Semantics-based approach for generating partial views from linked life-cycle highway project data

by

Tuyen Thanh Le

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Civil Engineering (Construction Engineering and Management)

Program of Study Committee:
H. David Jeong, Major Professor
Charles T. Jahren
Yelda Turkan
Shashi K. Gadia
Omar G. Smadi

Iowa State University

Ames, Iowa

2017

DEDICATION

TO MY WIFE AND MY PARENTS

FOR THEIR OVERWHELMING SUPPORT

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of the research and the writing of this dissertation.

First and foremost, I am grateful to my advisor, Dr. H. David Jeong, for his guidance and support throughout this research and the writing of this dissertation. Regular discussion with him enabled me to successfully explore the research problem and to identify appropriate research approaches. His insights and words of encouragement inspired me for completing my doctoral program and prepared me with a course of necessary skills for my future professional career.

My sincere appreciation also goes to my committee members for their efforts and contributions to this work: Drs. Charles Jahren, Yelda Turkan, Shashi Gadia, and Omar Smadi. Their constructive comments on my program of study significantly helped me select appropriate classes to fulfill the fundamental knowledge needs for my study. Additionally, without their reviews of my proposal and dissertation, I would not be able to reach the current stage.

Moreover, I wish to thank Drs. Stephen B. Gilbert, and Evgeny Chukharev-Hudilainen. The collaboration with them in an interdisciplinary team allowed me to acquire expertise in linguistics and machine learning which are required to complete this study. Their insightful questions, suggestions, and reviews considerably help to produce publishable papers.

Finally, I would like to acknowledge the financial support from the National Science Foundation. I would additionally like to express my gratefulness to anonymous reviewers who have provided insightful comments on the publications resulting from this study.

ABSTRACT

The purpose of this dissertation is to develop methods that can assist data integration and extraction from heterogeneous sources generated throughout the life-cycle of a highway project. In the era of computerized technologies, project data is largely available in digital format. Due to the fragmented nature of the civil infrastructure sector, digital data are created and managed separately by different project actors in proprietary data warehouses. The differences in the data structure and semantics greatly hinder the exchange and fully reuse of digital project data. In order to address those issues, this dissertation carries out the following three individual studies.

The first study aims to develop a framework for interconnecting heterogeneous life cycle project data into an unified and linked data space. This is an ontology-based framework that consists of two phases: (1) translating proprietary datasets into homogeneous RDF data graphs; and (2) connecting separate data networks to each other. Three domain ontologies for design, construction, and asset condition survey phases are developed to support data transformation. A merged ontology that integrates the domain ontologies is constructed to provide guidance on how to connect data nodes from domain graphs.

The second study is to deal with the terminology inconsistency between data sources. An automated method is developed that employs Natural Language Processing (NLP) and machine learning techniques to support constructing a domain specific lexicon from design manuals. The method utilizes pattern rules to extract technical terms from texts and learns their representation vectors using a neural network based word embedding approach. The study also includes the development of an integrated method of minimal-supervised machine learning, clustering analysis, and word vectors, for computing the term semantics and classifying the relations between terms in the target lexicon.

In the last study, a data retrieval technique for extracting subsets of an XML civil data schema is designed and tested. The algorithm takes a keyword input of the end user and

returns a ranked list of the most relevant XML branches. This study utilizes a lexicon of the highway domain generated from the second study to analyze the semantics of the end user keywords. A context-based similarity measure is introduced to evaluate the relevance between a certain branch in the source schema and the user query.

The methods and algorithms resulting from this research were tested using case studies and empirical experiments. The results indicate that the study successfully address the heterogeneity in the structure and terminology of data and enable a fast extraction of sub-models of data. The study is expected to enhance the efficiency in reusing digital data generated throughout the project life-cycle, and contribute to the success in transitioning from paper-based to digital project delivery for civil infrastructure projects.

CHAPTER 1. INTRODUCTION

The primary goal of this study is to develop a computational platform for automated data extraction from heterogeneous digital project data sources for the highway sector. The research offers a novel semantics-based approach to integrating project data that are structured and named differently from diverse silos. The study also includes a data retrieval algorithm that can interpret the end user's intention from their input queries and automatically extract the desired subset from a large civil data model. This study is expected to enable a seamless data exchange among project stakeholders, and ultimately to help reduce data duplication throughout the project life cycle. The following sections depict the importance of the research activity, its contributions to the body of knowledge, and the research methodology.

1.1 Importance of the research activity

The rapid implementation of such information technologies as Civil Information Modeling (CIM), Geographic Information Systems (GISs), or LIDAR has transformed how transportation project information is created, exchanged, and managed throughout the life cycle. The growing availability of digital data offers undeniable benefits to individual project stakeholders (engineers, contractors, asset managers, etc.). However, a transportation asset as a whole has not yet fully benefited from the potentials of digital project data as an accessible, reusable and reliable information source for life-cycle decision making. In current practices, a majority of the data transactions among project actors are still on a manual basis using papers or electronic papers instead of digital datasets [JBKnowledge (2016)]. An estimate by the National Institute of Standard and Technology (NIST) indicates that the manual work spent for extracting data from non-machine-readable as-designed and as-built documents and re-entering into facil-

ity management systems costs the U.S. capital facilities industry at least $15.8 billion per year [Gallaher et al. (2004)]. This evidence from the vertical sector demonstrates the importance of the ability to directly use heterogeneous project datasets to reducing data re-creation wastes and improving productivity throughout the life cycle of horizontal projects.

Data extraction is a key task in a digital data centric project delivery. Despite growing digital data, such data resources can not be fully exploited without the ability to extract the desired data. The data and information of a highway project are typically contributed by various project stakeholders from different domains of knowledge. Data sharing can either occur at within the same stage (i.e., structural engineers and mechanical engineers) or across different project stages (i.e., designers and contractors). For a given data sharing use case, only a subset of data rather than the entire data is needed [Froese (2003); East et al. (2012)]. Thus, the efficiency and effectiveness in extracting a subset of data from the life-cycle data space are key to successfully facilitate seamless exchange of digital project data and full computer-to-computer communication. However, due to the fragmented nature of the civil infrastructure industry, digital project data are largely heterogeneous, leading to various technical challenges for reusing data over the life cycle.

A primary problem is the lack of interoperability between software applications to communicate to each other. Project data are generated by different project partners, being archived in proprietary platforms and formats [Harrison et al. (2016)]. Due to the the syntactic discrepancy, the data obtained from a certain system is not readable to one another. Addressing the interoperability issue has been widely recognized as a pressing need to allow for computer-to-computer data exchange and seamless integration of heterogeneous data from multiple sources [Karimi et al. (2003); Gallaher et al. (2004); Bittner et al. (2005)]. The transportation sector, however, has not yet successfully facilitated a high degree of interoperability [Lefler (2014)]. In order to reuse digital data, much laborious work is required for finding, verifying, and transforming facility and project information from a certain format to one another [Gallaher et al. (2004)].

The diversity of data terminology is another big hurdle to the computer-to-computer communication. Names of things might vary across different data sources. A unique term can refer to different meanings in different contexts, and a single concept can be tagged with various la-

bels. Data integration in such a heterogeneous environment is highly problematic [Karimi et al. (2003)]. A lack of common understanding of the same data or similar data given in different terms can lead to the extraction and use of wrong data. Since computer systems are not yet able to understand the semantics of data, this issue creates a heavy burden to end user who must play as a middleware in the digital data exchange especially in cases of large and complex datasets.

In order to enhance the efficiency of data sharing, there is a demand for advanced computational techniques that can allow for automated extraction of data with minimized human interference [Venugopal et al. (2012b); Eastman (2012)]. The simplicity of data extraction critically decides the degree of reusability of up-stream digital models. Recent advances in the field of data science, linguistics processing, and machine learning have evidently shown the possibility to allow computers to understand the meaning of data and to enable user-friendly interfaces that accept natural language queries. However, the body-of-knowledge regarding data extraction for the civil infrastructure sector is not able to support fully-automated extraction of project data and still requires a high level of direct human intervention. The purpose of this research is to propose a novel approach for enhancing the ease of reuse of digital models generated through the life cycle of a civil infrastructure project. This study aims to enable the unambiguous use of data from heterogeneous data sources by developing a semantics-based integrated platform capable of linking and extracting partial models from isolated data spaces.

1.2 Background and gap of knowledge

This section briefly presents previous work on enhancing the reusability of heterogeneous project data. More detailed discussions on those studies will be described in the next manuscript-based chapters. This section also specifies research gaps and what are needed to overcome the limitations. The related studies can be divided into three areas of knowledge each of which is respectively discussed in the following sub-sections.

1.2.1 Overview of open data standards for sharing project data

Open data standards are widely accepted as a solution for the interoperability issue which hinders the integration and merging of project data generated from different software applications. Data standards use a certain data modeling language (e.g., EXPRESS, XML) to develop a common data schema for all associated knowledge domains of a construction project. Examples of standardized data schemas for transportation assets include LandXML, TransXML, and IFC Alignment. These data models serve as a neutral language between proprietary applications. One of the primary drawbacks of these standards is the semantic insufficiency [Niknam and Karshenas (2014); Yang and Zhang (2006); Venugopal et al. (2012b)]. Neither EXPRESS nor XML supports an explicit formulation of the relations between classes. Due to this reason, the context of a class is not formally defined and therefore its semantics is not directly provided. The shortcoming of semantics creates an important challenge for integrating and using data from separate sources. Both the data creator and receiver are required to have a deep understanding of the meaning of every single class in the neutral data schema to ensure a proper translation of data from or to a proprietary format.

This study employs a different approach to sharing life-cycle project data. In this research, ontology is a core means to present life-cycle project data. Ontology is referred as an explicit formalization of real-life conceptualization [Gruber (1995)]. It models data in a network format which consists of nodes and links, where nodes represent data entities and links represent the semantic relations between entities [McGuinness and Van Harmelen (2004)]. Ontology helps visualize the relationships between classes within a data schema. Since the context of a class is explicitly depicted, its meaning becomes machine and human-interpretable. Another advantage of the graph representation of data is that distinct data models can be merged and linked together by simply establishing additional links crossing from one to another. The implementation of this semantic data modeling approach will allow professionals in a certain discipline to be able to read external data sources and unambiguously integrate to their own systems.

1.2.2 Overview of efforts on data terminology classification

In order to help the civil infrastructure overcome the problem of terminological discrepancy between data sources, a few construction domain specific data dictionaries have been proposed for example the buildingSMART dictionary (ISO 12006-3) [buildingSMART (2016a)]. Digital dictionaries, which provide formal definitions for terms; can enable computers to interpret the meaning of data and to avoid mismatches when merging a multitude of data. However, since these semantic resources are largely handcrafted, their vocabulary size is yet relatively limited. Therefore, there is a demand for a computational technique that can automatically develop and maintain digital dictionaries to keep up with the growth of new terms.

The state-of-the-art shows a variety of research efforts on automated methods for detecting the semantic relations between construction project terms. Examples are those conducted by Marcus (1995); Navigli and Velardi (2010); Rezgui (2007), and Zhang and El-Gohary (2016). These studies employed natural language processing techniques to partially assist in computing the semantics of technical terms from a corpus of domain texts. However, no method can completely eliminate the involvement of human. Research is needed to fully automate the process of collecting technical terms and identifying their semantic relations for constructing transportation asset data dictionaries.

1.2.3 Overview of methods for model view extraction

Model View Definition (MVD) is a common approach to support extracting a specific subset, from a large digital project dataset, for a specific data exchange use case. An MVD is a specification for a specific view of a large open data standard. It is a subschema of the neutral data schema, defining data classes and attributes relevant to particular users. The conventional MVD development method includes two major steps: developing an Information Delivery Manual (IDM) and converting the IDM into an MVD. IDM is a text-based document that specifies what data needed for a specific data exchange, and MVD is a formal subschema of the required data reflecting a partial view of the neutral data standard. Developing IDMs and MVDs are laborious and time-consuming and require the involvement of various industry practitioners,

researchers, and software developers [Venugopal et al. (2012c); Eastman (2012); Hu (2014); Lee et al. (2016a)]. MVD developers are required to have considerable programming expertise and a deep understanding of data structure and meanings in a given discipline-specific data schema. These requirements become a big burden, especially for large and complex data models.

A few studies on the automated translation of IDMs into machine-readable MVDs are found in the literature. Previous studies focused on syntactic validation methods to assist developers in finding syntactically referenced entities for those semantically relevant to a data need. Examples of these research include the methods proposed by Yang and Eastman (2007); Lee (2009); Yang and Eastman (2007). Computer-aided MVD validation can reduce the burden on MVD developers. However, identifying data entities that reflect a specific data interest, which is a major task, is still performed manually. In order to ensure a fully-automated process of MVD generation, a computational technique for measuring the relatedness between IDM data requirements and the entities of the source schema is needed.

1.3 Research Questions

The efficiency and effectiveness of data integration and extraction are key to reusability of data throughout the life-cycle of a transportation project. The goal of this research is to offer effective methods and tools for extracting required data from heterogeneous data sources. The overall research question this research addresses is: *"How to enable computers to automatically generate a partial data view, given a keyword-based query, from heterogeneous life-cycle data spaces of a highway project?"* In order to answer that question, the following sub-questions must be addressed.

> **Question 1:** *What techniques and how they can be employed to inter-connect the life-cycle data spaces of a highway project?* Decision making requires data from multiple sources throughout the project life cycle. Once the above question is addressed, a method is available for the civil industry to integrate proprietary data generated by individual stakeholders.

Question 2: *How to achieve an automated method for constructing a digital knowledge base that provides formal definitions of heterogeneous terms used by different transportation agencies?* Data from distinct sources are presented using different technical terms, thus they can not be properly reused without inferring data meanings. By answering this questions, the study offers an automated method that can support researchers in translating data definitions in domain text documents into digital dictionaries.

Question 3: *What algorithm should be designed to allow computers to understand the semantics of the end user's queries and automatically generate subschemas from a large civil data schema?* The traditional method for subschema formulation is time-consuming and requires considerable programming expertise and a deep understanding of the source data schema. Addressing this question provides professionals with a user-friendly solution for finding semantically relevant data classes from complex and large data standards.

1.4 Research Objectives and Deliverables

The overall objective of this research is to develop an automated data retrieval platform that can interpret the end user's intention from their queries and automatically extract the desired data from heterogeneous sources of highway project data. The following are specific research objectives and expected deliverables.

Objective 1: Develop a framework for inter-connecting life cycle data spaces that can support data translation from proprietary formats into a unified format and allow for linking distinct datasets.

Deliverable 1: A life cycle data inter-connection framework.

Objective 2: Develop a method that can automatically extract technical terms, classify their semantic relations, and construct a knowledge base for the civil infrastructure domain from engineering text documents.

Deliverable 2: An automated method for classifying transportation asset terminology.

Objective 3: Design and test a context-based searching algorithm that can measure the semantic relevance between a source data element and the end user' keyword-based query, and return a subset of the source civil information schema.

Deliverables 3: A semantics-based partial schema extracting algorithm.

1.5 Research Methodology

Figure 1.1 presents the overall research approach. In order to achieve the first deliverable, the research employs Semantic Web techniques including Ontology Web Language (OWL) and Resource Description Framework (RDF) to develop a framework for interconnecting life-cycle project data. A case study is conducted to illustrate the success of the framework in linking heterogeneous data. The second deliverable of this study is obtained by utilizing Natural Language Processing (NLP) techniques and machine learning to develop an integrated method that can extract the semantics of terms and their semantic relations from texts. The final research deliverable is achieved by developing a semantics-based information retrieval system that utilizes a domain lexicon to interpret the end user's query and measures its semantic relatedness with the source data elements. The last two deliverables were evaluated by conducting experiments comparing the algorithm outputs and manually crafted test sets. The following is the specific research procedure to obtain the above deliverables.

Stage 1: Synthesize the body of knowledge regarding methods and tools in the following topics: digital project data exchange, natural language processing techniques, civil engineering data dictionaries, data query, model view extraction, and information retrieval systems.

Stage 2: Develop a framework for unifying heterogeneous life-cycle project data and integrate them into a connected data space.

– Select tools for interlinking heterogeneous data spaces generated through a highway project life cycle. Semantic Web is an emerging technology for linked data. Semantic Web provides tools, such as Ontology Web Language(OWL) and Resource Description Framework (RDF), to present data in a graph-based format and allows isolated data

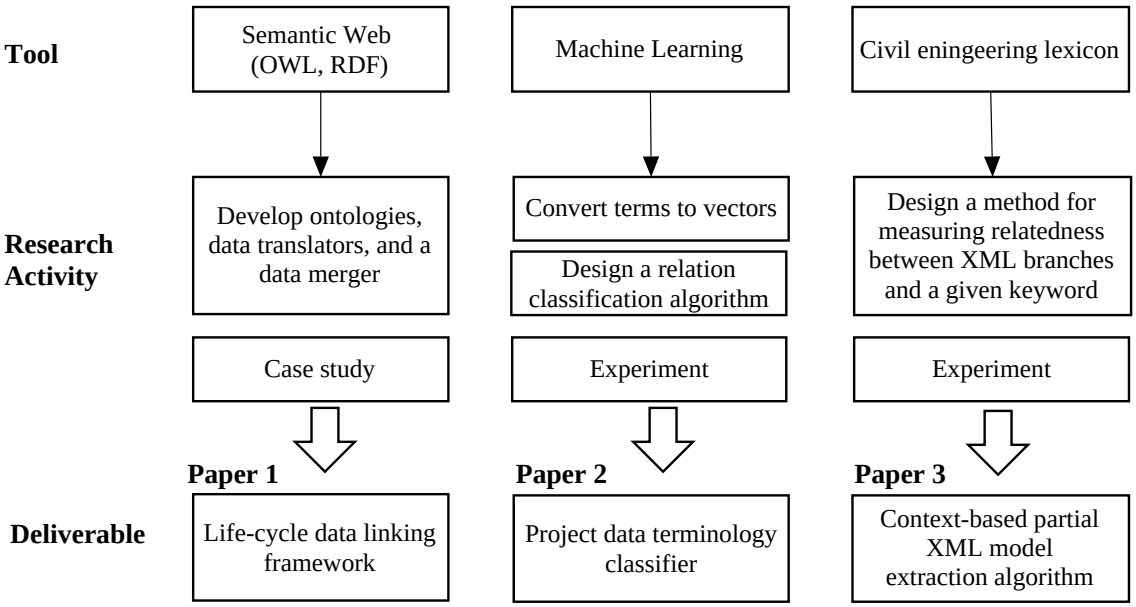

Figure 1.1: Overall research methodology

sources to be connected. Hence, this technology is a potential tool for linking separate project life cycle data islands.

- Develop a framework for inter-connecting life cycle data spaces using Semantic Web technology. Life-cycle ontologies for three project phases including design, construction, and asset management are developed. A merged ontology is then developed to provide guidance on how to connect individual highway data spaces into a unified space.
- Develop software prototype to support professionals in implementing the proposed framework. The prototype includes two translators for converting life-cycle data in proprietary formats (Landxml and relational database) into RDF.
- Undertake a case study to illustrate the procedure of interconnecting life-cycle data spaces using the proposed framework.

Stage 3: Develop an automated method for classifying heterogeneous technical terms in the field of highway construction.

- Analyze and select tools for measuring semantic similarity between technical terms commonly used for representing data through the infrastructure project life cycle.
- Collect highway-specific technical documents and build a text corpus for NLP tasks.
- Employ NLP to extract and rank technical terms commonly used in the highway industry.
- Utilize machine learning to develop a model for measuring semantic similarity between highway technical terms.
- Implement a minimal-supervised machine learning technique to learn the syntactic relation between terms.
- Integrate semantic models and syntactic relations to detect semantic relations among terms.
- Evaluate the method for automated construction of a transportation project data dictionary using a gold standard.

Stage 4: Develop a keyword-driven method for automated generation of partial views from a civil information data schema.

- Develop a method for identifying semantically related terms of a given keyword using a domain lexicon.
- Propose a context-based measure to evaluate the semantic relatedness between the user's query and a class in the source XML data schema.
- Implement a traversal method to find syntactically associated entities with those identified in the previous step to ensure the completeness of a generated MVD.
- Merge the XML branches retrieved to form a unique XML civil data suchschema.
- Conduct an experiment to evaluate the algorithm performance using a gold standard of hand-crafted model views.

1.6 Research Contribution

The primary contributions of this research are methods and tools that can enhance the efficiency of data exchange throughout the life cycle of a highway project. The improvement in data and information sharing between project participants and across various project development stages, which, will in turn, translate into increased productivity, efficiency in project delivery and accountability. The research success will translate into the willingness to embrace and effectively use digital datasets by all project stakeholders, and the seamless digital data exchange throughout the project life cycle can be achieved. Specifically, this dissertation contributes to the body of knowledge in the following areas.

First, the research develops a novel framework that can enable life-cycle digital data sources to be interconnected and translated into meaningful information for decision makers in highway asset management. The framework includes the development of various data wrappers that convert proprietary project data into a unique and mergeable format. The framework also allows for the integration of diverse project data into a single unified data space. An implementation tool built on this framework is expected to effectively assist project actors in directly reusing digital data generated by one another. As a result, a digital data exchange paradigm can be successfully established to replace the conventional costly and time-consuming paper-based project delivery method.

The next contribution is a novel automated method for classifying technical terms in the highway sector. This method can support the development of digital data dictionaries which are needed to ensure a proper integration of data from multiple sources. Since the method automatically extracts terms and their semantics directly from texts, less effort is required by developers to construct a domain specific knowledge base.

Additionally, as a result of a case study that implements the proposed terminology classifier on a highway corpus, an extensive highway lexicon which provides machine-readable definitions for a large number of domain terms is also developed. This resource can benefit the industry in various ways. It can be used by data integration platforms to match the same data given in different labels from diverse data sources. This extensive term library offers practitioners with

a rich resource to identify keywords, synonyms, and functionally related terms when searching for data from external databases.

Last but not least, this study contributes to the body of knowledge an effective method for automated generation of model views from an XML civil engineering data standard. The method is a semantic information retrieval technique that can explore the user's data interest from their input keywords and return a corresponding subtree of the source schema. The system developed in this study is expected to offer an enabling tool for MVD developers. A ranked list of related source entities generated by the system allows developers to work on a short list rather to manually scroll and examine the entire large and complex standard. With a list of the most semantically related items, the focus is paid on only a limited number of items; thus less effort is required to generate MVDs. In addition, less restriction is required for the end user to choose a keyword for searching relevant entities. Users with little background in the target domain are still able to extract a subschema without needing a deep understanding of the source schema. Once data extraction from digital models becomes straightforward, the bottleneck regarding MVD will be removed.

1.7 Dissertation organization

This dissertation is organized in a manuscript-based format, including 5 individual chapters. The dissertation first provides an introduction of the study in Chapter 1, followed by three separate published or submitted journal papers respectively compiled in chapters 2-4. Of which, each article addresses one research question of this dissertation. Specifically, chapter 2 presents the development of a framework for interconnecting life-cycle project data. Chapter 3 describes a method for learning semantics of technical terms from design manuals. Chapter 4 discusses a method for generating MVDs from a civil information model. The final chapter, chapter 6, concludes the dissertation with the major findings and future research opportunities.

CHAPTER 2. INTERLINKING LIFE-CYCLE DATA SPACES TO SUPPORT DECISION MAKING IN HIGHWAY ASSET MANAGEMENT

A paper published in *Automation in Construction*, Volume 64, Pages 54-64, (2016)

Tuyen Le, H. David Jeong

Abstract

Technology advances have changed highway project delivery and asset management from relying on 2D paper documents to n-D digital data sets. However, the implementation of diverse software applications imposes big challenges for integrating life-cycle data to support decision making in highway asset management due to the potential inconsistencies of levels of detail, data syntax and semantics. This paper presents an ontology based exchange mechanism that enables unification and interconnection of life-cycle data spaces to support decision making in highway asset management. The mechanism consists of the following key components: (1) domain and merged ontologies, (2) data wrappers and (3) a data query and reasoning system. The mechanism was tested on a sample roadway project retrieved from Landxml.org, and the results indicated the success in integrating fragmented life-cycle data spaces and extracting information for asset management.

2.1 Introduction

There has been a progressive trend of adopting advanced technologies in the highway industry. Digital models (3D, 4D, and nD) have been widely implemented in various types of projects (bridges, roadways and other transportation projects) for a wide range of purposes (visualization, clash detection, constructability review, etc.) and have changed project delivery process

and asset management from 2D paper-based documents to digital model based systems. This technology offers undeniable benefits to individual project stakeholders (engineers, contractors, owners, asset managers, etc.); however, due to the fragmented nature of the highway industry, a highway asset as a whole has not yet fully benefited from the potentials of digital models as a shared and reliable information source for life-cycle decision making. Since different project participants may use proprietary software platforms with different data structures, exchange of data becomes very challenging. Data exchange in a heterogeneous environment may lead to data loss, damage and requires time consuming processing in downstream phases. According to a research conducted by the National Institute of Standard and Technology (NIST), the un-interoperability issue was reported to cost the U.S. capital facilities industry at least $15.8 billion per year, and two thirds of those costs were incurred during the operation and maintenance stages [Gallaher et al. (2004)]. The major cost was time spent finding, verifying facility and project information, and transferring that information into a useful format. This finding indicates that the failure of collecting and transferring project data from upstream design and construction stages to asset management stage in proper format results in high operational costs. Therefore, a change from the traditional ad-hoc exchange mechanism to an interoperable exchange has become one of the top priorities in the vision of Information and Communication Technology (ICT) implementation in the construction sector [Zarli et al. (2003)]. By seamlessly using electronic engineered files generated during planning, design and construction phases, a significant amount of efforts can be saved as assets are managed in order to provide superior results.

