers Inc.).
Watson, Ian, and Farhi Marir. 1994. 'Case-based reasoning: A review', *The knowledge engineering review*, 9: 327-54.
Wu, Zhibiao, and Martha Palmer. 1994. "Verbs semantics and lexical selection." In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, 133-38. Association for Computational Linguistics.
Yalcinkaya, Mehmet, and Vishal Singh. 2015. 'Patterns and trends in building information modeling (BIM) research: a latent semantic analysis', *Automation in construction*, 59: 68-80.
Yang, S-A, and Dave Robertson. 1994. "A case-based reasoning system for regulatory information." In *Case Based Reasoning: Prospects for Applications (Digest No. 1994/057), IEE Colloquium on*, 3/1-3/3. IET.

Yeh, I-Cheng. 1998. 'Quantity estimating of building with logarithm-neuron networks', *Journal of Construction Engineering and Management*, 124: 374-80.
Zeeshan, Aziz, Anumba Chimay, Ruikar Darshan, and Carrillo Patricia. 2004. 'Semantic web based services for intelligent mobile construction collaboration', *Journal of Information Technology in Construction (ITcon)*, 9: 367-79.
Zhang, Jiansong, and Nora M El-Gohary. 2013. 'Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking', *Journal of Computing in Civil Engineering*, 30: 04015014.
Zhang, Jinyue. 2010. 'A Social Semantic Web System for Coordinating Communication in the Architecture, Engineering & Construction Industry'.

Appendices

Appendix A: Developed WBS samples

Table 10. Bridge construction$_1$

Row	WBS codes	Tasks	Row	WBS codes	Tasks
1	1	Bridge project	33	1.3.4.6	backfilling
2	1.1	Site preparation	34	1.4	super structure
3	1.1.1	site mobilization	35	1.4.1	piers
4	1.1.2	transport equipment	36	1.4.1.1	reinforcement installation
5	1.1.3	surveying	37	1.4.1.2	form work
6	1.1.4	securing the site	38	1.4.1.3	concrete pouring
7	1.1.5	temporary buildings	39	1.4.1.4	form work removal
8	1.2	earthworks	40	1.4.2	column caps
9	1.2.1	vegetation removal	41	1.4.2.1	reinforcement installation
10	1.22	stripping ground	42	1.4.2.2	form work
11	1.2.3	excavation	43	1.4.2.3	concrete pouring
12	1.3	substructure	44	1.4.2.4	form work removal
13	1.3.1	pile driving	45	1.4.2.5	bearing pads
14	1.3.2	pile caps	46	1.4.3	girders
15	1.3.2.1	reinforcement installation	47	1.4.3.1	girder installation
16	1.3.2.2	form work	48	1.4.3.2	bracing
17	1.3.2.3	concrete pouring	49	1.4.4	decks
18	1.3.2.4	curing concrete	50	1.4.4.1	form work
19	1.3.2.5	form work removal	51	1.4.4.2	reinforcement installation
20	1.3.3	left abutment	52	1.4.4.3	concrete pouring
21	1.3.3.1	reinforcement installation	53	1.4.4.4	form work removal
22	1.3.3.2	form work	54	1.5	road works
23	1.3.3.3	concrete pouring	55	1.5.1	guardrails
24	1.3.3.4	curing concrete	56	1.5.2	asphalt
25	1.3.3.5	form work removal	57	1.5.3	surface marking

26	1.3.3.6	backfilling	58	1.5.4	electrical wiring
27	1.3.4	right abutment	59	1.5.5	lightings
28	1.3.4.1	reinforcement installation	60	1.6	finishing works
29	1.3.4.2	form work	61	1.6.1	landscaping
30	1.3.4.3	concrete pouring	62	1.6.2	testing systems
31	1.3.4.4	curing concrete	63	1.6.3	cleaning
32	1.3.4.5	form work removal			

Table 11. Bridge construction$_2$

Row	WBS codes	Tasks	Row	WBS codes	Tasks
1	1	bridge construction	23	1.2.2.2.3	concrete pouring
2	1.1	site preparation	24	1.2.2.2.4	form work removal
3	1.1.1	procurement	25	1.2.2.3	piers
4	1.1.2	surveying	26	1.2.2.3.1	form work installation
5	1.1.3	fencing	27	1.2.2.3.2	reinforcement placement
6	1.1.4	equipment mobilization	28	1.2.2.3.3	concrete pouring
7	1.2	construction	29	1.2.2.3.4	form work removal
8	1.2.1	foundation	30	1.2.3	superstructure
9	1.2.1.1	excavation	31	1.2.3.1	steel beams installation
10	1.2.1.2	piles	32	1.2.3.2	deck slab
11	1.2.1.2.1	cast piles	33	1.2.3.2.1	form work installation
12	1.2.1.2.2	drive piles	34	1.2.3.2.2	reinforcement placement
13	1.2.1.2.3	piles installation	35	1.2.3.2.3	concrete pouring
14	1.2.2	substructure	36	1.2.3.2.4	form work removal
15	1.2.2.1	piles cap	37	1.2.4	roads
16	1.2.2.1.1	form work installation	38	1.2.4.1	pavement
17	1.2.2.1.2	reinforcement placement	39	1.2.4.2	barriers
18	1.2.2.1.3	concrete pouring	40	1.2.4.3	line painting
19	1.2.2.1.4	form work removal	41	1.2.4.4	lighting
20	1.2.2.2	abutments	42	1.3	finishing
21	1.2.2.2.1	form work installation	43	1.3.1	landscaping
22	1.2.2.2.2	reinforcement placement	44	1.3.2	cleaning

Table 12. Bridge construction$_3$

Row	WBS codes	Tasks	Row	WBS codes	Tasks
1	1	bridge construction	29	1.2.2.3	pier caps
2	1.1	site preparation	30	1.2.2.3.1	form work
3	1.1.1	survey	31	1.2.2.3.2	reinforcing installation
4	1.1.2	permits	32	1.2.2.3.3	casting
5	1.1.3	restricting the construction site area	33	1.2.2.3.4	curing
6	1.1.4	setup equipment	34	1.2.2.3.5	form work removal
7	1.1.5	setup crane	35	1.2.2.3.6	bearing pad installation
8	1.2	construction	36	1.2.2.4	girders
9	1.2.1	earth work	37	1.2.2.4.1	girders installation
10	1.2.1.1	excavation level	38	1.2.2.4.2	sealing of girders
11	1.2.2	structural	39	1.2.2.5	decks
12	1.2.2.1	foundation	40	1.2.2.5.1	form work
13	1.2.2.1.1	deep foundation	41	1.2.2.5.2	reinforcing installation
14	1.2.2.1.1.1	drilling	42	1.2.2.5.3	casting
15	1.2.2.1.1.2	installation of reinforcing cage	43	1.2.2.5.4	curing
16	1.2.2.1.1.3	casting	44	1.3.2.5.5	form work removal
17	1.2.2.1.1.4	curing	45	1.2.3	road works
18	1.2.2.1.2	pile caps	46	1.2.3.1	side walks
19	1.2.2.1.2.1	form work	47	1.2.3.2	pavement
20	1.2.2.1.2.2	reinforcing installation	48	1.2.3.3	painting the lines
21	1.2.2.1.2.3	casting	49	1.2.3.4	installing guards
22	1.2.2.1.2.4	curing	50	1.2.3.5	lighting system installation
23	1.2.2.1.2.5	form work removal	51	1.2.3.6	drainage system
24	1.2.2.2	columns	52	1.3	finishing
25	1.2.2.2.1	form work	53	1.3.1	tests
26	1.2.2.2.2	reinforcing installation	54	1.3.2	cleaning

27	1.2.2.2.3	casting	55	1.3.3	landscaping
28	1.2.2.2.4	curing			

Table 13. Steel structure building$_1$

Row	WBS codes	Tasks	Row	WBS codes	Tasks
1	1	steel building	28	1.6.5	doors installation
2	1.1	site preparation	29	1.7	second floor Architectural
3	1.1.1	site mobilization	30	1.7.1	external walls
4	1.1.2	surveying	31	1.7.2	separation walls
5	1.1.3	fencing	32	1.7.3	partition walls
6	1.2	earthworks	33	1.7.4	windows installation
7	1.2.1	stripping ground	34	1.7.5	doors installation
8	1.2.2	excavation	35	1.7.6	skylight structure
9	1.3	foundation	36	1.8	mechanical systems
10	1.3.1	reinforcing installation	37	1.8.1	mechanical room water heater
11	1.3.2	form work	38	1.8.2	mechanical room heating ventilation and air conditioning unit
12	1.3.3	concrete pouring	39	1.8.3	heating ventilation and air conditioning ducts ground floor
13	1.3.4	curing concrete	40	1.8.4	heating ventilation and air conditioning ducts second floor
14	1.3.5	form work removal	41	1.8.5	plumbing ground floor
15	1.4	steel structure	42	1.8.6	plumbing second floor
16	1.4.1	installation of columns	43	1.8.7	elevator room
17	1.4.2	installation of beams first floor	44	1.8.8	elevator installation
18	1.4.3	installation of beams roof	45	1.9	electrical systems
19	1.5	floor slabs	46	1.9.1	distribution boards