One of the earlier approaches to addressing the interoperability issue in the construction sector is the development of open data standards using Object-Oriented Modeling (OOM) techniques or EXtensible Markup Language (XML). Examples of those standards include Industry Foundation Classes (IFC) for the building sector and LandXML for the civil sector. Although these common standards consist of rich lists of concepts covering a wide range of phases and disciplines throughout the life cycle of a project, they are still insufficient to facilitate efficient data exchange [Froese (2003); East et al. (2012)]. One of the primary drawbacks of this approach is the lack of formal definitions of conceptualizations [Niknam and Karshenas (2014);

Yang and Zhang (2006); Venugopal et al. (2012b)]. This limitation is likely to lead to ambiguity and semantic inconsistency between the data creator and the receiver. Moreover, the lack of explicit presentation of relationships in a complex set of concepts imposes big challenges on the end user since they must have a deep understanding about the data schema in order to correctly extract desired data.

Recently, ontology has emerged as a solution to the issue of poor semantics in the existing open data standards. An ontology is an explicit formalization of a conceptualization which reflects several parts of the world [Gruber (1995)]. Under the view of data modeling, ontology is regarded as an abstract model consisting of formal definitions of classes and relationships among them. The implementation of this approach in data modeling has been accelerated by the availability of semantics supported modeling languages such as Ontology Web Language (OWL) [McGuinness and Van Harmelen (2004)] and Resource Description Framework (RDF) [Manola et al. (2014)] which are both developed by the World Wide Web Consortium (W3C). While OWL is meant to support modeling of classes, attributes and relationships, RDF offers a platform for describing individual metadata instances. Various authors have employed OWL and RDF to restructure IFC classes in the building sector, such as [Beetz et al. (2009); Karshenas and Niknam (2013); Zhang and Issa (2011)], whereas few research implementing these technologies have been carried out in the highway sector. Additionally, current ontology related research in the highway sector is mainly for knowledge management purposes. There is a lack of research that implements these technologies to formalize highway specific data elements for digital data exchange throughout the asset life cycle.

This paper presents an analysis of how an ontology based exchange mechanism can facilitate the interlinking of disparate and heterogeneous life-cycle data spaces so that digital data generated in upstream phases can be fully reused in asset management. Specifically, three domain ontologies and one merged ontology were developed using OWL to formulate the local conceptualizations and interrelationships involved in the design, construction and condition survey business processes. These ontologies are the crucial components of the mechanism as they provide sets of vocabularies for the translation of data instances from proprietary formats to the format of RDF triples. A prototype system was also built on the Jena API in Java environment

to support data translating, querying and information reasoning. A use case was finally applied and analyzed to demonstrate the success of the proposed exchange mechanism in facilitating semantic interoperability between applications involved in a highway project.

The paper is organized as follows. This section provides the background of the topic and rationale for the research. Section 2.2 presents the state of the art regarding solutions to the interoperability problem. Section 2.3 discusses the overall architecture of the semantic exchange framework. Section 2.4, 2.5, and 2.6 respectively explain the development of ontologies, data translator protocols and information extraction mechanism. Section 2.7 shows the results of the validation test. The final section summaries the main findings of the research and discusses the limitations and potential future works.

2.2 Literature review

2.2.1 A brief introduction to data interoperability

Data interoperability is defined as the ability of heterogeneous sources to communicate each other [Wegner (1996)] so that data generated from one platform can be sharable and fully reused. Research efforts to address the interoperability problem can be classified into: syntax and semantic levels [Sheth (1999)]. While the generation of syntactic interoperability aims to handle the mismatch between data formats, the semantic generation is to ensure the meanings and perspectives of data are precisely and unambiguously translated.

In attempts to address the syntactic interoperability issue, a variety of open data modeling languages have been developed. These languages offer common platforms for structuring abstract data models. Examples of these standards include STEP (also known as ISO 10303-11), Unified Modeling Language (UML) and eXentensible Markup Language (XML). Since these modelling standards are purely limited to syntax and structure, relations among data elements which provide context for the data are not explicitly represented. The lack of declarative semantics imposes big challenges on data exchange between disparate sources as they may use different sets of vocabularies. Exchanging of data relying on a common format would be straightforward if participants in each transaction have approval of mapping rules [Heflin and Hendler (2000)].

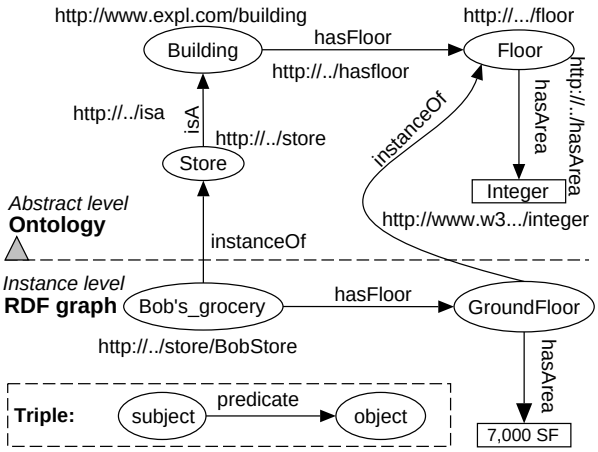

Figure 2.1: An example of ontology and RDF structure

But establishing and managing such a standard for data integration of enormous numbers of distinct sources in the global level is challenging and time consuming.

Semantics is the next generation of interoperability research. Ontology based methods have been widely studied and demonstrated as an effective solution for achieving semantic interoperability. From the database point of view, ontology, as illustrated in Figure 2.1, serves as an abstract schema for describing data instances in RDF format. An ontology consists of a set of nodes representing real-world concepts (classes or entities) and edges representing concept attributes (literal edges) or relations among concepts (object edges). RDF uses the triple structure which mimics the structure of a simple sentence to present resources (things, concepts) [Heath and Bizer (2011)]. Each triple comprises three elements including: (1) subject, (2) predicate and (3) object. To allow for interaction in the global network, unification resource identifier (URI) is used to identify a concept, relation or resource.

2.2.2 Open standard based exchange mechanism

A variety of research efforts have been made for the last two decades to establish open data standards for the highway industry. Most of the existing standards were developed adopting XML technique. LandXML [landxml.org (2017)], a result of early international collaboration efforts in facilitating interoperability in the civil industry, covers the following main groups

of data: survey data, ground model, parcel map, alignment, roadway and pipe network. As an effort to improve LandXML and propose a new standard specialized for the transportation industry, TransXML (NCHRP Project 20-64) project was chartered by the US National Cooperative Highway Research Program. TransXML focused on 4 business areas: survey/road design, construction/materials, bridge structures, and transportation safety [Scarponcini (2006)]. Of these domains, survey and geometric roadway classes are mainly derived from Land XML and are included suggestions for improvement [Ziering et al. (2007)]. But, similar to LandXML, the domains of pavement design and asset management have not been exploited yet.

In addition to the XML based standards, several extensions of IFC for road have been developed for a variety of purposes. Shen et al. (2014) developed an IFC model for highway projects; based on this structured data, a 3D model was also proposed for visualization purpose. Kim et al. (2014) developed another roadway model which focuses on embankment and subgrade classes to support automatic extraction of fill and cut quantity. In an attempt to enhance data exchange between structural engineers and designers in road structures (e.g. bridges, tunnels), Lee and Kim (2011) introduced a data model with the integration of structural components. In spite of these considerable research efforts, existing highway data standards still lack non-geometric information. As this information is crucial to asset management, it should be involved in further development of open data standards.

Open data standards consist of rich sets of data elements across many disciplines and phases. But, for a specific data exchange scenario, only a subset of instances rather than a whole set is required. Therefore the open data standards alone are insufficient to fully facilitate the seamless data exchange [Froese (2003); East et al. (2012)].

In order to leverage the use of neutral data formats in practices, several Model View Definition (MVD) have been developed. MVD, as defined by buildingSMART [buildingSMART (2016b)], is a subset of the IFC schema (entities, attributes and relations) that is required for a specific data exchange transaction. Several MVD specifications have been released by buildingSMART such as Coordination View, Structural Analysis View and Facility Management View (Cobie). The concept of MVD has also been adopted by the infrastructure sector recently. The VTT Technical Research Center of Finland has proposed the Inframodel that will be the

Finish national application specification for subsets of LandXML schema. The latest version Inframodel 3, released in 2013 for public review, provides model views for several infrastructure projects including roadways, railways, waterways and area planning.

The neutral format and MVD-based exchange mechanism has a significant drawback related to re-usability. A project funded by NIST concluded that the current practice of writing translators for data exchange which is on a case-by-case basis is not efficient and involves much redundancy and overhead efforts [Venugopal et al. (2012b)]. In order to support the extraction of partial models, significant efforts are needed for developing model view definitions which take years to be completed [Eastman (2012)]. As a huge number of business processes is involved through the project life cycle, such a number of MVDs is required. Moreover, since business processes are dynamic and tend to change over time, data requirements would consequently change. For this reason, more efforts would be required to tailor existing model views when the defined subsets of data become not sufficient, redundant or unsuitable to new business processes. Hence current MVD based exchange method needs to be transitioned from the ad-hoc manner to a more rigorous methodology [Eastman (2012)].

2.2.3 Ontology based exchange mechanism

Studies implementing ontology for the construction industry started in the early 2000s and this technology has gained an increasing attention from worldwide researchers.

In the building sector, a large number of ontology related studies have been conducted. Ontology was initially applied to formalize the construction knowledge. e-COGNOS [Lima et al. (2005)] which is one of the pioneering construction domain ontologies was developed to support construction-specific knowledge management. The ontology describes the top layer of construction knowledge which consists of four key elements including actors, resources, processes, and products and their relationships. Recently, ontology related research efforts focus more on describing building information. The research by Beetz et al. (2009), Pauwels et al. (2011), and Zhang and Issa (2013) developed ontologies of building information and proposed methods for converting IFC to RDF. Beside these broad topics, ontology has been implemented for such specific applications as cost estimation [Niknam and Karshenas (2013); Karshenas and Niknam

(2013)], cost related risk analysis Fagin et al. (2005), extracting construction features from the design models [Nepal et al. (2013)] and building facility management [Curry et al. (2013)].

The infrastructure sector is lagging behind the building sector in this research area. A few domain ontologies for the infrastructure industry have been introduced, for instance urban utilities [Osman (2007)] and highway projects [El-Diraby and Kashif (2005)]. However, these ontologies mainly target the knowledge representation purpose. Since data exchange requires deep levels of detail with a focus on data elements, the current ontologies are insufficient to allow for effective communications between software applications. Thus, there is a need for the development of such an ontology-based information model for highway projects.

2.3 Life cycle exchange mechanism

2.3.1 Motivating scenario

Applying proper maintenance activities at the right time would effectively extend the service life of a pavement asset [Hicks et al. (1999)]. Hence the process of analyzing maintenance and rehabilitation needs becomes one of the typical tasks of pavement management programs. Pavement treatments could be categorized into the following levels: needs nothing, preventive maintenance, light rehabilitation, medium rehabilitation and heavy rehabilitation (or reconstruction) [Wang et al. (2003); AASHTO (2001)]. Preventive maintenance activities are normally applied in the early years of the operation phase to address minor distresses, and to extend the cycle of major rehabilitations. Once preventive levels are ineffective, rehabilitation activities which include structural and operational treatments are considered.

In this research, the preventive treatment selection process for flexible pavements was selected as the motivating scenario. The selection process is a comprehensive analysis of multiple factors including distress type (rutting, cracking, bleeding, raveling, etc.), climate, cost, pavement age, pavement type, traffic volume, expected life, constructability, etc. [Hicks et al. (1999)]. Since these data are generated from many phases and disciplines through the asset life cycle, data integration becomes a critical task for evaluating treatment alternatives. To improve this decision making process, this research developed a conceptual framework and an implemen-

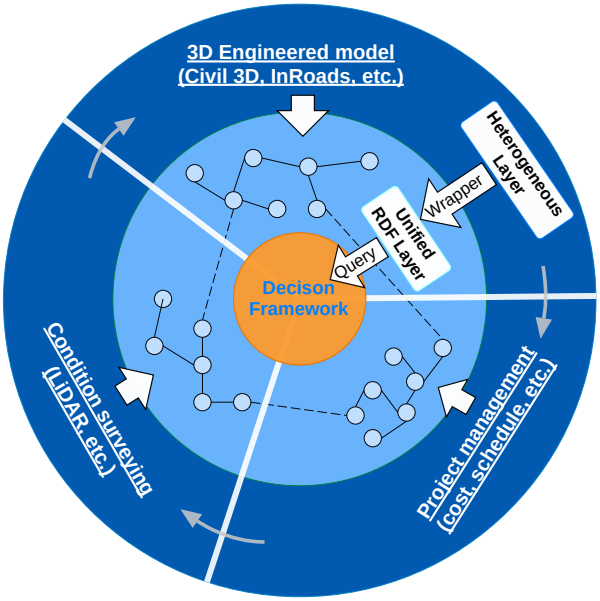

Figure 2.2: Life-cycle data exchange mechanism

tation prototype that supports integration and real-time access to life-cycle data sources and assists instant extraction of information about feasible treatment options.

2.3.2 Data exchange architecture

The proposed life-cycle data exchange mechanism to support asset management functions is shown in Figure 2.2. The ultimate goal of this framework is to transform heterogeneous data sources through the asset life cycle into meaningful information for the end user. This data flow is comprised of three stages, as follows.

2.3.2.1 Stage 1 - Data transformation

The first stage aims to transfer data from the heterogeneous layer (outer data space) to the homogeneous layer (inner data space). The outer layer involves data generated from multiple sources (e.g. 3D engineered models, project management tools, condition surveying technologies) which are required for asset management purposes. The translation of data from proprietary formats into mergeable RDF graphs in the inner layer is responsible by three data

wrappers. A data wrapper is a means to restructure a set of data into a new format. The key components of a data wrapper include a domain ontology and a set of mapping rules. The development of domain ontologies and data wrappers are presented in Section 2.4 and 2.5 respectively.

2.3.2.2 Stage 2 - Data inter-connection

In this stage, the separate RDF graphs of data that result from the previous stage are inter-connected together to create a unique data space. The interconnection network is created by adding external relationships between data elements across separate data graphs. An inter-relationship can be either an explicit property relation (e.g., A *consist* B) or an implicit inferring/reasoning rule (e.g., *If* A=a, *then* B=b). In Figure 2.2, local relations (between data elements generated by a single phase/stakeholder) are represented as the solid lines, and global relations (between phases/stakeholders) are represented as the dashed lines. With these global relations, local data elements and relations are fully visible and are readily available for external users. The details of this interlinking process will be discussed in Section 2.6.

2.3.2.3 Stage 3 - Data query and information reasoning

The final stage of the data flow is the transfer of a required part of the inter-connected data space to the core layer which contains the treatment selection framework. The data extraction is performed using the graph-supported query language SPARQL. There are two ways in which the asset manager can used the extracted data. The decision making framework can be utilized to analyze these data to determine feasible treatments. Alternatively, the framework can be translated into logic statements and embedded directly into the unified data graph using SWRL reasoning language. One advantage of the second method is that the inferred information about feasible treatments can be constructed as a new data graph and linked to other data graphs. Section 2.6 will presents more details about these data and information extraction processes.

2.4 Ontology development

Ontology is crucial to the conversion of a proprietary format into RDF format as it provides vocabularies of a domain. To support the development of data wrappers, this research developed three domain ontologies including: (1) design product, (2) construction event, and (3) condition survey event. In addition, a merged ontology was also developed to guide the process of merging distributed RDF graphs. This section, first presents the adopted methodology for ontology development, followed by detailed descriptions of the proposed ontologies.

2.4.1 Ontology development methodology

Ontology development methodologies have been suggested by several authors such as Grüninger and Fox (1995); Uschold and Gruninger (1996); Noy and McGuinness (2001). Although there is variation among these methods, they all include the following two key steps: identifying motivating scenarios, and determining a list of domain concepts and relations among them. The ontology development framework deployed in this study was derived from an extensive review of those methods and includes the following steps.

1. Determine the domain, scope and use cases of the ontology. This step aimed to: (1) determine the domain of knowledge; (2) identify the use case purposes/objectives of the ontology which will decide the ontology scope; and (3) determine the user and operator of the ontology.

2. Enumerate important terms in the ontology. In this step, a list of all terms involved in the domain area and suitable for the motivating scenario was developed. The development of this list was supported by answering the following questions. What are the things involved? What are the properties/aspects of those things that should be included for the objectives identified in the previous step?

3. Define class hierarchy. This step aimed to develop a hierarchy of classes based on the set of concepts determined in the previous step. This study employed the top-down development

process to build up the class hierarchy. The most general concepts were first identified and were then subdivided into specialized concepts.

4. Define class properties. This step was to provide detailed information about the defined concepts. For example, thickness and material are two properties should be attached to the *pavement layer* class.

5. Define domain and range of the properties. A property of a ontology class is presented as a directional edge going from the class to another class or a data type node. The class at which an edge begins is called *domain*, and the target node is called *range* that defines allowed values for the property. Class properties can be classified into literal and object properties depending on the type of *range*. For a literal property, its *range* is a literal data type, like string, number, boolean and date. The *range* of an object property is a class. In the example shown in Figure 2.1, the class *Building* is the *domain*, and the class *Floor* is the *range* of the object property *hasFloor*.

The objective of step 1 can be achieved by developing a set of competency questions representing for the queries that the ontology must be able to answer in the motivating scenario [Grüninger and Fox (1995)]. Table 2.1 shows the list of competence questions that arises when selecting feasible pavement treatment using the framework proposed by Zaghloul et al. (2006). Based on these questions, the scope of the ontology for the proposed exchange framework was identified to include the following three domain areas: (1) design product, (2) construction event and (3) condition survey event. The OWL ontologies proposed in this research were developed using the Jena API embedded in a Java program. The details of these domain ontologies as the results of step 2 to 5 are presented in the sections below.

2.4.2 Design product ontology

Figure 2.3 shows the proposed design product ontology for road routes. Based on the competence questions, The *Alignment* and *Pavement* concepts were considered sufficient for defining the *Route* concept. These basic concepts were then further defined by adding other associated concepts. The definition of the *Alignment* concept was derived from LandXML 1.2.

Table 2.1: Competency questions for the pavement selection process

Question	Answer examples
Which route does the section belong to?	I20
Where is the pavement section located?	Mile post 29.8
Which is the latest year that the pavement was resurfaced?	2005
What type of seal coat does the pavement have?	None, chip seal
How many layers does the pavement include?	3
What type of material is the pavement layer made of?	asphalt, stone
How thick is the pavement layer?	8 inch
How severe is the distress?	moderate raveling

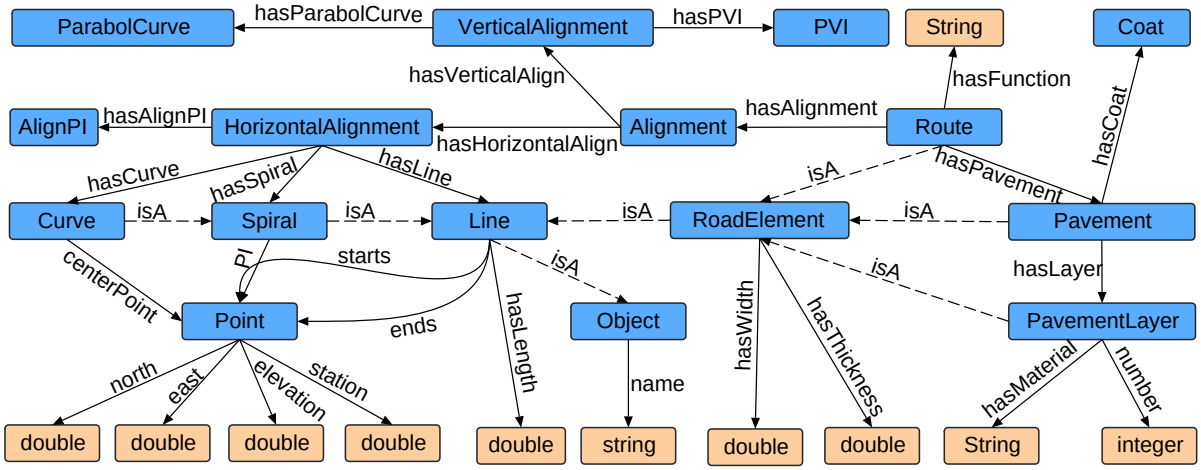

Figure 2.3: Design product ontology

One suggestion of this paper is the inclusion of two pavement related classes which are *Pavement* and *PavementLayer*.

Beyond the real physical concepts (e.g. pavement, alignment), this research constructed several superclasses, for instance *Object* and *RoadElement* which are composed of common attributes of real-world physical concepts. By simply applying the inheritance relationship, the properties of a subclass can be derived from its superclass. The inheritance relationship between two concepts is defined by using the *subClassOf* property (isA-dashed arrows) going from the child to the parent concept. Using the inheritance feature, the proposed design product ontology eliminates the issue of replication of attributes which occurs in the LandXML schema.

The following is a part of the design product ontology in Turtle format built upon the Jena API.

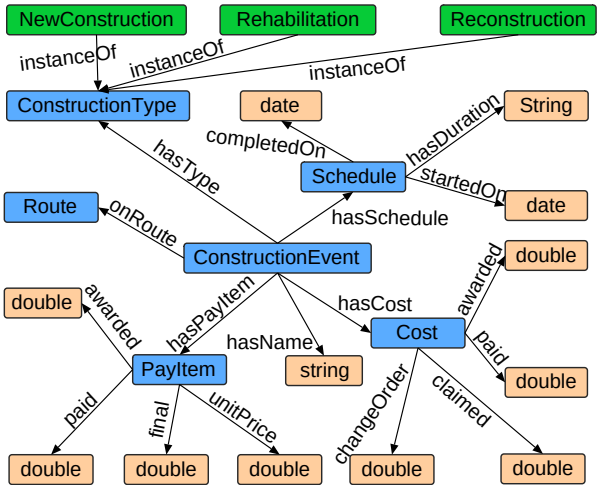

Figure 2.4: Construction event ontology

```
@prefix owl:    <http://www.w3.org/2002/07/owl#> .
@prefix rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix design: <http:example.com/Route/design/> .
...
design:Route  a           owl:Class;
rdfs:subClassOf  design:RoadElement. --This defines a node
...
design:hasAlignment  a  owl:ObjectProperty;
rdfs:domain  design:Route ;
rdfs:range   design:Alignment. --This defines an edge
design:Alignment  a       owl:Class;
rdfs:subClassOf  design:RoadElement.
....
design:hasPavement  a  owl:ObjectProperty ;
rdfs:domain  design:Route ;
rdfs:range   design:Pavement .
...
```

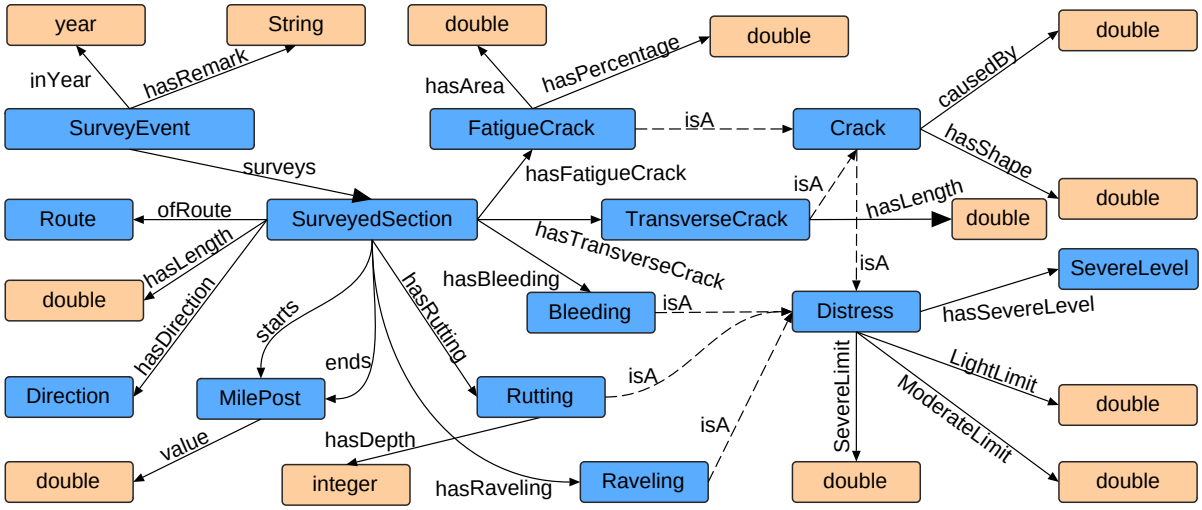

Figure 2.5: Condition ontology

2.4.3 Construction event ontology

As-built plans are used in the construction industry to represent project state at the completion of construction phase. In pavement projects, as-built plans include the following key information sheets: pay items, physical design features (drainage, typical section, profile, etc.), quantities, soil survey, and traffic control plans [Florida Department of Transportation (2014)]. Of these data, physical as-built data mainly relies on as-designed drawings and change orders that occur during the construction stage. Non-physical data such as actual quantity, schedule, and cost can be obtained from project monitoring systems. The proposed ontology for construction events, as shown in Figure 2.4, is limited to non-physical data and is mainly based on TransXML. In details, *ConstructionEvent* is the root concept that represents a construction event (new construction, reconstruction or rehabilitation). This concept is defined thanks to the relations with the following five associated concepts: *ConstructionType*, *Schedule*, *Cost*, *PayItem* and *Route*. These relevant concepts are broken down into subitems (awarded, paid, etc.) so that the ontology can best reflect execution planning and monitoring systems.

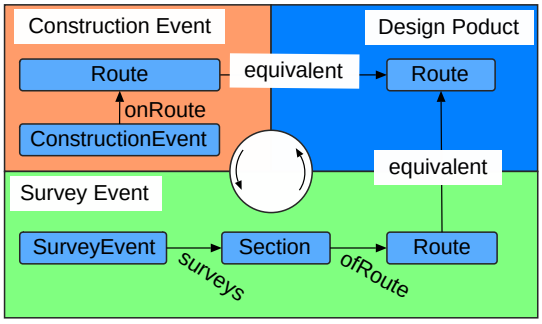

Figure 2.6: Merged life-cycle highway ontology

2.4.4 Condition survey ontology

Figure 2.5 shows the ontology formalizing the concepts related to pavement condition surveys. The classes involved in this ontology were built on the HPMS condition data elements developed by the Federal Highway Administration (FHWA) [U.S. DOT (2014)]. The head class of this ontology is *SurveyEvent*. This class relates to *SurveyedSection* which contains all data relevant to the survey event. The properties of the *SurveyedSection* class can be specialized into two groups. The first group, including the *Route, Direction* and *MilePost* represents inventory information. The second group contains distress related concepts such as *Crack, Rutting, Raveling* and *Bleeding*. Similarly to the design ontology, several superclasses (e.g. *Crack* and *Distress*) were defined in the condition ontology to utilize the advantages of the inheritance feature.

2.4.5 Merged ontology

Domain ontologies provide only data architectures and semantic formalizations for diverse data sources. Thus, in order for the instances of these isolated data islands to be properly interlinked into a unified data space, inter-linked relationships are needed to be formally defined. In this research, the local ontologies were merged by matching synonymous concepts using the *equivalentClass* property. The mapping process could be automated by algorithms that can automatically identify entities having contextual equivalence. However, the state of the art regarding automated ontology learning techniques are not yet been able to support a full and

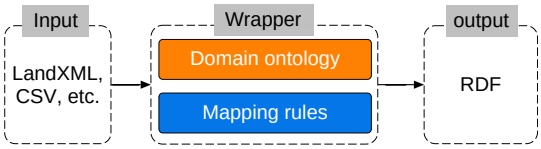

Figure 2.7: Wrapper architect

exact merging. In order to assure the accuracy of the mappings, this research used a manual method to define semantic equivalence among entities from the proposed life-cycle ontologies. Figure 2.6 illustrates the concept mapping. The *Route* concepts in the construction and condition ontologies are linked to the *Route* in the design ontology through the *equivalentClass* property. Based on this merged ontology, any *Route* instances generated by engineers can be legally linked to data graphs in the downstream phases.

2.5 Data wrapper development

To support the merging of data from isolated sources, these data are required to be converted into the common and linkable format of RDF. Such a number of wrappers are required to translate data from different application platforms into RDF format. A wrapper, as shown in Figure 2.7, is composed of two components: a domain ontology and a set of mapping rules. In the data translation process, ontology serves as the source of vocabularies, and mapping rules define semantic equivalence between terms of the source and target languages.

In this research, three wrappers were developed to translate design data (in LandXML format), construction and condition data which are usually in relational tables to RDF format. Sections below present the development of these wrappers.

2.5.1 LandXML to RDF

LandXML is an open standard which describes a wide range of design data of civil projects in XML format. XML data is structured in the tree format where data nodes are organized in a hierarchical structure (parent-child relationships) (see Figure 2.8). In this structure, a child node in the lower level is nested to one and only one parent node in the upper level. RDF is a

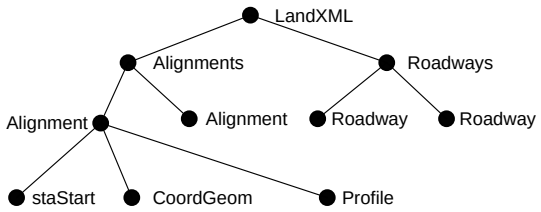

Figure 2.8: Partial LandXML tree

graph structure which also consists of vertices and edges. But, the relation between two vertices in a graph is more flexible. Two nodes can be connected by any meaningful edges rather than restricted to the nested relation. The following rules are proposed for translating LandXML design data into RDF graphs.

- For each node (in all levels) in a Landxml dataset, one node is produced in the RDF graph; the mappings are based on the data label mappings presented in the Table 2.2.
- All literal attributes of a Landxml node are derived to create RDF literal edges.
- For each nested relationship in Landxml, one edge is created to connect corresponding nodes in the RDF graph.

2.5.2 Relational data to RDF

Although the data generated from construction and condition events are both usually stored in the tabular format, two separate wrappers are required because of the difference in vocabulary sets. As the development of these translators are analogous, this paper discusses only the translation of the condition table to RDF format.

Table 2.2: Landxml to RDF mapping

Landxml name space	**RDF node**
Roadway	Route
Alignments	N/A
Alignment	Alignment
CoordGeom	HorizontalAlignment
Profile	VerticalAlignment

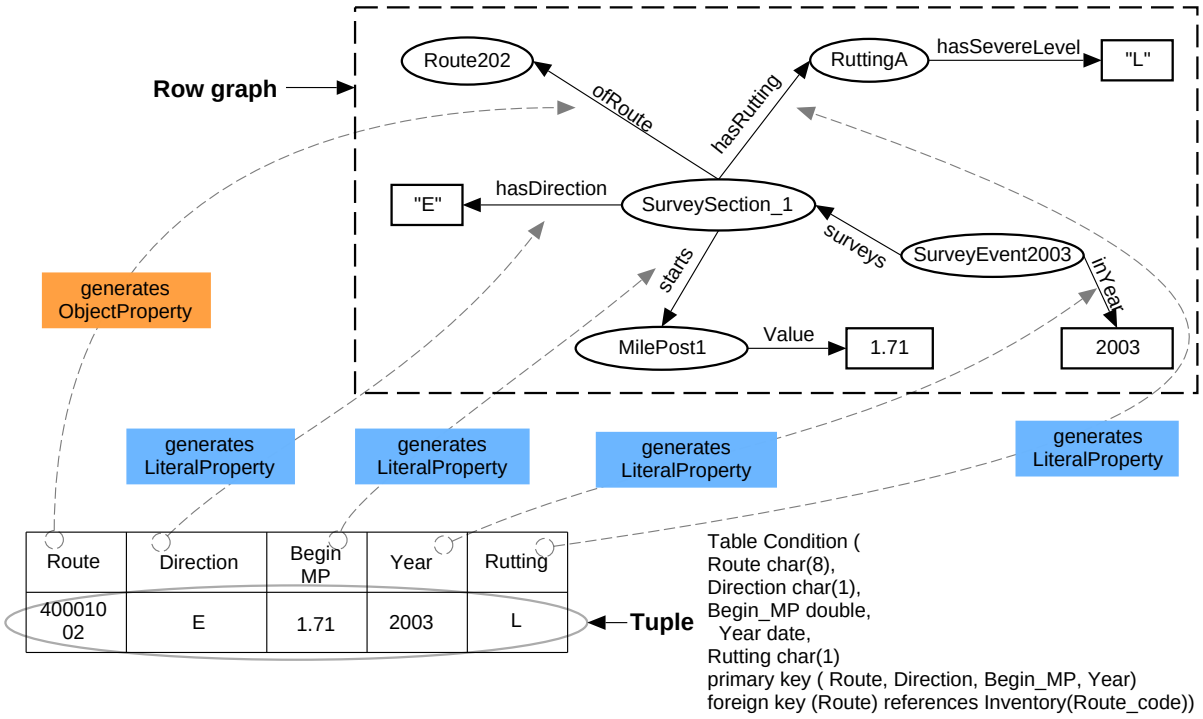

Figure 2.9: Translation from table to RDF graph

This study employed the transformation rules (see Table 2.3) recommended by W3C [Arenas et al. (2012)] to develop a wrapper for translating relational data to RDF. The application of these rules to the pavement condition table is illustrated in Figure 2.9. In details, each row (tuple) of the table generates a *row graph* consisting of a set of triples. The union of these row graphs generates a *table graph*. The process of converting a tuple to a *row graph* is initiated by creating a *row node* representing the tuple, followed by adding triples to the *row node*. The predicates of these triples are based on the mapping rules between the table column names and the properties in the condition ontology, as shown in Figure 2.9. If the column field doesn't serves as a foreign key, it's corresponding predicate is a literal property; otherwise, the corresponding predicate is a object property of which the *range* is a row node in another table graph. The last information needs to be preserved is uniqueness which is defined by the primary key. In the condition table, the primary key is composed of multiple attributes including *route, direction, year, begin_MP*. To maintain the uniqueness for the pavement condition row graph, the set of following triples in each row graph must be unique: $section - ofRoute \rightarrow route, section -$

$hasDirection \rightarrow string, surveyedsection \leftarrow surveys - SurveyEvent - inYear \rightarrow year$ and $section - startsAt \rightarrow MilePost - hasValue \rightarrow double$.

2.6 Interlinking data space and information extraction

The result of the data translation is a set of disparate RDF graphs. To fully support the decision making, these disparate resources are required to be connected to each other. Once these data are linked, query strategies and reasoning rules can be applied to extract specific information based on the objective of the decision making framework. Sections below present the data linking and information extraction process.

2.6.1 Linking diverse data spaces

The linking of data is performed by implementing the mappings provided in the merged ontology. Based on the guidance from the merged ontology; in order to link these data graphs, the URI of the *Route* instance in the design RDF file is assigned as the value of the *onRoute* and *ofRoute* properties in the construction and condition graphs.

2.6.2 Query over linked data space

This study employed the SPARQL [Prud'hommeaux and Seaborne (2008)] which is a query language for RDF graphs. SPARQL is a graph based language and the process of developing a query strategy is almost like a natural searching procedure, as follows:

Table 2.3: Relational table to RDF graph rules

Relational databases	**RDF graph**
table	table graph
tuple	row graph
column name	literal property
column as foreign key	object property

2.6.2.1 Step 1-Identity target nodes and constraint nodes

Target nodes are defined as the data to be queried, and constraints nodes are the data that are involved in query constraints. The target and constraint nodes can belong to multiple graphs. In the motivating scenario, the target data include: *type of seal coat* in the design product graph, *completion date* of the last construction in the construction event graph, and *distress severity* in the condition survey event graph. In asset management, roadways are commonly managed by sections. Asset manager might be interested in only several sections rather than the whole route; thus, the possible constraint nodes could be the beginning and ending mile posts.

2.6.2.2 Step 2-Develop informal node patterns

After the target and constraint nodes are determined, an informal node pattern will be constructed. A node pattern is a searching path going through all the target and constraint nodes identified in the first step. The illustration below shows the informal searching path for the completion year data of the last construction event on a surveyed section.

```
SurveyedSection-ofRoute-Route
ConstructionEvent-onRoute-Route
ConstructionEvent-hasSchedule-Schedule
Schedule-completedIn-xyear
```

2.6.2.3 Step 3-Formulate formal node patterns using SPARQL

The informal node pattern constructed in the previous step needs to be formalized using SPARQL syntax. The following presents the formal searching path containing all target and constraint nodes identified in Step 1.

```
FROM <rdf_condition.ttl>  --condition survey graph
FROM NAMED <rdf_design.ttl> --design product graph
FROM NAMED <rdf_construction.ttl> --construction event graph
WHERE {
?Survey condition:inYear ?SurveyedIn.
?Survey condition:surveying ?EvaluatedSection.
```

```
?EvaluatedSection condition:ofRoute ?Route.
?EvaluatedSection condition:hasRaveling ?Raveling.
?Raveling distress:hasSevereLevel ?RavSevereLevel.
?EvaluatedSection condition:hasBleeding ?Bleeding.
?Bleeding distress:hasSevereLevel ?BldSevereLevel.
?EvaluatedSection condition:startsAt ?BeginMP.
?EvaluatedSection condition:endsAt ?EndMP.
?EvaluatedSection condition:hasFatigueCrack ?FatigueCrack.
?FatigueCrack distress:hasSevereLevel distress:NoneLevel.
?FatigueCrack distress:hasPercent ?NoneCrackPercent.
GRAPH ?gd --design product graph
{?Route design:hasName ?RouteName.
?Route design:hasPavement ?Pavement.
?Pavement design:hasCoat ?Coat.
?Route design:constructedIn ?Project.}
GRAPH ?gc --construction event graph
{?Project construction:hasSchedule ?Schedule.
?Schedule construction:completedOn ?ConstructedIn.}}
```

In highway asset management, different types of data may be stored using different sectioning methods. Figure 2.10 illustrates the variety of methods used for inventory, distress and traffic data. The decision making process may also use a specific sectioning method different from those methods in the database. In this case, the following mile post constraint can be used to extract all the information related to the evaluated section.

$$StartMP_{q,Section} \leq MP^{ij}_{s,Section} \leq EndMP_{q,Section}$$

where $StartMP_{q,Section}$ and $EndMP_{q,Section}$ are respectively the start and end mile posts of the section to be queried. $MP^{ij}_{s,Section}$ is the mile post (either start or end) of the section i for the attribute j. The following example describes this constraint in SPARQL.

```
FILTER
(((?BeginMP >= 0.5) && (?BeginMP <=2.5)) || ((?EndMP >= 0.5) && (?EndMP
    <=2.5)))
```

Where && and || respectively represent for the *and* and *or* logic operations.

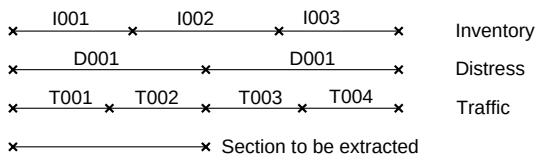

Figure 2.10: Query over database with multiple sectioning methods

-

2.6.2.4 Step 4-Construct query result as a new data graph

SPARQL offers three ways to form query results including: *SELECT, CONSTRUCT, and DESCRIBE*. Of these methods, the *CONSTRUCT* is able to allows for the construction of a new data graph based on the target variables in the node patterns. Once query results are formed under a new RDF graph, reasoning rules can be applied to extract further information. The SPARQL commands to build the RDF data graph of target data for an evaluated section is as follows.

```
CONSTRUCT
{treat:evaluation treat:hasInput ?EvaluatedSection.
?EvaluatedSection treat:ofRoute ?Route.
?EvaluatedSection treat:startsAt ?BeginMP.
?EvaluatedSection treat:endsAt ?EndMP.
?EvaluatedSection treat:constructedIn ?ConstructedIn.
?EvaluatedSection treat:surveyedIn ?SurveyedIn.
?EvaluatedSection treat:hasBleeding ?BldSevereLevel.
?EvaluatedSection treat:hasRaveling ?RavSevereLevel.
?EvaluatedSection treat:hasSealCoat ?Coat.
?EvaluatedSection treat:hasNoneCrackPercent ?NoneCrackPercent}
```

2.6.3 Information reasoning

To assist inferring information from an ontology or a data graph, W3C has developed the SWRL language [Horrocks et al. (2004)] to formulate reasoning rules. In this study, the proposed rules were developed using the Jena inference API which supports RDFS and OWL reasoners syntax. The generic form of a reasoning statement in SWRL language is as follows:

$$antecedent \rightarrow consequent$$

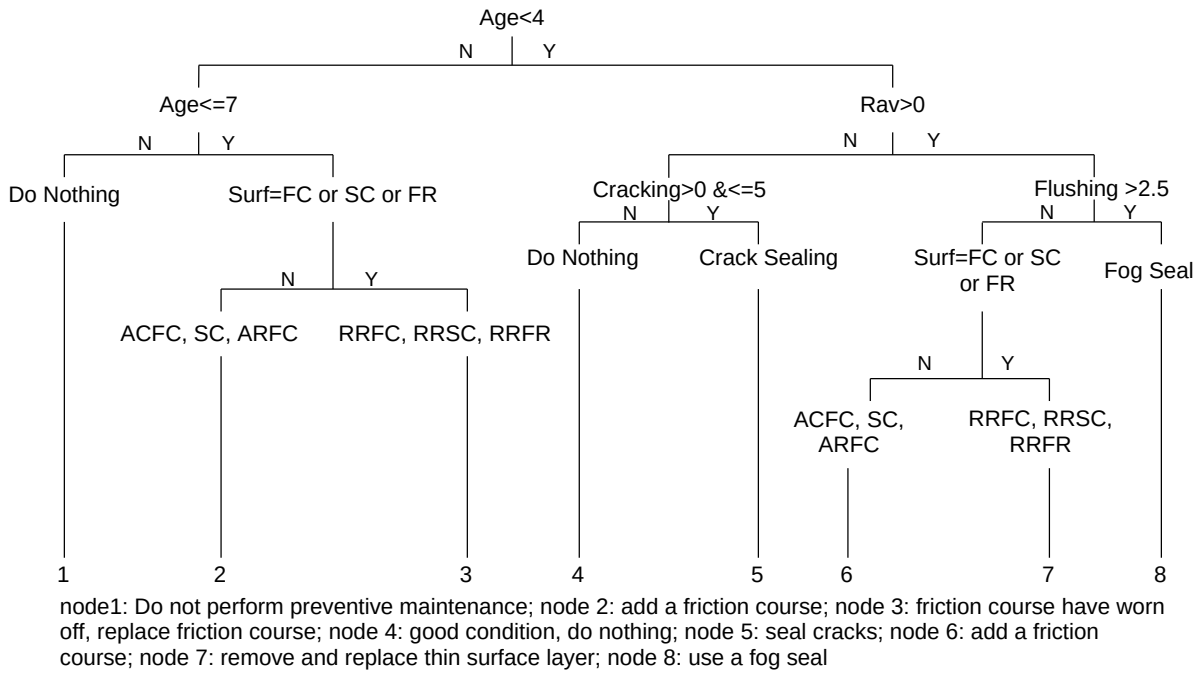

Figure 2.11: Preventive maintenance decision tree [Zaghloul et al. (2006)]

where, antecedent and consequent are conjunctions of atoms.

This research adopted the treatment selection framework developed by Zaghloul et al. (2006) (Figure 2.11). The following rule represents node 6 of the decision tree in SWRL language.

```
[Node 6:(?sect treat:constructedIn ?constYear) ^ (?sect treat:surveyedIn ?
    surYear) ^ difference(?constYear, ?surYear, ?age) ^ lessThan(?age, 4), (?
    sect treat:hasRaveling ?raveling) ^ notEqual(?raveling, distress:
    NoneLevel) ^ (?sect treat:hasBleeding ?bleeding) ^ equal(?bleeding,
    distress:NoneLevel) ^ (?sect treat:hasSealCoat ?coat) ^ equal(?g, design:
    noneCoat)
-> (?sect treat:needsTreatment treat:addFrictionCourse)]
```

2.7 Case study

2.7.1 Input data

The purpose of this experiment is to illustrate how the framework can be implemented in integrating isolated and proprietary data sources and from there utilizing the linked data space

for asset management businesses, specifically for the motivating use case. The case study does not aim to demonstrate the practical implementation readiness of the framework for a real project. To achieve the objective, a roadway project with example data sets regarding different stages (design, construction event, and condition survey event) was used. Firstly, the design data set was selected from the sample files provided by Landxml.org. A Landxml file includes common geometry data created by the designer such as alignment and roadway administration information. The data related to construction and condition survey activities were then assumed based on the design data. In this example, only as-built schedule information such as start date and end date were considered in the construction event data set. Regarding the condition data set, this type of data is normally consisted of distress information such as cracks, raveling, etc. In current state of practices, although condition data collection methods vary from manual to semi-automated or full-automated using laser scanning technology, the collected data are usually stored in tabular format. Hence, in this experiment, the example condition data set was assumed to be stored in CSV (Comma Separated Values) which is a common format for exchanging tabular data. A part of the input data from the three project phases are presented as follows:

```
--Route202 design data in LandXML format
<LandXML ...>
<Alignments>
<Alignment name="202cl" ...>
<CoordGeom>
<Line length="412.86000000" dir="293.253611111111">
<Start>1853.71000000 489.49000000</Start>
<End>1474.38813329 652.48785593</End>
</Line>... </CoordGeom> </Alignment> </Alignments>
<Roadways name="Route 202 Project" ...>
<Roadway name="Route 202" alignmentRefs="202cl" gradeModelRefs="Route 202"
    staStart="1000.000000"...>...
</Roadway> </Roadways>
...
</LandXML>
-------------------------
```

```
--Route202 project information in CSV format
Project_Name,Started_On,Completed_On
Route202,2009,2011
-------------------------
--Data resulted from a condition survey in CSV format
Route_Code,Dir,Begin_Mile_Post,Section_Length,None_Crack_Percent,
    Light_Crack_Percent,Moderate_Crack_Percent,Severe_Crack_Percent,Raveling,
    Bleeding
40601002,West,0,0.97,6,2,1,1,N,N
40601002,West,1.71,1.76,6,2,1,1,N,N
40601003,West,0,0.68,2,5,3,0,N,N
40601003,West,0.68,2.91,10,0,0,0,N,N
```

The experiment procedure was in accordance with the data flow specified in the proposed exchange mechanism. Specifically, the life-cycle data generated from the design works, construction events and condition surveys of the sample project was first translated to RDF graphs. These graphs were then connected to create an inter-linked data space. This step was followed by the process of querying necessary data and information reasoning. All of these steps were performed using a proposed Java prototype and the results are presented in the following section.

2.7.2 Results

Firstly, the separate and proprietary data files were converted into the RDF files using the developed conversion prototypes. These separate data files were then merged into a unique RDF file based on the linking guidance provided in the merged ontology. For example, by setting the URI of the *route202* node in the design RDF graph as the value of the *onRoute* edge in the construction event graph, these graphs become linked together. The following snippet is the description of the linked RDF graph in Turtle format.

```
--Design product RDF graph
@prefix design: <http:example.com/Route/design/> .
design:Route202  design:endsAt  design:Route202_endPoint ;
design:hasAlignment  design:Alignment0 ;
```

```
...
--Construction event RDF graph
@prefix construction: <http:example.com/Route/constructionProject/> .
construction:project202
construction:hasSchedule  construction:as_builtSchedule ;
construction:onRoute      <http:example.com/Route/design/Route202>  .--this
    is the URI of the route in the design graph
...
--Condition survey event RDF graph
@prefix distress: <http:example.com/Route/Distress/> .
@prefix condition: <http:example.com/Route/conditionSurvey/> .
condition:Section1  condition:endsAt  condition:Section1_endMP ;
condition:hasDirection     condition:West ;
...
condition:ofRoute          <http:example.com/Route/design/Route202>  ; --
    This is the URI of the route in the design graph
condition:startsAt         condition:Section1_startMP .
...
```

The linked data space built up above can be used by the asset manager in two ways: (1) querying and (2) reasoning. Data requirements can be extracted from the linked data space using SPARQL language (see Section 2.6.2). These extracted data then can serve as the input data in a decision making framework. Alternatively, the decision making framework can be translated into formal rules and integrated right into the extracted dataset as explained in Section 2.6.3. This feature of the proposed framework allows for the direct translation of isolated data into value information ready for use without middle steps of data processing and analyzing. In this case study, two methods were implemented. The following expression shows the extracted data graph resulted from a query statement and information reasoning process for the evaluated section from the mile post 0.5 to 2.5 of the *route202*.

```
@prefix treat: <http:example.com/Route/treatmentSelection/>.
...
condition:Section2  treat:constructedIn
"2011"^^xsd:long;
```

```
treat:endsAt  "3.47"^^xsd:double;
treat:hasBleeding  distress:NoneLevel;
treat:hasNoneCrackPercent  "6.0"^^xsd:double;
treat:hasRaveling  distress:NoneLevel;
treat:hasSealCoat  design:fogSeal;
treat:needsTreatment  treat:seal_cracks;
treat:ofRoute  design:Route202;
treat:startsAt  "1.71"^^xsd:double;
treat:surveyedIn  "2014"^^xsd:long.

condition:Section1  treat:constructedIn
"2011"^^xsd:long;
...
treat:evaluation  treat:hasInput  condition:Section2, condition:Section1.
```

The graph above consists of two pavement sections with their attributes regarding location (endsAt and startsAt), condition (hasBleeding, etc.), survey year (surveyedIn) and treatment needed (needsTreatment). Of these information, the value of the *needsTreatment* property is the result of the reasoning process; and the remaining information was derived from the query strategy.