20	1.5.1	ground floor concrete	47	1.9.2	electrical wiring ground floor
21	1.5.2	first floor concrete	48	1.9.3	electrical wiring second floor
22	1.5.3	roof floor concrete	49	1.9.4	smoke detectors
23	1.6	ground floor Architectural	50	1.9.5	outlets and switches
24	1.6.1	external walls	51	1.10	finishing works
25	1.6.2	separation walls	52	1.10.1	landscaping
26	1.6.3	partition walls	53	1.10.2	testing systems
27	1.6.4	windows installation	54	1.10.3	cleaning

Table 14. Steel structure building$_2$

Row	WBS codes	Tasks	Row	WBS codes	Tasks
1	1	steel building	35	1.4.1.2	heating ventilation and air conditioning installation
2	1.1	Site preparation	36	1.4.2	second floor
3	1.1.1	surveying	37	1.4.2.1	plumbing
4	1.1.2	equipment mobilization	38	1.4.2.2	heating ventilation and air conditioning installation
5	1.1.3	fences installation	39	1.4.3	elevator
6	1.2	earthworks	40	1.5	electrical works
7	1.2.1	stripping	41	1.5.1	ground floor
8	1.2.2	excavation	42	1.5.1.1	install wiring
9	1.3	structure	43	1.5.1.2	install outlets switches
10	1.3.1	foundation	44	1.5.1.3	install light fixtures
11	1.3.1.1	form work installation	45	1.5.1.4	fire alarm systems
12	1.3.1.2	reinforcing placement	46	1.5.2	second floor
13	1.3.1.3	concrete pouring	47	1.5.2.1	install wiring
14	1.3.1.4	curing	48	1.5.2.2	install outlets switches
15	1.3.1.5	form work removal	49	1.5.2.3	install light fixtures
16	1.3.2	steel structure	50	1.5.2.4	fire alarm systems
17	1.3.2.1	ground floor steel installation	51	1.6	finishing works

18	1.3.2.1.1	steel columns installation	52	1.6.1	ground floor finishing
19	1.3.2.1.2	steel beams installation	53	1.6.1.1	exterior walls installation
20	1.3.2.1.3	joints	54	1.6.1.2	dry walls installation
21	1.3.2.2	second floor steel installation	55	1.6.1.3	windows installation
22	1.3.2.2.1	steel columns installation	56	1.6.1.4	doors installation
23	1.3.2.2.2	steel beams installation	57	1.6.1.5	flooring
24	1.3.2.2.3	joints	58	1.6.1.6	painting
25	1.3.3	slabs	59	1.6.2	second floor finishing
26	1.3.3.1	ground floor slab	60	1.6.2.1	exterior walls installation
27	1.3.3.1.1	concrete pouring	61	1.6.2.2	dry walls installation
28	1.3.3.1.2	concrete curing	62	1.6.2.3	windows installation
29	1.3.3.2	second floor slab	63	1.6.2.4	doors installation
30	1.3.3.2.1	concrete pouring	64	1.6.2.5	flooring
31	1.3.3.2.2	concrete curing	65	1.6.2.6	painting
32	1.4	mechanical works	66	1.6.3	cleaning
33	1.4.1	ground floor	67	1.6.4	testing systems
34	1.4.1.1	plumbing			

Table 15. Steel structure building$_3$

Row	WBS codes	Tasks	Row	WBS codes	Tasks
1	1	steel building	28	1.4.2.7	curing floor slab
2	1.1	site preparation	29	1.5	non structural
3	1.1.1	survey	30	1.5.1	ground floor
4	1.1.2	fencing	31	1.5.1.1	surrounding walls
5	1.2	earthwork	32	1.5.1.2	interior walls
6	1.2.1	excavation	33	1.5.1.3	piping
7	1.3	foundations	34	1.5.1.4	heating ventilation and air conditioning installation
8	1.3.1	reinforcing installation	35	1.5.1.5	hardwood flooring
9	1.3.2	form work	36	1.5.1.6	electrical works and wiring
10	1.3.3	pouring concrete	37	1.5.1.7	installing doors
11	1.3.4	curing	38	1.5.1.8	installation windows

12	1.4	steel Structural	39	1.5.1.9	plastering and painting
13	1.4.1	ground floor	40	1.5.2	second floor
14	1.4.1.1	installing base plates	41	1.5.2.1	surrounding walls
15	1.4.1.2	installing steel columns	42	1.5.2.2	interior walls
16	1.4.1.3	beam installation	43	1.5.2.3	piping
17	1.4.1.4	bracing installation	44	1.5.2.4	heating ventilation and air conditioning installation
18	1.4.1.5	metal deck installation	45	1.5.2.5	hardwood flooring
19	1.4.1.6	concrete pouring floor slab	46	1.5.2.6	electrical works and wiring
20	1.4.1.7	curing floor slab	47	1.5.2.7	installing doors
21	1.4.2	second floor	48	1.5.2.8	installation windows
22	1.4.2.1	installing base plates	49	1.5.2.9	plastering and painting
23	1.4.2.2	installing steel columns	50	1.6	finishing
24	1.4.2.3	beam installation	51	1.6.1	testing
25	1.4.2.4	bracing installation	52	1.6.2	landscape
26	1.4.2.5	metal deck installation	53	1.6.3	cleaning
27	1.4.2.6	concrete pouring floor slab			

Table 16. concrete structure building$_1$

Row	WBS codes	Tasks	Row	WBS codes	Tasks
1	1	concrete building	52	1.7.7	concrete pouring slab
2	1.1	site preparation	53	1.7.8	form work removal slab
3	1.1.1	site mobilization	54	1.8	ground floor architectural
4	1.1.2	surveying	55	1.8.1	external walls
5	1.1.3	fencing	56	1.8.2	separation walls
6	1.2	earthworks	57	1.8.3	partition walls
7	1.2.1	stripping ground	58	1.8.4	windows installation
8	1.2.2	excavation	59	1.8.5	doors installation
9	1.3	foundation	60	1.9	second floor architectural
10	1.3.1	form work	61	1.9.1	external walls
11	1.3.2	reinforcing installation	62	1.9.2	separation walls
12	1.3.3	concrete pouring	63	1.9.3	partition walls

13	1.3.4	curing concrete	64	1.9.4	windows installation
14	1.3.5	form work removal	65	1.9.5	doors installation
15	1.3.6	base slab form work	66	1.10	third floor architectural
16	1.3.7	base slab reinforcing installation	67	1.10.1	external walls
17	1.3.8	base slab concrete	68	1.10.2	separation walls
18	1.4	ground level structure	69	1.10.3	partition walls
19	1.4.1	reinforcing installation columns	70	1.10.4	windows installation
20	1.4.2	form work columns	71	1.10.5	doors installation
21	1.4.3	concrete pouring columns	72	1.11	fourth floor architectural
22	1.4.4	form work removal columns	73	1.11.1	external walls
23	1.4.5	shoring and form work slab	74	1.11.2	separation walls
24	1.4.6	reinforcing installation slab	75	1.11.3	partition walls
25	1.4.7	concrete pouring slab	76	1.11.4	windows installation
26	1.4.8	form work removal slab	77	1.11.5	doors installation
27	1.5	second floor structure	78	1.12	mechanical systems
28	1.5.1	reinforcing installation columns	79	1.12.1	mechanical room water heater
29	1.5.2	form work columns	80	1.12.2	mechanical room heating ventilation and air conditioning unit
30	1.5.3	concrete pouring columns	81	1.12.3	heating ventilation and air conditioning ducts ground floor
31	1.5.4	form work removal columns	82	1.12.4	heating ventilation and air conditioning ducts second floor
32	1.5.5	shoring and form work slab	83	1.12.5	heating ventilation and air conditioning ducts third floor
33	1.5.6	reinforcing installation slab	84	1.12.6	heating ventilation and air conditioning ducts fourth floor
34	1.5.7	concrete pouring slab	85	1.12.7	plumbing ground floor

35	1.5.8	form work removal slab	86	1.12.8	plumbing second floor
36	1.6	third floor structure	87	1.12.9	plumbing third floor
37	1.6.1	reinforcing installation columns	88	1.12.10	plumbing fourth floor
38	1.6.2	form work columns	89	1.12.11	elevator room
39	1.6.3	concrete pouring columns	90	1.12.12	elevator installation
40	1.6.4	form work removal columns	91	1.13	electrical systems
41	1.6.5	shoring and form work slab	92	1.13.1	distribution boards
42	1.6.6	reinforcing installation slab	93	1.13.2	electrical wiring ground floor
43	1.6.7	concrete pouring slab	94	1.13.3	electrical wiring second floor
44	1.6.8	form work removal slab	95	1.13.4	electrical wiring third floor
45	1.7	fourth floor structure	96	1.13.5	electrical wiring fourth floor
46	1.7.1	reinforcing installation columns	97	1.13.6	smoke detectors
47	1.7.2	form work columns	98	1.13.7	outlets and switches
48	1.7.3	concrete pouring columns	99	1.14	finishing works
49	1.7.4	form work removal columns	100	1.14.1	landscaping
50	1.7.5	shoring and form work slab	101	1.14.2	testing systems
51	1.7.6	reinforcing installation slab	102	1.14.3	cleaning