2.8 Conclusions

Technology advances have noticeably changed highway project delivery and asset management from relying on 2D paper documents to n-D digital datasets. Since a highway project involves various phases, participants, disciplines which are using proprietary platforms, integrating data from the multiple resources for decision making becomes challenging. This paper proposes a novel framework that enables the interconnection of life-cycle digital data sources and translate into meaningful information for decision maker in highway asset management. To do this, this research developed three wrappers which are able to convert data generated in three main stages including design, construction and condition surveying into RDF format. The crucial components of these wrappers are the ontologies which provide class and relation definitions for the local domains. A merged ontology was developed to provide a standard guidance

on linking of isolated data spaces in the unified layer. SPARQL query and SWRL reasoning rules were then integrated to allow asset managers to extract required data or directly derive information from the linked data space.

An experiment on a sample project was conducted to illustrate how the proposed framework can assist the asset manager in integrating and deriving key information from fragmented and heterogeneous life-cycle data sets. Its life-cycle data, including design product (in LandXML format), construction event (in CSV format) and condition survey (in CSV format) were transformed into RDF graphs. To assist efficient data transition, the research developed three programs using Jena API in the Java environment. These RDF data were then linked together using the mapping rules provided in the merged ontology. The proposed SPARL based query strategy was applied to extract required data from the unified RDF data space. The extracted data was then interpreted using a proposed set of SWRL reasoning rules to infer feasible alternative treatments. The result shows that the proposed framework successfully interlinks life-cycle data spaces, extracts a subset of data and infers extra information.

The results of this research are expected to provide an effective and efficient means to facilitate seamless digital data exchange throughout the life cycle of a highway project. The proposed mechanism can be a potential solution for digitally handing over as-designed and as-built data to the operation phase and eliminate the costly and time consuming paper-based process. This approach can also leverage the effective concurrent collaboration between multiple partners not only within the same project but also with other construction sectors such as city planning, and other civil infrastructure (pipeline, railway, water supply, etc.). Once local data sets can be instantly accessed by other related disciplines, better decision making with holistic and long-term benefits would be achieved.

This research is limited to the integration of digital data from only three main phases including design, construction and survey monitoring. In order to enable a fully digital data exchange through the highway asset life cycle, future research is still needed to develop domain ontologies and wrappers for other business processes and platforms involved.

CHAPTER 3. NLP-BASED APPROACH TO SEMANTIC CLASSIFICATION OF HETEROGENEOUS TRANSPORTATION ASSET DATA TERMINOLOGY

A paper accepted for publication in *Journal of Computing in Civil Engineering, ASCE*, (2017)

Tuyen Le and H. David Jeong

Abstract

The inconsistency of data terminology has imposed big challenges on integrating transportation project data from distinct sources. Differences in meaning of data elements may lead to miscommunication between data senders and receivers. Semantic relations between terms in digital dictionaries, such as ontologies can enable the semantics of a data element to be transparent and unambiguous to computer systems. However, due to the lack of effective automated methods, identifying these relations is labor intensive and time consuming. This paper presents a novel integrated methodology that leverages multiple computational techniques to extract heterogeneous American-English data terms used in different highway agencies and their semantic relations from design manuals and other technical specifications. The proposed method implements Natural Language Processing (NLP) to detect data elements from text documents, and employs machine learning to determine the semantic relatedness among terms using their occurrence statistics in a corpus. The study also consists of developing an algorithm that classifies semantically related terms into three different lexical groups including synonymy, hyponymy and meronymy. The key merit in this technique is that the detection of semantic relations uses only linguistic information in texts and does not depend on other existing hand-coded semantic resources. A case study was undertaken that implemented the proposed method on a

16-million-word corpus of roadway design manuals to extract and classify roadway data items. The developed classifier was evaluated using a human-encoded test set and the results show an overall performance of 92.76% in precision and 81.02% recall.

3.1 Introduction

The implementation of advanced technologies such as 3D modeling, Geographic Information System (GIS), mobile devices, or LIDAR throughout the life cycle of a transportation asset has enabled data to be increasingly available in digital format. Due to the fragmented nature of the transportation industry, life-cycle data are generated individually by project partners and are archived in their own repositories [Harrison et al. (2016)]. The efficiency of data sharing and integration is crucial to enhance data reusability which will translate into reduced data re-creation, enhanced productivity, and better decision making. Addressing the interoperability issue has been widely recognized as a pressing need to allow for computer-to-computer data exchange and seamless integration of heterogeneous data from multiple sources [Karimi et al. (2003); Gallaher et al. (2004); Bittner et al. (2005)]. The transportation sector, however, has not yet successfully facilitated a high degree of interoperability [Lefler (2014)]. In order to reuse digital data, much laborious work is required for finding, verifying, and transforming facility and project information from a certain format to one another [Gallaher et al. (2004)].

Semantic interoperability is the highest level of interoperability that is concerned with the issue whereby two computer systems may not share a common understanding of the same data item [Heiler (1995)]. In the fragmented civil infrastructure domain, names of things might vary across data sources. *Polysemy* and *synonymy* are two major linguistic obstacles to the semantic integration of a multitude of data sources [Noy (2004)]. Polysemy refers to cases when a unique data term has distinct meanings in different contexts. The difference in meaning is due to the diversity and temporary of definitions, and the variation in data collection methods [Walton et al. (2015)]. For example, *rail* can mean a transportation mode or a barrier structure. Synonymy, in contrast, is associated with the disparity of names for the same data across systems. For instance, the data element of roadway type is named *functional system* in the Highway Performance Monitoring System (HPMS), but *functional class* in the Highway Safety Information

System (HSIS). Data integration in such a heterogeneous environment is highly problematic [Karimi et al. (2003)]. Polysemy may lead to a wrong match of two semantically different data items; and synonymy can cause a failure of aggregating similar elements. Explicitly specifying the semantic equivalence or relatedness between data terminologies becomes critical to proper integration of disparate data [Ouksel and Sheth (1999)].

Previous studies on semantic similarity and relatedness between data items lie in the development of data libraries, taxonomies and ontologies. A semantic resource specializes the meaning of terms through their lexical relations with each of other. Examples in this area include the Civil Engineering Thesaurus [Abuzir and Abuzir (2002)], the e-Cognos Ontology [Wetherill et al. (2002)], and the buildingSMART Data Dictionary [buildingSMART (2016a)]. As shown in the literature review, their coverages are still limited especially in the transportation sector in spite of years of efforts. This is because of the reliance on conventional methods which are labor-intensive and time consuming. To develop a knowledge base, developers are required to manually determine important terms and their relations by interviewing domain experts or examining technical documents. The shortage of such semantic resources has become a bottleneck for semantic integration. There is a need for an automated data classification method that will allow digital dictionaries to be quickly constructed for specific needs and to keep up with the growth of terms [Mounce et al. (2010)].

To fulfill that demand, this study aims to propose a novel linguistic approach for automatically classifying the semantic relations among heterogeneous data elements associated with a transportation asset. The study leverages Natural Language Processing (NLP) to extract key data items and their meanings by analyzing the statistical data of context words in technical documents. This process generates a vector space in which each point represents the semantics of a data item. The research also includes a new integrated classification algorithm that utilizes syntactic rules, cluster analysis, and word embedding to categorize related elements into three different lexical groups that are synonymy (similar-to), hyponymy (is-a), and meronymy (part-of). To demonstrate the success of the proposed method, the framework was implemented on a corpus of roadway design manuals. A Java package and several datasets resulting from the study can be found at https://github.com/tuyenbk/CeTermClassifier.

3.2 Background

3.2.1 Natural Language Processing (NLP)

NLP is a research area developing techniques that can be used to analyze and derive valuable information from natural languages like text and speech. Some of the major applications of NLP include language translation, information extraction, opinion mining [Cambria and White (2014)]. These applications are embodied by a rich set of NLP techniques ranging from syntactic processing such as Tokenization (breaking a sentence into individual tokens) [Webster and Kit (1992); Zhao and Kit (2011)], Part-of-Speech (POS) tagging (assigning tags: adjective, noun, verb, etc. to each token of a sentence) [Toutanova et al. (2003); Cunningham et al. (2002)], and Dependency parser (identifying relationships between linguistic units) [Chen and Manning (2014)], to the semantic level, for instance word sense disambiguation [Lesk (1986); Yarowsky (1995); Navigli (2009)]. NLP methods can be classified into two main groups: (1) rule-based and (2) machine-learning (ML) based methods. Rule-based systems, which rely solely on hand-coded syntax rules, are not able to fully cover all human rules [Marcus (1995)]; and their performance, therefore, is relatively low. Whereas, the ML-based approach is independent of languages and linguistic grammars [Costa-Jussa et al. (2012)] as patterns can be quickly learned from even un-annotated training examples. Thanks to its impressive out-performance, NLP research is shifting to statistical ML-based methods [Cambria and White (2014)].

3.2.2 Vector Representation of Word Semantics

Measuring semantic similarity, which is an important NLP-related research topic, aims at determining how much two linguistic units (e.g., words, phrases, sentences, concepts) are semantically alike. For example, a *railway* might be more similar to a *roadway* than to a *train*. The state-of-the-art methodology for this task can be divided into two categories that are (1) thesaurus-based methods and (2) vector space models (VSM) (also known as word embedding) [Harispe et al. (2013)]. The former approach relies on a hand-coded digital dictionary (e.g., WordNet) which formally structures terms in a network of semantic relations. In this method, the semantic similarity between a given pair of words can be measured based on the distance

between them in the hierarchical structure. The method is an ideal solution if digital dictionaries are available. However, digital dictionaries are typically hand-crafted; they are, therefore, not available to many domains [Kolb (2008)]. The latter technique assesses the meaning of words or phrases by analyzing their occurrence frequency in natural language text documents. VSM outperforms the dictionary-based method especially in terms of time saving as a semantic model can be automatically obtained from a text corpus and corpus collecting is much easier than manually constructing a digital dictionary [Turney and Pantel (2010)].

VSM estimates semantic similarity based on the *distributional model* which represents the meaning of a word through its context (co-occurring words) in a corpus [Erk (2012)]. The distributional model stands on the *distributional hypothesis* that states that two similar terms tend to occur in the same context [Harris (1954)]. The output of this approach is a vector space, in which each numeric vector represents a word in the vocabulary. The similarity between semantic units in this model can be represented by the Euclidean distance between the corresponding points [Erk (2012)].

The conventional method to construct a VSM is to use the 'word-context' matrix which shows how frequent a word is the context of one another in a given text corpus. These raw data of frequencies are used to estimate the co-occurrence probabilities. This statistical process results in a matrix in which each row is a vector representation. Pointwise Mutual Information (PMI) [Church and Hanks (1990)] or it's variant, Positive PMI (PPMI) is a popular method to calculate co-occurrence probabilities. A more advanced approach uses machine learning to train representation vectors. The two leading state-of-the-art ML based word embedding techniques are named Word2Vec and Glove. Word2Vec model [Mikolov et al. (2013)], which is a neural network model, learns vector representation of words from their surrounding words. Mikolov et al. (2013) proposed two opposite network architectures, including Continuous Bag-of-Words (CBOW) and skip-gram. CBOW predicts a word given a set of context words, whereas skip-gram aims to predict the context of a given word. The training objective of both models is to minimize the overall prediction error. Glove or Global Vectors [Pennington et al. (2014)] trains on the global co-occurrence matrix with the objective that the probability of co-occurrence between two words equals the dot product of their vector representations. There are conflicting

recommendations on the wining model in the literature. The authors of Glove argue that their model out-performs Word2Vec. However, a number of independent benchmarking experiments provide an opposite suggestion. For example, a comparative study by Levy et al. (2015) on the accuracy in various tasks and golden standards reveals that the skip-gram in Word2Vec is superior to Glove in most of the experiments, especially on similarity evaluation. The best precision of Skip-gram is .793, while Glove achieves the highest score of .725. The out-performance of Mikolov's models on the similarity task is confirmed in another benchmarking study [Hill et al. (2015)] where this model is also found as the winner.

The VSM approach has been progressively implemented in recent NLP related studies in the construction industry. Yalcinkaya and Singh (2015) utilized VSM to extract principle research topics related to BIM from a corpus of nearly 1,000 paper abstracts. This approach was also used for information retrieval to search for text documents [Lv and El-Gohary (2015)] or CAD documents [Hsu (2013)]. The increasing number of successful use cases in the construction industry has evidently demonstrated that the VSM method can be successfully implemented to tackle the issue of semantic interoperability in sharing digital data across the life cycle of a highway project.

3.2.3 Related Studies

A popular solution to semantic interoperability is to develop taxonomies, ontologies or other forms of digital dictionaries that can provide machine-readable definitions of domain concepts. A plethora of such semantic resources have been developed for the highway industry. However, conventional development methods require significant human efforts on knowledge retrieval, and ontology construction and validation. The pioneer in this line of research is the e-Cognos ontology [Wetherill et al. (2002); Lima et al. (2005)] which formulates the execution process of a construction project as an explicitly interactive network of the following principal concepts: Actors, Resources, Products, Processes and Technical Topics. The ontology developers of this project reviewed existing taxonomies (BS61000, UniClass, IFC) and construction specific documents, and interacted with the end users to identify relevant concepts and their semantic relations. Industry experts were invited to validate e-Cognos's concept names and relations.

Since the introduction of e-Cognos, a plenty of other ontologies have been built for various aspects of a highway project, for instance, construction taxonomy [El-Diraby et al. (2005)], freight ontology [Seedah et al. (2015a)], and the ontology of urban infrastructure products [Osman and Ei-Diraby (2006)]. These studies also relied on domain experts [El-Diraby et al. (2005); Osman and Ei-Diraby (2006)] or existing knowledge bases [Seedah et al. (2015a)] to construct their semantic products. The limitations on time and resources of the traditional knowledge-based methodology have created a bottleneck in semantic interoperability. In addition, existing ontologies primarily focus on concept description and neglect the heterogeneity of concept names. Therefore, there is a need to develop a data-driven method that can automate the process of formulating domain concepts and also incorporate term diversity into ontologies.

Another line of research on semantic interoperability targets at the heterogeneity of concept names rather than concept description. A few frameworks to assist developers in precisely mapping data labels from heterogeneous sources have been introduced for various construction sectors. In the building sector, buildingSMART proposed a novel framework, namely the International Framework for Dictionaries (IFD) or ISO 12006-3 for developing a multilingual data schema in which each concept can have multiple names in different languages. With IFD, the identity of a concept is defined by a Global Unique ID (GUID) instead of its name; hence an IFD-based exchange mechanism is able to avoid data mismatches due to name inconsistency [IFD Library Group (); Hezik (2008)]. The buildingSMART Data Dictionary (bSDD) [buildingSMART (2016a)] is the first digital library of building concepts organized in IFD format. Each concept in bSDD consists of a set of synonymy names not only in English but also in computer-coded languages (e.g., IFC) and in other human languages (e.g., French, Norwegian). Therefore, a complete bSDD would enable digital data in regardless of languages to be sharable and unambiguously reusable. Yet, its size remains limited as the identification of these sets of synonyms is laborious and time intensive. In the transportation sector, there has been a shortage of research efforts on the heterogeneity of data names at the database level until recently. Seedah et al. (2015b) proposed a role-based classification schema (RBCS) to classify data in freight databases. RBCS defines nine distinct groups of roles that are time (year, month), place (city name, population), commodity (liquid, value), link (roadway name, width), mode

(truck, rail), industry (company name, sales), event (accident, number of fatalities), human (officer, driver age), and unclassified. The authors argue that once data elements across separate databases are categorized using this standard system, it becomes easier for practitioners to identify the semantic relatedness between items. However, even if RBCS is successfully applied to all freight databases, much more effort is still needed to further specify the relation type (e.g., synonym, functional related) between two data elements in the same category.

In attempts to reduce laborious work on defining concepts, a few researchers have sought to propose semi-automated and automated methods for identifying semantic relations among technical terms. Abuzir and Abuzir (2002) developed the ThesWB system which utilizes hand-coded syntax patterns to detect lexical relations between civil engineering terms from HTML web pages. The performance of ThesWB was not reported, but it is not likely to be high since rule-based approaches are repeatedly criticized for not being able to capture all the variant ways to present relations among terms in natural language [Marcus (1995); Navigli and Velardi (2010)]. Rezgui (2007) suggested a more sophisticated approach that is based the statistics of word occurrence in domain text documents rather than predefined rules. This method implements Term Frequency-Inverse Document Frequency (TF-IDF) to evaluate the importance degree of a keyword to the examined domain. The method computes the relatedness between a given pair of important keywords using the 'Metric Clusters' measure which estimates the association based on the distance between them in the text. These potential relationships are then validated and categorized by domain experts. Since Rezgui's methodology detects relations between words occurring in the same sentence, equivalent terms which are used interchangeably could not be captured. In another study, Zhang and El-Gohary (2016) proposed a machine learning based methodology for identifying the semantic relation between a new concept and the existing IFC entities. This algorithm was reported to achieve an average precision of nearly 90 percent. The algorithm identifies potentially related concepts based on the pre-defined lexical relations provided in WordNet. Since WordNet is a generic lexicon that lacks concepts in many construction sectors including the civil infrastructure, this algorithm would not be scalable well on matching terms in those domains.

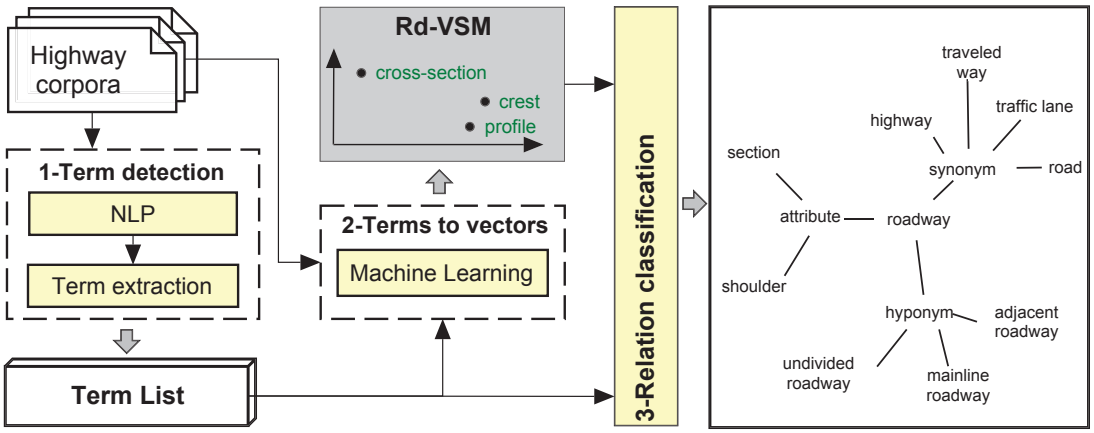

Figure 3.1: Overview of the proposed methodology.

As shown in the literature review, there are numerous research efforts in developing ontologies for the highway sector. However, existing ontologies are mainly hand-coded through the manual processes of knowledge acquisition and translation into a digital format. Relying on this traditional approach has created a bottleneck in facilitating semantic interoperability. A few efforts have been made to automate the process of constructing or extending existing semantic resources. The most rigorous methodology in the state-of-the-art is the one developed by Zhang and El-Gohary (2016) that has a high level of accuracy. One limitation of this algorithm is the reliance on a semantic resource; it, therefore, would not be well applicable to such domains as civil infrastructure and transportation which are beyond the vocabulary scope of existing lexical databases. Thus, it is essential to develop an automated method that can allow for fast development of domain lexicons and also reduce dependence on other existing semantic resources.

3.3 NLP-based Methodology to Classification of Heterogeneous Data Terms

The goal of this research is to propose an NLP-based methodology that can automate the process of extracting diverse data elements and their semantic relations from American-English technical guideline documents. As shown in Figure 3.1, the proposed method consists of three

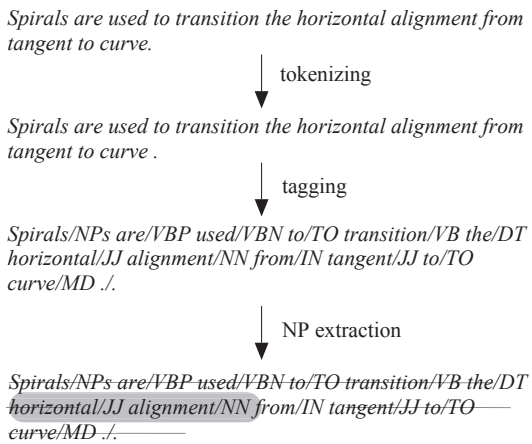

Figure 3.2: Linguistic processing procedure to detect NPs.

major stages that are to: (1) utilize NLP techniques to extract multi-word data items from a domain text corpus, (2) implement machine learning to generate vector representation of the extracted terms, and (3) design an algorithm integrating various linguistic patterns, clustering, and semantic vectors to detect the semantic relation between a given pair of terms. The below sections discuss these phases in detail.

3.3.1 Multi-word Data Element Extraction

Technical documents such as design manuals, guidelines, and specifications are great sources of domain data elements which occur as technical terms. Linguists argue that a technical term is either a noun (e.g., road, AADT) or a noun phrase (NP) (e.g., right of way, sight distance) that frequently occurs in domain text documents [Justeson and Katz (1995)]. The meaning of a multi-word term may not be directly interpreted from the meaning of its constituents; therefore, it must be treated as an individual word. As mentioned, a multi-word term must be an NP; thus, NPs are good multi-word term candidates. To detect this type of terms, the corpus is first scanned to search for NPs, followed by assessing their importance to the domain. The process of extracting multi-word terms is discussed in detail as follows.

Table 3.1: Term candidate filters.

Pattern	Examples
(Adj|N)*N	road, roadway shoulder, vertical alignment
(Adj|N)*N Prep (Adj|N)*N	right of way, type of roadway
Note: |, * respectively denote 'or', and 'zero or more'. Prep is a prepostion	

3.3.1.1 NP extraction

Figure 3.2 illustrates how NPs are extracted from a natural language sentence. This process includes the following steps.

i Word tokenizing: In this step, the text corpus is broken down into individual units (also called tokens) based on the OpenNLP Tokenizer. Tokenizing is to separate punctuation marks, for instance periods (.), commas (,), semicolons (;), parentheses, etc., from words. The tokenizer is capable of distinguishing between marks in acronyms (e.g., r.o.w., r/w) and punctuation symbols; this kind of words will remain in the corpus.

ii Part of Speed (POS) tagging: The purpose of this step is to determine the Part of Speech (POS) tag (e.g., NN-noun, JJ-adjective, VB-verb, etc.) for each unit of the tokenized corpus obtained from the previous step. A full set of POS tags can be found in the Penn Treebank [Marcus et al. (1993)].

iii Noun phrase detection: This phase aims to collect NPs using syntactic rules. Table 3.1 presents the employed patterns which are reformulated from the one suggested by Justeson and Katz (1995) for better human-readability. The tagged corpus is thoroughly scanned to collect sequences matching those patterns. This study assumes that sequences of more than 6 words are not likely to be a technical term; they, therefore, are automatically discarded. In addition, in order to reduce the discrimination between syntactic variants of the same term, the collected NPs need to be normalized. This study considers the following two types of syntactic variation.

- Type 1 - Plural forms, for example 'roadways' and 'roadway'. Stemming is a popular process to reduce words to their stems. Over and under-stemming are two common

errors. Over-stemming refers to the removal of true suffixes (e.g., 'divided highway' → 'divide highway'); under-stemming occurs when pre-defined rules fail to handle irregular forms for instance 'foot' - 'feet'. Despite the fact that, none of the existing algorithms can completely eliminate these errors, they are good enough to not degrade the overall performance of NLP applications [Jivani et al. (2011)]. This study implements the Pling stemmer [Suchanek et al. (2006)], which stems an English noun to its singular form, to normalize plural nouns in the corpus. One advantage of this algorithm is the utilization of both syntactic rules and dictionaries. Dictionaries are to verify the outcomes from purely pattern-based stemming and allow for the inclusion of irregular plural nouns; therefore, stemming errors can be reduced. Furthermore, as only nouns are impacted, mis-stemming on such terms as 'divided highway' can be prevented.

- Type 2 - Prepositional noun phrases, for example 'type of roadway' and 'roadway type'. In order to normalize this type of variation, the form with a preposition is converted into the non-preposition form by removing the preposition and reversing the order of the remaining portions. For example, 'type of roadway' will become 'roadway type'. However, blindly applying normalization will create unreal instances since not every prepositional NP is paraphrasable. 'Right of way' is one example of such non-paraphrasable NPs. Therefore, this study implements paraphrasing for only those NPs whose the reversed form also exists in the extracted list.

The instances obtained from the above process may include errors. To eliminate incorrectly extracted sequences, the following two discard criteria are used. First, a valid NP must not contain any 'minimal stop' word. The 'minimal stop' list consists of frequent words and phrases that carry obviously no meaning for a technical term, including determiners (e.g., another, any, particular), coordinating conjunctions (e.g., nor, or, and), comparative adjectives (e.g., largest, longest, best), and stop phrases (e.g., lack of, set of, kind of). The list is called 'minimal stop' list to distinguish it from the large stop list commonly used in NLP applications. The second constraint for filtering out 'bad' NPs is occurrence frequency. This study assumes that instances

that are not a randomly combined sequence appear at least twice in the corpus. Items that appear only once are eliminated. This hypothesis might be not applicable for a small corpus (let's say 10,000 words) as the frequency of true NPs tends to be low.

3.3.1.2 NP Ranking and Term Selection

Multi-word term definition varies between authors, and there is a lack of formal and widely accepted rules to determine if an NP is a multi-word term [Frantzi et al. (2000)]. There are a number of methods for estimating termhood (the degree that a linguistic unit is a domain-technical concept), such as TF-IDF [Sparck Jones (1972); Salton and Buckley (1988)], C-Value [Frantzi et al. (2000)], and Termex [Sclano and Velardi (2007)]. Of these methods, Termex outperforms others on the Wikipedia corpus, and C-Value is the best on the GENIA medical corpus [Zhang et al. (2008)]. One notable observation from these studies is that C-value is more suitable for term extraction from a domain corpus rather than a generic one. For this reason, C-value has been used in various studies in the biomedical field, for instance works conducted by Ananiadou et al. (2000), Lossio-Ventura et al. (2013), and Nenadić et al. (2002). Since the methodology proposed in this paper aims to extract data elements from highway guidelines and manuals which are domain specific documents, C-value would be the most suitable for the termhood determination task. The C-value measure, as formulated in Equation 3.1, suggests that the longer an NP is, the more likely that is a term; and the more frequently it appears in a domain corpus, the more likely it will be a domain term.

$$C-value(a) = \begin{cases} log_2|a| \cdot f(a), & \text{if a is not nested} \\ log_2|a|(f(a) - \frac{1}{P(T_a)} \sum_{b \in T_a} f(b)), & \text{otherwise} \end{cases} \tag{3.1}$$

Where:

a is a candidate noun phrase

|**a**| is the length of noun phrase a

f is the frequency of a in the corpus

Ta is the set of extracted noun phrases that contain a

P(Ta) is the size of Ta set.

The C-value measure is employed to compute termhood for every term candidate generated from the previous stage. This process results in a dataset of terms along with their C-value scores. These term candidates are ranked by C-value.

To automatically remove candidates that are unlikely to be a domain term, a C-value threshold can be used as an acceptance limit. However, choosing a proper absolute threshold is challenging as it typically depends on the corpus size. A high limit can help to significantly reduce 'bad' candidates, but real terms that appear at the bottom due to their low frequency will be excluded. Manual evaluation of the entire sorted list would avoid the removal of real terms with low C-values, but it might be too laborious especially for large corpora. To minimize both laborious work and the number of true terms wrongly discarded, this study adopts a relative cut-off policy proposed by Lopes and Vieira (2015) which is based on the optimal trade-off point between wrong discard of true domain terms and wrong inclusion of irrelevant ones. The policy suggests that the bottom 85% of the ranked list should be discarded.

3.3.2 Data Element Vector Space Model

This phase aims at converting the vocabulary of a domain corpus into a vector space that presents the semantics of a term as a vector. This study employs the unsupervised Word2Vec model [Mikolov et al. (2013)] to learn representation vectors. As discussed earlier in the Background section, Word2Vec and Glove are the two leading state-of-the-art word embedding techniques. Word2Vec is usually found to outperform Glove, despite the fact that there is a lack of conclusive evidence in the literature for the superiority of one to the other. Since the objective of this research is not to propose an optimized embedding method, we arbitrarily selected Word2Vec for the vector representation learning task in the proposed classifier.

In Word2Vec model, a training data point is corresponding to a target word and its context words in the corpus. The sentence below illustrates how surrounding words are captured using a context window which limits how many words to the left and to the right of the target word. In the example, the context of the term 'roadway' with the window size of 5 will be {bike, lane, width, on, a, with, no, curb, and, gutter}. Any context word that is in the stop list (a list that

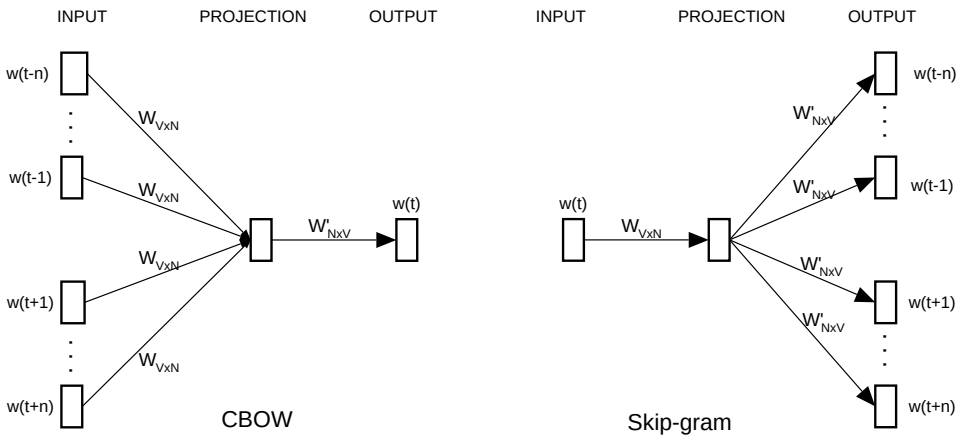

Figure 3.3: Word2Vec neural network structures.

contains frequent words in English such as 'a', 'an', and 'the' that have little meaning) will be neglected, and the context set becomes {bike, lane, width, curb, gutter}.

"*The minimum [bike lane width on a roadway with no curb and gutter] is 4 feet .*"

Prior to data collection, an additional step is needed to handle the issue related to multi-word terms. Since document scanning is on a word-by-word basis, the tokenized and stemmed corpus resulted from the NP extraction phase must be adjusted so that multi-word data elements can be treated as single words. To meet that requirement, white spaces within a multi-word term are replaced by minus signs (-) to connect its individual words into a single unit. For instance, 'vertical alignment' becomes 'vertical-alignment'.

This study trains vector representation using both CBOW and skip-gram network types of Word2Vec. Figure 3.3 illustrates these learning networks, where V and N respectively denote the size of the corpus vocabulary and the hidden layer. In CBOW, context words are at the input layer and target word is at the output layer; whereas skip-gram reverses the role of the data components. Word2Vec encodes a word as a 'one-hot' vector in which only one element at the index of the word in the vocabulary is set to one, and all other items are zero. For example, the one-hot vector of the k^{th} word in the vocabulary with the size of V will be $\{x_1 = 0, x_2 = 0, ..., x_k = 1, ...x_V = 0\}$. The outcome of this machine learning process is a set of N-dimensional representation vectors each of which is corresponding to a row in the learned parameter matrix $W_{N.V}$. The similarity between a pair of vectors represents the similarity in

Table 3.2: Skip-gram model parameters.

Parameter	Value
Frequency threshold	0-100
Hidden layer size	100-500
Context window size	5-15

context between their corresponding words, and can be measured by the angle between word representation vectors (Equation 3.2) or the distance between word points (Equation 3.3).

$$cosine\_similarity = \frac{A \cdot B}{||A||\,||B||} \tag{3.2}$$

$$dis\_similarity = \sqrt{(xA_1 - xB_1)^2 + (xA_2 - xB_2)^2 + ... + (xA_n - xB_n)^2} \tag{3.3}$$

Where: n is the vector dimension which is also the hidden layer size.

The learning model includes three major parameters that are *frequency threshold*, *hidden layer size* and *window size* (see Table 3.2). *Frequency threshold* is used in this phase to eliminate from the training data those input words that are unimportant to the domain. As discussed earlier in the NP extraction stage, low-frequency words are unlikely to be a technical term. Words with the occurrence below a threshold will be excluded from the input vocabulary. Radim (2014) suggests a frequency limit ranging from 0 to 100 depending on the corpus size, where 0 means to accepts everything. Setting this parameter high can enhance the accuracy, but many true technical terms would be out of vocabulary. The second important parameter is *layer size* which determines the number of nodes in the hidden layer. This parameter highly affects the training accuracy and processing time. A larger layer size is better in terms of accuracy, but this will be paid off by the running time. A reasonable figuration for this parameter is from tens to hundreds [Radim (2014)]. The final major parameter, *context window size*, decides how many context words to be considered. Google recommends a size of 10 for the Skip-gram model [Google Inc. (2016)]. In our experiments, these parameters are subject to be changed so that the best model can be achieved. The selection of an optimal parameter setting is discussed later in the 'Implementation and Performance Evaluation' section.

3.3.3 Semantic Relation Classification Algorithm

This section explains a designed classification algorithm for automated identification of semantic relations between data terms. This study focuses on the following three semantic relations: *synonymy* (similar-to), *hyponymy* (is-a), and *meronymy* (part-of). The relation *similar-to* refers to a pair of terms that share similar meanings. Since very few instances have exactly the same meaning; in this study, the *similar-to* category also includes near-synonyms which can be used interchangeably to a certain extent [Inkpen and Hirst (2006)]. For example, two terms 'highway' and 'street' are equivalent in the context where geometry is the only attribute considered. Another type to be detected is the *is-a* tag which relates to concept-superconcept pairs, for instance 'highway-facility'. Finally, *part-of* is associated with instances where a concept represents a component (or a property) of one another concept (i.e., 'shoulder-road', 'volume-traffic').

Terms that relate to each other via one of the above semantic relations are expected to have a high similarity score. Thus, a collection of nearest terms generated by the vector space model is an excellent source of semantically related terms. To support automated detection of relation type, this study designs a classifying algorithm of which the pseudo code is shown in Algorithm 1. Given a pair of the target t and a near term n, the algorithm returns one of the following tags *similar-to*, *is-a*, *part-of*, and *non-related*. First, a surfacing rule-based checking is performed. The rule herein is that if the target word t (e.g., road) is the head noun of a near term n (e.g., 'public road'), a triple *(n is-a t)* is correspondingly harvested. In cases where t (e.g. road) matches the modifier component of n (e.g., 'road facility'), the modifier is eliminated from n. Second, the algorithm detects the relation between pair *(n-t)* by checking its occurrence in a syntactically related pair dataset. The syntactic resource consists of *is-a* and *part-of* term pairs which are extracted from the input corpus using a minimal-supervised training method (explained in the section below). The algorithm also considers *reverse is-a* (hypernym) and *reverse part-of* (whole-of) when the input pair in reverse order exists in the syntactic resources. If the input pair does not belong to those categories, it is temporarily tagged *similar-to*. Clustering is then applied on the temporary *similar-to* list after being sorted

Algorithm 1 Semantic relation classification algorithm

```
1: Inputs: term t, list of nearest terms N, list of partof pairs P, list of isa pairs I
2: Outputs: list of Parts, list of Wholes, list of Hyponyms, list of Hypernyms, list of Synonyms
3: procedure TERM CLASSIFICATION PROCEDURE
4:     for all n ∈ N do
5:         x ← pair n:t
6:         h ← headOf(n); m ← modifierOf(n)
7:         if h = t then
8:             add x to Hyponyms
9:         else if m = t then
10:            n ← h
11:        else
12:            if n : t ∈ P then
13:                add x to Parts
14:            else if t : n ∈ P then
15:                add x to Wholes
16:            else if n : t ∈ I then
17:                add x to Hyponyms
18:            else if t : n ∈ I then
19:                add x to Hypernyms
20:            else
21:                add x to Synonyms
22:    Clusters ← Kmeans(Synonyms)
23:    Synonyms ← top_C(Clusters)
```

by similarity in descending order. As similar terms tend to have a high similarity score, accepting only items occurring in the top c clusters helps to eliminate other *non-related* terms. Sections below discuss in detail the collection of *is-a* and *part-of* instances and the clustering of *similar-to* items.

3.3.3.1 Part-of and is-a instance extraction

Using syntactic patterns like the ones developed by Hearst (1992) is a popular method for automated detection of lexical relations. This method is straightforward as instances can be quickly captured and can yield a high precision. However, a typical issue of using pre-defined rules is the low recall as generic patterns are usually ignored Pantel and Pennacchiotti (2006). Generic patterns are those that are applicable to multiple types of relations. For instance, the

pattern 'X of Y' can be found in both *part-of* (e.g., 'shoulder of roadway') and *is-a* (e.g., 'facility of highway'). In addition, existing patterns are usually induced from generic corpora, they might not well applicable for a domain corpus. This study adopts a widely used minimal-supervised technique proposed by Pantel and Pennacchiotti (2006) to learn reliable patterns for *is-a* and *part-of* relations from the highway corpus. The selection of this particular method is because of its computational efficiency and recall improvement as more patterns can be discovered from domain-specific texts. The pattern learning is an iterative procedure of the following steps: (1) pattern induction, (2) pattern ranking/selection, and (3) instance extraction.

Pattern learning starts with extracting word sequences connecting the constituents of each pair instance for a certain relation (e.g., part-of). In order to initiate the first iteration, seed pairs, which are found by examining engineering glossaries from various State Departments of Transportation (DOTs), are used. For example, with the seed 'median-roadway' of *part-of*, one extracted sequence is 'roadway without a median' which correspondingly yields a pattern *'WHOLE without a PART'*. Along with 'good' chains, 'bad' ones (e.g., 'of the roadway when median is') are also collected. Similar to the NP extraction task, a frequency threshold of 2 is used to reduce random sequences. The reliability of a pattern p in P patterns collected is measured as the average association with all instances in I using the Equation 3.4 below:

$$r_\pi(p) = \sum_{i \in I} \frac{\frac{pmi(i,p)}{\max_{\forall i \in I} pmi(i,p)} * r_l(i)}{I} \tag{3.4}$$

where $r_l(i)$ is the instance reliability score which is defined later in Equation 3.6. The reliability of initial seed pairs is set to 1. The association between instance i and pattern p, $pmi(i,p)$, is based on their occurrence frequencies as follows:

$$pmi(i,p) = log \frac{|x,p,y|}{|x,*,y||*,p,*|} \tag{3.5}$$

where the asterisk (*) represents a wildcard.

The patterns induced in step 1 are ranked according to their reliability scores and only the top-k are accepted, where k is set to 1 in the first iteration and increases by 1 over each iteration. The algorithm runs until k meets a given desired number of patterns, τ, which is 5 for all experiments in this study.

In step 3, instances of related pairs are extracted from the corpus using those patterns accepted in step 2. The reliability of an instance i is measured based on an equation analogous to the pattern reliability, as in the equation below. Subsequent iterations will use the top m instances extracted for the pattern induction phase. In our experiments, we set $m = 100$. At the last iteration when τ patterns are induced, the extracted pairs are accepted as lexical-syntactic resources which will be used by the relation classifier.

$$r_l(i) = \sum_{p \in P} \frac{\frac{pmi(i,p)}{\max pmi} * r_\pi(p)}{P} \tag{3.6}$$

3.3.3.2 Similar-to instance clustering

In this phase of the classification algorithm, the system implements cluster analysis on the temporary list to separate *similar-to* terms from other *non-related* items. This study employs K-mean clustering algorithm MacQueen (1967) to split the list into multiple clusters according to their similarity scores with the target word. The objective of K-mean clustering, as illustrated in Equation 3.7, is to minimize the sum of squared distances between words and the corresponding cluster centroid, where μ_i is the mean of points in the cluster C_i and k is the number of clusters. Since similar words tend to have a higher similarity score than other non-related words, items in the top clusters are more likely to be similar to the target word. Those terms beyond the top-c clusters are unlikely to be a similar term; they are, thus, removed from the temporary *similar-to* list and are classified as *non-related*. Increasing k would provide a better separation of near words, we choose the value of k as high as the total *similar-to* candidates divided 2 in our experiments.

$$arg \min_{C} \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2 \tag{3.7}$$

3.4 Implementation and Performance Evaluation

This section presents an implementation case study on classifying roadway transportation data terms using the domain text. An empirical comparison between the proposed model and several baseline methods are also discussed.

3.4.1 Experiment setup

We performed experiments on a highway corpus composed of 48 engineering manuals and guidelines from 30 State DOTs. The content in a manual document in the civil engineering field is commonly presented in various formats such as plain text, tables, and equations. Since the structures of words in tables and equations are not yet supported by the state-of-the-art NLP techniques, they were removed from the text corpus. The removal may slightly reduce the corpus size, and accordingly affects the training dataset. However, since sequences of words in tables and equations are not organized in the formal structure of a sentence, many unreal noun phrases would be captured when applying NP patterns on those features. The final plain text corpus consists of nearly 16 million words. This dataset was utilized to extract multi-word technical terms which were then trained and transformed into representation vectors.

In this study, a Java prototype was built to assist researchers in implementing the proposed methodology to extract heterogeneous domain data elements and their semantic relations from plain text technical documents. The implementation procedure was according to the phases described in the proposed methodology. Specifically, the plain text roadway corpus was first fed into the system to generate a bag of roadway data elements, a dataset of their representation vectors, and a collection of syntactically related pairs. This was followed by an evaluation of the semantic classifier algorithm and a comparison to several baseline models. The classifier was also tested with different parameter settings.

To evaluate the system performance, we developed a test dataset consisting of 22,500 pairs. Of which, there are 332 related pairs of words (88 is-a, 176 part-of, and 68 similar-to) and 22,168 non-related instances. The vocabulary of the test pairs was extracted from 1,000 sentences randomly selected from the highway corpus. By manually reviewing the automatically generated terms from the test sentences, 150 domain technical terms that appear two or more times were collected. Three Ph.D. students in Civil Engineering, including the first author, worked as annotators who independently identified and labeled the semantic relations among 150 words in the test vocabulary. They were asked to assign one of the following three tags to a certain semantically related pair: *part-of*, *is-a*, and *similar-to*. Other pair combinations among 150

words beyond those discovered and tagged by annotators were automatically assigned 'non-related' tag. The knowledge base WordNet and various DOT roadway transportation glossaries were used during the annotation process. As a result, 332 pairs that at least two annotators agree were obtained for the validation purpose. For a given pair of terms, the system returns one of the following tags: *is-a*, *part-of*, *similar-to*, and *non-related*. In this study, the following three measures are used to evaluate the semantic classifier: precision, recall, and F-measure. Let S_i denote a set of true pairs labeled with relation i in the test set, and S'_i is a set of pairs classified as relation i by the system. The evaluation metrics for a certain relation are defined in Equations 3.8-3.10. The overall system performance is evaluated using the same equations, but is based on the total correctly classified pairs for all types of relations.

$$Precision_i = \frac{S_i \cap S'_i}{S'_i} \tag{3.8}$$

$$Recall_i = \frac{S_i \cap S'_i}{S_i} \tag{3.9}$$

$$F_i = \frac{2 \cdot Precision_i \cdot Recall_i}{Precision_i + Recall_i} \tag{3.10}$$

To evaluate the success of the system, experiments were conducted to compare the performance between the proposed classifier and other two baseline methods. The first baseline model is one that purely uses lexical patterns learned in this study to detect the semantic relation between a given pair of terms. Since this uses only rules, *similar-to* is not applicable. The second baseline method employs Word2Vec without integrating pattern features. This model is basically the same as the proposed method but all near words generated by Word2Vec are accepted as similar terms. Therefore, the comparison with this baseline method was only on the *similar-to* relation.

3.4.2 Output from interim steps

The first output from the system is a domain terminology set. There are almost 288,000 NPs extracted, of which over 17,000 were accepted as technical terms after removing instances with stop words and applying the 15% cut-off policy. Table 3.3 shows the distribution of terms

Table 3.3: Total number of extracted terms. C-values are between brackets.

N-gram	Count	Percentage	Top 5 (c-value)
Bigrams	11,446	65.62%	sight distance (9701); design speed (9376); traffic control (6142); cross section (5280); clear zone (4837)
Trigrams	4,421	25.35%	right of way (7945); traffic control device (3188); contract unit price (2836); left turn lane (1976); portland cement concrete (1930)
4-grams	1,180	6.76%	right of way line (1147); uniform traffic control device (924); highway right of way (907); portland cement concrete pavement (737); right of way acquisition (564)
5-grams	306	1.75%	two way left turn lane (303), mdt statewide integrated roadside vegetation (241); portable precast concrete barrier rail (163); right of way control section (149); effective modulus of subgrade reaction (130)
6-grams	68	0.39%	positional accuracy of as built record (65); right turn fixed object pedestrian night (46); bridge rehabilitation technique steel superstructure reference (46); air void of compacted bituminous mixture (38); continuous two way left turn lane (38)
Total	17,443	100%	

by sequence length along with the top 5 examples for each category. As shown, the majority are bigrams (65.62%), while lengthy NPs account for a relatively small portion in the corpus, 1.75% and 0.39% respectively for 5 and 6 grams. Using this terminology dataset, the corpus was modified by connecting the tokens in the multiple-word terms with the minus sign (-) to ensure that they are treated as single tokens.

The system was then applied on the modified corpus to extract lexical pairs. Table 3.4 shows the patterns learned and examples of instances harvested for the *part-of* and *is-a* relations. We used 10 and 7 seeds respectively to collect pairs related through the two relations. Those seeds were obtained by reviewing various roadway transportation glossaries. As shown in the table, 3 groups of patterns were induced for each relation *part-of* and *is-a*. Using these patterns, around 30,000 *part-of* and 8,000 *is-a* pairs were collected.

Another important product generated by the system is a term space. Figure 3.4 presents the vector space of roadway data elements derived from the word embedding training process

Table 3.4: Patterns learned and examples of pairs extracted.

Relation	Seeds	Patterns learned	Extracted pairs
X part-of Y	alignment::roadway median::roadway ramp::interchange (Total seeds: 10)	X (of|at|in) (a|an|the) Y Y (with|with no|without) (a|an) X Y ('s|where) X	curb::roadway sidewalk::bridge radius::horizontal curve (Total pairs: 30,423)
X is-a Y	highway::facility culvert::drainage facility sign::traffic control device (Total seeds: 7)	X {,} (NP,)* (and|or) other Y Y {,} such as (NP,)* X Y, including (NP,)* X	cracking::damage bridge::structure crane::equipment (Total pairs: 8,339)

Table 3.5: Examples of top nearest words.

Target term	**Nearests**	**Cosine**	**Rank**
street	highway	0.658	1
	direct-access	0.583	2
	collector-road	0.557	3
	public-street	0.533	4
	local-street	0.561	5
	...	...	...
	curb-extension	0.526	13
	on-street-parking	0.491	23

when the parameters, *frequency threshold*, *hidden layer size* and *window size* were set to 5, 100 and 5 respectively. To present those high-dimensional vectors in a 2D graph, PCA (Principle Component Analysis) was used to reduce the dimension. Based on the distance between terms visualized in Figure 3.4, the most related data elements for a certain data type can be quickly identified. For example, an *inlet* (bottom right corner) can be inferred to be more similar to an *outlet* (bottom right corner) than to a *pavement* (upper right corner). Table 3.5 shows a partial ranked list of the nearest terms of 'street' in order of similarity score.

3.4.3 System performance

Before evaluating the system and comparing the performance with baseline methods, several experiments were carried out to identify the optimal value for three model parameters, *frequency threshold*, *vector size*, and *context window size*, and to select a better network type (CBOW or skip-gram) of the Word2Vec training model. To examine the effect of a certain parameter we

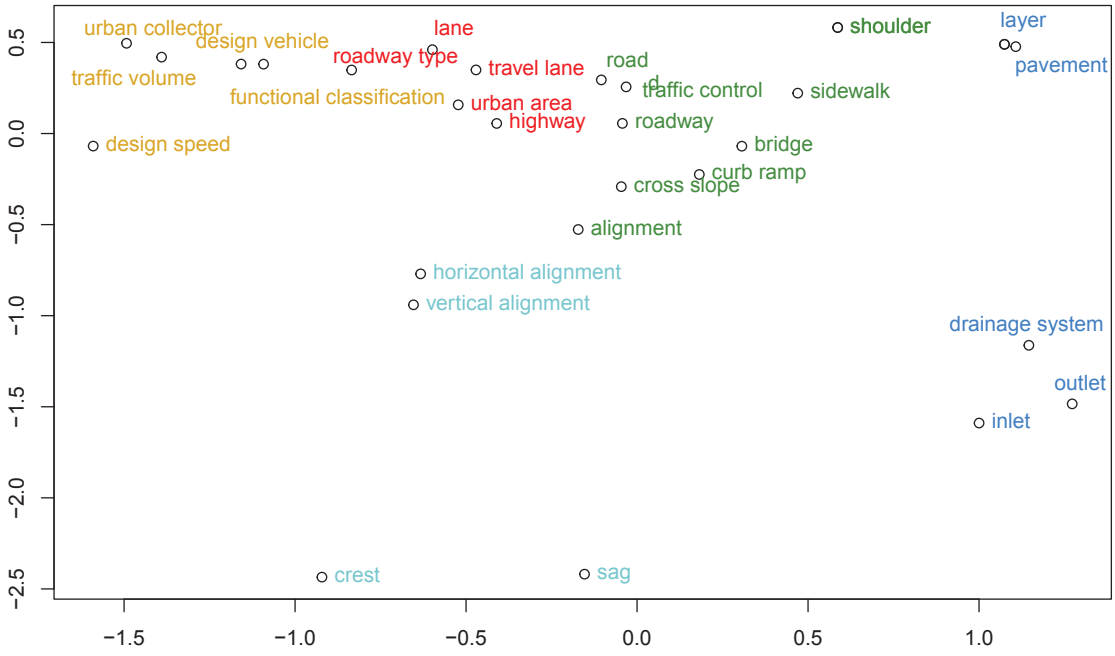

Figure 3.4: PCAs representation of roadway term vectors

Table 3.6: Overall system performance with different parameter settings and training network type.