Table 17. Concrete structure building$_2$

Row	WBS codes	Tasks	Row	WBS codes	Tasks
1	1	concrete building	68	1.4	mechanical works
2	1.1	Site preparation	69	1.4.1	ground floor
3	1.1.1	surveying	70	1.4.1.1	plumbing
4	1.1.2	equipment mobilization	71	1.4.1.2	heating ventilation and air conditioning installation
5	1.1.3	fences installation	72	1.4.2	second floor
6	1.2	earth works	73	1.4.2.1	plumbing
7	1.2.1	stripping	74	1.4.2.2	heating ventilation and air conditioning installation
8	1.2.2	excavation	75	1.4.3	third floor

9	1.3	structure	76	1.4.3.1	plumbing
10	1.3.1	foundation	77	1.4.3.2	heating ventilation and air conditioning installation
11	1.3.1.1	form work installation	78	1.4.4	fourth floor
12	1.3.1.2	reinforcing placement	79	1.4.4.1	plumbing
13	1.3.1.3	concrete pouring	80	1.4.4.2	heating ventilation and air conditioning installation
14	1.3.1.4	curing	81	1.4.5	elevator
15	1.3.1.5	form work removal	82	1.5	electrical works
16	1.3.2	ground floor structure	83	1.5.1	ground floor
17	1.3.2.1	columns	84	1.5.1.1	install wiring
18	1.3.2.1.1	form work installation	85	1.5.1.2	install outlets switches
19	1.3.2.1.2	reinforcing placement	86	1.5.1.3	install light fixtures
20	1.3.2.1.3	concrete pouring	87	1.5.1.4	fire alarm systems
21	1.3.2.1.4	curing	88	1.5.2	second floor
22	1.3.2.1.5	form work removal	89	1.5.2.1	install wiring
23	1.3.2.2	beams and slab	90	1.5.2.2	install outlets switches
24	1.3.2.2.1	form work installation	91	1.5.2.3	install light fixtures
25	1.3.2.2.2	reinforcing placement	92	1.5.2.4	fire alarm systems
26	1.3.2.2.3	concrete pouring	93	1.5.3	third floor
27	1.3.2.2.4	curing	94	1.5.3.1	install wiring
28	1.3.2.2.5	form work removal	95	1.5.3.2	install outlets switches
29	1.3.3	second floor structure	96	1.5.3.3	install light fixtures
30	1.3.3.1	columns	97	1.5.3.4	fire alarm systems
31	1.3.3.1.1	form work installation	98	1.5.4	fourth floor
32	1.3.3.1.2	reinforcing placement	99	1.5.4.1	install wiring
33	1.3.3.1.3	concrete pouring	100	1.5.4.2	install outlets switches
34	1.3.3.1.4	curing	101	1.5.4.3	install light fixtures
35	1.3.3.1.5	form work removal	102	1.5.4.4	fire alarm systems
36	1.3.3.2	beams and slab	103	1.6	finishing works
37	1.3.3.2.1	form work installation	104	1.6.1	ground floor finishing
38	1.3.3.2.2	reinforcing placement	105	1.6.1.1	exterior walls installation
39	1.3.3.2.3	concrete pouring	106	1.6.1.2	dry walls installation

40	1.3.3.2.4	curing	107	1.6.1.3	windows installation
41	1.3.3.2.5	form work removal	108	1.6.1.4	doors installation
42	1.3.4	third floor structure	109	1.6.1.5	flooring
43	1.3.4.1	columns	110	1.6.1.6	painting
44	1.3.4.1.1	form work installation	111	1.6.2	second floor finishing
45	1.3.4.1.2	reinforcing placement	112	1.6.2.1	exterior walls installation
46	1.3.4.1.3	concrete pouring	113	1.6.2.2	dry walls installation
47	1.3.4.1.4	curing	114	1.6.2.3	windows installation
48	1.3.4.1.5	form work removal	115	1.6.2.4	doors installation
49	1.3.4.2	beams and slab	116	1.6.2.5	flooring
50	1.3.4.2.1	form work installation	117	1.6.2.6	painting
51	1.3.4.2.2	reinforcing placement	118	1.6.3	third floor finishing
52	1.3.4.2.3	concrete pouring	119	1.6.3.1	exterior walls installation
53	1.3.4.2.4	curing	120	1.6.3.2	dry walls installation
54	1.3.4.2.5	form work removal	121	1.6.3.3	windows installation
55	1.3.5	fourth floor structure	122	1.6.3.4	doors installation
56	1.3.5.1	columns	123	1.6.3.5	flooring
57	1.3.5.1.1	form work installation	124	1.6.3.6	painting
58	1.3.5.1.2	reinforcing placement	125	1.6.4	fourth floor finishing
59	1.3.5.1.3	concrete pouring	126	1.6.4.1	exterior walls installation
60	1.3.5.1.4	curing	127	1.6.4.2	dry walls installation
61	1.3.5.1.5	form work removal	128	1.6.4.3	windows installation
62	1.3.5.2	beams and slab	129	1.6.4.4	doors installation
63	1.3.5.2.1	form work installation	130	1.6.4.5	flooring
64	1.3.5.2.2	reinforcing placement	131	1.6.4.6	painting
65	1.3.5.2.3	concrete pouring	132	1.6.5	cleaning
66	1.3.5.2.4	curing	133	1.6.6	testing systems
67	1.3.5.2.5	form work removal			

Table 18. Concrete structure building$_3$

Row	WBS codes	Tasks	Row	WBS codes	Tasks
1	1	concrete building	62	1.3.5.1	ground floor
2	1.1	site preparation	63	1.3.5.1.1	installing joist
3	1.1.1	equipment mobilization	64	1.3.5.1.2	forming
4	1.1.2	surveying	65	1.3.5.1.3	concrete pouring
5	1.1.3	fence installation	66	1.3.5.1.4	curing
6	1.2	earthwork	67	1.3.5.2	second floor
7	1.2.1	excavation	68	1.3.5.2.1	installing joist
8	1.3	Structural	69	1.3.5.2.2	forming
9	1.3.1	foundations	70	1.3.5.2.3	concrete pouring
10	1.3.1.1	forming	71	1.3.5.2.4	curing
11	1.3.1.2	reinforcing assembly	72	1.3.5.3	third floor
12	1.3.1.3	casting concrete	73	1.3.5.3.1	installing joist
13	1.3.1.4	curing	74	1.3.5.3.2	forming
14	1.3.2	columns	75	1.3.5.3.3	concrete pouring
15	1.3.2.1	ground floor columns	76	1.5.3.3.4	curing
16	1.3.2.1.1	forming	77	1.3.5.4	fourth floor
17	1.3.2.1.2	reinforcing assembly	78	1.3.5.4.1	installing joist
18	1.3.2.1.3	casting concrete	79	1.3.5.4.2	forming
19	1.3.2.1.4	curing	80	1.3.5.4.3	concrete pouring
20	1.3.2.2	second floor columns	81	1.3.5.4.4	curing
21	1.3.2.2.1	forming	82	1.4	non structural
22	1.3.2.2.2	reinforcing assembly	83	1.4.1	ground floor
23	1.3.2.2.3	casting concrete	84	1.4.1.1	outer walls
24	1.3.2.2.4	curing	85	1.4.1.2	inner walls
25	1.3.2.3	third floor columns	86	1.4.1.3	piping
26	1.3.2.3.1	forming	87	1.4.1.4	heating ventilation and air conditioning installation
27	1.3.2.3.2	reinforcing assembly	88	1.4.1.5	floor tiling
28	1.3.2.3.3	casting concrete	89	1.4.1.6	electrical works and wiring
29	1.3.2.3.4	curing	90	1.4.1.7	plastering and painting

30	1.3.2.4	fourth floor columns	91	1.4.1.8	installing doors and windows
31	1.3.2.4.1	forming	92	1.4.2	second floor
32	1.3.2.4.2	reinforcing assembly	93	1.4.2.1	outer walls
33	1.3.2.4.3	casting concrete	94	1.4.2.2	inner walls
34	1.3.2.4.4	curing	95	1.4.2.3	piping
35	1.3.3	beams	96	1.4.2.4	heating ventilation and air conditioning installation
36	1.3.3.1	ground floor beams	97	1.4.2.5	floor tiling
37	1.3.3.1.1	forming	98	1.4.2.6	electrical works and wiring
38	1.3.3.1.2	reinforcing assembly	99	1.4.2.7	plastering and painting
39	1.3.3.1.3	casting concrete	100	1.4.2.8	installing doors and windows
40	1.3.3.1.4	curing	101	1.4.3	third floor
41	1.3.3.2	second floor beams	102	1.4.3.1	outer walls
42	1.3.3.2.1	forming	103	1.4.3.2	inner walls
43	1.3.3.2.2	reinforcing assembly	104	1.4.3.3	piping
44	1.3.3.2.3	casting concrete	105	1.4.3.4	heating ventilation and air conditioning installation
45	1.3.3.2.4	curing	106	1.4.3.5	floor tiling
46	1.3.3.3	third floor beams	107	1.4.3.6	electrical works and wiring
47	1.3.3.3.1	forming	108	1.4.3.7	plastering and painting
48	1.3.3.3.2	reinforcing assembly	109	1.4.3.8	installing doors and windows
49	1.3.3.3.3	casting concrete	110	1.4.4	fourth floor
50	1.3.3.3.4	curing	111	1.4.4.1	outer walls
51	1.3.3.4	fourth floor beams	112	1.4.4.2	inner walls
52	1.3.3.4.1	forming	113	1.4.4.3	piping
53	1.3.3.4.2	reinforcing assembly	114	1.4.4.4	heating ventilation and air conditioning installation
54	1.3.3.4.3	casting concrete	115	1.4.4.5	floor tiling
55	1.3.3.4.4	curing	116	1.4.4.6	electrical works and wiring
56	1.3.4	shear walls and stairs	117	1.4.4.7	plastering and painting
57	1.3.4.1	forming	118	1.4.4.8	installing doors and windows
58	1.3.4.2	reinforcing assembly	119	1.4.5	finishing
59	1.3.4.3	casting concrete	120	1.4.5.1	landscape

60	1.3.4.4	curing	121	1.4.5.2	system tests
61	1.3.5	slabs			

Appendix B: Detailed results of mapped nodes

Table 19. Mapped nodes in comparison of the pair (S_1 , S_2) (second scenario)

Row	$steel\ building_1$		$steel\ building_2$		$sim_{semantic}$	$sim_{parents}$	$sim_{siblings}$	$sim_{average}$
	WBS code	Task	WBS code	Task				
1	1.1	site preparation	1.1	Site preparation	1.000	1.000	0.361	0.787
2	1.1.1	site mobilization	1.1.2	equipment mobilization	0.853	1.000	0.971	0.941
3	1.1.2	surveying	1.1.1	surveying	1.000	1.000	0.898	0.966
4	1.1.3	fencing	1.1.3	fences installation	0.942	1.000	0.926	0.956
5	1.2	earthworks	1.2	earthworks	1.000	1.000	0.361	0.787
6	1.2.1	stripping ground	1.2.1	stripping	0.850	1.000	1.000	0.950
7	1.2.2	excavation	1.2.2	excavation	1.000	1.000	0.850	0.950
8	1.3.1	reinforcing installation	1.3.1.2	reinforcing placement	0.816	0.786	0.946	0.850
9	1.3.2	form work	1.3.1.1	form work installation	0.952	0.786	0.912	0.884
10	1.3.3	concrete pouring	1.3.1.3	concrete pouring	1.000	0.786	0.900	0.896
11	1.3.4	curing concrete	1.3.1.4	curing	0.833	0.786	0.942	0.854
12	1.3.5	form work removal	1.6.2	second floor finishing	0.801	0.899	0.471	0.724
13	1.4	steel structure	1.3	structure	0.917	1.000	0.393	0.770
14	1.4.1	installation of columns	1.3.2.1.3	joints	0.624	0.781	0.785	0.730
15	1.4.2	installation of beams first floor	1.3.2.2	second floor steel installation	0.835	0.786	0.503	0.708
16	1.4.3	installation of beams roof	1.3.2.2.3	joints	0.590	0.778	0.803	0.724

17	1.5.1	ground floor concrete	1.3.1	foundation	0.822	0.954	0.755	0.844
18	1.5.2	first floor concrete	1.3.2	steel structure	0.804	0.954	0.764	0.841
19	1.5.3	roof floor concrete	1.3.3	slabs	0.706	0.954	0.813	0.824
20	1.6.1	external walls	1.6.2.6	painting	0.594	0.688	0.823	0.702
21	1.6.2	separation walls	1.6.2.2	dry walls installation	0.808	0.688	0.758	0.751
22	1.6.3	partition walls	1.6.2.1	exterior walls installation	0.897	0.688	0.738	0.774
23	1.6.4	windows installation	1.6.2.3	windows installation	1.000	0.688	0.715	0.801
24	1.6.5	doors installation	1.6.2.4	doors installation	1.000	0.688	0.683	0.790
25	1.7.2	separation walls	1.6.1.6	painting	0.656	0.680	0.843	0.726
26	1.7.3	partition walls	1.6.1.1	exterior walls installation	0.897	0.680	0.664	0.747
27	1.7.4	windows installation	1.6.1.3	windows installation	1.000	0.680	0.644	0.775
28	1.7.5	doors installation	1.6.1.4	doors installation	1.000	0.680	0.615	0.765
29	1.10	finishing works	1.6	finishing works	1.000	1.000	0.400	0.800
30	1.10.1	landscaping	1.6.1	ground floor finishing	0.452	1.000	0.800	0.751
31	1.10.2	testing systems	1.6.4	testing systems	1.000	1.000	0.400	0.800
32	1.10.3	cleaning	1.6.3	cleaning	1.000	1.000	0.400	0.800

Similarity scores and measurements

mapped nodes	Unmapped nodes	Mapped not relevant	Node similarity	Structural similarity	Total similarity
64	57	2	0.430	0.785	0.607
Mapping precision			0.97		

Table 20. Mapped nodes in comparison of the pair (S_1 , S_2), (Third scenario)

Row	*steel building*$_1$		*steel building*$_2$		$sim_{semantic}$	$sim_{parents}$	$sim_{siblings}$	$sim_{average}$
	WBS code	Task	WBS code	Task				
1	1	steel building	1	steel building	1.000	1.000	0.000	0.750
2	1.1	site preparation	1.1	Site preparation	1.000	1.000	0.533	0.883
3	1.1.1	site mobilization	1.1.2	equipment mobilization	0.853	1.000	0.971	0.919
4	1.1.2	surveying	1.1.1	surveying	1.000	1.000	0.898	0.974
5	1.1.3	fencing	1.1.3	fences installation	0.942	1.000	0.926	0.953
6	1.2	earthworks	1.2	earthworks	1.000	1.000	0.533	0.883
7	1.2.1	stripping ground	1.2.1	stripping	0.850	1.000	1.000	0.925
8	1.2.2	excavation	1.2.2	excavation	1.000	1.000	0.850	0.963
9	1.3	foundation	1.5	electrical works	0.529	1.000	0.601	0.665
10	1.3.1	reinforcing installation	1.5.1	ground floor	0.801	0.812	0.317	0.683
11	1.3.2	form work	1.6.2	second floor finishing	0.788	0.899	0.568	0.761
12	1.3.3	concrete pouring	1.3.1.3	concrete pouring	1.000	0.786	0.900	0.922
13	1.3.4	curing concrete	1.3.1.4	curing	0.833	0.786	0.942	0.849
14	1.3.5	form work removal	1.6.1	ground floor finishing	0.840	0.899	0.600	0.795
15	1.4	steel structure	1.3	structure	0.917	1.000	0.565	0.850
16	1.4.1	installation of columns	1.3.1	foundation	0.742	0.967	0.685	0.784
17	1.4.2	installation of beams first floor	1.5.2	second floor	0.837	0.845	0.474	0.748
18	1.4.3	installation of beams roof	1.3.3	slabs	0.647	0.967	0.732	0.748

19	1.5	floor slabs	1.4	mechanical works	0.567	1.000	0.595	0.682
20	1.5.1	ground floor concrete	1.4.1	ground floor	0.952	0.827	0.765	0.874
21	1.5.2	first floor concrete	1.4.2	second floor	0.911	0.827	0.786	0.858
22	1.5.3	roof floor concrete	1.3.2	steel structure	0.788	0.954	0.758	0.822
23	1.6	ground floor Architectural	1.3.3.1	ground floor slab	0.786	0.461	0.139	0.543
24	1.6.1	external walls	1.5.1.3	install light fixtures	0.621	0.611	0.567	0.605
25	1.6.2	separation walls	1.6.1.2	dry walls installation	0.808	0.617	0.758	0.748
26	1.6.3	partition walls	1.6.1.1	exterior walls installation	0.897	0.617	0.738	0.788
27	1.6.4	windows installation	1.6.1.3	windows installation	1.000	0.617	0.715	0.833
28	1.6.5	doors installation	1.6.1.4	doors installation	1.000	0.617	0.683	0.825
29	1.7	second floor Architectural	1.5.2.2	install outlets switches	0.583	0.519	0.320	0.501
30	1.7.1	external walls	1.6.1.6	painting	0.594	0.577	0.886	0.662
31	1.7.2	separation walls	1.6.2.2	dry walls installation	0.808	0.614	0.809	0.759
32	1.7.3	partition walls	1.6.2.1	exterior walls installation	0.897	0.614	0.791	0.800
33	1.7.4	windows installation	1.6.2.3	windows installation	1.000	0.614	0.770	0.846
34	1.7.5	doors installation	1.6.2.4	doors installation	1.000	0.614	0.741	0.839
35	1.7.6	skylight structure	1.6.2.6	painting	0.632	0.614	0.844	0.680
36	1.8	mechanical systems	1.3.2.1.3	joints	0.625	0.588	0.258	0.524