Model	**Precision (%)**	**Recall(%)**	**F (%)**
CBOW 5-100-5	**92.76**	**81.02**	**86.50**
CBOW 5-300-5	93.70	77.37	84.76
CBOW 50-100-5	84.44	85.71	85.07
skip-gram 50-100-5	80.60	65.06	72.00
skip-gram 50-100-15	76.15	54.82	63.75

increased its value in the standard setting (5, 100, 5) while other parameters stayed unchanged. The training network type was also changed to determine the optimal setting. Table 3.6 shows the results from those experiments when the top synonym cluster parameter, c, was set to 2. The results indicate that neither increasing frequency threshold, hidden layer size, nor window size necessarily improves the system performance. In addition, CBOW shows its strong superiority to skip-gram in our classifying system. Thus, in the comparative testing with other baseline methods, we used the standard parameter set with the CBOW structure.

Table 3.7: System performance. P, R, and F respectively denote precision, recall and F measure.

Model	part-Of			is-A			similar-To		
	P(%)	R(%)	F (%)	P(%)	R(%)	F (%)	P(%)	R(%)	F (%)
Pattern only	80.00	95.45	87.05	81.93	77.27	79.53	-	-	-
CBOW only	-	-	-	-	-	-	70.0	61.76	65.63
CBOW+Pattern	94.74	81.82	**87.80**	94.87	84.09	**89.16**	85.0	75.0	**79.69**

Table 3.8: Excerpts of extracted near-synonym set.

No.	Synonym set
1	highway; road; street
2	crosswalk; crosswalk-line; pedestrian-crossing
3	roundabout; traffic-circle; splitter-island
4	traffic-island; refuge-island; pedestrian-refuge
5	subbase; subgrade; base-course; base-layer
6	grade-separation; at-grade-crossing; interchange; overpass

Table 3.7 shows the performance of the proposed method in comparison to other two baseline models. The performance for *similar-to* in this table is in accordance with the best case (F-score reaching max) when varying the number of top c clusters accepted (see Figure 3.5 and 3.6 respectively for CBOW and CBOW+Pattern models). As shown in the table, the integration between syntactic patterns and semantic word vectors significantly improves both recall and precision for the *is-a* and *similar-to* relations. A slight F enhancement is also observed for the *part-of* relation. Among those three relations, *is-a* has the best performance with a precision of nearly 95% and a recall of around 85%. These impressive figures yield a 14% F improvement over the pattern-based approach, in which a major contribution is from the precision. It is evident that once semantic relatedness is considered, incorrect instances matching the syntactic *is-a* patterns can be effectively eliminated. Detecting synonymy, which is the most challenging task, also achieves a relative F score of 79.69% compared with 65.63% when solely using CBOW. With respect to *part-of* detection, the integrated method greatly enhances the precision from 80% to 94.74%; however, due to a considerable drop in recall, the overall F improvement is just 0.75%. This result indicates that the induced *part-of* patterns are highly reliable; thus the inclusion of semantic features gives only a slight improvement.

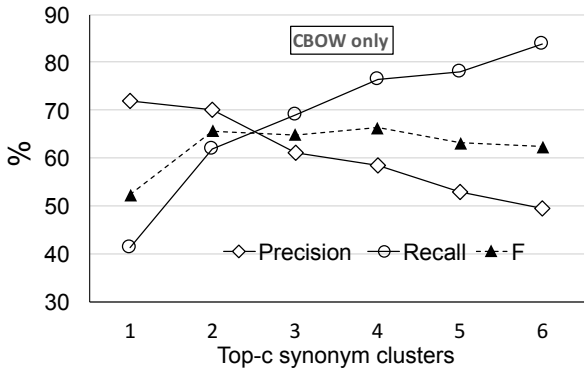

Figure 3.5: Synonym detection performance for CBOW model.

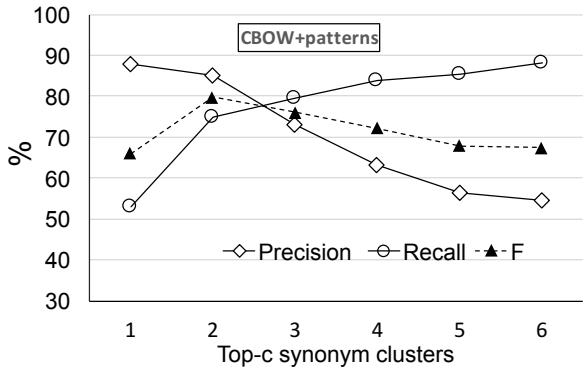

Figure 3.6: Synonym detection performance for CBOW+Pattern model.

3.5 Research findings, implications and limitations

This paper provides many important contributions to the area of integrating transportation asset data. The disparity of data names and semantics is a major hurdle to merging disconnected transportation data sources. This study provides a novel linguistic methodology to assist in classifying heterogeneous data items using linguistic information in technical text documents. Specifically, this study contributes to the body of knowledge by: (1) developing an NLP-based method for automated extraction of data types and their name variants from design manuals, (2) introducing a machine-learning approach that can learn the similarity in meaning among data items using their context words in texts, and (3) designing an algorithm that integrates syntactic rules, clustering, and word embedding to classify lexical relations among heterogeneous terms. The main merit of the study lies in the detection of linguistic inconsistency in naming

the same data element. This capability enables data integration to precisely combine similar data even given different terms in different systems. Another key advantage is the use of only linguistic information in domain texts for semantic relatedness identification. By purely using the occurrence of data elements in domain documents, the classifying algorithm overcomes the limitations of costly hand-crafted rules as used by Abuzir and Abuzir (2002) and Rezgui (2007), and eliminates the reliance on other existing dictionaries like in the work by Zhang and El-Gohary (2016).

The present framework is not to completely eliminate human involvement, but it is expected to offer an enabling tool that can assist researchers in developing supporting ontologies, taxonomies and other forms of semantic resources with the inclusion of alternative names for a concept. Using the method presented in this paper, less effort is required as the only major requirement is collecting domain documents. Researchers may need to pay some effort on validating the automatically generated datasets, but it is much less time-consuming than interviewing domain experts or manually examining written documents. Although the methodology has been tested only on a roadway corpus, it is generic and its applicability is broad. For example, the developed system can be implemented to extend the buildingSmart building data dictionary [buildingSMART (2016a)]. The findings of this study would accelerate the process of removing the current bottleneck of machine readable dictionaries which are required for unambiguous data sharing, integration, and exchange.

In addition to theoretical implications, the outcome of this study offers practical value to the highway industry. The datasets resulted from the experiment in this study provide name variants and related items for over 17,000 roadway data elements. For example, some of the alternative ways to present 'right of way' include 'row', 'r/w', or 'r.o.w.'. Several other examples of synonym sets generated from the system are shown in Table 3.8. The full library of terminology network generated from this study provides practitioners with suggestions on data keywords, their variations, and related data when finding data from external databases.

The current study has a number of limitations. The classifying algorithm covers only three types of semantic relations that are synonymy, hyponymy, and meronymy. Several other important relations that are not considered include siblings, functional associations, etc. The

inclusion of these relations into the classifier would reduce incorrect synonym matching, which will enhance the precision value. In addition, this study only targets at the synonymy issue, the polysemy obstacle is not yet addressed. Further research is needed to detect different senses of terms. Since a term that has multiple meanings would occur in different context, one potential solution is to cluster the instances of context words. A spread of contexts is a strong indication that a given term may refer to multiple things.

3.6 Conclusions

Data manipulation from multiple sources is a challenging task in transportation asset management due to the inconsistency of data terminology. The key contribution of this study is a novel approach for automated classification of semantic relations among heterogeneous data elements. In the proposed framework, machine learning is used to train the semantic similarity between technical terms. An algorithm is also designed to classify the nearest terms resulted from the semantic similarity model into distinct groups in accordance with their lexical relationships.

The developed system was tested and evaluated on a 16-million-word corpus of roadway design manuals collected from 30 State DOTs across the United States. The system performance was assessed by comparing automatically classified relations with those in a man-crafted gold standard. The result shows an overall performance of 92.76% in precision and 81.02% in recall. The best model is associated with the CBOW training structure and a parameter setting of 5, 100, and 5 respectively for frequency threshold, hidden layer size, and window size. One area for future studies is to improve the recall score which can be done by considering additional relation types. In addition, this paper focuses only on synonymy, research is needed to address the polysemy issue among data elements.

The proposed automated methodology for detecting semantic relations between data elements from texts is expected to significantly reduce human efforts in developing semantic resources for specific use cases in, but not limited to the field of transportation asset management. Once digital data dictionaries become readily available, the level of semantic interoperability can be fully achieved in the construction industry.

Acknowledgement

This research was funded by the National Science Foundation (NSF) via Award NSF-CIS 420-60-83. The authors gratefully acknowledge NSF's support. Any opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

CHAPTER 4. GENERATING PARTIAL CIVIL INFORMATION MODEL VIEWS USING A SEMANTIC INFORMATION RETRIEVAL APPROACH

A paper submitted for publication in *Automation in Construction* (May 2017)

Tuyen Le, H. David Jeong, Stephen B. Gilbert, and Evgeny Chukharev-Hudilainen

Abstract

Open data standards (e.g. LandXML, TransXML, CityGML) are a key to addressing the interoperability issue in exchanging civil information modeling (CIM) data throughout the project life-cycle. Since these schemas include rich sets of data types covering a wide range of assets and disciplines, model view definitions (MVDs) which define subsets of a schema are required to specify what types of data to be shared in accordance with a specific exchange scenario. The traditional method for generating MVDs is time-consuming and tedious as developers have to manually search for entities and attributes relevant to a particular data exchange context. This paper presents a computational method that can locate relevant information based on the user's keyword and return a subset of relevant nodes from a source XML data schema. The study employs a semantic resource of civil engineering terms to understand the semantics of a keyword-based query. The study also introduces a novel context-based search technique for retrieving related entities and their referenced objects. The developed method was tested on a gold standard of several LandXML subschemas. The experiment results show that the semantic MVD retrieval algorithm achieves a mean average precision of nearly 0.9.

4.1 Introduction

Neutral data standards have been widely accepted as an effective means for transferring the Civil Information Modeling (CIM) data of a civil infrastructure asset between project stakeholders. Several open standards have been proposed, for instance the Industry Foundation Classes (IFC) Extension for alignment [buildingSMART (2017)], LandXML [landxml.org (2017)], and TransXML [Ziering et al. (2007)]. These standardized data models contain rich sets of data elements covering various business processes and disciplines during the project life-cycle. However, a specific data exchange scenario needs only a subset of data. For example, among the digital design models of a certain corridor project created by the designer, automated machine guidance needs only the data associated with the earthmoving work including 3D surface models and alignment lines. Neutral data standards alone are insufficient to facilitate seamless digital data exchange among project stakeholders [Froese (2003); East et al. (2012)]. There is a need for formal definitions of subschemas specifying what types of data is needed for specific data exchange use cases.

Model View Definition (MVD), a concept introduced by the buildingSMART alliance [buildingSMART (2016b)], aims to fulfill the above need. MVDs are subsets of a standard data schema. These subsets represent only such data that is directly relevant in the context of a particular use case. The availability of these model views underpins the extraction of data from complicated sets generated throughout the project life cycle. The horizontal sector has adopted this MVD concept to develop a number of CIM schema views. A pioneering effort in this area is the InfraModel project carried out by the Technical Research Center of Finland that aims to define subsets of LandXML for different types of transportation assets [inframodel.fi (2017)]. In spite of significant research efforts, currently existing MVDs are yet inadequate to meet the large demand from the industry. This is because that the current method for developing MVDs is on a manual basis which is time-consuming and labor-intensive [Venugopal et al. (2012c); Eastman (2012); Hu (2014); Lee et al. (2016a)]. Developers are required to manually translate data exchange requirements presented in a paper-based Information Delivery Manual (IDM) into a machine-readable MVD. Much effort is also needed to tailor the existing MVDs to reflect

changes in industry practices. In order to remove this bottleneck, it is imperative to develop a more effective methodology [Venugopal et al. (2012c)]. The need for an automated methodology has been raised by various researchers.

One of the primary steps in the MVD development process is to identify classes, properties, and their referenced elements to be included in the view. In the current practice, developers need to interpret the semantics of the data keywords in an IDM and look for relevant entities, attributes, and types in the source schema. This task becomes extremely challenging especially for such large standards as LandXML and IFC which keep growing every year. These schemas are composed of thousands of classes and attributes along with complex relations such as superclasses and subclasses. Manually finding relevant classes for a given data need is tedious, time-consuming, and error-prone [Yang and Eastman (2007); Lee et al. (2016a)]. Although open standards are structured using a systematic categorization method, i.e., by assets in CityGML or by disciplines in IFC, developers may still need to go through the entire schema since a use case typically requires data from across different groups. This is even more problematic when the terms used in the IDM are inconsistent with the labels used in the source schema. Such terminology discrepancies may lead to a wrong inclusion of irrelevant entities, or to a failure to those are actually relevant. Because of those reasons, the task of identifying relevant entities and properties becomes an important hurdle for developers and possibly involves semantic errors. A method that can assist them in identifying related items in the source schema can afford significant time savings and reduce the number of errors in MVD development.

Previous work on enhancing the efficiency and effectiveness of the MVD development has focused on providing tools and methods that support syntactic validation of MVDs [Yang and Eastman (2007)] and improve their reusability [Lee et al. (2016a)]. With the state-of-the-art methods, selecting data in the source schema to meet the user's need for a specific use case still heavily relies on developers. To date, no methods of automating the process of finding relevant entities, properties, and relations for MVD development has been proposed. Automatically binding the end user's queries to the source schema would considerably accelerate the development of an MVD and reduce semantic mismatch errors.

To fill that gap, this research aims to develop a computer-assisted MVD methods for generating partial views from a CIM schema that is encoded in the eXtensible Markup Language (XML). XML is chosen because it is an international open standard used as the basis of various neutral data schemas for the civil infrastructure sector, such as, LandXML, TransXML, and RoadXML. The proposed method is an Information Retrieval (IR) technique that generates a ranked list of XML branches related to a given keyword query. The top retrieved results serve as suggested branches that should be used to form the desired XML subschema. This study adapts a semantic IR approach for generating MVDs. Specifically, the proposed technique consists of the following three stages: (1) interpreting the semantics of an input keyword using a civil engineering-specific knowledge base, (2) designing a context-based measure for assessing the semantic relevance of an XML class with a keyword query, (3) and developing an algorithm that can find referenced entities in the source tree schema to ensure syntactic completeness in the retrieved MVD.

4.2 Background

4.2.1 Neutral Data Standards for Civil Information Modeling

Building Information Modeling (BIM) for infrastructure, which is referred to as Civil Information Modeling (CIM) in this paper, is lagging behind the building industry, but has increasingly gained attention from both academic and practical communities. A large body of research has been undertaken for the last two decades to establish open data standards for the highway industry, of which, the majority are constructed in XML. LandXML [landxml.org (2017)], for example, is a result of early international collaboration efforts in facilitating interoperability in the civil industry, covering the following main groups of data: survey data, ground model, parcel map, alignment, roadway, and pipe network. As an attempt to improve LandXML and propose a new standard specialized for the transportation industry, the US National Cooperative Highway Research Program chartered the TransXML (NCHRP Project 20-64) project which focused on four business areas: survey/road design, construction/materials, bridge structures, and transportation safety [Scarponcini (2006)]. Of these domains, survey and geometric roadway

classes are mainly derived from LandXML in addition to various suggestions for improvement [Ziering et al. (2007)]. The buildSMART alliance is also actively participating in developing standards for infrastructure assets. This agency has recently released the IFC Alignment for the exchange of alignment information and is carrying out several other ongoing projects such as IFC Extension for roads and bridges.

4.2.2 Model View Definition

Model View Definition (MVD), which is a concept introduced by the building sector, is a formal subset of a data schema in accordance with the data requirements for a specific data use case [See et al. (2012)]. A subset of schema includes such elements as entities, types, attributes, and reference relations among entities in the source schema [Yang and Eastman (2007)]. MVDs help ensure that only required information instead of the entire dataset is shared with a target consumer. In addition, as an MVD provides a semantic map between the data elements for a domain user and the source entities, ambiguity in recycling data can be eliminated [Jiang et al. (2015)].

In the vertical sector, an extensive amount of research efforts on MVD development has been undertaken. Most MVDs aim to support the transfer of data from up-stream to down-stream phases. Example use cases are energy modeling [Jeong et al. (2014)] and building asset operation [East et al. (2012); East (2007)]. There are also MVDs for the reversed flow of information (e.g., construction methods, product details) from downstream actors to enhance early planning and design [Berard and Karlshoej (2012)]. Some of the results from the research community have become buildingSMART International (bSI) standards such as IFC coordination view, facility management handover view, structural analysis view, etc.

Recently, the MVD concept has also been adopted by the infrastructure sector. The VTT Technical Research Center of Finland develops a national specification for subsets of the LandXML schema, namely Inframodel [inframodel.fi (2017)]. The current version Inframodel 4 provides various MVDs for different types of infrastructure assets for instance waterways, water supply and sewage, roadways and streets, railways, and pipe networks.

4.2.3 MVD Development Process

The traditional process of developing an MVD has been well explained in the literature by various authors, for instance, Eastman et al. (2009), See et al. (2012), and Venugopal et al. (2012a). The typical procedure includes the following three major steps: (1) professional experts investigate industry business workflows and data exchange requirements to develop an Information Delivery Manual (IDM); (2) software developers translate the IDM in natural language into a computer-readable MVD by mapping the required information to those entities in the source schema and re-structuring them in a formal computerized format so that software vendors are able to develop the data exchange application; and (3) software applications are implemented and the translation results are validated. This approach to developing MVDs is on a manual basis which is labor-intensive [Venugopal et al. (2012a); Eastman (2012); Hu (2014); Venugopal et al. (2015)]. This leads to the shortage of MVDs in comparison with the large number of data exchange use cases involved in a single project or asset in the construction industry. Thus, there is a need for computational methods that can support automated generation of data schema subsets [Venugopal et al. (2012a)].

The conventional method shows various important drawbacks. Lee et al. (2016a) criticized the paper-based presentation format of IDMs. They pointed out that being presented in such an unstructured format, exchange requirements are not manageable and may include inconsistency and redundancy. In addition, constructing a model view from an IDM document is laborious and error-prone as developers are required to manually collect the exchange information described in natural language texts and translate them into formal classes and properties in the model view. Critics also argue that it is challenging to identify error sources when validating a model view [Lee et al. (2016b)]. To detect bugs in a translator, software vendors need to manually review a complex set of mapping pairs. With respect to the reusability aspect, the current development method is found to be a duplicated process [Venugopal et al. (2012a)]. They explain that the current practice lacks an explicit definition for a concept; an MVD, therefore, can be used only for a single use case and is not able to be reused for other exchange scenarios.

4.3 Related Studies and Knowledge Gap

This section presents an extensive review of related studies on automated generation and validation of MDVs. Their advancements and limitations will be discussed hereafter.

4.3.1 Previous Studies on Automated MVD generation

A few studies on automated translation of IDMs into machine-readable MVDs are found in the literature. A common objective of these works is to reduce the manual task performed by developers in finding syntactical referenced relations to those base entities that are identified to match the data exchange need.

The method proposed by Yang and Eastman (2007) is one of the notable studies on automated generation of IFC subsets. The study defines various rules to construct two types of subsets including 'base sets' and valid subsets. A 'base set' is a base IFC entity that is included with a set of dependent data types, whereas a legal subset is an aggregation of a 'base set' and its syntactically referenced 'base sets'. With respect to the semantic level, the method of Yang and Eastman (2007) offers developers with a mechanism for defining semantic rules, such as *ifcDoor* must be added when *ifcFireExit* exists, to create semantically complete subsets. While the developed syntactic rules are generic and can be applied in broad applications, knowledge rules are largely dependent on the domain of interest [Yang and Eastman (2007)]. Thus a valid subset formulated for a context might be not reusable to others.

As an attempt to address the reusability weakness of the above method, Lee (2009) introduced the concept of 'minimal set' that serves as a basic legal semantic unit and can be shared by different contexts. The rules used to define a 'minimal set' are mainly based on those proposed by Yang and Eastman (2007). A 'minimal set' is a complete set of entities, properties, and their references, presenting a single real life concept such as 'wall' or 'building.' Since this basic set itself is a complete semantic subset, it can be shared between model views. To successfully implement this method, however, a standard ontology of concepts that is agreed by all relevant domains is needed. The mapping between the concepts in the target ontology and the entities in the source schema is also required.

A common limitation of the above methods is that they are designed for extracting views from only a single data model. To overcome this drawback, Katranuschkov et al. (2010) introduced a multi-model view generation technique that can support filtering and aggregating object properties and relations to generate a single view from different models. They proposed various filtering rules to automatically find referenced classes (e.g., abstract types, superclasses) from multiple schemas. This method is particularly useful for a domain where the required information is from multiple sources.

As discussed, validating the syntactic correctness of a model view is currently well supported by various rule-based algorithms. This automated validation function can significantly reduce the burden on software developers; they, however, are still required to have a deep understanding of the semantics of the IFC schema to properly match with the data exchange elements in an IDM.

4.3.2 Knowledge Gap

Although a large number of studies have been undertaken, there is still a lack of a methodology supporting automated identification of semantically relevant source entities for a data exchange requirement item. The state-of-the-art on model view generation focuses on validating the syntax of a subset. The task of finding relevant entities for particular information need is still heavily dependent on developers. Prior studies assume that developers are aware of the semantics of the entity terminology in the source schema. Under this assumption, it is especially challenging for developers to find relevant source entities in a large and complex schema. Therefore, research is needed to offer effective tools and methods that can assist developers in quickly retrieving relevant classes given an input query. Generating MVDs using a language closed to human being would provide ease of use to the end user [Jiang et al. (2015)].

4.4 Keyword-driven Methodology for Generating XML Subschemas

As discussed earlier, the majority of the existing data standards in the civil engineering sector are presented in the XML format such as LandXML, TransXML, CityGML. The buildingSMART IFC schema, which is originally developed in EXPRESS structure, is also available

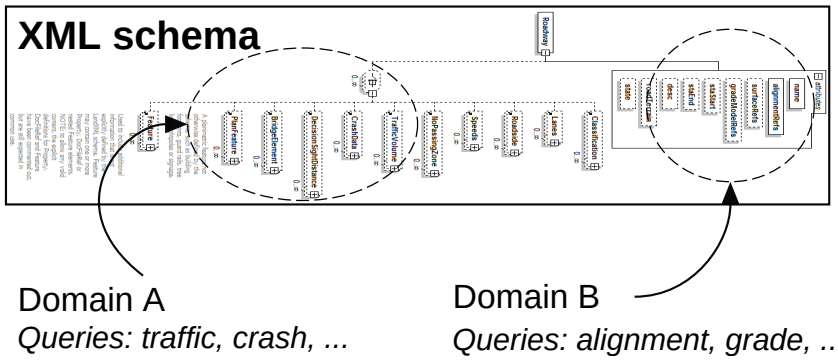

Figure 4.1: Partial views of XML schema

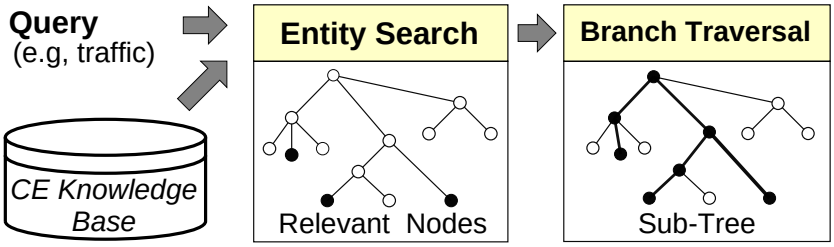

Figure 4.2: Overall method architecture.

in the ifcxml format. Because of the popularity of XML, the method proposed in this study is specially designed for XML schemas.

Figure 4.1 shows the conceptualization of model view extraction for a certain domain. Keyword-based queries are a popular means for the user to express their information needs in finding relevant data and information. Most of the queries are found to be shorter than 3 words [Arampatzis and Kamps (2008)]. Exploring the semantics behind an input keyword would help to capture the user's data and information interests. Once the desired context is identified, the associated portion of the source schema can be extracted. For example, the query 'traffic' refers to the partial view of LandXML that involves several entities such as 'Lanes', 'TrafficControl', 'Speeds'. The goal of this study is to propose a method that can take the input from the user and generate a corresponding subschema from a CIM schema. Since XML schema organizes classes and attributes in a tree structure. The proposed method is to specifically assist users in extracting partial branches of entities from a CIM neutral data standard using only keywords as the input information.

The keyword-driven MVD generator is an IR-like system that aims to obtain, from the source schema tree, a ranked list of XML schema branches relevant to a keyword-based query. In this study, we will look at single keyword queries rather than free-length ones. The top branches retrieved from the system are expected to be a useful starting point for developers to select entities and attributes that will form a schema subset. The architecture of the system, as shown in Figure 4.2, encompasses the following key components which will be discussed in more detail in the next sections.

1. Query expansion. The system searches for semantically related entities rather pure name matches. Thus, automatic interpretation of the semantics of the user input is needed. A lexicon specific to the civil engineering domain is used as a knowledge base for exploring the semantics of the query. This step generates an expanded query that adds to the original keyword with semantically related terms (e.g., synonyms, hyponyms, hypernyms, meronyms, etc.). This new set of keywords is utilized in the following phase for finding related source entities.