37	1.8.1	mechanical room water heater	1.3.3.2	second floor slab	0.710	0.554	0.182	0.539
38	1.8.2	mechanical room heating ventilation and air conditioning unit	1.3.1.5	form work removal	0.702	0.474	0.363	0.560
39	1.8.3	heating ventilation and air conditioning ducts ground floor	1.4.1.2	heating ventilation and air conditioning installation	0.942	0.484	0.164	0.633
40	1.8.4	heating ventilation and air conditioning ducts second floor	1.4.2.2	heating ventilation and air conditioning installation	0.935	0.462	0.164	0.624
41	1.8.5	plumbing ground floor	1.6.1.5	flooring	0.911	0.510	0.627	0.740
42	1.8.6	plumbing second floor	1.6.2.5	flooring	0.889	0.492	0.633	0.725
43	1.8.7	elevator room	1.3.2.1	ground floor steel installation	0.807	0.607	0.189	0.603
44	1.8.8	elevator installation	1.4.3	elevator	0.917	0.375	0.330	0.635
45	1.9	electrical systems	1.3.2.2.3	joints	0.625	0.587	0.258	0.524
46	1.9.1	distribution boards	1.5.1.4	fire alarm systems	0.839	0.484	0.571	0.683
47	1.9.2	electrical wiring ground floor	1.3.1.1	form work installation	0.728	0.474	0.623	0.638
48	1.9.3	electrical wiring second floor	1.3.2.2	second floor steel installation	0.771	0.607	0.306	0.614
49	1.9.4	smoke detectors	1.5.2.4	fire alarm systems	0.741	0.462	0.579	0.630

50	1.9.5	outlets and switches	1.5.1.2	install outlets switches	0.774	0.484	0.623	0.664
51	1.10	finishing works	1.6	finishing works	1.000	1.000	0.568	0.892
52	1.10.1	landscaping	1.5.2.3	install light fixtures	0.493	0.571	0.607	0.541
53	1.10.2	testing systems	1.6.4	testing systems	1.000	1.000	0.400	0.850
54	1.10.3	cleaning	1.6.3	cleaning	1.000	1.000	0.400	0.850

Similarity scores and measurements

mapped nodes	Unmapped nodes	Mapped not relevant	Node similarity	Structural similarity	Total similarity
108	13	36	0664	0.818	0.741
Mapping precision			0.67		

Table 21. Mapped nodes in comparison of the pair (S_1 , S_2), (Fourth scenario)

Row	$steel\ building_1$		$steel\ building_2$		$sim_{semantic}$	$sim_{parents}$	$sim_{siblings}$	$sim_{average}$
	WBS code	Task	WBS code	Task				
1	1	steel building	1	steel building	1.000	1.000	0.000	0.750
2	1.1	site preparation	1.1	Site preparation	1.000	1.000	0.361	0.840
3	1.1.1	site mobilization	1.1.2	equipment mobilization	0.853	1.000	0.971	0.919
4	1.1.2	surveying	1.1.1	surveying	1.000	1.000	0.898	0.974
5	1.1.3	fencing	1.1.3	fences installation	0.942	1.000	0.926	0.953
6	1.2	earthworks	1.2	earthworks	1.000	1.000	0.361	0.840
7	1.2.1	stripping ground	1.2.1	stripping	0.850	1.000	1.000	0.925
8	1.2.2	excavation	1.2.2	excavation	1.000	1.000	0.850	0.963
9	1.3.1	reinforcing installation	1.6.1	ground floor finishing	0.795	0.899	0.473	0.740
10	1.3.2	form work	1.3.1.1	form work installation	0.952	0.786	0.912	0.901

11	1.3.3	concrete pouring	1.3.1.3	concrete pouring	1.000	0.786	0.900	0.922
12	1.3.4	curing concrete	1.3.1.4	curing	0.833	0.786	0.942	0.849
13	1.3.5	form work removal	1.6.2	second floor finishing	0.801	0.899	0.471	0.743
14	1.4	steel structure	1.3	structure	0.917	1.000	0.393	0.806
15	1.4.1	installation of columns	1.3.2.1.3	joints	0.624	0.781	0.785	0.704
16	1.4.2	installation of beams first floor	1.3.2.2	second floor steel installation	0.835	0.786	0.503	0.740
17	1.4.3	installation of beams roof	1.3.2.1	ground floor steel installation	0.793	0.786	0.498	0.718
18	1.5.1	ground floor concrete	1.3.1	foundation	0.822	0.954	0.755	0.838
19	1.5.2	first floor concrete	1.3.2	steel structure	0.804	0.954	0.764	0.832
20	1.5.3	roof floor concrete	1.3.3	slabs	0.706	0.954	0.813	0.795
21	1.6.2	separation walls	1.6.2.2	dry walls installation	0.808	0.688	0.758	0.765
22	1.6.3	partition walls	1.6.2.1	exterior walls installation	0.897	0.688	0.738	0.805
23	1.6.4	windows installation	1.6.2.3	windows installation	1.000	0.688	0.715	0.851
24	1.6.5	doors installation	1.6.2.4	doors installation	1.000	0.688	0.683	0.843
25	1.7.2	separation walls	1.6.1.2	dry walls installation	0.808	0.680	0.682	0.744
26	1.7.3	partition walls	1.6.1.1	exterior walls installation	0.897	0.680	0.664	0.785
27	1.7.4	windows installation	1.6.1.3	windows installation	1.000	0.680	0.644	0.831
28	1.7.5	doors installation	1.6.1.4	doors installation	1.000	0.680	0.615	0.824

29	1.8.5	plumbing ground floor	1.6.1.5	flooring	0.911	0.548	0.634	0.751
30	1.8.6	plumbing second floor	1.6.2.5	flooring	0.889	0.511	0.639	0.732
31	1.9.1	distribution boards	1.5.1.4	fire alarm systems	0.839	0.579	0.624	0.720
32	1.10	finishing works	1.6	finishing works	1.000	1.000	0.400	0.850
33	1.10.2	testing systems	1.6.4	testing systems	1.000	1.000	0.400	0.850
34	1.10.3	cleaning	1.6.3	cleaning	1.000	1.000	0.400	0.850

Similarity scores and measurements

mapped nodes	Unmapped nodes	Mapped not relevant	Node similarity	Structural similarity	Total similarity
68	53	6	0.46	0.795	0.628
Mapping precision			0.91		

Table 22. Mapped nodes in comparison of the pair (B_1 , C_2), (First scenario)

Row	$steel\ building_1$		$steel\ building_2$		$sim_{semantic}$	$sim_{parents}$	$sim_{siblings}$	$sim_{average}$
	WBS code	Task	WBS code	Task				
1	1	bridge project	1	concrete building	0.784	0.784	0.000	0.522
2	1.1	site preparation	1.1	Site preparation	1.000	0.784	0.675	0.820
3	1.1.1	site mobilization	1.1.2	equipment mobilization	0.853	0.870	0.578	0.767
4	1.1.2	transport equipment	1.1.3	fences installation	0.733	0.870	0.618	0.740
5	1.1.3	surveying	1.1.1	surveying	1.000	0.870	0.529	0.800
6	1.1.4	securing the site	1.5.4	fourth floor	0.370	0.715	0.623	0.569
7	1.1.5	temporary buildings	1.6.2	second floor finishing	0.702	0.795	0.575	0.691

8	1.2	earthworks	1.4	mechanical works	0.500	0.784	0.815	0.700
9	1.2.1	vegetation removal	1.4.2	second floor	0.569	0.670	0.522	0.587
10	1.2.2	stripping ground	1.4.3	third floor	0.736	0.670	0.446	0.617
11	1.2.3	excavation	1.2.2	excavation	1.000	0.726	0.560	0.762
12	1.3	substructure	1.3	structure	0.923	0.784	0.718	0.808
13	1.3.1	pile driving	1.5.3	third floor	0.673	0.682	0.778	0.711
14	1.3.2	pile caps	1.3.5	fourth floor structure	0.729	0.839	0.673	0.747
15	1.3.2.1	reinforcement installation	1.3.1.2	reinforcing placement	0.603	0.796	0.946	0.782
16	1.3.2.2	form work	1.3.1.1	form work installation	0.952	0.796	0.808	0.852
17	1.3.2.3	concrete pouring	1.3.1.3	concrete pouring	1.000	0.796	0.815	0.870
18	1.3.2.4	curing concrete	1.3.1.4	curing	0.833	0.796	0.857	0.829
19	1.3.2.5	form work removal	1.3.1.5	form work removal	1.000	0.796	0.834	0.877
20	1.3.3	left abutment	1.3.1	foundation	0.833	0.839	0.660	0.778
21	1.3.3.1	reinforcement installation	1.6.1.4	doors installation	0.871	0.718	0.567	0.719
22	1.3.3.2	form work	1.6.3.6	painting	0.768	0.692	0.591	0.684
23	1.3.3.3	concrete pouring	1.3.2.1.3	concrete pouring	1.000	0.706	0.725	0.810
24	1.3.3.4	curing concrete	1.6.1.2	dry walls installation	0.629	0.718	0.620	0.656
25	1.3.3.5	form work removal	1.6.1.1	exterior walls installation	0.698	0.718	0.605	0.674
26	1.3.3.6	backfilling	1.6.2.1	exterior walls installation	0.225	0.696	0.740	0.554