2. Entity search. At this stage, classes that are semantically most relevant to the user's input are located. The relatedness between an entity or property in the source schema and the expanded query set is measured using a concept-based matching procedure that we proposed (see below for details). The retrieved entities are then ranked by this relatedness measure.

3. MVD branch traversal. As a model view is a valid subschema, it must include all the referenced entities to ensure its syntactic completeness. This phase aims to implement a traversal technique to collect syntactically related classes (superclasses, referenced types, etc.) for those semantically related entities found in the previous stage. This procedure returns a set of XML branches connecting the relevant nodes to the root node.

4.5 Indexing Classes in XML Schema

Indexing the information sources is the first step in most IR systems. This step represents the source items in a such format that the search engine can effectively evaluate their relevance to the user's query. The relevance is measured by analyzing the representative features of a

source item. In this study, a source entity e is represented by the following features: (1) class name, (2) parent nodes, and (3) children nodes.

$$e = (e.name, e.parents, e.children) \tag{4.1}$$

where *e.parents* is a set of the class names of upper nodes within two levels of the target node e, and *e.children* set includes the labels of children and grandchildren nodes in the XML tree hierarchy. Stop nodes that are common geometric attributes of physical objects (e.g., width, length, area, type) are eliminated from the *e.children* set. Stop nodes are shared information of various classes, they provide little representative character. Since this study evaluates the semantic similarity based on the number of common features (as presented later), the discard of unnecessary features will help to dismisses their effects on the similarity score.

Entity names in a data standard do not necessarily follow the grammatical rules of English. Developers can define their own rules to name classes. Therefore, they may not match technical terms verbatim. For example, the label 'ProfAlign' in the LandXML schema stands for the real term 'profile alignment'. Searching for entities based on such abbreviated labels might fail to properly evaluate their relevance. In the index structure of the source schema, classes are renamed to a form called 'natural name' that is more close to the domain terminology. The 'natural name' of an entity is inferred by comparing its original computer-friendly label with the description texts provided in the referenced meta data document of a data schema. This process includes two steps. First, a label is splitted into tokens, with token boundaries before uppercase characters. Second, the referenced description texts are scanned for the full-word versions of the tokens. For example, with the label 'ProfAlign', these steps will respectively generate 'Prof Align' and 'Profile Alignment'.

4.6 Query Semantics Interpretation

This stage aims to infer the intention of the user by computing the semantics of the input query. Keyword-based queries are a common way for the user to express their needs when searching for information. To interpret the meaning of a query, this study implements a civil engineering lexicon, namely CeLex, as a domain knowledge base. This is a lexical resource of

civil infrastructure concepts, storing the semantics of the technical keywords in the domain. Using this knowledge base, the context terms relevant to a query can be identified. By using the relevant terms as supplementary queries during the retrieval process, the system is able to capture all source entities related to the user's need. Sections below explain in detail CeLex and the query expansion process.

4.6.1 Domain knowledge base

A domain knowledge is critical to understand the user's input keyword. To support inferring the semantics of a user query, this study utilizes CeLex[1] as the underlying domain knowledge base. CeLex is a lexicon that we constructed using our NLP toolkit, namely CeTermClassifer [Le and Jeong (2017)]. This is a machine learning based system that can automatically extract civil engineering terms and learn their lexical relations from a corpus of domain texts. The outcome provided by the system is a lexical space in which the semantics of a term is represented as a high-dimensional vector. The closeness between points in this space represents the semantic relatedness between the corresponding terms. In this model, terms are connected to one another through one of the following lexical links: synonymy (similar-to), hyponymy (is-a), hypernym (reverse is-a), meronymy (part-of), and holonymy (reverse part-of). A pair of terms that are close to each other but their specific lexical type is not detected by the system are called 'fuzzynyms'.

The CeLex lexicon utilized in this study was obtained by implementing the CeTermClassifier system on a 16-million-word corpus comprising 38 highway design manuals for 30 State Departments of Transportation. CeLex provides semantically equivalent and related terms for 17,000 individual technical keywords. Figure 4.3 illustrates the classified related terms of 'roadway' in the CeLex lexical space. By mapping the user's keyword query to this space, synonyms and context terms where the target keyword occurs can be identified. Those related terms are used for expanding the original input query.

[1] https://github.com/tuyenbk/CeTermClassifier

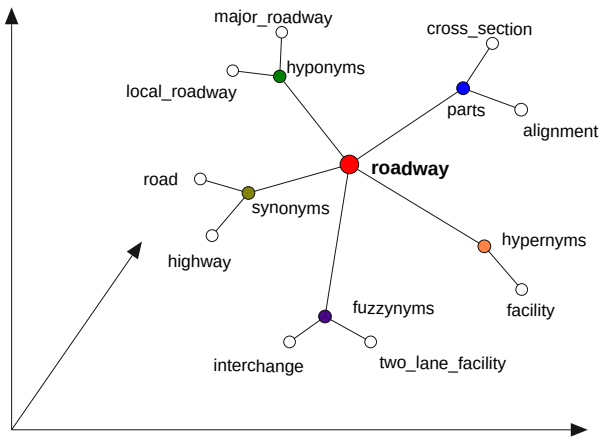

Figure 4.3: Partial Civil Engineering Lexicon

4.6.2 Keyword expansion and query concept formulation

Keyword search, which returns only items that contain the input term, is widely recognized to have a number of important weaknesses. Most of the problems of this approach are associated with terminological discrepancy. For example, synonymy can lower the recall of relevant entities. Also, polysemy, which refers to the multiplicity of term meanings, would lead to an inclusion of wrong items. In addition, it is not able to capture those source entities that are hypernyms or hyponyms of the input term. To overcome these problems, the user tends to conduct various search queries. However, they are required to have a deep knowledge of the target domain vocabulary. Additionally, finding relevant entities by manually trying all possible keywords is time-consuming and a number of good input options can still be missed.

We developed a semantic model view generation method that can allow for the retrieval of all relevant XML branches given a single keyword by the end user. For example, with the keyword 'drainage', all the related entities representing different drainage structures such as 'ditch', 'channel', 'pipe' should be obtained. This semantic search feature provides flexibility to the end user and helps to minimize the number of keywords used for entity retrieval. Given this requirement, interpreting the semantics of a query keyword to understand the user's intent is crucial to the quality of retrieved list.

In order to analyze the data needs suggested by an input keyword, its semantically equivalent and related terms need to be identified and included in the query. These additional keywords

are those terms that directly link to the user's original term in the CeLex knowledge base. As a result, an original query q_0 is extended to a set of queries Q^t which is the union of itself and the k nearest terms, where the latter are those items (e.g., synonyms, hyponyms) that are connected to q_0 in the lexicon. The equation below presents the expanded query set for a certain single keyword. If the vocabulary does not contain q_0, the nearest set becomes empty and Q^t includes only the original query.

$$Q^t = q_0 \cup \{t_1, t_2, ..., t_k\} \tag{4.2}$$

The expanded query set includes terms semantically related to the user's query. By simply mapping these keywords to the entity labels in the source schema, wrong entities might be captured due to the *polysemy* issue. To take into account this ambiguity problem, this study introduces the idea of *concept query*. A concept query q^c for a keyword query q^t is defined as a triple of concept name, parent context terms, and children context terms as follow:

$$q^c = (q^c.name, q^c.parents, q^c.children) \tag{4.3}$$

where $q^c.name$ is a keyword in the expanded keyword set, $q^c.parents$ is a set of CeLex terms to which $q^c.name$ relates through the $is-a$ edge, and $q^c.children$ are the parts or hyponyms of $q^c.name$. With the above 'keyword to concept query' transformation method, the expanded keyword query set Q^t correspondingly generates a set of $(k+1)$ concept queries Q^c as defined below.

$$Q^c = \{q_1^c, q_2^c, ..., q_k^c, q_{k+1}^c\} \tag{4.4}$$

4.7 Entity Matching and Ranking

The proposed algorithm for finding relevant entities in an XML schema is illustrated in Figure 4.4. The relatedness measure of a certain source entity e, as defined in Equation 4.5, is the accumulation of its similarity with every target concept query q^c in Q^c. The source entities are ranked by their relatedness scores and those lower than a threshold σ are eliminated.

$$\alpha_e = \sum_{q^c \in Q^c} \alpha_{e,q^c} \tag{4.5}$$

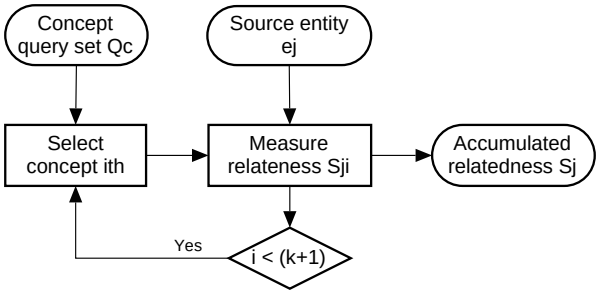

Figure 4.4: Relatedness measure approach

In the equation above, α_{e,q^c} is the similarity between a target concept query q^c and a source class e. As given in Equation 4.6, the concept similarity α_{e,q^c} is a weighted sum of two different measures of similarity: concept name matching (α^n_{e,q^c}) and context matching (α^c_{e,q^c}), where w_n and w_c respectively represent the weight of each matching type. Concept name matching measures the similarity in name between a pair of a concept query and a source entity, and context matching is based on the commonality of their parents and children features. Both of these measures are based on 'label string similarity'. Sections below explain the measures in detail.

$$\alpha_{e,q^c} = w_n \alpha^n_{e,q^c} + w_c \alpha^c_{e,q^c}, \quad where \quad w_n + w_c = 1 \tag{4.6}$$

4.7.1 Label String Similarity

Classes and properties in a data schema are represented by their names. The matching between the target and source sequences is a key metric for finding matched instances. The Levenshtein edit-distance algorithm Gale and Church (1993) is one of the most common methods for measuring the similarity between a given pair of sequences. However, like other popular string-based algorithms, this measure is computed on characters instead of words. Thus, using this method, a certain high level of similarity might be given to a pair of two different labels that share little common semantics such as 'trail' and 'rail'. To eliminate such matching errors, this study proposes to evaluate based on words rather characters.

In this study, the similarity between a pair of labels (a, b) is determined by the ratio of the number of common words over the total unique words, as defined in Equation 4.8, where the

$words(x)$ function returns a list of words composing a given label name. For example, with the pair *(traffic sign, road sign)*, there is 1 common word (sign) in the total of 3 unique words (traffic, sign, and road); thus, their similarity is 1/3 (33%).

$$sim_{string}(a, b) = \frac{|words(a) \cap words(b)|}{|words(a) \cup words(b)|} \tag{4.7}$$

4.7.2 Concept name matching - α^{n}_{e,q^c}

Concept name is an important indicator of semantic similarity. The degree of overlapping in name between an input concept query q^c and a source entity e reflects a certain level of semantic similarity between them. Concept name similarity α^{n}_{e,q^c} is based on the label string similarity measure and is computed as:

$$\alpha^{n}_{e,q^c} = sim_{string}(e.name, q^c.name) = \frac{|words(e.name) \cap words(q^c.name)|}{|words(e.name) \cup words(q^c.name)|} \tag{4.8}$$

4.7.3 Context matching - α^{c}_{e,q^c}

This measure compares the context items of the target and source concepts. The consideration of context is to reduce mismatches due to the polysemy issue. By comparing their attributes and other related entities, their meaning difference can be detected. The similarity from this viewpoint can be measured by the commonality and difference between their context entities. In this study, context is classified into parent and children contexts. The overall context similarity measure is the average of parent α^{cp}_{e,q^c} and children similarity α^{cc}_{e,q^c} as follows.

$$\alpha^{c}_{e,q^c} = \frac{\alpha^{cp}_{e,q^c} + \alpha^{cc}_{e,q^c}}{2} \tag{4.9}$$

Context similarity is commonly measured as a function of common and distinctive features of entities compared [Tversky (1977)]. In this study, the context similarity between a concept query and a source entity disregards the degree of difference in their features. Since the existing civil data schemas reflect only a small number of disciplines, the attributes included to define a class in the source schema are still incomplete. Whereas, a concept query which is formulated based on the CeLex lexicon, includes a large number of attributes. Assessing the similarity using the distinction information may involve a great bias. This paper evaluates the similarity based

only on what they share in common. A context similarity is defined as a logarithm function of the total string similarity score for all pairs of a context term in the concept query with one another in the source entity. The context similarity score is within the range [0-1]. The context similarity measures for parent and children context similarity are respectively shown in Equations 4.10 - 4.11:

$$\alpha^{cp}_{e,q^c} = \min(1, \log_{10}[1 + \sum_{m \in M_p} \sum_{n \in N_p} sim_{string}(e_{p,m}, q^c_{p,n})]) \tag{4.10}$$

$$\alpha^{cc}_{e,q^c} = \min(1, \log_{10}[1 + \sum_{m \in M_c} \sum_{n \in N_c} sim_{string}(e_{c,m}, q^c_{c,n})]) \tag{4.11}$$

where M_p and N_p respectively denote the collection of parent context terms of a source entity e and a concept query q^c. M_c and N_c represent children context term sets.

4.8 Branch Search and MVD Composition

A subschema must include referenced entities (e.g., datatypes, superclasses) to be syntactically complete. In an XML data schema, classes are constructed in a tree-like structure which encompasses of nodes and edges. The superclass of an arbitrary class is its parent node in the tree. Thus a legal MVD corresponds to a subtree of which the leaf-nodes are those entities obtained in the previous phase. Sections below describe the proposed method for generating a subtree given a list of leaf-nodes.

4.8.1 Branch Traversal Algorithm

In a tree schema, branches are defined as a path starting from a target node to the root. To retrieve XML branches for those semantically relevant nodes, this study adopts a traversal algorithm for ontology view extraction proposed by Seidenberg and Rector (2006). The algorithm is illustrated in Figure 4.5. As shown, the traversal algorithm aims to search for the path that connects a certain target class to the root of the schema. The search starts with a target node and goes upwards on the link connecting to its superclasses to accumulate related elements. When traversing throughout the tree is performed, a collection of paths will be collected. As an XML-based schema is a tree format, a given single node e will generate one and only one

path l from node e to the root node. The relatedness of a schema path with the user query is inherited from the relatedness score of the target leaf node. The path relatedness β is defined in Equation 4.12 below.

$$\beta_l = \alpha_e \tag{4.12}$$

4.8.2 Branch Merging for Subtree Formation

For a given input keyword, multiple relevant classes and accordingly various branches will be obtained. Since the matched paths might be overlapped, merging is necessary to eliminate duplication and allow for the generation of a single subschema.

In order to merge different branches, they are broken down into a set of separate segments each of which is an edge linking a certain pair of vertices. For example, the path 'Landxml → Roadway → Alignments → Alignment' will accordingly generate the following segments {Landxml → Roadway, Roadway → Alignments, Alignments → Alignment}. The final subtree is defined as a union of all sets of branch segments. A segment g inherits the relatedness score of its original path l, defined as follows.

$$\lambda_{g,l} = \beta_l \tag{4.13}$$

Since a unique segment may appear in multiple branches, its score is aggregated from all retrieved paths. The equation below shows the method for accumulating the score of a branch

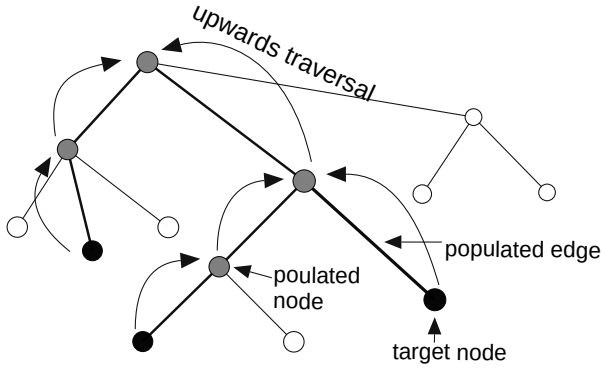

Figure 4.5: Traversal approach to populate schema branches

segment in the set of accepted paths L:

$$\lambda_g = \sum_{l \in L} \beta_l \cdot \pi(g, l) \tag{4.14}$$

where $\pi(g, l)$ is 1 if segment g appears in path l, and 0 otherwise.

4.9 Implementation and Discussion

4.9.1 Experiment setup

To evaluate the proposed method, we carried out an experiment on the LandXML 2.0 schema which consists of nearly 2,5000 entities and attributes. In this experiment, a gold standard which serves as a testing data set was developed to evaluate the accuracy of the developed subschema generator. The gold standard contains seven queries and their corresponding manually-constructed LandXML subschemas which are represented as a list of tree segments. Table 4.1 shows the test keywords along with several excerpts of segments in the gold standard[2]. To construct this testing dataset, seven keywords representing different contexts of civil engineering were selected. The test queries selected must represent a domain of knowledge that is covered in the source schema. We manually identified all the relevant tree branches and segments in the source schema for each of the keywords to construct the test MVDs. The existing Finnish specification of LandXML subschemas was used as a reference resource to find the relevant information to be included in the views. We also interviewed several local contractors who actively implementing automated construction technologies such as Automated Machine

[2] https://github.com/tuyenbk/CeTermClassifier

Table 4.1: Gold standard of LandXML subsets

No.	Query	Segment Count	Segment example
1	alignment	2	LandXML→Alignments
2	pavement	4	GradeSurface→Zones
3	surface model	31	Surface→SourceData
4	roadside	7	Roadway→Roadside
5	drainage	18	Roadside→Ditch
6	bridge	7	GradeSurface→Zones
7	traffic	28	Roadway→Lanes

Guidance and Stringless Paving, to verify those views regarding the construction domain. The developed testing data set was compared with the retrieved MVDs returned by the system to evaluate the system's performance.

This study adopted IR evaluation measures to assess the proposed method. The Mean Average Precision (MAP) measure (as defined in Equation 4.15) was used to evaluate the ranked list of retrieved segments. MAP is a unique measure that represents both precision and recall of the system. MAP is the mean of the average precision at different recall levels over the entire testing query set.

$$MAP = \frac{1}{Q}\sum_{j=1}^{j=Q}\frac{1}{N_j}\sum_{k=1}^{k=N_j} Precision(R_{jk}) \tag{4.15}$$

In the equation above Q is the test query set, N_j is the number of relevant tree segments for query j in the gold standard. $Precision(R_{jk})$ represents the precision of the top results which contains k relevant branch segments for query j, as given in the equation below.

$$Precision(R_k) = \frac{k}{\sum \text{Top results until k relevant items found}} \tag{4.16}$$

To demonstrate the success of this study, a comparison in retrieval accuracy between the proposed context-based model and a baseline keyword search model was conducted. This baseline is purely based on the matching of an input keyword with entity names. We compared these two models using the MAP metric. The precisions at several recall levels including 10%, 30%, and 50% were also reported. In addition, to explore the importance of concept name matching and context matching to the system performance, we run the system with different weight settings. The evaluation results are discussed in the following section.

4.9.2 Results and discussions

Tables 4.2 and 4.3 illustrate the LandXML subschema retrieved by the designed system for the query 'drainage' in two different representation formats. With the first format (see Table 4.2), the retrieved subtree is represented as a ranked list of full branches connecting the root and the relevant data entities. Presenting the results in this way helps to visualize all the referenced nodes for a semantical leaf node, but the relatedness measure for a specific segment of the subtree is not explicitly presented. Alternatively, the system can present the MVD subset in a

Table 4.2: Top retrieved branches for query 'drainage'

No.	Top Retrieved Branches	β_l	Relevant?
1	LandXML→Roadways→Roadway→Roadside→Ditch	3.33	yes
2	LandXML→GradeModel→GradeSurface→Zones→Zone	3.19	yes
3	LandXML→PipeNetworks→PipeNetwork→Pipes→Pipe→PipeFlow	3.10	yes
4	LandXML→PipeNetworks→PipeNetwork→Pipes→Pipe	2.06	yes
5	LandXML→PipeNetworks→PipeNetwork	2.05	yes

Table 4.3: Top retrieved segments for query 'drainage'

No.	Top Retrieved Segments	λ_g	Relevant?
1	LandXML→PipeNetworks	9.08	yes
2	PipeNetworks→PipeNetwork	8.93	yes
3	PipeNetwork→Pipes	5.44	yes
4	LandXML→Roadways	5.13	yes
5	Pipes→Pipe	5.10	yes

manner of a ranked collection of segments (see Table 4.3). The advantage of this method is the visualization of the relevance for each of the segments in the tree. However, users are required to link related segments if the they need to find referenced parents. To fully benefit from the advantages of these two formats, users would need to read the results in both ways.

Table 4.4 shows the comparison results between the prosed semantics-based model and the baseline model. As shown, the incorporation of semantic features significantly enhances the performance of the system whereby the MAP value improves from 58.43% to 86.75%. The semantic algorithm utilizes related terms for entity retrieval; it, therefore, allows the system to capture relevant entities with names different from the input keyword. For example, for the query 'drainage', the baseline model fails to capture relevant entities since the source LandXML schema does contain any classes or attributes with the name 'drainage'. Whereas the semantic algorithm is able to obtain related entities such as 'PipeNetworks' and 'Ditch' (see Table 4.2).

We also analyzed the system performance variation over the change to the weights of the matching components. We run the system with three different combinations of weights. The results of this experiment are illustrated in Table 4.5. As shown, the system performance largely depends on the weight setting. The MAP score was found to significantly vary from

Table 4.4: Effect of semantic search on performance. Precisions (%) are calculated for different recall levels. The semantic model performance is according with the weights w_n and w_c are both set to 0.5.

Model	R@10%	R@30%	R@50%	MAP (%)
keyword-base search	86.23	65.94	47.63	58.43
Semantic search	100.00	93.22	93.54	**86.75**

Table 4.5: Effect of weight setting on the system performance. Precisions (%) are calculated for different recall levels

Weight setting	R@10%	R@30%	R@50%	MAP (%)
w_n=1.0; w_c=0.0	100.00	89.59	82.13	83.63
w_n=0.5; w_c=0.5	100.00	93.22	93.54	**86.75**
w_n=0.0; w_c=1.0	85.71	78.78	73.38	68.32

just nearly 70% to over 86%. The results also show that there is a notable difference in the level of importance between the matching factors. A significant increase from 0 to 50% towards the context matching weight w_c leads to only a slight improvement of 3% in the overall system performance. When concept name matching weight becomes zero, the system performance noticeably falls to just below 70%. These observations indicate that the two matching factors both important to the quality of the retrieved results. However, it is evident that the major contribution of the semantic algorithm's outperformance over the keyword-base search is from the concept name matching. In other words, the expansion of the user keyword to consider other related terms during retrieval process plays a significant role in the system enhancement. The best performance in our experiment is corresponding to the case where their weights were both set to 0.5.

4.10 Research contributions and implications

The main contribution of this study is an effective method for automated generation of model views from an XML civil engineering data standard. The method allows for building a semantic IR system that can analyze the user's data interest from their single input keyword and return a corresponding subtree of the source schema. Previous studies focused on developing rule-based

methods for automatically validating the syntax completeness of model views. Because of the lack of an effective method, the construction of a semantically correct view still relies on the MVD developers. This article has fulfilled that need by providing a methodology to support developers in searching for entities relevant to the context of a domain.

The system developed in this study is expected to offer an enabling tool for MVD developers. A ranked list of related source entities generated by the system allows developers to work on a short list rather to manually scroll and examine the entire large and complex standard. With a list of the most semantically related items, the focus is paid on only a limited number of items; thus less effort is required to generate MVDs. In addition, less restriction is required for the end user to choose a keyword for searching relevant entities. The knowledge base utilized in this system is an extensive resource that covers a large number of domain terms. Users with little background in the domain are still able to extract a subschema without needing a deep understanding of the source schema.

Moreover, the system is expected to allow for a considerable reduction of time in IDM development. In the current practice, as discussed earlier, MVD development is a long process in which interviewing experts to develop an IDM is a critical step. IDMs help to specify what data elements relevant to a certain topic, but the development process is labor-intensive and may take several years. The present study offers a new approach in which the end user can use keywords for exploring relevant information. Rather than conducting a costly and time-consuming process of IDM development, the end user can simply use a query to search for subschemas. The proposed technique would help to transform the way that MVDs are developed as it enables an eliminate of IDM documentation.

4.11 Conclusions

Model view definition has been widely recognized as a means for facilitating seamless information exchange throughout the project life-cycle. Although a large body of MVDs have been developed, they are still limited compared with the large demand in the construction industry. This is because that the current ad-hoc practices of MVD development is time-consuming, laborious, and error-prone. The contribution of this study is an automated method for generating

MVDs using the user's keywords. The designed algorithm leverages a domain data dictionary to interpret the user's intention. It also utilizes a context-aware approach to match the interpreted concepts to those entities in the source XML data schema. The algorithm takes into account the variation in the name of concepts, thus it can reduce mismatches due to the inconsistent use of terminology between the user and the data standard.

The proposed method was tested on a gold standard of 7 subtrees manually extracted from the LandXML schema for different input keywords. The result shows that the algorithm can serve as an effective tool for extracting subsets of data schema when the mean average precision approaches .9. As using keywords is one preferable method for information search, the algorithm is expected to become a fundamental tool assisting professionals in extracting data from complex digital datasets. The technique presented in this article offers a foundation platform for future studies on transforming the way the end user interacts with CIM models. As keywords is a basic unit of human language, the capacity of understanding of this basic semantic unit allows computers to interpret the user's need in a more complex input.