27	1.3.4	right abutment	1.3.4	third floor structure	0.816	0.839	0.665	0.774
28	1.3.4.1	reinforcement installation	1.6.1.3	windows installation	0.889	0.755	0.567	0.737
29	1.3.4.2	form work	1.6.1.6	painting	0.768	0.755	0.591	0.704
30	1.3.4.3	concrete pouring	1.6.4.5	flooring	0.714	0.728	0.602	0.681
31	1.3.4.4	curing concrete	1.6.1.5	flooring	0.726	0.755	0.594	0.691
32	1.3.4.5	form work removal	1.6.3.4	doors installation	0.728	0.736	0.600	0.688
33	1.3.4.6	backfilling	1.6.4.6	painting	0.455	0.728	0.727	0.637
34	1.4	super structure	1.5	electrical works	0.529	0.784	0.774	0.695
35	1.4.1	piers	1.5.1	ground floor	0.805	0.682	0.775	0.754
36	1.4.1.1	reinforcement installation	1.5.3.4	fire alarm systems	0.827	0.714	0.655	0.732
37	1.4.1.2	form work	1.5.1.1	install wiring	0.695	0.716	0.720	0.710
38	1.4.1.3	concrete pouring	1.5.1.3	install light fixtures	0.598	0.716	0.731	0.682
39	1.4.1.4	form work removal	1.5.2.2	install outlets switches	0.734	0.714	0.707	0.718
40	1.4.2	column caps	1.3.3	second floor structure	0.892	0.815	0.652	0.786
41	1.4.2.1	reinforcement installation	1.6.2.3	windows installation	0.889	0.750	0.647	0.762
42	1.4.2.2	form work	1.6.2.6	painting	0.768	0.750	0.674	0.731
43	1.4.2.3	concrete pouring	1.6.3.5	flooring	0.714	0.750	0.686	0.717
44	1.4.2.4	form work removal	1.6.2.4	doors installation	0.728	0.750	0.683	0.720
45	1.4.2.5	bearing pads	1.6.2.5	flooring	0.774	0.750	0.659	0.728
46	1.4.3	girders	1.5.2	second floor	0.639	0.682	0.831	0.717
47	1.4.3.1	girder installation	1.3.3.2	beams and slab	0.705	0.747	0.857	0.770
48	1.4.3.2	bracing	1.3.3.1	columns	0.857	0.747	0.705	0.770

49	1.4.4	decks	1.3.2	ground floor structure	0.935	0.815	0.624	0.791
50	1.4.4.1	form work	1.5.2.4	fire alarm systems	0.777	0.756	0.688	0.740
51	1.4.4.2	reinforcement installation	1.5.1.4	fire alarm systems	0.827	0.769	0.655	0.750
52	1.4.4.3	concrete pouring	1.5.4.4	fire alarm systems	0.721	0.724	0.705	0.717
53	1.4.4.4	form work removal	1.5.1.2	install outlets switches	0.734	0.769	0.702	0.735
54	1.5	road works	1.2	earth works	0.843	0.784	0.711	0.779
55	1.5.1	guardrails	1.6.3	third floor finishing	0.580	0.792	0.630	0.667
56	1.5.2	asphalt	1.6.1	ground floor finishing	0.748	0.792	0.595	0.712
57	1.5.3	surface marking	1.4.1	ground floor	0.864	0.737	0.632	0.744
58	1.5.4	electrical wiring	1.4.5	elevator	0.553	0.737	0.707	0.666
59	1.5.5	lightings	1.6.5	cleaning	0.700	0.792	0.592	0.695
60	1.6	finishing works	1.6	finishing works	1.000	0.784	0.724	0.836
61	1.6.1	landscaping	1.6.4	fourth floor finishing	0.392	0.870	0.571	0.611
62	1.6.2	testing systems	1.6.6	testing systems	1.000	0.870	0.286	0.719
63	1.6.3	cleaning	1.2.1	stripping	1.000	0.796	0.543	0.780

Similarity scores and measurements

mapped nodes	**Unmapped nodes**	**Mapped not relevant**	**Node similarity**	**Structural similarity**	**Total similarity**
126	70	82	0.467	0.685	0.576
Mapping precision			0.35		

Table 23. Mapped nodes in comparison of the pair (B_1 , C_2), (Second scenario)

Row	$steel\ building_1$		$steel\ building_2$		$sim_{semantic}$	$sim_{parents}$	$sim_{siblings}$	$sim_{average}$
	WBS code	Task	WBS code	Task				
1	1.1	site preparation	1.1	Site preparation	1.000	0.784	0.453	0.745
2	1.1.2	transport equipment	1.1.2	equipment mobilization	0.861	0.870	0.572	0.768
3	1.1.3	surveying	1.1.1	surveying	1.000	0.870	0.529	0.800
4	1.1.5	temporary buildings	1.1.3	fences installation	0.715	0.870	0.618	0.734
5	1.3.1	pile driving	1.3.2	ground floor structure	0.694	0.839	0.700	0.745
6	1.3.2.1	reinforcement installation	1.3.1.2	reinforcing placement	0.603	0.796	0.946	0.782
7	1.3.2.2	form work	1.3.1.1	form work installation	0.952	0.796	0.650	0.800
8	1.3.2.3	concrete pouring	1.3.1.3	concrete pouring	1.000	0.796	0.658	0.818
9	1.3.2.4	curing concrete	1.3.1.4	curing	0.833	0.796	0.700	0.776
10	1.3.2.5	form work removal	1.3.1.5	form work removal	1.000	0.796	0.675	0.824
11	1.3.3	left abutment	1.3.1	foundation	0.833	0.839	0.474	0.715
12	1.3.3.1	reinforcement installation	1.3.4.1.1	form work installation	0.868	0.701	0.614	0.728
13	1.3.3.2	form work	1.3.2.1.1	form work installation	0.952	0.706	0.578	0.746
14	1.3.3.3	concrete pouring	1.3.2.1.3	concrete pouring	1.000	0.706	0.585	0.764
15	1.3.3.4	curing concrete	1.3.3.1.4	curing	0.833	0.705	0.622	0.720
16	1.3.3.5	form work removal	1.3.2.1.5	form work removal	1.000	0.706	0.600	0.769

17	1.3.4.1	reinforcement installation	1.3.5.1.1	form work installation	0.868	0.697	0.614	0.726
18	1.3.4.2	form work	1.3.3.1.1	form work installation	0.952	0.705	0.578	0.745
19	1.3.4.3	concrete pouring	1.3.3.1.3	concrete pouring	1.000	0.705	0.585	0.763
20	1.3.4.4	curing concrete	1.3.2.1.4	curing	0.833	0.706	0.622	0.721
21	1.3.4.5	form work removal	1.3.3.1.5	form work removal	1.000	0.705	0.600	0.768
22	1.4	super structure	1.3	structure	0.861	0.784	0.569	0.738
23	1.4.1.1	reinforcement installation	1.6.3.3	windows installation	0.889	0.700	0.553	0.714
24	1.4.1.2	form work	1.3.3.1.2	reinforcing placement	0.637	0.735	0.819	0.730
25	1.4.1.3	concrete pouring	1.3.4.1.3	concrete pouring	1.000	0.725	0.514	0.746
26	1.4.1.4	form work removal	1.3.4.1.5	form work removal	1.000	0.725	0.534	0.753
27	1.4.2	column caps	1.3.4	third floor structure	0.892	0.815	0.485	0.731
28	1.4.2.1	reinforcement installation	1.6.2.3	windows installation	0.889	0.751	0.647	0.762
29	1.4.2.2	form work	1.6.2.6	painting	0.768	0.751	0.674	0.731
30	1.4.2.3	concrete pouring	1.6.2.5	flooring	0.714	0.751	0.686	0.717
31	1.4.2.4	form work removal	1.6.2.4	doors installation	0.728	0.751	0.683	0.721
32	1.4.2.5	bearing pads	1.6.2.1	exterior walls installation	0.701	0.751	0.689	0.714
33	1.4.3.1	girder installation	1.3.3.2	beams and slab	0.705	0.635	0.857	0.733
34	1.4.3.2	bracing	1.3.3.1	columns	0.857	0.635	0.705	0.733

35	1.4.4	decks	1.3.3	second floor structure	0.914	0.815	0.479	0.736
36	1.4.4.1	form work	1.6.1.6	painting	0.768	0.787	0.583	0.712
37	1.4.4.2	reinforcement installation	1.6.1.3	windows installation	0.889	0.787	0.553	0.743
38	1.4.4.3	concrete pouring	1.5.1.1	install wiring	0.580	0.801	0.753	0.712
39	1.4.4.4	form work removal	1.6.1.4	doors installation	0.728	0.787	0.593	0.703
40	1.5	road works	1.2	earth works	0.843	0.784	0.550	0.726
41	1.5.3	surface marking	1.3.5	fourth floor structure	0.773	0.804	0.540	0.706
42	1.6	finishing works	1.6	finishing works	1.000	0.784	0.491	0.758
43	1.6.1	landscaping	1.2.2	excavation	0.815	0.796	0.667	0.759
44	1.6.2	testing systems	1.6.6	testing systems	1.000	0.870	0.286	0.719
45	1.6.3	cleaning	1.2.1	stripping	1.000	0.796	0.543	0.780

Similarity scores and measurements

mapped nodes	Unmapped nodes	Mapped not relevant	Node similarity	Structural similarity	Total similarity
90	106	26	0.342	0.669	0.505
Mapping precision			0.71		

Table 24. Mapped nodes in comparison of the pair (B_1 , C_2), (Third scenario)

Row	$steel\ building_1$		$steel\ building_2$		$sim_{semantic}$	$sim_{parents}$	$sim_{siblings}$	$sim_{average}$
	WBS code	Task	WBS code	Task				
1	1	bridge project	1	concrete building	0.784	0.784	0.000	0.588
2	1.1	site preparation	1.1	Site preparation	1.000	0.784	0.675	0.865
3	1.1.1	site mobilization	1.1.2	equipment mobilization	0.853	0.870	0.578	0.788

4	1.1.2	transport equipment	1.1.3	fences installation	0.733	0.870	0.618	0.739
5	1.1.3	surveying	1.1.1	surveying	1.000	0.870	0.529	0.850
6	1.1.4	securing the site	1.4.2	second floor	0.505	0.715	0.513	0.560
7	1.1.5	temporary buildings	1.6.2	second floor finishing	0.702	0.795	0.575	0.693
8	1.2	earthworks	1.5	electrical works	0.500	0.784	0.815	0.650
9	1.2.1	vegetation removal	1.6.4	fourth floor finishing	0.638	0.686	0.458	0.605
10	1.2.2	stripping ground	1.6.5	cleaning	0.861	0.686	0.406	0.703
11	1.2.3	excavation	1.2.2	excavation	1.000	0.726	0.560	0.821
12	1.3	substructure	1.3	structure	0.923	0.784	0.718	0.837
13	1.3.1	pile driving	1.5.2	second floor	0.673	0.682	0.778	0.701
14	1.3.2	pile caps	1.3.4	third floor structure	0.801	0.839	0.673	0.779
15	1.3.2.1	reinforcement installation	1.6.3.3	windows installation	0.889	0.721	0.630	0.782
16	1.3.2.2	form work	1.3.1.1	form work installation	0.952	0.796	0.808	0.877
17	1.3.2.3	concrete pouring	1.3.1.3	concrete pouring	1.000	0.796	0.815	0.903
18	1.3.2.4	curing concrete	1.3.1.4	curing	0.833	0.796	0.857	0.830
19	1.3.2.5	form work removal	1.3.1.5	form work removal	1.000	0.796	0.834	0.908
20	1.3.3	left abutment	1.3.5	fourth floor structure	0.669	0.839	0.671	0.712
21	1.3.3.1	reinforcement installation	1.6.1.4	doors installation	0.871	0.718	0.567	0.757
22	1.3.3.2	form work	1.6.1.6	painting	0.768	0.718	0.591	0.711
23	1.3.3.3	concrete pouring	1.3.2.1.3	concrete pouring	1.000	0.706	0.725	0.858

24	1.3.3.4	curing concrete	1.6.3.4	doors installation	0.654	0.692	0.614	0.654
25	1.3.3.5	form work removal	1.3.2.1.5	form work removal	1.000	0.706	0.742	0.862
26	1.3.3.6	backfilling	1.3.2.1.4	curing	0.364	0.706	0.762	0.549
27	1.3.4	right abutment	1.3.1	foundation	0.851	0.839	0.631	0.793
28	1.3.4.1	reinforcement installation	1.6.4.3	windows installation	0.889	0.728	0.567	0.768
29	1.3.4.2	form work	1.3.1.2	reinforcing placement	0.637	0.852	0.822	0.737
30	1.3.4.3	concrete pouring	1.6.4.5	flooring	0.714	0.728	0.602	0.690
31	1.3.4.4	curing concrete	1.6.1.5	flooring	0.726	0.755	0.594	0.700
32	1.3.4.5	form work removal	1.6.2.4	doors installation	0.728	0.741	0.600	0.699
33	1.3.4.6	backfilling	1.6.4.6	painting	0.455	0.728	0.727	0.591
34	1.4	super structure	1.4	mechanical works	0.529	0.784	0.774	0.654
35	1.4.1	piers	1.5.1	ground floor	0.805	0.682	0.775	0.767
36	1.4.1.1	reinforcement installation	1.5.1.4	fire alarm systems	0.827	0.716	0.655	0.756
37	1.4.1.2	form work	1.5.3.4	fire alarm systems	0.777	0.714	0.688	0.739
38	1.4.1.3	concrete pouring	1.3.3.1.3	concrete pouring	1.000	0.735	0.694	0.857
39	1.4.1.4	form work removal	1.5.2.2	install outlets switches	0.734	0.714	0.707	0.722
40	1.4.2	column caps	1.3.3	second floor structure	0.892	0.815	0.652	0.813
41	1.4.2.1	reinforcement installation	1.6.2.3	windows installation	0.889	0.750	0.647	0.794
42	1.4.2.2	form work	1.6.2.6	painting	0.768	0.750	0.674	0.740

43	1.4.2.3	concrete pouring	1.6.3.5	flooring	0.714	0.750	0.686	0.716
44	1.4.2.4	form work removal	1.6.3.6	painting	0.737	0.750	0.669	0.723
45	1.4.2.5	bearing pads	1.6.2.5	flooring	0.774	0.750	0.659	0.739
46	1.4.3	girders	1.4.5	elevator	0.700	0.682	0.733	0.704
47	1.4.3.1	girder installation	1.3.4.1	columns	0.806	0.747	0.752	0.778
48	1.4.3.2	bracing	1.3.3.1	columns	0.857	0.747	0.705	0.792
49	1.4.4	decks	1.3.2	ground floor structure	0.935	0.815	0.624	0.827
50	1.4.4.1	form work	1.5.2.4	fire alarm systems	0.777	0.756	0.688	0.750
51	1.4.4.2	reinforcement installation	1.6.1.3	windows installation	0.889	0.772	0.553	0.775
52	1.4.4.3	concrete pouring	1.5.4.4	fire alarm systems	0.721	0.724	0.705	0.718
53	1.4.4.4	form work removal	1.5.1.2	install outlets switches	0.734	0.769	0.702	0.734
54	1.5	road works	1.2	earth works	0.843	0.784	0.711	0.795
55	1.5.1	guardrails	1.5.3	third floor	0.654	0.737	0.637	0.671
56	1.5.2	asphalt	1.6.1	ground floor finishing	0.748	0.792	0.595	0.721
57	1.5.3	surface marking	1.4.1	ground floor	0.864	0.737	0.632	0.774
58	1.5.4	electrical wiring	1.4.3	third floor	0.497	0.737	0.714	0.611
59	1.5.5	lightings	1.6.3	third floor finishing	0.682	0.792	0.578	0.684
60	1.6	finishing works	1.6	finishing works	1.000	0.784	0.724	0.877
61	1.6.1	landscaping	1.5.1.3	install light fixtures	0.493	0.608	0.607	0.550
62	1.6.2	testing systems	1.6.6	testing systems	1.000	0.870	0.286	0.789
63	1.6.3	cleaning	1.2.1	stripping	1.000	0.796	0.543	0.835

Similarity scores and measurements

mapped nodes	Unmapped nodes	Mapped not relevant	Node similarity	Structural similarity	Total similarity
126	70	72	0.479	0.694	0.587
Mapping precision			0.43		

Table 25. Mapped nodes in comparison of the pair (B_1 , C_2), (Fourth scenario)

Row	$steel\ building_1$		$steel\ building_2$		$sim_{semantic}$	$sim_{parents}$	$sim_{siblings}$	$sim_{average}$
	WBS code	Task	WBS code	Task				
1	1.1	site preparation	1.1	Site preparation	1.000	0.784	0.453	0.809
2	1.1.2	transport equipment	1.1.2	equipment mobilization	0.861	0.870	0.572	0.791
3	1.1.3	surveying	1.1.1	surveying	1.000	0.870	0.529	0.850
4	1.1.5	temporary buildings	1.1.3	fences installation	0.715	0.870	0.618	0.729
5	1.2	earthworks	1.6.4.2	dry walls installation	0.802	0.667	0.601	0.718
6	1.2.3	excavation	1.2.2	excavation	1.000	0.639	0.560	0.800
7	1.3	substructure	1.3	structure	0.923	0.784	0.503	0.783
8	1.3.2.1	reinforcement installation	1.3.1.2	reinforcing placement	0.603	0.796	0.946	0.737
9	1.3.2.2	form work	1.3.1.1	form work installation	0.952	0.796	0.650	0.838
10	1.3.2.3	concrete pouring	1.3.1.3	concrete pouring	1.000	0.796	0.658	0.864
11	1.3.2.4	curing concrete	1.3.1.4	curing	0.833	0.796	0.700	0.791
12	1.3.2.5	form work removal	1.3.1.5	form work removal	1.000	0.796	0.675	0.868
13	1.3.3.1	reinforcement installation	1.6.2.4	doors installation	0.871	0.676	0.439	0.714