However, the present study has several limitations. Namely, this study focuses on the abstract level and is not yet suitable for extracting partial views from an instance model. As instance extraction is the ultimate requirement for data exchange, future research is needed to develop additional rules to map the instance and abstract models. In addition, this study is limited to a single keyword per query. It is necessary to develop such operators as 'or' and 'and' to merge views from sophisticated queries.

Acknowledgement

This research was funded by the National Science Foundation (NSF) under the Award Number 1635309. The authors gratefully acknowledge NSF's support. Any opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

CHAPTER 5. CONCLUSIONS

This chapter concludes the dissertation with a summary and discussions on the contribution of the study, limitations and future studies. The key findings from the study will be presented first, followed by discussions on how the study impacts the highway and other construction sectors. Research limitations and suggestions for future research will then be discussed.

5.0.1 Summary

Project data has become increasingly available in digital formats. In spite of the fact that most stakeholders work on digital computerized systems, data sharing is not yet to be exchanged in digital format. The project information is mainly handed over in a non-machine-readable format. A key challenge for the computer-computer communication is that project partners use different syntax and terminology to represent their data. Due to the lack of tools and methods to overcome those problems, direct reuse of digital data generated throughout the project life-cycle is a human-relied practice which is time-consuming, costly and error-prone. Once digital data can be fully reused, data re-creation will be reduced and productivity will be improved.

This study offers an integrated computational platform that helps link heterogeneous life-cycle project data and enable computers to automatically extract sub-model given a keyword-based query. This study specifically answers the following questions.

- How to unify the life-cycle data in heterogeneous formats of a highway project into a unified and linked space?
- How can highway specific terms and their semantic relations including synonymy, hyponymy, meronymy be automatically extracted from engineering text documents?

- How to enable computers to automatically generate a partial view from a civil information data schema given a keyword-based query?

By answering the above questions, this study contributes to the body of knowledge the following methods and tools.

- An ontology-based data exchange mechanism for unifying and linking project data generated by different participants throughout the life cycle of a highway project. The study includes various data wrappers that convert proprietary data into RDF graphs. A set of linking rules is also developed to support linking separate data graphs. This framework is tested on a case study of pavement treatment selection which needs data from design, construction, and condition survey. The framework successfully enables linking diverse life cycle datasets and allows for automated generation of treatment information.

- An NLP methodology for automated classification of technical terms used by different highway agencies and disciplines. The method is tested on a gold standard of hand-coded pairs of related terms and achieves a relatively high overall F-score of 0.86. The experiment also results in a civil engineering lexicon which consists of nearly 17,000 highway terms. This new method uses purely texts as the input data, it helps to reduce the reliance on costly hand-crafted rules as used by Abuzir and Abuzir (2002) and Rezgui (2007) and to eliminate the dependence on existing semantic resource such as one proposed by Zhang and El-Gohary (2016).

- A novel semantic data retrieval system that can take the end user's keyword and return a list of the semantically most relevant data items from a source civil information data schema. This method looks for data elements based on their meanings rather than words. The system provides users with a model view in which entities and attributes are ranked by the relatedness score. This subschema data generator is tested on the LandXML standard using a gold standard consisting of the subschemas manually developed for 7 keyword-based queries. The system is evaluated on the ranked list of LandXML branches and the results show that the system achieves a mean average precision of nearly 0.9.

5.0.2 Research impact

The findings from this study are expected to offer the industry with enabling tools and methods for exchanging digital data throughout the highway project life-cycle.

For the data interconnection framework, since the RDF format is both machine and human-readable, practitioners with little programming background is still able to read and properly merge data. This framework would remove the burden on professionals to examine and extract data from complex datasets. The framework provides a foundation for a fully digital data exchange paradigm in which project data is shared and directly used across different stages.

The automated terminology classification developed in this study offers an enabling means for future research on semantic resource development for the construction industry. This tool is expected to effectively assist researchers in developing a dictionary for a specific domain. Using this system, the reliance on laborious work is reduced as the major effort required is only to obtain domain text documents. Thus this method is expected to accelerate the development of digital dictionaries which are critical for unambiguous data exchange and integration from heterogeneous and separate data sources in the highway industry. Since the proposed method is generic, it is applicable for developing new or expanding current semantic resources (e.g, the buildingSmart Data Dictionary) for other sectors.

Moreover, the research is expected to allow for a considerable reduction of time in MVD development. In the current practice, as discussed earlier, IDM development is a critical step to MVD formulation. IDMs help to specify what data elements relevant to a certain topic, but the development process is labor-intensive and may take several years. The present study offers a new approach in which MVD developers can quickly identify relevant information for a certain topic represented by a keyword. Rather than conducting a costly and time-consuming process of IDM development, the developer can simply use keyword-based queries to generate related subschemas. Thus, less effort would be required to find relevant data for a certain data need. The proposed technique would help to transform the way that MVDs are developed as it enables an eliminate of IDM documentation.

5.0.3 Limitation and future research

This study has several limitations. First, this research supports the data integration from only three main phases including design, construction, and asset condition survey. In order to achieve a complete data centric project delivery through the highway asset life cycle, research is needed to develop domain ontologies and wrappers for other domains of knowledge in the highway industry, for example, preliminary survey and asset management. The next limitation of this study is that the term classifying algorithm detects only three types of semantic relations that are synonymy, hyponymy, and meronymy. Other important relations that are not considered include siblings, functional associations, etc. The inclusion of these relations into the classifier would reduce incorrect synonym matching, which will enhance the precision value. In addition, this study only targets at the synonymy issue, the polysemy obstacle is not yet addressed. Further research is needed to detect different senses of terms. Since a term that has multiple meanings would occur in different context, one potential solution is to cluster the instances of context words. A spread of contexts is a strong indication that a given term may refer to multiple things. Finally, this study does not support data extraction at the instance level. The data extraction algorithm is only for partial extraction of subschemas. As instance extraction is the ultimate requirement for data exchange, further research is needed to construct rules to map the instance and abstract models. In addition, in this study, a query is limited to a single keyword. It is essential to develop such operators as 'or' and 'and' to merge views from sophisticated queries.

BIBLIOGRAPHY

AASHTO (2001). *Pavement management guide*. AASHTO, Washington, D.C.

Abuzir, Y. and Abuzir, M. O. (2002). Constructing the civil engineering thesaurus (cet) using the theswb. *Computing in Civil Engineering*.

Ananiadou, S., Albert, S., and Schuhmann, D. (2000). Evaluation of automatic term recognition of nuclear receptors from medline. *Genome Informatics*, 11:450–451.

Arampatzis, A. and Kamps, J. (2008). A study of query length. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 811–812. ACM.

Arenas, M., Bertails, A., Prud, E., Sequeda, J., et al. (2012). *A Direct Mapping of Relational Data to RDF*. World Wide Web Consortium.

Beetz, J., Van Leeuwen, J., and De Vries, B. (2009). IfcOWL: A case of transforming EXPRESS schemas into ontologies. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 23(01):89–101.

Berard, O. B. and Karlshoej, J. (2012). Information delivery manuals to integrate building product information into design. In *CIB W78-W102 2011: International Conference*.

Bittner, T., Donnelly, M., and Winter, S. (2005). Ontology and semantic interoperability. *Large-scale 3D data integration: Challenges and Opportunities*, pages 139–160.

buildingSMART (2016a). buildingsmart data dictionary. (Accessed: March 15, 2016).

buildingSMART (2016b). Model view definition summary. (accessed April 12, 2016).

buildingSMART (2017). Ifc alignment. Accessed: April 1, 2017.

Cambria, E. and White, B. (2014). Jumping nlp curves: a review of natural language processing research [review article]. *Computational Intelligence Magazine, IEEE*, 9(2):48–57.

Chen, D. and Manning, C. D. (2014). A fast and accurate dependency parser using neural networks. In *EMNLP*, pages 740–750.

Church, K. W. and Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

Costa-Jussa, M. R., Farrús, M., Mariño, J. B., and Fonollosa, J. A. (2012). Study and comparison of rule-based and statistical catalan-spanish machine translation systems. *Computing and Informatics*, 31(2):245–270.

Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. (2002). Gate: an architecture for development of robust hlt applications. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 168–175. Association for Computational Linguistics.

Curry, E., O'Donnell, J., Corry, E., Hasan, S., Keane, M., and O'Riain, S. (2013). Linking building data in the cloud: Integrating cross-domain building data using linked data. *Advanced Engineering Informatics*, 27(2):206–219.

East, E. W. (2007). Construction operations building information exchange (cobie). Technical report, DTIC Document.

East, E. W., Nisbet, N., and Liebich, T. (2012). Facility management handover model view. *Journal of computing in civil engineering*, 27(1):61–67.

Eastman, C. (2012). The future of ifc: Rationale and design of a sem ifc layer. Presentaion at the IDDS workshop.

Eastman, C., Jeong, Y., Sacks, R., and Kaner, I. (2009). Exchange model and exchange object concepts for implementation of national bim standards. *Journal of Computing in Civil Engineering*, 24(1):25–34.

El-Diraby, T. and Kashif, K. (2005). Distributed ontology architecture for knowledge management in highway construction. *Journal of Construction Engineering and Management*, 131(5):591–603.

El-Diraby, T., Lima, C., and Feis, B. (2005). Domain taxonomy for construction concepts: Toward a formal ontology for construction knowledge. *Journal of Computing in Civil Engineering*, 19(4):394–406.

Erk, K. (2012). Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 6(10):635–653.

Fagin, R., Kolaitis, P., Miller, R., and Popa, L. (2005). Data exchange: semantics and query answering. *Theoretical Computer Science*, 336(1):89 – 124. Database Theory.

Florida Department of Transportation (2014). Preparation and documentation manual. Technical Report 700-000-000, Florida, U.S.

Frantzi, K., Ananiadou, S., and Mima, H. (2000). Automatic recognition of multi-word terms: the c-value/nc-value method. *International Journal on Digital Libraries*, 3(2):115–130.

Froese, T. (2003). Future directions for ifc-based interoperability. *ITcon*, 8:231–246.

Gale, W. A. and Church, K. W. (1993). A program for aligning sentences in bilingual corpora. *Computational linguistics*, 19(1):75–102.

Gallaher, M. P., O'Connor, A. C., Dettbarn, J. L., and Gilday, L. T. (2004). *Cost analysis of inadequate interoperability in the U.S. capital facilities industry*. U.S. Department of Commerce Technology Administration, National Institute of Standards and Technology, Gaithersburg, MD.

Google Inc. (2016). word2vec. (accessed May 12, 2016).

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies*, 43(5):907–928.

Grüninger, M. and Fox, M. S. (1995). Methodology for the design and evaluation of ontologies. In *IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal, Canada.

Harispe, S., Ranwez, S., Janaqi, S., and Montmain, J. (2013). Semantic measures for the comparison of units of language, concepts or instances from text and knowledge base analysis. *arXiv preprint arXiv:1310.1285*.

Harris, Z. S. (1954). Distributional structure. *Word*.

Harrison, F., Gordon, M., and Allen, G. (2016). Nchrp report 829 - leadership guide for strategic information management for state departments of transportation.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545. Association for Computational Linguistics.

Heath, T. and Bizer, C. (2011). Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, 1(1):1–136.

Heflin, J. and Hendler, J. (2000). Semantic interoperability on the Web. In *Extreme Markup Languages 2000*.

Heiler, S. (1995). Semantic interoperability. *ACM Computing Surveys (CSUR)*, 27(2):271–273.

Hezik, M. (2008). Ifd library background and history. In *The IFD Library/IDM/IFC/MVD Workshop*.

Hicks, R. G., Moulthrop, J. S., and Daleiden, J. (1999). Selecting a preventive maintenance treatment for flexible pavements. *Transportation Research Record: Journal of the Transportation Research Board*, 1680(1):1–12.

Hill, F., Reichart, R., and Korhonen, A. (2015). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., and Dean, M. (2004). Swrl: A semantic web rule language combining owl and ruleml. In *W3C Member submission*. Word Wide Web Consortium.

Hsu, J.-y. (2013). Content-based text mining technique for retrieval of cad documents. *Automation in Construction*, 31:65–74.

Hu, H. (2014). *Development of interoperable data protocol for integrated bridge project delivery*. Ph.d. Copyright - Copyright ProQuest, UMI Dissertations Publishing 2014 Last updated - 2015-03-18 First page - n/a.

IFD Library Group. Ifd library white paper. Accessed: 2015-07-06.

inframodel.fi (2017). Inframodel. Accessed: April 1, 2017.

Inkpen, D. and Hirst, G. (2006). Building and using a lexical knowledge base of near-synonym differences. *Computational linguistics*, 32(2):223–262.

JBKnowledge (2016). *The 5th annual construction technology report*.

Jeong, W., Kim, J. B., Clayton, M. J., Haberl, J. S., and Yan, W. (2014). Translating building information modeling to building energy modeling using model view definition. *Scientific World Journal*, 2014.

Jiang, Y., Yu, N., Ming, J., Lee, S., DeGraw, J., Yen, J., Messner, J., and Wu, D. (2015). Automatic building information model query generation. *Journal of Information Technology in Construction*.

Jivani, A. G. et al. (2011). A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl*, 2(6):1930–1938.

Justeson, J. S. and Katz, S. M. (1995). Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1(01):9–27.

Karimi, H. A., Akinci, B., Boukamp, F., and Peachavanish, R. (2003). Semantic interoperability in infrastructure systems. *Information Technology*, pages 42–42.

Karshenas, S. and Niknam, M. (2013). Ontology-based building information modeling. In *Computing in Civil Engineering (2013)*, pages 476–483. American Society of Civil Engineers.

Katranuschkov, P., Weise, M., Windisch, R., Fuchs, S., and Scherer, R. J. (2010). Bim-based generation of multi-model views. *CIB W78*.

Kim, H., Orr, K., Shen, Z., Moon, H., Ju, K., and Choi, W. (2014). Highway alignment construction comparison using object-oriented 3d visualization modeling. *Journal of Construction Engineering and Management*, 140(10):05014008.

Kolb, P. (2008). Disco: A multilingual database of distributionally similar words. *Proceedings of KONVENS-2008, Berlin*.

landxml.org (2017). About landxml.org. Accessed: April 1, 2017.

Le, T. and Jeong, H. D. (2017). Nlp-based approach to semantic classification of heterogeneous transportation asset data terminology. *Journal of Computing in Civil Engineering*.

Lee, G. (2009). Concept-based method for extracting valid subsets from an express schema. *Journal of Computing in Civil Engineering*, 23(2):128–135.

Lee, S.-H. and Kim, B.-G. (2011). IFC extension for road structures and digital modeling. *Procedia Engineering*, 14:1037–1042. The Proceedings of the Twelfth East Asia-Pacific Conference on Structural Engineering and ConstructionEASEC12.

Lee, Y.-C., Eastman, C. M., and Solihin, W. (2016a). An ontology-based approach for developing data exchange requirements and model views of building information modeling. *Advanced Engineering Informatics*, 30(3):354–367.

Lee, Y. C., Eastman, C. M., Solihin, W., and See, R. (2016b). Modularized rule-based validation of a bim model pertaining to model views. *Automation in Construction*, 63:1–11.

Lefler, N. X. (2014). Nchrp synthesis 458: Roadway safety data interoperability between local and state agencies. Technical report, Transportation Research Board.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26. ACM.

Levy, O., Goldberg, Y., and Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Lima, C., El-Diraby, T., and Stephens, J. (2005). Ontology-based optimization of knowledge management in e-construction. *Journal of IT in Construction*, 10:305–327.

Lopes, L. and Vieira, R. (2015). Evaluation of cutoff policies for term extraction. *Journal of the Brazilian Computer Society*, 21(1):9.

Lossio-Ventura, J. A., Jonquet, C., Roche, M., and Teisseire, M. (2013). Combining c-value and keyword extraction methods for biomedical terms extraction. In *LBM'2013: 5th International Symposium on Languages in Biology and Medicine*.

Lv, X. and El-Gohary, N. M. (2015). Semantic annotation for context-aware information retrieval for supporting the environmental review of transportation projects. In *Computing in Civil Engineering 2015*, pages 165–172. ASCE.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

Manola, F., Miller, E., and McBride, B. (2014). RDF 1.1 primer. *W3C recommendation*, 10(1-107):6.

Marcus, M. (1995). New trends in natural language processing: statistical natural language processing. *Proceedings of the National Academy of Sciences*, 92(22):10052–10059.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

McGuinness, D. L. and Van Harmelen, F. (2004). OWL web ontology language overview. In *W3C recommendation*. Word Wide Web Consortium.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mounce, S., Brewster, C., Ashley, R., and Hurley, L. (2010). Knowledge management for more sustainable water systems. *Journal of information technology in construction*, 15:140–148.

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):10.

Navigli, R. and Velardi, P. (2010). Learning word-class lattices for definition and hypernym extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1318–1327. Association for Computational Linguistics.

Nenadić, G., Spasić, I., and Ananiadou, S. (2002). Automatic acronym acquisition and term variation management within domain-specific texts. In *Third International Conference on Language Resources and Evaluation (LREC2002)*, pages 2155–2162.

Nepal, M. P., Staub-French, S., Pottinger, R., and Zhang, J. (2013). Ontology-based feature modeling for construction information extraction from a building information model. *Journal of Computing in Civil Engineering*, 27(5):555–569.

Niknam, M. and Karshenas, S. (2013). A semantic web service approach to construction cost estimating. In *Computing in Civil Engineering (2013)*, pages 484–491. American Society of Civil Engineers.

Niknam, M. and Karshenas, S. (2014). A social networking website for AEC projects. In *Computing in Civil and Building Engineering (2014)*, pages 2208–2215. American Society of Civil Engineers.

Noy, N. F. (2004). Semantic integration: a survey of ontology-based approaches. *ACM Sigmod Record*, 33(4):65–70.

Noy, N. F. and McGuinness, D. L. (2001). *Ontology development 101: A guide to creating your first ontology*. Stanford knowledge systems laboratory technical report KSL-01-05 and Stanford medical informatics technical report SMI-2001-0880.

Osman, H. and Ei-Diraby, T. (2006). Ontological modeling of infrastructure products and related concepts. *Transportation Research Record: Journal of the Transportation Research Board*, 1984(-1):159–167.

Osman, H. M. (2007). *A knowledge-enabled system for routing urban utility infrastructure*. PhD thesis, University of Toronto.

Ouksel, A. M. and Sheth, A. (1999). Semantic interoperability in global information systems. *ACM Sigmod Record*, 28(1):5–12.

Pantel, P. and Pennacchiotti, M. (2006). Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 113–120. Association for Computational Linguistics.

Pauwels, P., Van Deursen, D., De Roo, J., Van Ackere, T., De Meyer, R., Van de Walle, R., and Van Campenhout, J. (2011). Three-dimensional information exchange over the semantic web for the domain of architecture, engineering, and construction. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 25(04):317–332.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Prud'hommeaux, E. and Seaborne, A. (2008). Sparql query language for rdf. In *W3C recommendation*. Word Wide Web Consortium.

Radim, R. (2014). Word2vec tutorial.

Rezgui, Y. (2007). Text-based domain ontology building using tf-idf and metric clusters techniques. *The Knowledge Engineering Review*, 22(04):379–403.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Scarponcini, P. (2006). Transxml: Establishing standards for transportation data exchange. In *Joint International Conference on Computing and Decision Making in Civil and Building Engineering*, Montreal, Canada.

Sclano, F. and Velardi, P. (2007). *Termextractor: a web application to learn the shared terminology of emergent web communities*, pages 287–290. Springer.

See, R., Karlshoej, J., and Davis, D. (2012). An integrated process for delivering ifc based data exchange.

Seedah, D. P., Choubassi, C., and Leite, F. (2015a). Ontology for querying heterogeneous data sources in freight transportation. *Journal of Computing in Civil Engineering*, page 04015069.

Seedah, D. P., Sankaran, B., and O'Brien, W. J. (2015b). Approach to classifying freight data elements across multiple data sources. *Transportation Research Record: Journal of the Transportation Research Board*, (2529):56–65.

Seidenberg, J. and Rector, A. (2006). Web ontology segmentation: analysis, classification and use. In *Proceedings of the 15th international conference on World Wide Web*, pages 13–22. ACM.

Shen, Z., Kevin, O., Choi, W., Kim, N., and Kim, H. (2014). *Object-based 3D Intelligent Model for Construction Planning/Simulation in a Highway Construction*, chapter 27, pages 259–268. American Society of Civil Engineers.

Sheth, A. P. (1999). Changing focus on interoperability in information systems:from system, syntax, structure to semantics. In *Interoperating geographic information systems*, volume 495 of *The Springer International Series in Engineering and Computer Science*, pages 5–29. Springer US.

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Suchanek, F. M., Ifrim, G., and Weikum, G. (2006). Leila: Learning to extract information by linguistic analysis. In *Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 18–25.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics.

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188.

Tversky, A. (1977). Features of similarity. *Psychological review*, 84(4):327.

U.S. DOT (2014). Highway performance monitoring system-field manual. Technical Report 2125-0028, Washington, D.C.

Uschold, M. and Gruninger, M. (1996). Ontologies: Principles, methods and applications. *The knowledge engineering review*, 11:93–136.

Venugopal, M., Eastman, C., and Sacks, R. (2012a). Configurable model exchanges for the precast/pre-stressed concrete industry using semantic exchange modules (sem). In *International Conference on Computing in Civil Engineering*, pages 269–276.

Venugopal, M., Eastman, C. M., Sacks, R., and Teizer, J. (2012b). Semantics of model views for information exchanges using the industry foundation class schema. *Advanced Engineering Informatics*, 26(2):411–428.

Venugopal, M., Eastman, C. M., Sacks, R., and Teizer, J. (2012c). Semantics of model views for information exchanges using the industry foundation class schema. *Advanced Engineering Informatics*, 26(2):411–428.

Venugopal, M., Eastman, C. M., and Teizer, J. (2015). An ontology-based analysis of the industry foundation class schema for building information model exchanges. *Advanced Engineering Informatics*, 29(4):940–957.

Walton, C. M., Seedah, D. P., Choubassi, C., Wu, H., Ehlert, A., Harrison, R., Loftus-Otway, L., Harvey, J., Meyer, J., Calhoun, J., et al. (2015). *Implementing the freight transportation data architecture: Data element dictionary*. Number Project NCFRP-47.

Wang, F., Zhang, Z., and Machemehl, R. B. (2003). Decision-making problem for managing pavement maintenance and rehabilitation projects. *Transportation Research Record: Journal of the Transportation Research Board*, 1853(1):21–28.

Webster, J. J. and Kit, C. (1992). Tokenization as the initial phase in nlp. In *Proceedings of the 14th conference on Computational linguistics-Volume 4*, pages 1106–1110. Association for Computational Linguistics.

Wegner, P. (1996). Interoperability. *ACM Computing Surveys*, 28(1):285–287.

Wetherill, M., Rezgui, Y., Lima, C., and Zarli, A. (2002). Knowledge management for the construction industry: the e-cognos project.

Yalcinkaya, M. and Singh, V. (2015). Patterns and trends in building information modeling (bim) research: A latent semantic analysis. *Automation in Construction*, 59:68–80.

Yang, D. and Eastman, C. M. (2007). A rule-based subset generation method for product data models. *Computer-Aided Civil and Infrastructure Engineering*, 22(2):133–148.

Yang, Q. and Zhang, Y. (2006). Semantic interoperability in building design: Methods and tools. *Computer-Aided Design*, 38(10):1099–1112.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 189–196. Association for Computational Linguistics.

Zaghloul, S., Helali, K., and Bekheet, W. (2006). *Development and Implementation of Arizona Department of Transportation (ADOT) Pavement Management System (PMS), Final Report 494*. Arizona Department of Transportation, Phoenix, Arizona.

Zarli, A., Bourdeau, M., Soubra, S., Rezgui, Y., Cooper, G., Aouad, G., Hannus, M., Bohms, M., Blasco, M., Hassan, T., Bouchlagem, D., Garas, F., Steinmann, R., and Gobin, C. (2003). Roadcon final report. Technical Report IST-2001-37278.

Zhang, J. and El-Gohary, N. (2016). Extending building information models semiautomatically using semantic natural language processing techniques. *Journal of Computing in Civil Engineering*, page C4016004.

Zhang, L. and Issa, R. R. (2011). Development of ifc-based construction industry ontology for information retrieval from ifc models. In *Proceedings of the 2011 Eg-Ice Workshop, University of Twente, The Netherlands*, pages 6–8.

Zhang, L. and Issa, R. R. (2013). Ontology-based partial building information model extraction. *Journal of Computing in Civil Engineering*, 27(6):576–584.

Zhang, Z., Iria, J., Brewster, C., and Ciravegna, F. (2008). A comparative evaluation of term recognition algorithms. In *LREC*.

Zhao, H. and Kit, C. (2011). Integrating unsupervised and supervised word segmentation: The role of goodness measures. *Information Sciences*, 181(1):163–183.

Ziering, E., Harrison, F., and Scarponcini, P. (2007). *TransXML: XML schemas for exchange of transportation data*, volume 576. Transportation Research Board.