14	1.3.3.2	form work	1.3.2.1.1	form work installation	0.952	0.706	0.578	0.797
15	1.3.3.3	concrete pouring	1.3.2.1.3	concrete pouring	1.000	0.706	0.585	0.823
16	1.3.3.4	curing concrete	1.3.2.1.4	curing	0.833	0.706	0.622	0.749
17	1.3.3.5	form work removal	1.3.2.1.5	form work removal	1.000	0.706	0.600	0.827
18	1.3.4	right abutment	1.3.1	foundation	0.851	0.839	0.448	0.747
19	1.3.4.1	reinforcement installation	1.6.1.4	doors installation	0.871	0.760	0.439	0.735
20	1.3.4.2	form work	1.3.3.1.1	form work installation	0.952	0.705	0.578	0.797
21	1.3.4.3	concrete pouring	1.3.3.1.3	concrete pouring	1.000	0.705	0.585	0.822
22	1.3.4.4	curing concrete	1.3.3.1.4	curing	0.833	0.705	0.622	0.748
23	1.3.4.5	form work removal	1.3.3.1.5	form work removal	1.000	0.705	0.600	0.826
24	1.4	super structure	1.6.4.4	doors installation	0.781	0.667	0.616	0.711
25	1.4.1	piers	1.4.2	second floor	0.802	0.529	0.714	0.711
26	1.4.1.1	reinforcement installation	1.6.3.3	windows installation	0.889	0.700	0.553	0.757
27	1.4.1.2	form work	1.3.4.1.1	form work installation	0.952	0.725	0.505	0.784
28	1.4.1.3	concrete pouring	1.3.4.1.3	concrete pouring	1.000	0.725	0.514	0.810
29	1.4.1.4	form work removal	1.3.4.1.5	form work removal	1.000	0.725	0.534	0.815
30	1.4.2	column caps	1.3.4	third floor structure	0.892	0.815	0.485	0.771

31	1.4.2.1	reinforcement installation	1.6.2.3	windows installation	0.889	0.751	0.647	0.794
32	1.4.2.2	form work	1.6.2.6	painting	0.768	0.751	0.674	0.740
33	1.4.2.3	concrete pouring	1.6.2.5	flooring	0.714	0.751	0.686	0.716
34	1.4.2.4	form work removal	1.6.3.6	painting	0.737	0.751	0.669	0.724
35	1.4.2.5	bearing pads	1.6.2.1	exterior walls installation	0.701	0.751	0.689	0.711
36	1.4.3.1	girder installation	1.3.4.1	columns	0.806	0.635	0.752	0.750
37	1.4.3.2	bracing	1.3.3.1	columns	0.857	0.635	0.705	0.764
38	1.4.4	decks	1.3.2	ground floor structure	0.935	0.815	0.479	0.791
39	1.4.4.1	form work	1.6.1.6	painting	0.768	0.787	0.583	0.726
40	1.4.4.2	reinforcement installation	1.6.1.3	windows installation	0.889	0.787	0.553	0.779
41	1.4.4.3	concrete pouring	1.6.1.5	flooring	0.714	0.787	0.596	0.703
42	1.4.4.4	form work removal	1.6.3.4	doors installation	0.728	0.770	0.593	0.705
43	1.5	road works	1.2	earth works	0.843	0.784	0.550	0.755
44	1.5.3	surface marking	1.3.3	second floor structure	0.836	0.804	0.536	0.753
45	1.5.5	lightings	1.6.4.6	painting	0.842	0.693	0.528	0.726
46	1.6	finishing works	1.6	finishing works	1.000	0.784	0.491	0.819
47	1.6.2	testing systems	1.6.6	testing systems	1.000	0.870	0.286	0.789
48	1.6.3	cleaning	1.2.1	stripping	1.000	0.796	0.543	0.835

Similarity scores and measurements

mapped nodes	Unmapped nodes	Mapped not relevant	Node similarity	Structural similarity	Total similarity
96	100	32	0.378	0.685	0.532
Mapping precision			0.67		

Appendix C: Precision scores in retrieving the query samples

Table 26. Retrieving precession scores (First scenario)

Threshold	Query sample	Total similarity		Node similarity		Structural similarity	
		Recall score					
		50%	100%	50%	100%	50%	100%
		Retrieving precision					
0.5	b1	0.5	0.66	0.5	0.66	0.5	0.5
	c1	0.5	0.4	0.5	0.33	0.5	0.66
	s1	1	1	1	1	1	1
Average		0.68		0.66		0.69	
0.55	b1	0.5	0.66	0.5	0.6	0.5	0.5
	c1	0.5	0.33	0.25	0.25	0.5	0.66
	s1	1	1	1	0.66	1	1
Average		0.66		0.54		0.69	
0.6	b1	0.5	0.66	1	0.66	0.5	0.66
	c1	0.33	0.33	0.33	0.28	0.5	0.66
	s1	1	1	0.33	0.66	1	1
Average		0.64		0.54		0.72	
0.65	b1	1	0.66	1	0.66	0.5	0.66
	c1	1	0.29	0.33	0.25	0.5	0.66
	s1	0.5	0.66	0.5	0.66	1	1
Average		0.68		0.57		0.72	
0.7	b1	1	0.66	1	0.66	1	0.66
	c1	0.5	0.25	0.5	0.25	0.5	0.66
	s1	0.5	0.66	0.5	0.66	1	1
Average		0.59		0.59		0.80	
0.75	b1	1	0.66	1	0.66	1	0.66
	c1	0.5	0.29	0.5	0.25	0.5	0.66
	s1	0.5	0.66	0.5	0.5	1	1
Average		0.6		0.56		0.80	
0.8	b1	1	0.4	1	0.29	1	0.5
	c1	0.25	0.33	0.17	0.25	0.5	0.66
	s1	0.5	0.66	1	0.66	1	1
Average		0.52		0.56		0.78	

Table 27. Retrieving precession scores (Second scenario)

Threshold	Query sample	Total similarity		Node similarity		Structural similarity	
		Recall score					
		50%	100%	50%	100%	50%	100%
		Retrieving precision					
0.5	b1	0.5	0.5	0.5	0.66	0.5	0.5
	c1	0.5	0.5	0.5	0.4	0.5	0.66
	s1	1	1	1	1	1	1
Average		0.67		0.68		0.69	
0.55	0.5	0.66	0.5	0.66	0.5	0.5	0.5
	0.5	0.4	0.4	0.4	0.5	0.66	0.5
	1	0.66	0.5	0.5	1	0.5	1
Average		0.62		0.49		0.61	
0.6	b1	0.5	0.5	0.5	0.5	0.5	0.5
	c1	0.33	0.4	0.25	0.33	0.5	0.66
	s1	1	1	1	0.5	1	1
Average		0.62		0.51		0.69	
0.65	b1	1	0.66	1	0.66	0.5	0.66
	c1	0.5	0.33	0.33	0.28	0.5	0.66
	s1	1	1	1	0.66	1	0.66
Average		0.74		0.65		0.66	
0.7	b1	1	0.66	1	0.66	0.5	0.66
	c1	0.5	0.33	0.5	0.25	0.5	0.66
	s1	1	0.66	0.5	0.66	1	1
Average		0.69		0.595		0.72	
0.75	b1	1	1	1	0.66	1	0.5
	c1	0.33	0.28	0.33	0.25	0.5	0.66
	s1	0.5	0.66	0.5	0.66	1	1
Average		0.62		0.56		0.78	
0.8	b1	1	0.28	1	0.25	0.6	0.66
	c1	0.33	0.33	0.25	0.25	0.5	0.66
	s1	0.5	0.66	1	0.66	1	0.66
Average		0.52		0.57		0.68	

Appendix D: Sample equation

The following calculations show how the semantic similarity of task node "doors installation" and "windows installation" is measured by means of the Equation 8. Table 28 and Table 29 show the comparison of these two tasks component words.

Eq. 8

$$sim_{semantic}(n_i, m_j) = \frac{1}{2}\left(\frac{\sum_{w\in\{n_i\}}(maxSim(w, m_j)_{wup})}{\sum_{w\in\{n_i\}} 1} + \frac{\sum_{w\in\{m_j\}}(maxSim(w, n_i)_{wup})}{\sum_{w\in\{m_j\}} 1}\right)$$

n_i ="doors installation", $w = \{$doors, installation$\}$

m_j = "windows installation", $w = \{$windows, installation$\}$

Table 28. Comparing the component words of n_i and m_j

$w \in \{n_i\}$	m_j	Similarity	$maxSim(w, m_j)_{wup}$
doors	windows	0.750	0.769
doors	installation	0.769	
installation	windows	0.769	1.0
installation	installation	1.0	

Table 29. Comparing the component words of $\boldsymbol{m_j}$ and $\boldsymbol{n_i}$

$w \in \{m_j\}$	n_i	Similarity	$maxSim(w, n_i)_{wup}$
windows	doors	0.750	0.769
windows	installation	0.769	
installation	doors	0.769	1.0
installation	installation	1.0	

Eq. 8

$$sim_{semantic}(n_i, m_j) = \frac{0.769 + 1.0}{2} + \frac{0.769 + 1.0}{2} = 0.84$$